



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## NÁVRH A REALIZACE SYSTÉMU PRO VÝPŮJČKU ZKUŠEBNÍCH ZAŘÍZENÍ

DESIGN OF A SYSTEM FOR BORROWING OF ELECTRONIC DEVICES

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Dominik Berčík

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Fiedler, Ph.D.

BRNO 2022

# Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

**Student:** Bc. Dominik Berčík

**ID:** 203194

**Ročník:** 2

**Akademický rok:** 2021/22

**NÁZEV TÉMATU:**

## Návrh a realizace systému pro výpůjčku zkušebních zařízení

### POKyny PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a vytvořit systém, který detekuje vybrané měřicí zařízení a přidělí jej v databázi k zaměstnanci, který zařízení vybral. Požadavek bude zadáván pomocí mobilní aplikace.

1. Analyzujte a konkretizujte požadavky zadavatele.
2. Na základě analýzy požadavků navrhnete koncepci řešení.
3. Navrhnete datový model databáze a způsob komunikace mezi komponentami.
4. Navržený systém realizujte a ověřte funkčnost a vyhodnoťte míru splnění požadavků zadání.

### DOPORUČENÁ LITERATURA:

Literatura podle požadavků zadavatele (ABB)

**Termín zadání:** 7.2.2022

**Termín odevzdání:** 18.5.2022

**Vedoucí práce:** doc. Ing. Petr Fiedler, Ph.D.

**doc. Ing. Petr Fiedler, Ph.D.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Cílem této diplomové práce je navrhnout a vytvořit systém, který vydává testovací zařízení ze skladu dle požadavků uživatele. Rovněž poskytuje službu pro navrácení zařízení na sklad a převod zařízení mezi uživateli. Systém dále umožňuje nahlížení do databáze zařízení, kde lze zjistit aktuální dostupnost a historii jednotlivých zařízení. Je ovládán pomocí mobilní aplikace, kterou má každý z uživatelů nainstalovanou ve svém mobilním telefonu. Potřebná data se ukládají do databáze, která je vyvinuta pomocí platformy Firebase. Ta poskytuje jednoduchou správu dat. Součástí práce je i podrobný rozbor zadání, teoretický popis již existujících výdejních zařízení, popis použitých softwarových nástrojů a vyhodnocení funkcionality systému.

## **Klíčová slova**

Výdejní automat, Webová aplikace, Mobilní aplikace, Android Studio, Flutter, Dart, Firebase

## **Abstract**

The aim of this thesis is to design and create a system that issues test equipment from the warehouse according to user requirements. It also provides a functionality for returning equipment to the warehouse and transferring the equipment between users. The system also allows you to view the device database, where you can find out the current availability and history of individual devices. It is controlled by a mobile application that each user has installed on their mobile phone. The necessary data is stored in a database that is developed using the Firebase platform. It provides easy data management. The work also includes a detailed analysis of the assignment, a theoretical description of existing dispensing equipment, a description of the software tools used and an evaluation of the system's functionality.

## **Keywords**

Vending machine, Web application, Mobile application, Android Studio, Flutter, Dart, Firebase

## **Bibliografická citace**

BERČÍK, Dominik. *Návrh a realizace systému pro výpůjčku zkušebních zařízení*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2022. 68 s., 2 s. příloh. Diplomová práce. Vedoucí práce: doc. Ing. Petr Fiedler, Ph.D.

# Prohlášení autora o původnosti díla

<b>Jméno a příjmení studenta:</b>	<i>Dominik Berčík</i>
<b>VUT ID studenta:</b>	<i>203194</i>
<b>Typ práce:</b>	<i>Diplomová práce</i>
<b>Akademický rok:</b>	<i>2021/22</i>
<b>Téma závěrečné práce:</b>	<i>Návrh a realizace systému pro výpůjčku zkušebních zařízení</i>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 17. května 2022

-----  
podpis autora

## **Poděkování**

Děkuji vedoucímu diplomové práce doc. Ing. Petru Fiedlerovi, Ph.D. za odborné konzultace, předání cenných rad a pomoc při formálním zpracování této diplomové práce. Rovněž bych rád poděkoval konzultantům z firmy ABB s.r.o. (Ing. Janu Palátovi, Ing. Jiřímu Holubovi, Ing. Michalu Picmausovi, Ing. Peteru Křištofovi) za odborné konzultace, předání užitečných myšlenek a nápadů a pomoc při realizaci diplomové práce. V neposlední řadě bych také rád poděkoval své manželce (Marii Berčíkové) za trpělivost a gramatickou opravu této práce.

V Brně dne: 17. května 2022

-----  
podpis autora

# Obsah

<b>SEZNAM OBRÁZKŮ .....</b>	<b>9</b>
<b>SEZNAM TABULEK.....</b>	<b>10</b>
<b>1. ÚVOD.....</b>	<b>11</b>
<b>2. ANALÝZA POŽADAVKŮ ZADAVATELE.....</b>	<b>12</b>
2.1 SHRNUÍ POŽADAVKŮ .....	14
<b>3. ROZBOR PODOBNÝCH ŘEŠENÍ.....</b>	<b>15</b>
3.1 VÝDEJ SKLADNÍKEM .....	15
3.2 KLASICKÉ VÝDEJNÍ AUTOMATY .....	15
3.3 VÝDEJNÍ BOXY .....	18
<b>4. NÁVRH ŘEŠENÍ .....</b>	<b>21</b>
4.1 MOBILNÍ APLIKACE OBECNĚ.....	23
4.1.1 <i>Nativní aplikace</i> .....	23
4.1.2 <i>Webová aplikace</i> .....	24
4.1.3 <i>Hybridní aplikace</i> .....	24
4.2 NÁVRH FUNKCIONALITY MOBILNÍ APLIKACE .....	25
4.2.1 <i>Přihlášení do mobilní aplikace</i> .....	25
4.2.2 <i>Menu mobilní aplikace</i> .....	26
4.2.3 <i>Vypůjčit zařízení</i> .....	26
4.2.4 <i>Vrátit zařízení</i> .....	26
4.2.5 <i>Rezervace</i> .....	27
4.2.6 <i>Zrušení rezervace</i> .....	27
4.2.7 <i>Předat zařízení</i> .....	27
4.2.8 <i>Další funkce mobilní aplikace</i> .....	28
4.3 WEBOVÁ APLIKACE OBECNĚ .....	28
4.3.1 <i>Databáze</i> .....	30
4.3.2 <i>Aplikační server</i> .....	31
4.3.3 <i>Webový server</i> .....	31
4.4 NÁVRH FUNKCIONALITY WEBOVÉ APLIKACE .....	32
4.4.1 <i>Menu Uživatel a Menu Administrátor webové aplikace</i> .....	33
4.4.2 <i>Moje výpůjčky</i> .....	34
4.4.3 <i>Náhled do databáze</i> .....	34
4.4.4 <i>Správa databáze</i> .....	34
<b>5. DATABÁZE SYSTÉMU .....</b>	<b>35</b>
5.1 FIREBASE .....	35
5.1.1 <i>Cloud Firestore</i> .....	36
5.1.2 <i>Firestore Authentication</i> .....	36
5.2 NASTAVENÍ DATABÁZE .....	37
5.2.1 <i>Nastavení cloudu Firestore</i> .....	38
5.2.2 <i>Nastavení nástroje Authentication</i> .....	40
<b>6. SOFTWARE.....</b>	<b>42</b>

6.1	ANDROID STUDIO.....	42
6.2	FLUTTER .....	44
6.2.1	<i>Jazyk Dart</i> .....	44
6.3	POPIS FUNKCIONALITY PROGRAMU .....	44
6.3.1	<i>Inicializace</i> .....	45
6.3.2	<i>Definice ovladačů</i> .....	45
6.3.3	<i>Přihlášení do mobilní aplikace</i> .....	46
6.3.4	<i>Menu mobilní aplikace</i> .....	47
6.3.5	<i>Výpůjčka</i> .....	48
6.3.6	<i>Vrácení</i> .....	49
6.3.7	<i>Převod</i> .....	50
6.3.8	<i>Volná zařízení</i> .....	52
6.3.9	<i>Historie výpůjček</i> .....	53
6.3.10	<i>Moje výpůjčky</i> .....	56
6.3.11	<i>Odhlášení</i> .....	56
<b>7.</b>	<b>ZHODNOCENÍ PŘÍNOSU SYSTÉMU .....</b>	<b>58</b>
7.1	VÝSLEDKY Z FIREBASE .....	58
7.2	PŘIPOMÍNKY ZAMĚSTNANCŮ .....	60
7.3	SROVNÁNÍ VÝSLEDKŮ S POŽADAVKY .....	61
<b>8.</b>	<b>ZÁVĚR.....</b>	<b>63</b>
	<b>LITERATURA.....</b>	<b>65</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>68</b>
	<b>PŘÍLOHA A - DATOVÝ MODEL .....</b>	<b>69</b>
	<b>PŘÍLOHA B - TABAULKA DAT Z FIREBASE .....</b>	<b>70</b>

# SEZNAM OBRÁZKŮ

3.1	Ukázka výdejních automatů od firmy Würth [4].....	16
3.2	Šnekovité dopravníky se spirálou [2] a výsuvnou stěnou [6].....	16
3.3	Pásový dopravník [7] .....	17
3.4	Tubový výdej [8].....	18
3.5	Ukázka komerčních výdejních boxů [9].....	18
3.6	Výdejní automat pro interní firemní výdej s šuplíky i boxy [10] .....	19
3.7	Automaty na ochranné pomůcky s prosklenými okénky [11] .....	19
4.1	Návrh funkcionality systému.....	21
4.2	Návrh struktury mobilní aplikace .....	25
4.3	Schéma výpůjčky zařízení.....	26
4.4	Schéma návratu zařízení.....	27
4.5	Schéma předání zařízení.....	28
4.6	Princip funkce dynamické stránky [13].....	29
4.7	Princip načítání dat z databáze pomocí dynamické stránky [13].....	30
4.8	Návrh struktury webové aplikace .....	33
5.1	Nástroje platformy Firebase [19] .....	35
5.2	Metody přihlašování – Firebase Authentication.....	36
5.3	Firestore projekt – Device Share System .....	37
5.4	Firestore projekt – Možnosti pro párování s operačními systémy .....	38
5.5	Firestore – kolekce „things“ .....	38
5.6	Příklad QR kódu.....	39
5.7	Nastavená pravidla v záložce Rules .....	40
5.8	Přidání uživatele do přihlašovací databáze.....	40
5.9	Náhled do databáze uživatelů.....	41
6.1	Ukázka funkcí Android studia [23].....	42
6.2	Ukázka testování programu pomocí Emulátoru [26] .....	43
6.3	Definice ovladačů v souboru pubspec.yaml .....	46
6.4	Obrazovka mobilní aplikace – Přihlášení .....	47
6.5	Obrazovka mobilní aplikace – Menu .....	48
6.6	Obrazovky mobilní aplikace – Výpůjčka .....	49
6.7	Obrazovky mobilní aplikace – Vrácení .....	50
6.8	Obrazovky mobilní aplikace – Převod .....	52
6.9	Obrazovka mobilní aplikace – Volná zařízení .....	53
6.10	Obrazovky mobilní aplikace – Filtrování – Všechny výpůjčky .....	53
6.11	Obrazovky mobilní aplikace – Filtrování – Všechna zařízení.....	54
6.12	Obrazovky mobilní aplikace – Filtrování – Všichni zaměstnanci .....	55
6.13	Obrazovky mobilní aplikace – Historie výpůjček – detail .....	55
6.14	Obrazovka mobilní aplikace – Moje výpůjčka – detail .....	56
6.15	Obrazovka mobilní aplikace – Odhlášení.....	57
7.1	Srovnání denního výčtu dat z databáze .....	58
7.2	Srovnání denního zápisu dat do databáze.....	59
7.3	Srovnání denního používání aplikace.....	60

## SEZNAM TABULEK

2.1	Shrnutí požadavků .....	14
2.2	Shrnutí požadovaných funkcí systému .....	14
7.1	Srovnání výsledků a požadavků ze zadání .....	61
7.2	Srovnání použitých a nepoužitých funkcí systému .....	62

# 1. ÚVOD

Cílem této diplomové práce je navrhnout a vytvořit systém, který detekuje vybrané měřicí zařízení a přidělí jej v databázi k zaměstnanci, který zařízení vybral. Diplomová práce vychází z poznatků semestrální práce [1]. Součástí práce je navrhnout demonstrační panel, který bude obsahovat prvky pro ovládání systému, elektronické zámky určené pro výdej a navrácení vypůjčeného zařízení a prvky pro komunikaci s databází. Do databáze se budou ukládat potřebné informace o převedení zodpovědnosti za vypůjčené zboží ze skladu na zaměstnance a případně předání zboží mezi zaměstnanci. Součástí diplomové práce je také vytvoření návrhu funkcionality a následně i realizace mobilní či webové aplikace. Webová aplikace má sloužit jako kontrolní prvek pro administrátora systému. Společně s mobilní aplikací má umožnit běžným uživatelům nahlížet do databáze vypůjčených zařízení, díky čemuž lze zjistit, jestli je požadované zařízení dostupné na skladišti či nikoliv.

Požadavek na vytvoření tohoto systému vzešel z firmy ABB z divize rozvaděčů, z pobočky sídlící v Brně, ulice Vídeňská, konkrétně z oddělení Programování ochrany. Systém má sloužit ke zpřehlednění dostupnosti testovacích zařízení na skladištích a zaručit rychlé a jednoduché dohledání těchto zařízení. Další úkol systému je donutit zaměstnance, kteří nedodržují pravidla sdílení testovacích zařízení, aby tak činili a nezdržovali svým sobeckým chováním jiné zaměstnance od práce.

## 2. ANALÝZA POŽADAVKŮ ZADAVATELE

Zadavatelem semestrální práce je firma ABB s.r.o. Konkrétní požadavek na zpracování tohoto tématu vzešel z oddělení Programátorů ochran, jež je v provozu na ulici Vídeňská v Brně. Tato pobočka patří do divize rozvaděčů a zabývá se právě výrobou rozvaděčů středního a vysokého napětí.

Úkolem programátorů z tohoto oddělení je vytvářet logiky pro ochranné terminály, ale také nastavovat měřicí funkce, zřizovat komunikace mezi jednotlivými terminály rozvodné sítě, nahrávat a testovat funkčnost programů a funkčnost ochranných funkcí. Zařízení, do kterých se programy nahrávají, se nacházejí na výrobní hale. Každý programátor, který jde nahrávat programy, si bere ze skladu vozík (pojízdný stůl), na který si položí svůj notebook, případně si s vozíkem vezme i testovací zařízení Freja 300. Testovací zařízení se využívá pro kontrolu správného nastavení ochranných funkcí. S vozíkem si programátoři mohou vzít i binární simulátory, případně převodníky pro testování logiky.

Vzhledem k velkému množství zařízení, která si zaměstnanec může ze skladu zapůjčit, a vzhledem k velkému počtu zaměstnanců, je třeba sledovat pohyb vypůjčených zařízení. Je třeba zamezit nekontrolovanému pohybu napříč různými halami. S tímto vědomím zadavatel požaduje, aby každý zaměstnanec zaznamenal v nově vytvořeném systému pracoviště (např. „Stará hala“ nebo „Nová hala“), na které si zařízení bere. Tímto by se zavázal k místu, kde bude vypůjčené zařízení používat. Toto opatření by mělo také zaručit kontrolovanější pohyb jednotlivých zařízení po halách a předcházet tím jejich ztrátě. Rovněž je zde snaha o zamezení volně se vyskytujících zařízení ve výrobních halách a k přiměnění zaměstnanců vracet vypůjčené vybavení zpět na sklad ihned po ukončení testování.

V systému bude možnost vyhledat informace o tom, kdo si jaké zařízení vypůjčil, aby si ten, kdo aktuálně zařízení pro testování potřebuje a již na něj žádné na skladě nezbylo, mohl domluvit převod zařízení. To znamená, že systém bude umožňovat převod již vypůjčeného zařízení z jedné osoby na druhou osobu.

Další funkcí, kterou bude systém nabízet, je rezervace zařízení. Ta je důležitá ve chvíli, kdy zákazník přijede na přejímku. Přejímkou se rozumí finální práce na rozvaděči. Přesněji řečeno jde o předvedení funkčnosti rozvaděče zákazníkovi. Na tuto ukázkou je potřeba mít vypůjčené testovací zařízení, které je velmi vhodné mít zamluveno dopředu, tedy rezervováno v systému. O tom, že zákazník přijede, se ví dopředu a je pro to pevně stanovené datum. Přejímky mají navíc mnohem větší prioritu oproti běžnému testování.

Požadovaný systém bude umožňovat identifikaci zaměstnance, a to pomocí čtečky karet nebo pomocí jiného identifikátoru, aby bylo jednoznačně určeno, kdo si zařízení vypůjčil. Výdejní zařízení bude obsahovat elektronické zámky, které budou sloužit jako zabezpečovací prvek proti nekontrolovanému odnášení zkušebních zařízení, či případnému odvážení pojízdných stolů. Zadavatel nepožaduje žádnou konkrétní řídicí

jednotku systému. Navrhuje vybrat vhodné PLC s moduly pro vstupy, výstupy a komunikační modul pro připojení k internetu tak, aby bylo možné data přijímat a odesílat do databáze. Pro jednoduché ovládání systému se přidá i HMI panel. V rámci návrhu výdejního zařízení je třeba vybrat všechny další potřebné komponenty podle zmíněných požadavků zadavatele. Jedním z rozhodujících faktorů výběru všech použitých komponent by měla být cena. Veškeré použité hardwarové prvky budou logicky uspořádány na demonstračním panelu, který bude sloužit pro testování správné funkcionality systému. Rovněž bude sloužit pro prezentaci systému vedení firmy.

Pro ovládací panel je dále třeba vytvořit i patřičné programy, jako je program pro PLC, nastavení vizualizace na HMI panelu, nastavení komunikační karty PLC a další. V rámci softwaru bude vytvořena databáze, která umožní propojování informací ze dvou datových tabulek do tabulky výpůjček. První datová tabulka bude sloužit pro zaměstnance a jejich identifikaci. Ve druhé datové tabulce budou uvedena všechna zařízení určená k výpůjčce. Dále by se mělo ošetřit navrácení správného přístroje, tedy identifikovat, které zařízení zaměstnanec vrací. Rovněž by mělo být navrženo, jak by výdejní zařízení řešilo případný dlouhodobý výpadek připojení k internetu, a tudíž i znemožnění přijímání informací z datového serveru.

Další přání zadavatele je zprostředkovat nahlížení do databáze a umožnit ovládání systému pomocí mobilní a webové aplikace. Měla by zde být rozdělena práva uživatelů na běžného uživatele a na administrátora. Administrátor by měl mít přístup ke správě databáze a možnosti editace veškerých tabulek. Kromě toho by měla tato aplikace mít možnost provozovat některé funkce systému jako je například rezervace zařízení. Proto je nutné vyřešit, jak bude výdejní zařízení (demonstrační panel) komunikovat s webovou či mobilní aplikací.

Po sestrojení všech potřebných komponent a po vytvoření potřebného softwaru bude systém testován přímo v provozu firmy. Nejdříve však bude třeba vysvětlit zaměstnancům, jak systém pracuje, jak jej mají používat a jak jim usnadní práci. Systém se po zavedení do provozu bude dva týdny testovat. Poté budou zaměstnanci požádáni o předání zpětné vazby. Z výsledků bude vyhodnoceno, jak je systém přínosný, kde jsou jeho úskalí a jestli je třeba něco vylepšit pro další používání.

Zadavatel vyžaduje, aby tato diplomová práce obsahovala i rozbor podobných již existujících řešení, která mají sloužit pro inspiraci při tvorbě systému. Je zde snaha poučit se z chyb a omylů, které při tvorbě těchto systémů vznikaly a naopak zužitkovat dobré myšlenky popisovaných systémů. V dalším kroku je třeba popsat teorii tvorby webových a mobilních aplikací pro uvědomění si všech souvislostí. Důležité je také vybrat vhodné programovací jazyky pro dané aplikace, aby jejich tvorba byla co nejjednodušší a nejrychlejší a vedla ke zdárnému zprovoznění systému.

## 2.1 Shrnutí požadavků

Analýza požadavků od zadavatele je velmi komplexní a může být nepřehledná. Pro lepší pochopení slouží následující tabulka, kde je bodově popsáno, co zadavatel očekává od vytvářeného systému a naznačuje přibližný postup řešení diplomové práce.

Tabulka 2.1 Shrnutí požadavků

<b>Číslo</b>	<b>Požadavek zadavatele</b>
1.	<i>Nalezení podobných řešení problému</i>
2.	<i>Zamykání zařízení na skladu pomocí elektronických zámků</i>
3.	<i>Identifikace zaměstnance</i>
4.	<i>Navržení a sestrojení ovládacího zařízení (PLC + vhodné komponenty)</i>
5.	<i>Navržení a sestrojení demonstračního panelu pro testování a prezentaci</i>
6.	<i>Návrh řešení požadovaných funkcí</i>
7.	<i>Vytvoření softwaru pro demonstrační panel (ovládací panel)</i>
8.	<i>Ukládání dat do databáze</i>
9.	<i>Vyřešit možnost výpadku přístupu na internet</i>
10.	<i>Popis webových a mobilních aplikací</i>
11.	<i>Nahlížení do databáze a ovládání systému pomocí mobilní a webové aplikace</i>
12.	<i>Zavedení systému do provozu ve firmě a testování funkčnosti</i>

Pro úplnost je rozbor zadání doplněn i o tabulku požadovaných systémových funkcí, které budou moci zaměstnanci provádět.

Tabulka 2.2 Shrnutí požadovaných funkcí systému

<b>Číslo</b>	<b>Požadovaná funkce systému</b>
1.	<i>Výpůjčka zařízení</i>
2.	<i>Návrat zařízení</i>
3.	<i>Nahlížení do databáze systému</i>
4.	<i>Převod zařízení na jiného zaměstnance</i>
5.	<i>Rezervace zařízení</i>

## **3. ROZBOR PODOBNÝCH ŘEŠENÍ**

Výdej drobného zboží nebo vybavení pro zaměstnance bývá jedna z prvních věcí, do kterých firma investuje. Firmy postupně přecházejí od výdeje zařízení pomocí zaměstnaného skladníka k výdejním automatům, které mohou pracovat nepřetržitě. Aktuálně je trh nasycen možnostmi nákupu nejrůznějších výdejních automatů a jim podobným technologiím. Některé z nich jsou popsány dále v textu.

### **3.1 Výdej skladníkem**

Jedním z nejjednodušších výdejních systémů je člověk, jenž sedí ve skladišti a podle papírového seznamu vydává požadované věci. Jako příklad lze uvést paní v kolejní prádelně, která vydává čisté povlečení na základě identifikace studenta ISIC kartou. Vezme od něj použité povlečení a předá mu čisté, za podmínky, že uplynul minimální počet dní od poslední výměny, který je předem stanoven. Podobně v dřívějších dobách fungoval i výdej potřebného pracovního materiálu a nářadí, např. obsluha vrtáku se zlomil nebo otupil vrták. Obsluha si tedy musela jít za skladníkem pro vrták nový, kde si skladník zapsal, co za předměty kolegovi vydává, a také kolik mu jich předal. Tento systém je velmi neefektivní hlavně proto, že hlavní náplní práce skladníka je čekat, až za ním někdo přijde a v tu chvíli zaměstnavatel platí skladníka, i když nevykonává žádnou produktivní činnost po většinu pracovní doby.

Dalším důvodem, proč se pro tuto práci hodí více stroj než člověk, je fakt, že stroj nevydá uživateli víc, než je povoleno. Nic neodcizí. Veškeré vydané zboží stroj eviduje a může ukládat na servery nebo do skladových databází, což zajišťuje možnost dohledat vydané zboží a konkrétní osoby, pokud by se něco ztratilo.

### **3.2 Klasické výdejní automaty**

Klasické výdejní automaty [2], [3], [4] určené pro výdej zboží ve firmách fungují na následujícím principu. Zaměstnanec přijde k automatu a přihlásí se k zařízení pomocí své karty zaměstnance nebo identifikačního čísla nebo jiným předem definovaným způsobem. Automat si ověří, že tomuto pracovníkovi může vydat zboží. Po kladném ověření umožní pracovníkovi vybrat zboží, které požaduje. Když je zboží zadáno a potvrzen výběr, automat vydá požadované zboží. Odemkne zámek od výdejních dveří a pracovník si vezme požadovaný předmět. Následně se zámek výdejních dveří opět uzamkne. Samotné vysunutí zboží z automatu funguje na několika typických principech.



Obrázek 3.1 Ukázka výdejních automatů od firmy Würth [4]

Prvním principem je výdej pomocí šnekovitého dopravníku [5]. Tato technologie u výdejních automatů funguje pomocí kovové spirály [6], ve které je umístěno zboží. Spirála je otáčena pomocí servomotoru. Jejím otočením o jednu otáčku vysune právě jeden kus zboží do výdejní části automatu a zboží v zadní části se tak posune o jednu pozici dopředu.

Využití šnekovitého dopravníku [5] může být různé. Nejzákladnější použití je vkládání zboží přímo do šneku. Mezery spirály jsou pak přizpůsobeny velikosti vydávaného zboží. Další možností je umístění šnekovitého dopravníku do dolní části v jednotlivých výdejních buňkách. K výdeji zboží dochází pomocí výsuvné stěny [6], která se vysunuje od zadní části do přední pomocí svého upevnění na spirále.



Obrázek 3.2 Šnekovité dopravníky se spirálou [2] a výsuvnou stěnou [6]

Vybrané zboží po vysunutí většinou padá volným pádem do prostoru výdejních dveří, kde si zboží uživatel vyzvedá. Pokud ale zboží není schopno vydržet volný pád a došlo by tak k jeho znehodnocení nebo zničení, využívá se pro přesun od dopravníku k výdejním dveřím výtah [3]. Ten pomocí zadaných souřadnic přijede k buňce, která

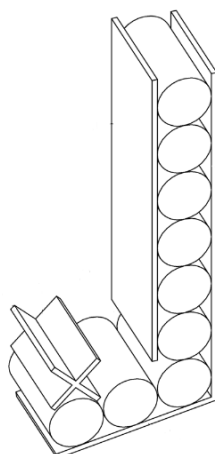
obsahuje požadované zboží a čeká, až se zboží dopraví do výtahu. Následně pak náklad dopraví ke dveřím. Obvykle bývá poháněn servomotory. Pohybuje se nejprve směrem nahoru (osa y) a následně do boku (osa x).

Další možností pro vysunutí zboží ze zásobníku jsou dopravní pásy [5], [7]. Tento princip funguje prakticky stejně jako šnekovitý dopravník, jen místo spirály (šneka) využívá posuvné pásy. Na ně pak mohou být připevněny profily pro oddělení jednotlivých prvků, čili pro oddělení jednotlivého výdejního zboží. Často využívaný způsob s pásovým dopravníkem je, že na konci pásu je umístěna stěna, které zabraňuje zaseknutí zboží a informuje uživatele o nepřítomnosti zboží.



Obrázek 3.3 Pásový dopravník [7]

Pro předměty kulatého tvaru, jako jsou například balené vody nebo zboží zabalené v tubách, se používají lopatkové vydavače [8]. Každým pootočením lopatky se vydá jedna tuba. Zásobník na tyto tuby je mírně nakloněný směrem k výdejním dveřím nebo jsou v něm tuby naskládány na sebe, čímž vyvíjí tlak na lopatku. Tuba nejbližší k výdejním dveřím je zadržována lopatkou, která je ovládána elektromagnetem. Pokud je požadavek dostat tubu ven, změní se pomocí elektromagnetu poloha bránícího tělesa, čímž se uvolní cesta k průchodu tuby k výdejním dveřím. Lopatka může být stejně tak otáčena a zadržována pomocí servomotoru.



Obrázek 3.4 Tubový výdej [8]

### 3.3 Výdejní boxy

Výdejní box [9], [10], [11] je samoobslužný automat, který otevře dvířka od úložného prostoru po zadání požadavku uživatelem a po potvrzení oprávnění. Poté vydá zboží, které bylo požadováno. V poslední době se staly výdejní boxy velmi populární, a to jak v odvětví komerčního obchodu, tak i ve firmách jakožto interní výdej potřebného vybavení pro práci zaměstnanců.

Popularita výdejních boxů je zapříčiněná pandemií Covid-19, kdy se zvedla poptávka po bezkontaktním převzetí nakoupeného zboží z internetových obchodů. Je zde rovněž velká výhoda toho, že zboží je dostupné 24 hodin denně, 7 dní v týdnu, stejně jako u klasických výdejních automatů [9].



Obrázek 3.5 Ukázka komerčních výdejních boxů [9]

V podnicích jsou výdejní boxy převážně určeny pro výpůjčku náradí [10], [11], např. vrtačku. Často se lze setkat i s výdejními regály, kde je mnoho malých šuplíků, které vydávají šroubky, matičky, podložky, hřebíčky a další drobný pracovní materiál. Dalším

obsahem těchto boxů bývají rozměrově větší ochranné pomůcky, pro příklad lze uvést svářečská maska, ochranné brýle, ochranná sluchátka a jiné.



Obrázek 3.6 Výdejní automat pro interní firemní výdej s šuplíky i boxy [10]

Princip funkce je téměř stejný jako u klasických výdejních automatů, jen namísto jednoho výdejního okna je k dispozici větší množství oken. Tato okna jsou blokována elektronickým zámek [10], [11]. Elektronický zámek povoluje řídicí jednotka dle možností a požadavků uživatele. Nejčastěji uživatel komunikuje s výdejním boxem pomocí dotykového displeje, který je umístěn na boxu. Nevýhodou takového displeje je jeho funkčnost na přímém slunci, kde jeho funkčnost nemusí být úplně optimální. Proto některé společnosti zabývající se výrobou výdejních boxů preferují komunikaci s boxem pomocí mobilní či webové aplikace v mobilním telefonu uživatele.

Jedna z hlavních výhod výdejního boxu je možnost jak vyzvednutí tak i vracení zboží [11]. V komerčním použití to znamená zjednodušení procesu reklamace zboží, ve výrobnách pak ke kontrolovatelnému sdílení pracovního nářadí. Při vypůjčce i navrácení se do systému uloží informace o tom, kdo a kdy si nářadí naposledy vypůjčil nebo jej vrátil zpět, čímž lze dohledat i to, kdo případně nástroj poškodil.

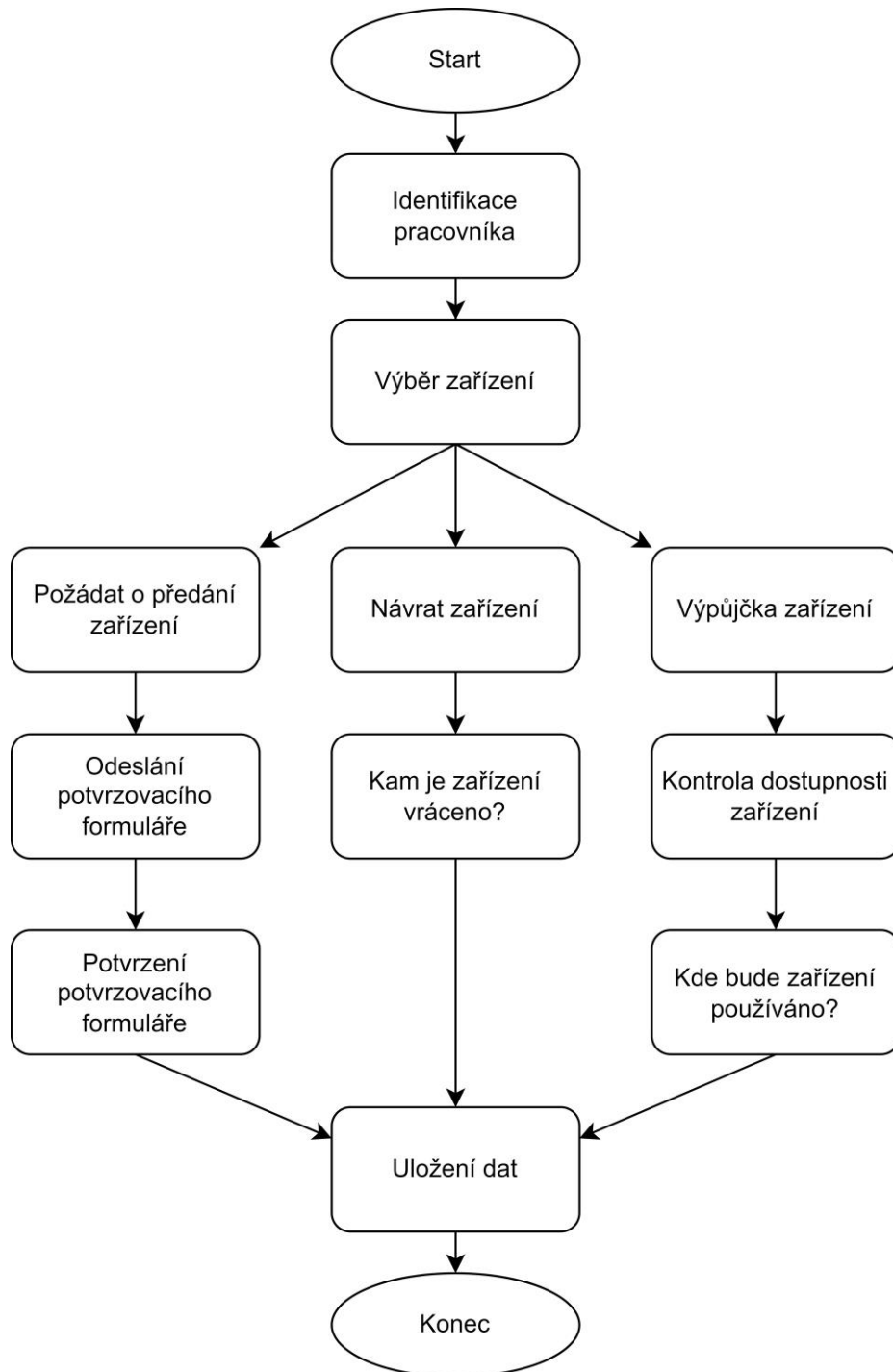


Obrázek 3.7 Automaty na ochranné pomůcky s prosklenými okénky [11]

Další z vlastností některých výdejních boxů je detekce přítomnosti zařízení [11]. Přítomnost se většinou určuje pomocí snímání hmotnosti obsahu jednotlivých boxů nebo pomocí optických snímačů. Dále některé automaty umožňují výdej zboží pomocí rotačních zásuvek, kdy automat pomocí motoru pootočí kolo se zásuvkami na pozici, která umožní uživateli vytáhnout požadovaný předmět.

## 4. NÁVRH ŘEŠENÍ

Před samotnou tvorbou výdejního systému je velmi důležité se zamyslet na tím, jak by měl systém fungovat a co vše by měl uživateli umožňovat. Návrh funkcionality je na následujícím obrázku.



Obrázek 4.1 Návrh funkcionality systému

System na začátku identifikuje pracovníka, který přichází na sklad s požadavkem na provedení změny počtu zařízení. Pracovník vybere požadované zařízení. Následně je na jeho rozhodnutí, co se má s vybraným zařízením stát. Jsou zde tři základní možnosti.

První možnost je výpůjčka zařízení. U této možnosti systém zkontroluje dostupnost zařízení, tedy jestli si někdo zařízení předem nerezervoval na pozdější dobu, případně zobrazí čas od kdy je zařízení rezervováno. Následně se systém zeptá, kde bude zařízení používáno. Pro firmu ABB na ulici Vídeňská jsou aktuálně dvě možnosti místa užití vybraného zařízení, a to „Stará hala“ a „Nová hala“. Po výběru uživatel potvrdí volbu. Systém uloží vše potřebné do databáze, tedy kdo si jaké zařízení a kdy vypůjčil. Následně odemkne zámek, kterým je zařízení zabezpečeno a umožní zaměstnanci zařízení využívat dle jeho potřeb.

Druhou možností je návrat již vypůjčeného zařízení. Zaměstnanec i zařízení bylo již identifikováno, tudíž nyní stačí jen vybrat možnost, na který ze dvou skladů je zařízení navraceno. Každá hala má totiž na pobočce firmy ABB na Vídeňské ulici svůj vlastní sklad. Může však nastat situace, kdy je třeba dané zařízení použít na jiné hale kvůli nedostupnosti na první hale. Díky tomu pak může zaměstnanec vrátit zařízení na jiný sklad. Pro zadavatele je důležité vědět, kde se zařízení nachází, i když zařízení není používáno, protože systém tímto usnadní jeho případné vyhledání. Na vybraném skladu pak vybere zaměstnanec číslo zámku, do kterého zařízení uložil. Po potvrzení místa uskladnění systém přeruší vazbu zařízení – zaměstnanec a uzamkne vybraný zámek. Avšak ve své historii si informaci o výpůjčce ponechá, a to kvůli případnému dohledávání informací při vyšetřování poškození nebo zničení zařízení.

Poslední ze základních možností je požádat o předání zařízení. Žadatel má dvě možnosti výběru zařízení, které má být předáno. V systému jsou viditelná všechna zařízení, tudíž zde stačí najít zařízení, jenž je požadováno a označit jej. Druhá možnost je zařízení vybrat naskenováním jeho QR kód, čímž jej identifikuje. Po výběru obdrží původní majitel upozornění, že jiný uživatel žádá o převod zařízení. Původní majitel převod potvrdí a v databázi dojde k převodu uživatelů. Pokud by volbu zamítl, převod se neuskuteční. Když je zařízení předáno, nový majitel musí ještě vyplnit, kde bude zařízení používat stejně jako při klasické výpůjčce. Na závěr se uloží potřebné informace do databáze.

Jedna ze situací, která by mohla nastat, je, že pracovník přijde do skladu a nenajde zde zařízení, které potřebuje nebo dokonce zde nebude vůbec žádné zařízení, případně tuto informaci zjistí pomocí náhledu do databáze vypůjčených zařízení. Tato situace bude řešena tak, že aplikace bude obsahovat skupinový chat. Do tohoto chatu bude možné napsat dotaz, jestli by někdo nemohl uvolnit své vypůjčené zařízení žadateli.

Během konzultací této práce se zadavatelem se dospělo k názoru, že vytvářet demonstrační panel není úplně nutné. Bylo rozhodnuto, že místo demonstračního panelu bude vytvořena mobilní aplikace, která bude funkce panelu nahrazovat. Při použití mobilní aplikace namísto demonstračního panelu by bylo komplikované zařízení

zabezpečit pomocí elektronických zámek. Od této funkcionality bude prozatím ustoupeno. Pro prvotní potřebu bude do zaměstnancům vložena důvěra k používání systému za předpokladu, že pokud by nastal problém a uzamykání zařízení by bylo vyloženě vyžadováno (kvůli nevracení zařízení nebo nezadávání výpůjčky), bude v budoucnu systém o tyto zámky obohacen. Díky tomu, že navrhovaný systém bude zaměřen hlavně na software, se ušetří hodně času návrhem demonstračního panelu a jeho následnou tvorbou. Ušetřený čas pomůže při vývoji softwaru a také k doladění chyb, které by zde mohly vznikat.

Výhoda mobilní aplikace je taková, že ji má každý uživatel systému ve svém telefonu, tudíž k ní má neustále přístup a může kdykoliv provádět potřebné úpravy. Hlavní výhodou je však to, že díky fotoaparátu na mobilním telefonu lze jednoduše skenovat QR kód. Samolepka s originálním QR kódem bude umístěna na všech zařízeních přidaných do systému. Rovněž bude umístěna na dveřích skladu pro jeho rychlou detekci. Mobilní aplikace bude propojena s databází webové aplikace a bude možné ji používat jen při připojení na internet. Firma zaručila, že síť Wi-Fi bude fungovat vždy, a pokud by se tak nestalo, každý zaměstnanec má možnost využití mobilních data na svém pracovním telefonu. Další výhodou mobilní aplikace je, že není potřeba nakupovat a vytvářet další hardware. Místo toho se využije již používaná technika, což sníží firmě náklady časové i finanční, kromě toho poskytne firmě možnost flexibilní úpravy algoritmu.

## 4.1 Mobilní aplikace obecně

Mobilní aplikace [12] je program určený k provozu na mobilních telefonech, tabletech či na chytrých hodinkách. Počáteční mobilní aplikace poskytovaly jednoduché, přímo účelné funkce, jako jsou čtení e-mailu, náhled do kalendáře nebo jednoduchý kalkulačtor. V dnešní době, kdy je mobilní telefon neodmyslitelnou součástí běžných lidí, jsou aplikace na velmi vysoké funkční, ale i bezpečnostní úrovni (např. mobilní internetové bankovníctví).

Mobilní aplikace se rozdělují na tři obecné kategorie, jejichž popis níže byl převzat z [12]:

- Nativní
- Webová
- Hybridní

### 4.1.1 Nativní aplikace

Tento typ mobilních aplikací je vyvíjen pro konkrétní operační systém, jako je např. Android nebo iOS. Z bezpečnostního hlediska jsou nativní aplikace vysoce chráněny proti zneužití díky různým vrstvám operačního systému. Kromě toho mají vlastní API (Application Programming Interface), což jsou již vytvořené a ověřené funkční celky, které programátor vkládá do programu. Jelikož jsou tyto aplikace vyvinuty na konkrétní operační systém, může uživatel aplikace přistupovat přímo k hardwaru zařízení, jako jsou

mikrofon, fotoaparát, GPS, atd. Díky vkládání těchto funkcí je u nativní aplikace menší chybovost v programech a je jednodušší naučit se na vývoji těchto aplikací pracovat.

Oproti jiným typům mobilních aplikací jsou nativní mobilní aplikace rychlejší, protože se nemusí spoléhat na prohlížeče, které by provoz aplikace zprostředkovaly. Rovněž je tento typ ideální pro práci bez připojení na internet, protože si vše ukládají do mezipaměti.

Nevýhodou nativních aplikací je individualita pro každý operační systém, což znamená tvorbu samostatných aplikací a jejich aktualizací. Tím roste doba a náklady vložené do vývoje.

#### **4.1.2 Webová aplikace**

Webové aplikace jsou přístupné pomocí internetového prohlížeče s hlavní funkcí přizpůsobit se libovolnému zařízení, na kterém má být aplikace prohlížena. Nejsou tedy vytvořeny pro konkrétní operační systém a ani se nestahují a neinstalují do mobilního zařízení.

Obecně jsou vytvářeny pomocí jazyku JavaScript, CSS a HTML5. Oproti běžným mobilním aplikacím jsou rychlejší. Aktualizace webové aplikace je výrazně jednodušší, stačí změnu publikovat jen jednou na webu, není třeba vyžadovat od uživatelů instalaci aktualizace. Další výhodou mobilních webových aplikací je jejich snadné vyhledání a lehké sdílení mezi uživateli.

Největší nevýhodou webové aplikace je to, že nebude fungovat bez připojení na internet, stejně jako prohlížeč, ve kterém se aplikace spouští. Také přístup k funkcím telefonu je pomalejší, jelikož dotazování se na funkci a následné zpracování, např. u fotoaparátu, probíhá na vzdáleném serveru.

#### **4.1.3 Hybridní aplikace**

Hybridní aplikace kombinují výhody mobilního webu a nativní aplikace. Jsou vytvářeny pomocí HTML, CSS, JavaScriptu běžícího na mobilním webu Webview, avšak stále mohou využívat funkce zařízení, jako jsou fotoaparát, určování polohy, svítidla a další.

Hybridní aplikace se podobají nativním aplikacím – nabízí stejné symboly, rychlý výkon nebo možnost pracovat i bez připojení k internetu. Ve skutečnosti ale pracují na principu webových aplikací.

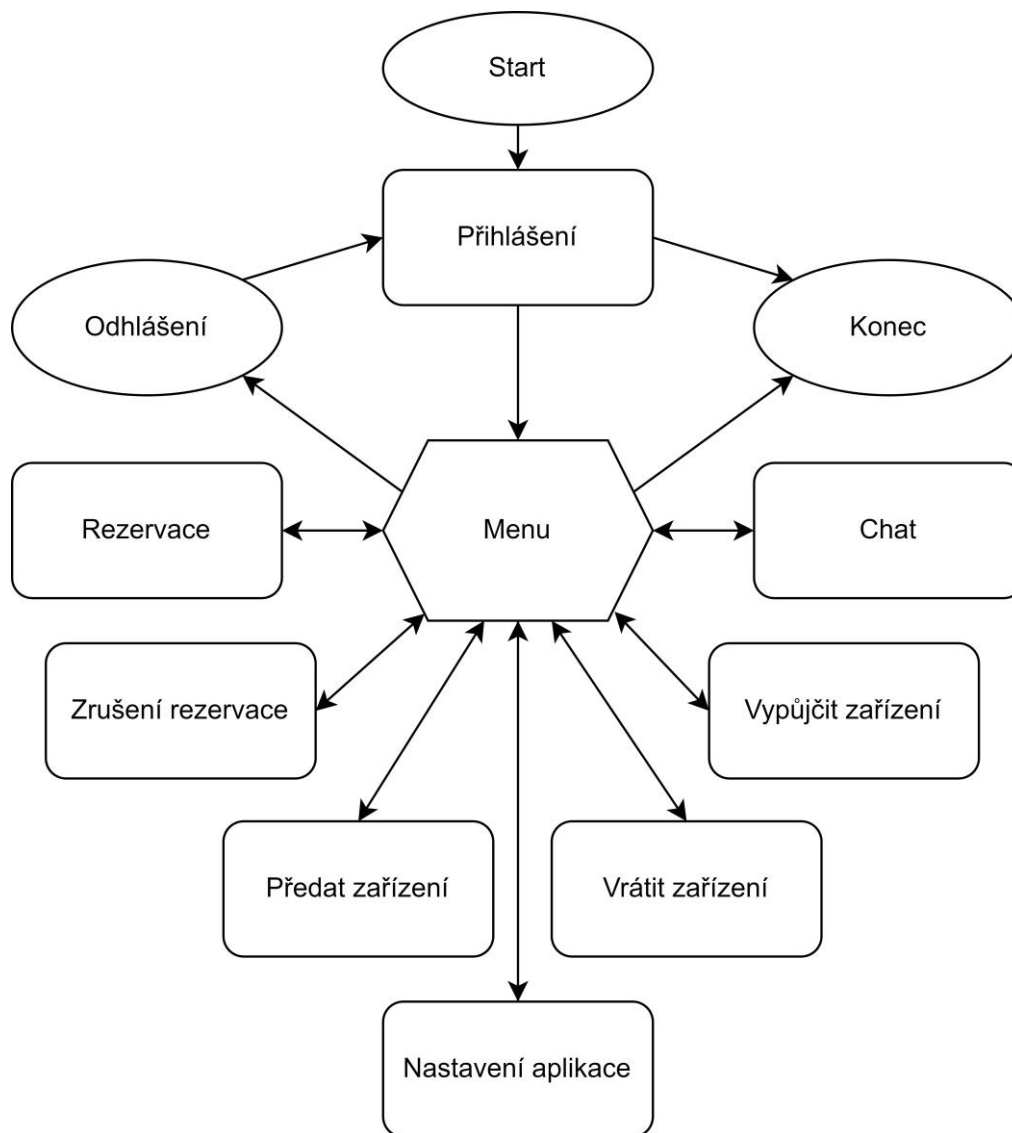
Mezi výhody těchto aplikací patří schopnost pracovat na více platformách. To znamená, že jedna kódová základna může fungovat jak v systému Android, tak i v systému iOS. Další výhody z toho vyplývající jsou rychlý vývoj, snadná údržba a cenová dostupnost.

Jednou z hlavních nevýhod je, že služby nabízené bez připojení k internetu jsou značně omezené. Omezení služeb může rovněž nastat při využívání pluginů, pomocí kterých aplikace přistupují k vestavěným funkcím zařízení (např. fotoaparát), a to pokud jsou pluginy zastaralé nebo nespolehlivé.

## 4.2 Návrh funkcionality mobilní aplikace

Mobilní aplikace má za úkol zprostředkovat uživateli možnost ovládání systému. Změny, které provádí uživatel, ukládá do databáze. Z databáze rovněž čerpá veškeré informace, proto je nutné při používání aplikace připojit mobilní zařízení k webové síti. Jedná se tedy o návrh hybridní mobilní aplikace.

Pro mobilní aplikace je navrženo několik funkcí, které má provádět. Jejich vzájemné propojení je ukázáno na následujícím obrázku.



Obrázek 4.2 Návrh struktury mobilní aplikace

### 4.2.1 Přihlášení do mobilní aplikace

Po spuštění aplikace bude uživatel vyzván k přihlášení do systému. Zde zadá svůj pracovní e-mail a heslo. Do databáze zaměstnanců bude uživatel přidán pomocí vedoucího oddělení, případně administrátora aplikace. Ostatní nebudou moci nové

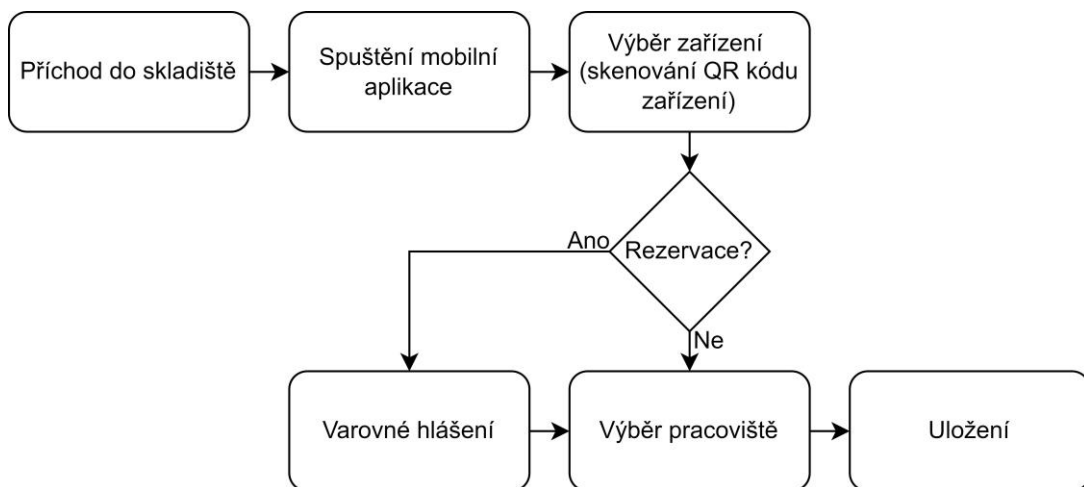
uživatele v systému registrovat. V nastavení aplikace pak bude moci uživatel změnit heslo, které mu na začátku bude vygenerováno. Z části „*Přihlášení*“ bude možné aplikaci ukončit („*Konec*“) nebo se přesunout do sekce „*Menu*“.

#### 4.2.2 Menu mobilní aplikace

Jak již název napovídá, sekce „*Menu*“ bude sloužit jako rozcestník mezi jednotlivými funkcemi aplikace. Po každé provedené akci se uživatel vždy vrátí zpět do „*Menu*“, kde bude moci zvolit, co se má provádět dál. Aplikaci bude možné odsud ukončit s tím, že uživatel zůstane přihlášený a při opětovném spuštění se nebude muset přihlašovat a bude tedy připuštěn zrovna do sekce „*Menu*“.

#### 4.2.3 Vypůjčit zařízení

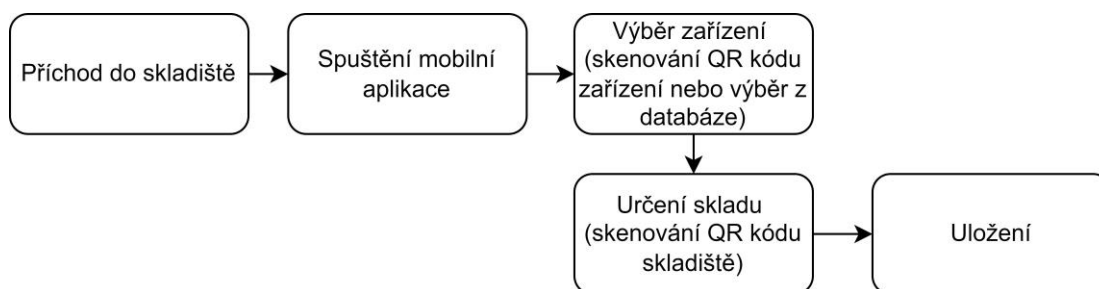
Po spuštění funkce „*Vypůjčení zařízení*“ bude uživatel vyzván, aby naskenoval QR kód zařízení, které si chce vypůjčit. Když aplikace zjistí, jaké zařízení je vybráno, zkontroluje, zda si zařízení nikdo nerezervoval. Pokud zjistí, že zařízení je rezervováno, zobrazí varovné hlášení, kde bude napsáno kdo a na kdy má zařízení rezervováno. Další pokyn, který zaměstnanec obdrží, bude zadat, kde bude zařízení využívat. Po potvrzení volby se uloží informace a v databázi se vytvoří vazba mezi zaměstnancem a přístrojem.



Obrázek 4.3 Schéma výpůjčky zařízení

#### 4.2.4 Vrátit zařízení

Při návratu zařízení musí uživatel naskenovat QR kód ze zařízení a následně bude vyzván, aby naskenoval QR kód skladu, do kterého chce zařízení uložit. Po potvrzení formuláře systém zruší vazbu mezi uživatelem a zařízením, avšak záznam o výpůjčce si ponechá ve své historii.



Obrázek 4.4 Schéma návratu zařízení

#### 4.2.5 Rezervace

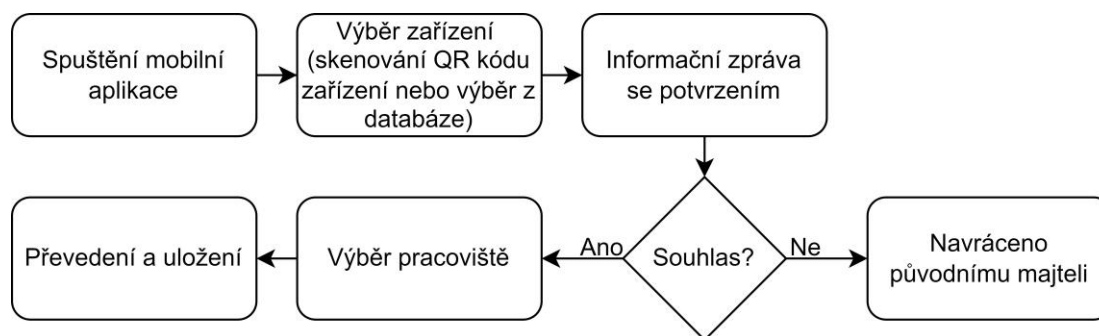
Po vybrání této funkce se zobrazí možnost naskenování QR kódu zařízení nebo možnost výběru zařízení z databáze pro určení, o jaké zařízení se jedná. Pokud zvolí možnost databáze, vypíše se mu na obrazovku zařízení s informacemi o aktuálním využívání a aktuálních rezervacích jednotlivých zařízení. Když má uživatel vybráno, provede potvrzení výběru a zadá, kdy bude zařízení potřebovat. Následně se informace uloží do databáze a 12 hodin před začátkem rezervace přijde uživateli e-mail s připomínkou rezervace. Pokud by si zařízení někdo vypůjčil dřív a nevrátil jej 24 hodin před začátkem rezervace, přijde mu též informační oznámení. Jestliže zařízení zůstane nevyzvednuté ve skladu 3 hodiny po začátku rezervace, systém ji automaticky ukončí.

#### 4.2.6 Zrušení rezervace

Ve funkci „Zrušení rezervace“ se uživateli vypíše z databáze všechna zařízení, která má rezervovaná, i s nastaveným časem vypůjčení pro lepší orientaci. Uživatel vybere zařízení, u kterého chce rezervaci zrušit. Opět zde bude možnost vybrat zařízení pomocí načtení QR kódu. Po potvrzení zrušení rezervace se v databázi smaže tento záznam.

#### 4.2.7 Předat zařízení

Požadavek na předání zařízení provádí ten, kdo chce zařízení používat. Pomocí databáze nebo naskenování QR kódu vybere požadované zařízení a požádá o předání zařízení. Původnímu majiteli přijde informační zpráva s požadavkem na potvrzení předání zařízení novému uživateli. Pokud potvrdí převzetí, nový majitel bude vyzván, aby určil pracoviště, kde bude se zařízením pracovat. Po potvrzení se v databázi změní vazba mezi zařízením a uživateli. V opačném případě, když původní majitel nepotvrdí předání, se změna v databázi neprovede. Pouze přijde žadateli oznámení o zamítnutí.



Obrázek 4.5 Schéma předání zařízení

#### 4.2.8 Další funkce mobilní aplikace

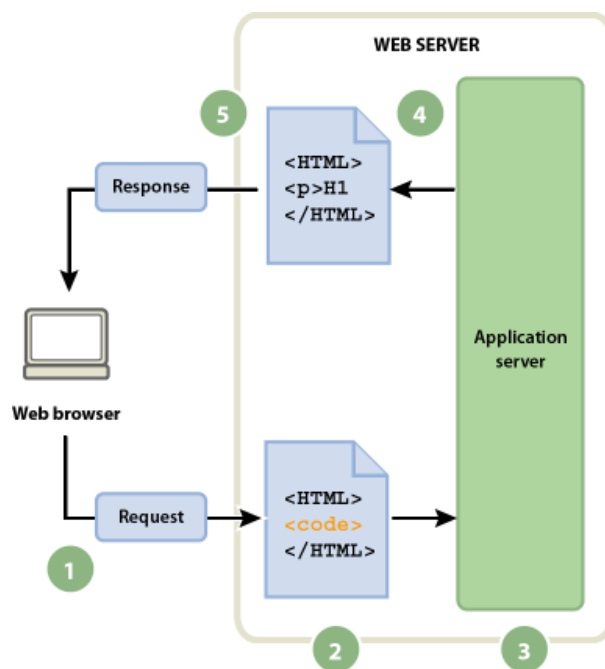
Mezi další funkce, které mobilní aplikace bude obsahovat, patří „*Odhlášení*“. Dále pak „*Nastavení aplikace*“, ve kterém si bude moci uživatel například změnit heslo nebo provádět jiná potřebná nastavení aplikace.

Poslední nepopsaná funkce je „*Chat*“. Tato funkce slouží pro posílání zpráv mezi zaměstnanci za účelem domluvy převodu zařízení na jinou osobu nebo pro rychlé nalezení osoby, která zařízení aktuálně nevyužívá. Funkcionalita by měla být podobná, jako u jiných existujících programů pro chatování. Vzhledem k tomu, že má tento chat sloužit pro domluvu ohledně toho, kdo má jaké zařízení napsané aktuálně na sebe, což není žádná tajná informace, nebude komunikace nijak šifrována.

### 4.3 Webová aplikace obecně

Webová aplikace [13] je softwarová služba poskytovaná uživateli internetového prohlížeče z webového serveru. Základním požadavkem je připojení na internet nebo k jeho vnitropodnikové podobě – intranetu. Oproti běžným webovým stránkám, které mají statický obsah, je webová aplikace [13] umístěna na webu bez úplného obsahu stránky. Konečný obsah webové aplikace se určí, až když návštěvník požádá o dynamickou stránku z webového serveru, protože návštěvník svými požadavky a provedenými akcemi ovlivňuje finální obsah aplikace. Dynamická stránka je proti statické stránce odlišná v tom, že obsahuje prázdná místa a sadu instrukcí, pomocí kterých se stránka má doplnit.

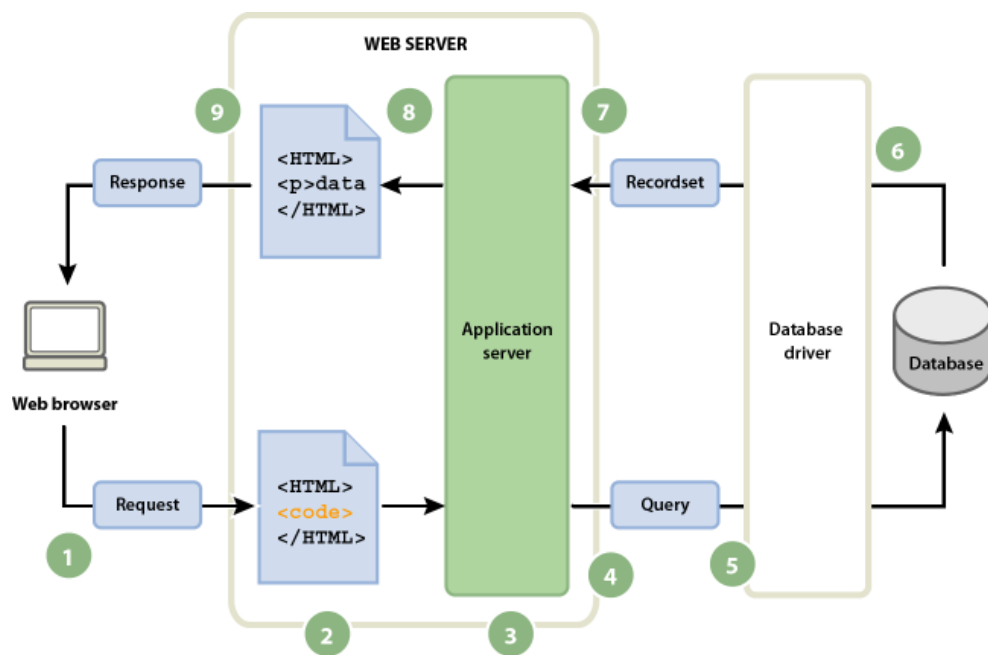
Funkcionalita statické stránky [13] je taková, že webový server přijme požadavek na statickou webovou stránku. Najde výsledek a pošle jej přímo do prohlížeče, který o něj požádal. Oproti tomu, při požadavku na dynamickou stránku [13], reaguje webový server tak, že předá stránku aplikačnímu serveru. Ten si přečte kód stránky a dokončí stránku podle instrukcí v kódu, který nakonec sám odstraní. Takto z dynamické stránky vznikne obyčejná statická stránka, která se přes webový server pošle návštěvníkovi do prohlížeče.



Obrázek 4.6 Princip funkce dynamické stránky [13]

Aplikační server [13] může čerpat data z databázi. Dynamická stránka pak může zadat příkaz aplikačnímu serveru s požadavkem na vložení dat z databáze. Využitím databáze je možné oddělit informace obsahu stránky od vizuálního návrhu webové stránky. Díky tomu stačí pro poskytnutí více informací vytvořit jen jeden vizuální návrh stránky. Nejrůznější změny a aktualizace se pak provádějí jen na jednom místě, není třeba ručně upravovat všechny stránky webu.

Jelikož software databáze má uzavřený kód [13], tak činí data nečitelnými. Není možné, aby databáze komunikovala s aplikačním serverem přímo. Pro tuto komunikaci se využívá systém řízení báze dat. Úkolem tohoto systému je, dle přání aplikačního serveru, vytvořit sadu záznamů, což je množina dat získaná z jedné nebo více databázových tabulek. Získanou sadu záznamů pak použije aplikační server k dokončení stránky.



Obrázek 4.7 Princip načítání dat z databáze pomocí dynamické stránky [13]

### 4.3.1 Databáze

Databáze [14] je organizovaná sbírka přesně strukturovaných informací nebo dat. Systém správy dat (DBMS – DataBase Management Systems) sloužící pro řízení databáze a samotná data jsou označována jako databázový systém. Data v databázích jsou obvykle řazena do řádků a sloupců a umísťována do tabulek tak, aby bylo zpracovávání a vyhledávání dat co nejefektivnější. Nejčastější jazyk, který databáze využívají, je SQL (Structured Query Language), neboli strukturovaný dotazovací jazyk.

Existuje velké množství databází. Mezi ně patří například [14], [15]:

- **Distribuované databáze** – data nejsou uložena na jednom místě, ale jsou rozdělena do paměti různých zařízení.
- **Relační databáze** – definují vztahy databází ve formě tabulek.
- **Objektově orientované databáze** – podporují ukládání všech datových typů ve formě objektů. Objekty obsahují atributy a metody, které říkají, co mají daná data dělat.
- **Centralizované databáze** – obsahují jedno místo, kam se všechna data ukládají. Ukládají se i aplikační postupy pro usnadnění vyhledávání dat z různých míst.
- **Databáze s otevřenými zdrojovými kódy** – uchovávají informace spojené s kódem, používají se hlavně v marketingu.
- **Cloudové databáze** – optimalizovány pro virtuální prostředí.
- **Datové sklady** – informační systémy obsahující historické a komutativní data. Slouží pro zjednodušení procesu analýzy organizace.

- **Osobní databáze** – pro ukládání dat v osobních počítačích. Jsou menší a jejich správa je jednoduchá. Data jsou přístupna jen malé skupině lidí.
- **Multimodální databáze** – podporují více datových modelů, které si definují, jak by měly být určité znalosti a informace v databázi organizovány.
- **Databáze dokumentů** – data jsou uchována ve sbírce dokumentů, obvykle pomocí formátu XML, JSON, BSON.
- **Hierarchické databáze** – využívají vztah „rodič-dítě“ při ukládání dat. Jejich struktura je jako strom s uzly, které reprezentují záznamy, a větve, reprezentující pole dat.

### 4.3.2 Aplikační server

Vrstva mezi operačním systémem a aplikacemi se nazývá aplikační server [16]. Poskytuje například přístup k souborům nebo ke správě procesů pro funkce enterprise aplikací. Pro zjednodušení psaní aplikací vytváří server vrstvu abstrakce. Jako příklad takových funkcí lze uvést podporu zpracování požadavku, výměnu zpráv mezi aplikacemi, perzistenci (neboli stálost) objektů a jiné. Zmíněná enterprise aplikace je běžná aplikace se zvýšenými nároky na spolehlivost, dostupnost, robustnost, výkonnost a typicky i na obsluhu velkého množství klientů.

Díky aplikačnímu serveru [16] je možné převádět a vytvářet různé typy jinak nekompatibilních komunikací a synchronizací. To znamená, že lze sjednocovat požadavky klientů na odlišných platformách tak, aby byla komunikace kompatibilní.

Požadavky klientů [16] aplikační server vyhodnocuje a směřuje do určených databázových struktur, kde získá potřebná data nebo data zde umístěná modifikuje dle požadavků klienta. Návrátová data jsou opět poslána klientským aplikacím.

Aplikační server [16] rovněž umožňuje realizovat jakékoli typy zabezpečení, jako je například určení přístupových práv uživatelům k jednotlivým prvkům databáze. Všechna práva a omezení, která jsou pro obsluhu databáze pomocí aplikačního serveru potřebná, si lze naprogramovat.

### 4.3.3 Webový server

Pojem webový server [17], [18] je možné chápat dvěma způsoby. První je program, který nabízí své služby klientům, což jsou vlastně požadavky z webového prohlížeče, které na webový server přicházejí pomocí komunikačních protokolů. Nejčastěji používaným protokolem pro přenos webových stránek je protokol http (HyperText Transfer Protocol). Druhý význam je hardwarový. Webový server je pak chápán jako počítač, jenž má na svém pevném disku uloženy jednotlivé statické webové stránky ve formě souborů. Tyto soubory pak posílá klientovi, dle jeho požadavků. I když existují dvě specifikace webového serveru, tak obvykle je tento pojem chápán oběma významy najednou.

Různé webové servery [18] se mohou od sebe lišit, avšak některé vlastnosti mají společné. Všechny webové servery vyžadují připojení k počítačové síti a dokážou

přijímat dotaz pomocí protokolu http. Dotaz je následně vyřízen a dotazovanému klientovi je vrácena odpověď nejčastěji formou HTML (HyperText Markup Language) dokumentu, případně jiného textového formátu či obrázku. Samotná odpověď serveru nejdříve obsahuje http protokol uveden hlavičkou se stavovým kódem. Až za ní následuje zmíněný dokument.

Stavový kód [17] přicházející jako odpověď ze serveru říká, že vznesený požadavek od klienta byl v pořádku vyřízen nebo že došlo k nějakým komplikacím. Stavový kód je trojciferné číslo. Mezi nejčastější stavové kódy patří kód 200, což znamená úspěšné vyřízení požadavku (OK) a kód 404, který říká, že během vyřizování požadavku nastala chyba (Not Found). Webový server tyto chyby často zaznamenává z důvodu optimalizace obsahu, způsobu uložení a prezentace požadovaných dat.

Jen zřídka nemá webový server [17], [18] možnost konfigurace. Tím je myšleno nastavení kořenového adresáře poskytovaného online a také spravování každého podadresáře jednotlivě. Např. v jaké časové zóně se nachází nebo jaké podporuje jazyky a přípony souborů. Dále pak může podporovat dynamický obsah. Pokud se tak děje, jsou součástí konfigurace i skriptovací jazyky, které tento požadavek zpracovávají. U sofistikovanějších webových serverů je možné se setkat s pokročilejším řízením požadavků nebo s různými zásuvnými moduly.

## 4.4 Návrh funkcionality webové aplikace

Další možností ovládání navrhovaného systému bude webová aplikace. Návrh funkcí je podobný jako u mobilní aplikace a tedy i zde bude možné provádět stejné změny v databázi. Tyto principiálně shodné funkce jsou „Rezervace“, „Zrušit rezervaci“, „Předat zařízení“, „Nastavení“ a „Chat“. Jsou popsány v kapitole 4.2.

Webová aplikace má sloužit hlavně jako kontrolní prvek pro vedení firmy. Pomocí této aplikace bude možné vkládat do databáze nové uživatele, kontrolovat správnost používání systému a odhalovat zaměstnance, kteří nedodržují pravidla pro vypůjčování zařízení. Struktura webové aplikace se tedy bude dělit na dvě části. První část je pro užívání běžným uživatelem a druhá část je určená pro administrátora, tedy pro správce systému a pro vedoucího příslušného oddělení. Kompletní strukturu webové aplikace ukazuje následující obrázek.



Obrázek 4.8 Návrh struktury webové aplikace

#### 4.4.1 Menu Uživatel a Menu Administrátor webové aplikace

Po zadání přihlašovacích údajů se uživatel dostane do „Menu“, které slouží opět jako rozcestník, kde si vybere, co v aplikaci bude provádět. Každý uživatel bude mít pomocí databáze určeno, jaká má práva. Pro vybrané uživatele bude „Menu Administrátor“, přes které bude možné přidávat nebo mazat ze systému uživatele nebo vypůjčované zařízení. V „Menu“ bude rovněž odkaz na odhlášení z aplikace.

#### 4.4.2 Moje výpůjčky

V této části budou vypsána zařízení, která má uživatel aktuálně na sebe převedena, a dále pak zařízení, jenž má uživatel zarezervována. Po výběru zařízení bude možné u rezervovaných zrušit rezervaci a u již vypůjčených zařízení bude možnost předat je jiné osobě stejně jako v mobilní aplikaci.

#### 4.4.3 Náhled do databáze

Funkce je určena pro běžného uživatele a slouží k vyhledávání zařízení v databázi a nalézání informací o tom, kdo má vypůjčené jaké zařízení. Pomocí tohoto náhledu uživatel zjistí, zdali je zařízení, které bude potřebovat, k dispozici a na jakém skladišti jej najde. Dále se může dozvědět, jestli je zařízení rezervováno, případně na kdy a rozhodnout se, zdali stihne vykonat svoji práci a vrátit zařízení před začátkem rezervace.

#### 4.4.4 Správa databáze

„*Správa databáze*“ obsahuje tři velmi důležité funkce webové aplikace. Je určena pouze pro administrátory. Právě administrativní databáze je důvod, proč je webová aplikace rozdělena na dvě úrovně uživatelů s různými právy.

První z funkcí „*Správy databáze*“ je „*Správa zařízení*“. Pomocí ní se do databáze dají přidat další zařízení nebo s její pomocí mazat záznamy o přístrojích. Stejně tak funguje funkce „*Správa uživatelů*“. Jediný rozdíl je, že místo zařízení spravuje informace o uživateli.

Třetí možností je zobrazit „*Všechny výpůjčky*“. To znamená, že se administrátor může podívat na kompletní historii jednotlivých zařízení i jednotlivých zaměstnanců. Náhled do historie se stane důležitým, když se bude hledat viník poškození zařízení.

## 5. DATABÁZE SYSTÉMU

Zadavatelská firma ukládá svá data na servery s podporou SQL databází. Při tvorbě systému byla snaha čerpat konkrétní data přímo z tabulek s daty zařízení a s daty zaměstnanců z firemní databáze. To však nebylo možné. Firma vysloveně zakazuje sdělovat citlivé údaje o svém majetku a zaměstnancích ve veřejném prostoru akademických prací, ale i jinde. Z tohoto důvodu nebylo možné data čerpat přímo z firemní databáze. Byla poskytnuta pouze vygenerovaná tabulka s nejnужnějšími daty pro chod systému. Přístup a připojení jakéhokoliv softwaru s možností editace dat firemní databáze byl zamítnut.

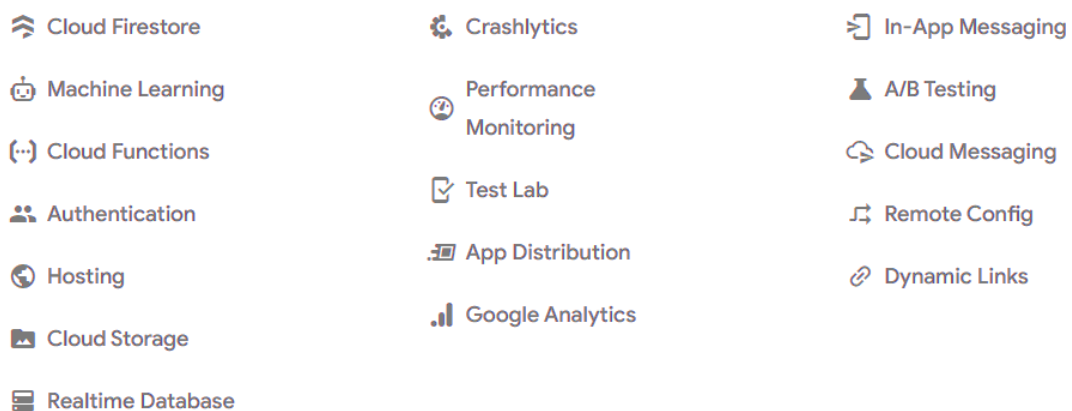
Z těchto důvodů se hledalo řešení, kde vytvořit potřebnou databázi. Po několika neúspěšných pokusech s SQL databázemi a jejich připojením v zabezpečené firemní síti bylo rozhodnuto, že pro prvotní vývoj systému bude použita databázová platforma Firebase, která je přímo určena pro uchovávání a práci s daty pomocí mobilních a webových aplikací.

V rámci návrhu databáze, byl vytvořen datový model. V tomto modelu je definována předpokládaný datový formát a také struktura dat. Datový model je umístěn v příloze A.

### 5.1 Firebase

Firestore [19], [20] je služba Backend-as-a-Service (BaaS) spravovaná společností Google, která umožňuje vývojářům soustředit se na frontendovou část (tvorba uživatelského prostředí a funkcí) svých aplikací a při tom využívat backendových služeb (serverové zpracování dat) bez jejich budování.

Databáze ve Firestore [19], [20] je kategorizován jako NoSQL („nerelační“) služba, která ukládá data do JSON (JavaScript Object Notation) dokumentů. Firestore poskytuje vývojářům celou řadu nástrojů a služeb, které pomáhají vyvíjet kvalitní aplikace a dokáží rozšířit jejich uživatelskou základnu. Přehled těchto nástrojů ukazuje následující obrázek.



Obrázek 5.1 Nástroje platformy Firebase [19]

Nástroje, které byly využity pro tvorbu zadaného systému, jsou Cloud Firestore a Authentication.

### 5.1.1 Cloud Firestore

Cloud Firestore [21] je flexibilní, škálovatelná databáze určená pro vývoj a provoz mobilních a webových aplikací. Udržuje data synchronizovaná napříč klientskými aplikacemi a nabízí i offline podporu pro mobilní aplikace a webové stránky, takže lze vytvářet aplikace, které fungují bez ohledu na latenci připojené sítě.

Podle datového modelu NoSQL Cloud Firestore [21] se ukládají data do dokumentů, které obsahují mapovaná pole s přiřazenými hodnotami. Tyto dokumenty jsou ukládány do kolekcí, které lze následně použít pro organizaci dat. Dokumenty podporují mnoho datových typů – od jednoduchých řetězců (String) a čísel (Int) až po složité vnořené objekty (map, array). Je zde možnost vytvořit i vnořené dokumenty a vytvářet tak hierarchické datové struktury.

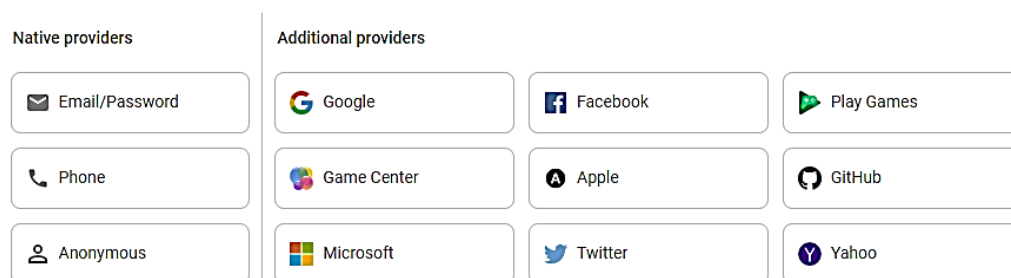
Další výhodou cloudu Firestore [21] je jeho rychlost, efektivita a flexibilita. Umožňuje získat datové dokumenty či vnořená data, aniž by bylo třeba načítat celou sbírku dokumentů do aktuálně používaného zařízení. Kromě toho cloud nabízí ve svém internetovém prohlížeči možnost filtrování a zobrazování požadovaných dat přímo v databázi, takže v mnoha případech není třeba vytvářet prostředí pro filtrování databáze a vše potřebné je pro správce snadno dohledatelné přímo na cloudu.

Mimo jiné je v cloudu Firestore [21] možnost nastavit bezpečnostní pravidla. Ta ve většině případů slouží pro nastavení práv uživatelů. Určeným uživatelům dovoluje číst, zapsat, mazat nebo aktualizovat data z cloudu.

### 5.1.2 Firebase Authentication

Mnoho aplikací potřebuje znát identitu uživatele, ať už pro určení jeho práv nebo pro záznam informace o tom, co provedl za změny v databázi. Pro tyto účely nabízí Firebase nástroj Authentication [22], který umožňuje přihlášení uživatele do aplikace. Podporuje ověřování identity pomocí hesla, telefonního čísla nebo populárních poskytovatelů identit, jako jsou Google, Facebook, Twitter a další. U jednoho uživatele je možné nastavit více přihlašovacích metod.

Get started with Firebase Auth by adding your first sign-in method



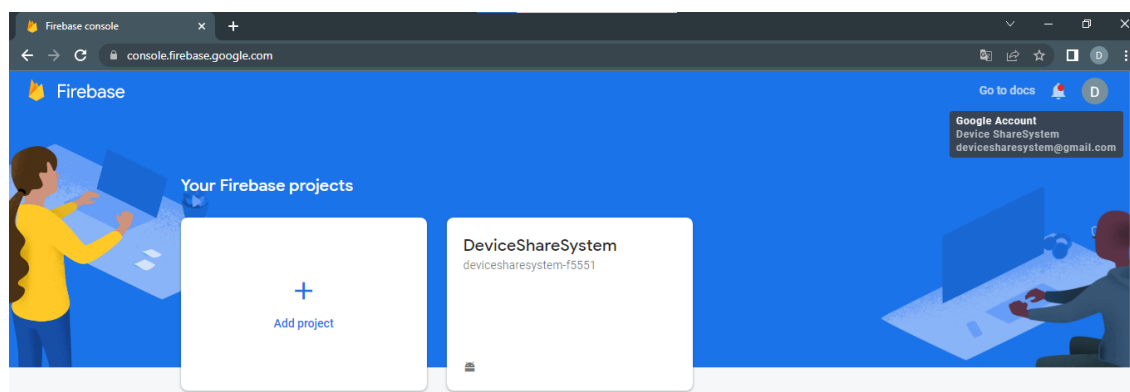
Obrázek 5.2 Metody přihlašování – Firebase Authentication

Firebase Authentication [22] umožňuje kompletní správu uživatelských účtů, např. vygenerovat nové heslo při zapomenutí starého, odesílat ověřovací e-maily nebo SMS zprávy, případně hromadné e-maily. V internetovém prohlížeči databáze si může správce prohlížet i minimální informace o uživateli, jako jsou uživatelské jméno, datum, kdy byl účet vytvořen a kdy se uživatel naposledy přihlásil. K informaci ohledně hesla se dostat nelze. Lze však zadat příkaz pro generování nového hesla, které se uživateli odešle na jeho e-mail nebo telefon dle použitého přihlašovacího systému.

## 5.2 Nastavení databáze

Pro používání potřebných nástrojů v platformě Firebase je třeba se přihlásit na Google účet pomocí g-mailu. Za tímto účelem byl vytvořen nový účet s názvem „*devicesharesystem@gmail.com*“. Uživatelské jméno odpovídá názvu systému, který zní „*Device Share System*“, zkráceně DSS, což v překladu znamená Systém sdílení zařízení. Tento účet bude sloužit pouze pro správu vytvářeného systému a bude k němu mít přístup pouze administrátor systému a vedoucí oddělení, pro které je tento systém vyvíjen.

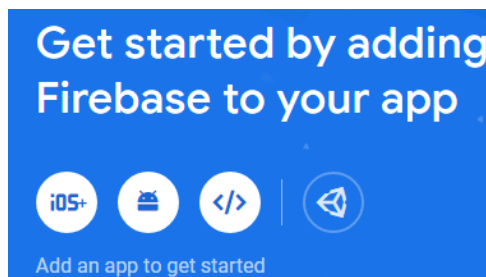
Pro přidání potřebných nástrojů je třeba na stránkách Firebase v záložce „console“ vytvořit nový projekt, ve kterém se nástroje nastaví. Opět zde byl vytvořen projekt s názvem „*Device Share System*“.



Obrázek 5.3 Firebase projekt – Device Share System

Po vytvoření nového projektu je třeba spárovat Firebase projekt s projektem vytvořeným v Android studiu. Jsou zde tři základní možnosti, kdy se určuje, z jakého operačního systému se bude do databáze přistupovat, a to buď Android nebo iOS. Může se taktéž určit přístup přes webový prohlížeč. Často se vytváří aplikace s přístupem jak pro Android, tak pro iOS, pro tyto účely slouží položka „*Unity*“, kdy se vytvoří přístup pro oba operační systémy současně. Všechny tři základní možnosti (bez položky „*Unity*“) lze kombinovat, tedy přidat do projektu všechny možnosti současně.

Velkou výhodou je, že se tyto přístupy mohou přidat i později. Proto pro aktuální testovací potřebu byl v tomto projektu vytvořen přístup pouze pro operační systém Android.



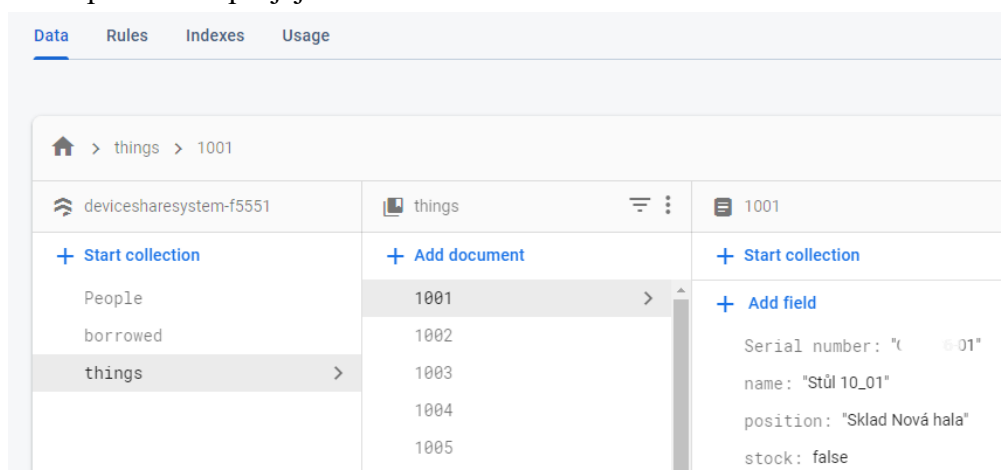
Obrázek 5.4 Firebase projekt – Možnosti pro párování s operačními systémy

Vytvoření kterékoliv z variant je spojeno s jednoznačným krokovým návodem, který Firebase vždy poskytne, a na jehož konci je výzva k uložení souboru na přesně určené místo v projektu Android studia. Pro každou možnost přístupu (operační systémy nebo web) se generuje jiný název souboru a ukládá se na jiné místo v projektu Android studia.

### 5.2.1 Nastavení cloudu Firestore

Po propojení Firebase s Android studiem je třeba vytvořit databázi pro ukládání dat. Při vytváření cloudu Firestore je třeba správně vybrat nejbližší server, na který se data mají ukládat. Pro DSS byl vybrán server „*eur3 (europe-west)*“ nacházející se ve Frankfurtu nad Mohanem, který byl vyhodnocen jako nejbližší možný. Druhý nejbližší server pro Českou republiku se nachází ve Varšavě.

Odsouhlasením se vytvoří prostředí pro ukládání dat, ve kterém lze jednoduše přidávat kolekce, ve kterých se následně vytvoří jednotlivé dokumenty označené buď automaticky vytvořeným ID, nebo ID definovaným uživatelem s požadovanými daty. Data uvnitř jednotlivých dokumentů nemusí nést stejný název, mohou být různá, avšak je třeba na to pamatovat při jejich filtrování.



Obrázek 5.5 Firestore – kolekce „things“

Data z kolekcí „*People*“ a „*things*“ byla převzata z vygenerované tabulky s daty od zadavatele. V kolekci „*People*“ je nastaveno ID dokumentu shodně s ID zaměstnance a každý z těchto dokumentů pak obsahuje údaje o zaměstnanci, jako jsou jméno, příjmení, id, e-mail a telefonní číslo. Všechny tyto informace jsou datového typu String.

Obdobně pak v kolekci „*things*“ jsou přidána data s informacemi o jednotlivých zařízeních. Opět je většina dat uložena v datovém typu String. Patří sem sériové číslo, jméno zařízení a aktuální pozice. Kromě toho obsahuje kolekce „*things*“ ještě boolean informaci o přítomnosti zařízení na skladě (true = přítomno, false = nepřítomno). ID jednotlivých dokumentů v databázi je pak tvořeno čtyřmi čísly, kde první dvě nebo tři číslice určují typ zařízení a další dvě nebo jedna číslice (označeny písmenem „x“) určují pořadové číslo zařízení dle poskytnutých dat.

Je zde několik základních typů zařízení:

- Malý stůl – označený 10xx
- Velký stůl – označený 11xx
- Tester Freja 300 – označený 300x
- Tester Freja 306 – označený 306x
- Tester Freja 406 – označený 406x
- Tester Freja 546 – označený 546x

Pro všechna takto zaznamenaná zařízení byl vytvořen unikátní QR kód, pomocí kterého je každé zařízení v mobilní aplikaci identifikováno. Každý QR kód je na patřičném zařízení nalepen. Samolepka s QR kódem obsahuje vždy i název zařízení, tak jak je zaznamenan v databázi.



**Freja 300\_1**

Obrázek 5.6 Příklad QR kódu

Poslední ještě nezmíněná kolekce v databázi se jmenuje „*borrowed*“ čili výpůjčky. Zde je ID dokumentu tvořeno automaticky, neboť data v těchto dokumentech jsou vytvářeny pomocí mobilní aplikace. Datovým typem String je zaznamenáno sériové číslo zařízení, název zařízení, jméno zaměstnance, ID zaměstnance, pracoviště a časový záznam, kdy bylo zařízení vypůjčeno a kdy bylo zařízení navráceno. Pomocí boolean proměnné „*active*“ se určuje, zdali je výpůjčka aktivní nebo neaktivní, tedy zda právě probíhá (true) nebo již byla ukončena (false). Pokud nedošlo k ukončení, je časový záznam s datem ukončení nahrazen řetězcem „*Nevráceno!*“.

Důležité nastavení, které je třeba provést, se nachází v záložce „*Rules*“, tedy v záložce pravidla. Zde se nastavují práva uživatelů. Pro jednoduchost bylo pro systém

DSS nastaveno, že všichni uživatelé mají možnost data číst i zapisovat. Toto nastavení není úplně správné, jelikož je nastaveno přes podmínku, která je vždycky pravdivá. Pro budoucí používání je tak třeba tuto podmínku upravit.

```
1 rules_version = '1';
2 service cloud.firestore {
3   match /databases/{database}/documents {
4     match /{document=**} {
5       allow read, write: if true;
6     }
7   }
8 }
```

Obrázek 5.7 Nastavená pravidla v záložce Rules

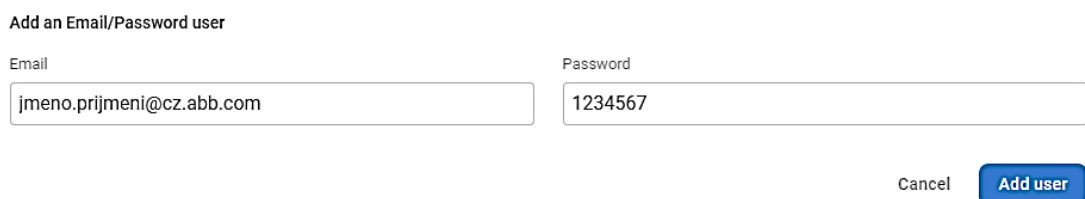
Jak je patrné, běžní uživatelé mohou provádět změny v aplikaci jen přes mobilní telefon, konkrétně jsou cíleny na kolekci výpůjček („*borrowed*“). Pro editaci uživatelů nebo zařízení je třeba mít přihlašovací údaje na systémový g-mail, ze kterého se správce dostane k editaci veškerých dat v databázi.

### 5.2.2 Nastavení nástroje Authentication

Pro přihlášení do mobilní aplikace pomocí nástroje Authentication je třeba ve Firebase uložit informace o uživateli a zadat typ přihlašovacích údajů, které budou použity. Možnosti typu přihlašovacích údajů jsou stejné jako na obrázku 5.2. Pro systém DSS byl vybrán způsob přihlášení pomocí e-mailu.

Všichni uživatelé, kteří mají systém používat, byli do této databáze přidáni. Zadávají se dvě informace. První z nich je pracovní e-mail uživatele, pomocí kterého se bude přihlašovat. Pracovní e-mail je ve tvaru „jméno.příjmeni@cz.abb.com“.

Druhý parametr je heslo uživatele. To bylo dle přání zadavatele nastaveno jako sedmimístné číslo odpovídající hodnotě ID jednotlivých zaměstnanců.



Add an Email/Password user


Email	Password
<input type="text" value="jmeno.prijmeni@cz.abb.com"/>	<input type="text" value="1234567"/>

Cancel

Obrázek 5.8 Přidání uživatele do přihlašovací databáze

Přihlašovací e-mail i heslo jsou voleny jednoduše, za účelem nejen zamezení zapomenutí přihlašovacích údajů, ale také kvůli tomu, že se pomocí zadaného ID načtou informace o uživateli z cloudu Firestore z kolekce „*People*“, se kterými pak mobilní aplikace dále pracuje.

Nástroj Authentication mj. umožňuje správci sledovat aktivitu jednotlivých zaměstnanců. U každého zaměstnance je uvedeno, kdy byl do systému přidán, ale také kdy naposledy se do systému přihlásil.

lukas...@cz.abb.com		Apr 21, 2022	Apr 28, 2022
rostislav...@cz.abb.com		Apr 21, 2022	May 2, 2022
jiri...@cz.abb.com		Apr 21, 2022	Apr 28, 2022
vladislav...@cz.abb.c...		Apr 21, 2022	Apr 22, 2022
fake.mail@cz.abb.com		Mar 17, 2022	May 2, 2022
dominik...@cz.abb.com		Mar 16, 2022	Apr 26, 2022

Obrázek 5.9 Náhled do databáze uživatelů

Jediný způsob, jak zadat nového zaměstnance nebo odstranit zaměstnance odcházejícího, je stejně jako u cloudu Firestore přes uživatelskou platformu Firebase, kam mají přístup pouze pověřené osoby.

## 6. SOFTWARE

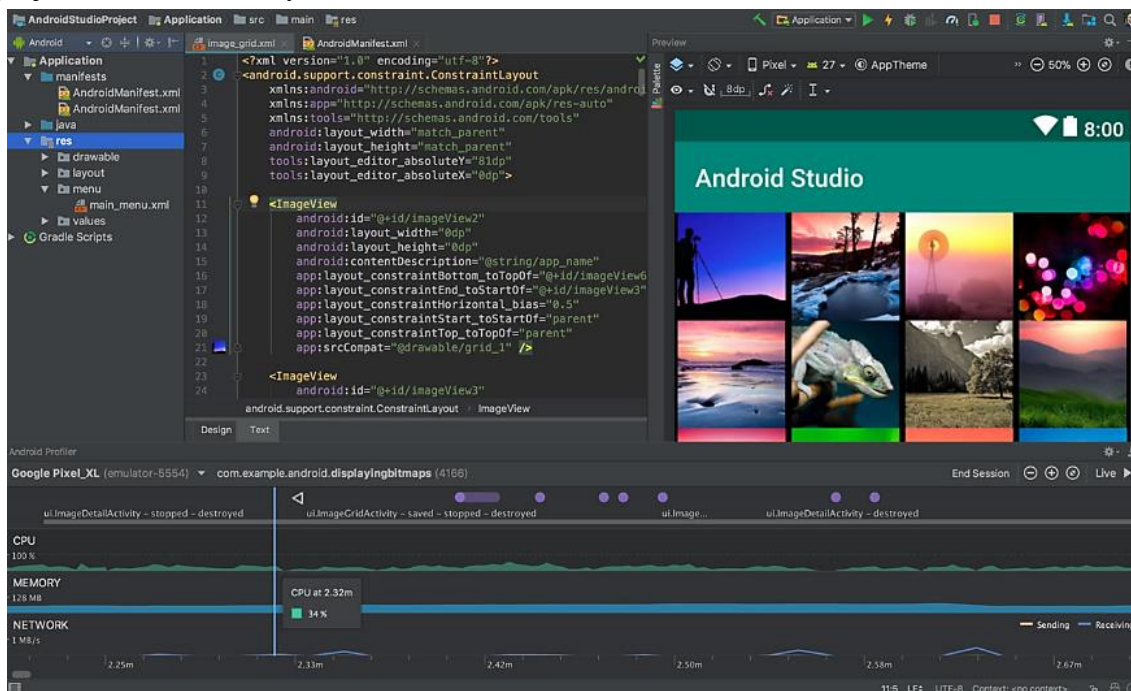
Jak vyplývá z předchozího textu, bylo domluveno, že se upustí od tvorby demonstračního panelu s elektronickými zámky pro uzamykání zařízení na skladišti. Pro první testování systému bude vytvořena pouze mobilní aplikace, která nahradí většinu plánovaných funkcí výdejního panelu.

Jako prostředí pro tvorbu mobilní aplikace byl zvolen program Android Studio s podpůrným softwarovým rámcem Flutter, který umožňuje jednoduchou tvorbu hybridních aplikací. Hybridní aplikace nejlépe splňuje požadavky zadání.

### 6.1 Android Studio

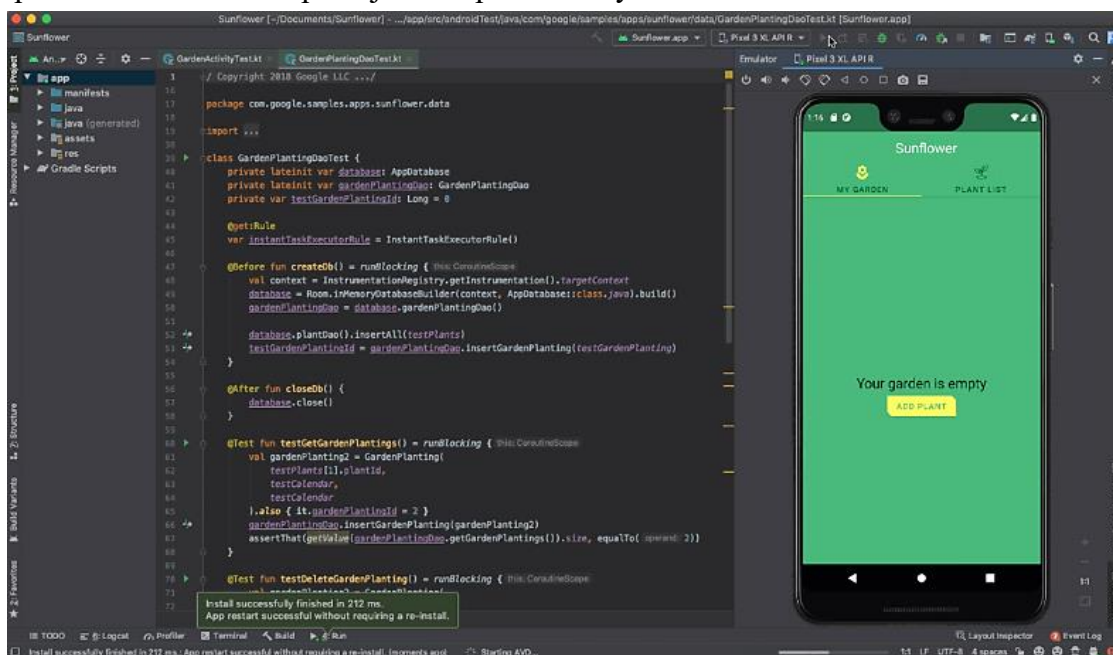
Android studio [23] je oficiální vývojové prostředí pro vývoj aplikací Android, ale umožňuje vytvářet aplikace i pro jiné operační systémy. Android studio je založeno na IntelliJ IDEA [24], což je komerční vývojové prostředí pro programování v jazyku Java, Groovy a dalších. Flexibilní sestavování systému je založeno na nástroji Gradle [25]. Tento nástroj je určen pro automatizaci sestavování programů vzniklých původně pro prostředí Java a staví na zkušenostech s nástroji Apache Ant, Apache Maven a programovacím jazykem Groovy.

Android studio [23] obsahuje nástroje pro sledování výkonu, kontrolu kompatibility verzí a rozsáhlé testovací nástroje a rámce. Dále pak umožňuje umístit kód na GitHub. Díky této vestavěné funkci je možné nejen zálohovat projekt, ale také importovat do projektu ukázkové kódy.



Obrázek 6.1 Ukázka funkcí Android studia [23]

Mezi základní funkce, které Android Studio nabízí, patří rychlý Emulátor [26], pomocí kterého si uživatel může prakticky okamžitě otestovat funkčnost své aplikace. Funkčnost testuje tak, že aplikaci nahraje do Emulátoru a po provedení jakékoliv změny v kódu a následného stisku tlačítka „Hot reload“ se v Emulátoru během několika málo sekund projeví změny provedené ve vytvářeném kódu. Emulátor nabízí všechny běžné funkce, které má mobilní telefon s operačním systémem Android, ale také si lze stáhnout a pracovat se zařízením pracující na operačním systému iOS.



Obrázek 6.2 Ukázka testování programu pomocí Emulátoru [26]

Další z funkcí Android Studia [23] je Editor rozložení, který umožňuje rychle vytvářet rozvržení obrazovky mobilního telefonu přetažením prvků uživatelského rozhraní do Editoru vizuálního designu namísto ručního psaní kódu. Editor umožňuje zobrazit rozvržení na různých zařízeních a verzích Androidu. Je možné dynamicky měnit velikost rozvržení pro kontrolu správné funkcionality na různých velikostech obrazovky.

Android Studio [23] nabízí volbu programovacích jazyků, a to buď jazyk Java, Kotlin, případně C++. Umožňuje zdarma vyvíjet mobilní aplikace na operačních systémech Windows, Mac, Linux a Chrome OS. Mimo jiné umožňuje přidat ke svým funkcím podpůrnou softwarovou strukturu (neboli „framework“), která přidává speciální funkce a rozšiřuje schopnosti mobilní aplikace. Mezi frameworky [27] patří např. PhoneGrapp, React Native, Flutter nebo Corona SDK. Každý z nich se snaží zaměřovat na speciální typ aplikací.

## 6.2 Flutter

Flutter [28], [29] je open-source uživatelské rozhraní od společnosti Google určené primárně pro vytváření nativně kompilovaných aplikací pro více platform z jediné kódové základny. To znamená, že jeden zdrojový kód je spustitelný na více operačních systémech. Umožňuje však vytvářet i aplikace webové či hybridní. Skládá se ze dvou základních částí [29]:

- SDK (software Development Kit), což je kolekce nástrojů, která kompiluje kód do nativního strojového kódu (kód pro iOS a Android),
- UI knihovny, která je založena na widgetech. Jedná se o kolekci opakovaně použitelných prvků uživatelského rozhraní. Patří sem např. tlačítka nebo textové pole, které lze přizpůsobovat pro vlastní potřeby.

Programování v tomto rozhraní probíhá pomocí jazyka Dart, u kterého je značná podobnost s jazykem C++, ze kterého vychází.

### 6.2.1 Jazyk Dart

Dart [30] je obecný objektově orientovaný open-source programovací jazyk se syntaxí ve stylu jazyka C vyvinutý společností Google v roce 2011. Důvodem vzniku tohoto jazyka bylo vytvořit frontendové uživatelské rozhraní pro web a pro mobilní aplikace. Neustále probíhá práce na jeho vývoji a vylepšování. Je kompilován do nativního strojového kódu pro vytváření mobilních aplikací. Byl inspirován jinými programovacími jazyky, jako jsou Java, JavaScript, C#, C++ a další. Jazyk Dart je silně typován.

Podporuje většinu běžných konceptů programovacích jazyků, jako jsou třídy a funkce. Jazyk Dart [30] přímo nepodporuje pole. Místo toho podporuje kolekci, která se používá k replikaci datových struktur, jako jsou obecné datové struktury, pole, či „*optional typing*“.

## 6.3 Popis funkcionality programu

Během konzultací se zadavatelem se dospělo k názoru, že vytvářená mobilní aplikace nemusí obsahovat všechny navržené funkce, ba dokonce, že jsou některé zbytečné. Rovněž bylo domluveno, že není třeba používat všechny navržené prvky jednotlivých funkcí, například ve funkci pro převod zařízení není třeba souhlas původního majitele. Mezi zbytečné funkce se řadí „*Chat*“ a „*Nastavení aplikace*“.

Funkce „*Chatu*“ je nahrazena jednoduchou možností vytočení telefonního čísla patřičného zaměstnance přímo v aplikaci. Telefonní rozhovor je rychlejší než čekání na zprávy z chatu. Stejně tak pro zjednodušení není vytvořena funkce „*Nastavení aplikace*“. Přišlo se na to, že kromě tmavého/světlého režimu aplikace by aplikace žádné další možné nastavení neobsahovala. Uživatelské heslo je nastaveno pro jednoduchost jako ID zaměstnance a se změnami se nepočítá. Proto byly obě funkce vypuštěny.

Další funkce, které nebyly po konzultaci vytvořeny, jsou „Rezervace“ a „Zrušení rezervace“. Od nich se upustilo, jelikož vytvářený systém neobsahuje elektronické zámky, které by zaručily, že požadované zařízení zůstane na skladišti. Navíc u zadavatele panovala obava, že by někteří zaměstnanci měli jedno zařízení neustále rezervováno, čímž by bylo znemožněno jeho sdílení.

Program pro mobilní aplikaci byl vytvořen v Android studiu s využitím uživatelského prostředí Flutter a je napsán v jazyce Dart. Obsahuje funkční program pro mobilní aplikace běžící pod operačním systémem Android. Projekt je předpřipraven také pro mobilní aplikace se systémem iOS, avšak pro tento operační systém zde chybí vložený inicializační soubor pro Firebase, jelikož k přidání tohoto souboru je třeba mít vývojové prostředí spuštěno pod operačním systémem od společnosti Apple. Rovněž je projekt připraven na možné rozšíření o webovou aplikaci. Projekt obsahuje 15 zdrojových souborů (+ další vytvořené při tvorbě projektu automaticky) s více než 3000 řádky kódu. Výsledná mobilní aplikace byla pojmenována jako DSS, což znamená „*Device Share System*“ (Systém sdílení zařízení).

Vytvořená mobilní aplikace vyžaduje neustálé připojení k internetu. Bez připojení na internet není možné, aby aplikace fungovala, jelikož je provázána s vytvořenou databází a neustále využívá její služby. V následujícím textu je vysvětlena její funkcionální.

### 6.3.1 Inicializace

Inicializační soubor „*main.dart*“ se spouští jako první a jeho hlavním úkolem je inicializovat použití Flutteru a rovněž i připojení do databáze Firebase. Jedná se o soubor s nejkratším kódem. Funkce `main` spouští třídu `NavApp`, která pak přesměrovává program do obrazovky pro přihlášení uživatelů.

Další významný soubor „*globals.dart*“ je určený pouze pro inicializaci globálních proměnných. Je zde definováno 23 proměnných typu `String`, `bool` nebo `int`. Tyto proměnné slouží pro předávání hodnot mezi funkcemi a pro ukládání potřebných dat do databáze.

### 6.3.2 Definice ovladačů

Definice všech důležitých ovladačů, které má mobilní aplikace používat, se provádí v projektovém souboru „*pubspec.yaml*“. Rovněž se v tomto souboru definuje cesta, která umožňuje nalézt používané obrázky a také se zde inicializuje změna ikonky aplikace na mobilním zařízení.

```
cupertino_icons: ^1.0.2
firebase_core: ^1.14.1
cloud_firestore: ^3.1.12
firebase_auth: ^3.3.14
qr_code_scanner: ^0.7.0
html: ^0.15.0
url_launcher: ^6.0.20
flutter_email_sender: ^5.1.0
build_runner: ^2.1.8
provider: ^6.0.2
flutter_launcher_icons: ^0.9.2
```

Obrázek 6.3 Definice ovladačů v souboru pubspec.yaml

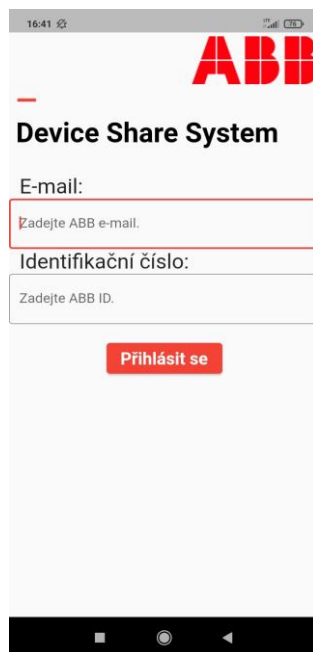
Ovladač „*flutter\_launcher\_icons*“ slouží pro přidání a vytvoření ikonek v aplikaci. Jedná se například o přidání šipky („*return*“) nebo obrázek domečku („*home*“). V rámci přípravy na zprovoznění aplikace na iOS je přidán i ovladač „*cupertino\_icons*“ spravující ikony právě pro systém iOS. „*Firebase\_core*“ slouží pro připojení databáze, další ovladače „*cloud\_firestore*“ a „*firebase\_auth*“ pak slouží pro přímé volání jednotlivých nástrojů z Firebase. Ovladač „*qr\_code\_scanner*“ umožňuje programu přístup k fotoaparátu a možnost číst snímaný čárový kód nebo QR kód. Ovladače „*html*“ a „*url\_launcher*“ jsou určeny pro sdílení odkazů. Využívají se při přesměrování aplikace k vytočení telefonního čísla zaměstnance. „*Flutter\_email\_sender*“ umožňuje mobilní aplikaci předpřipravít a odeslat e-mailovou zprávu. Další použitý ovladač je „*build\_runner*“, jenž slouží ke generování souboru. V tomto případě tedy k přidání a zobrazení obrázků uložených ve složce projektu „*image*“. Ovladač „*provider*“ vytváří rozcestník pro tvorbu menu aplikace.

### 6.3.3 Přihlášení do mobilní aplikace

První obraz, který uživatel po zapnutí aplikace uvidí, je výzva k přihlášení. Tato část kódu využívá ovladače „*firebase\_auth*“. Slouží k jednoznačné identifikaci uživatele pro možnosti budoucího záznamu.

Po stisku tlačítka „*Přihlásit se*“ jsou obě textová pole, která uživatel vyplnil e-mailem a identifikačním čísle, programem překontrolována. Nejdříve se zkontroluje, zdali jsou obě pole vyplněna. Pokud nejsou, objeví se červeným písmem výzva k vyplnění polí. Stejně tak nastane, když je špatně zadán formát e-mailu. Ověření, zdali uživatel existuje a zdali zadal správně heslo (identifikační číslo) se provádí pomocí funkcí zmíněného ovladače. Ten již komunikuje s databází a kontroluje zadané parametry. Pokud jsou parametry zadány špatně nebo je problém s připojením telefonu na internetovou síť, vypíše se varovné hlášení.

Po úspěšném přihlášení je uživatel přesměrován do „*Menu*“ a jeho ID se uloží do globální proměnné, pomocí které bude dále identifikován.



Obrázek 6.4 Obrazovka mobilní aplikace – Přihlášení

#### 6.3.4 Menu mobilní aplikace

Před zobrazením samotného „*Menu*“ program nejdříve vezme ID uživatele, který se přihlásil a vyhledá pomocí něj informace v cloudu Firestore, přesněji v kolekci „*People*“. Veškeré nalezené informace o uživateli se uloží do globálních proměnných. Jméno, příjmení a ID uživatele se pak vypíše v horní části obrazovky.

Rozcestník „*Menu*“ umožňuje spustit sedm různých operací. Po kliknutí na ikonu požadované akce, je uživatel přesměrován dále do programu, dle svého výběru. Všechny z nich jsou popsány níže v textu.

Ze všech vybraných operací má uživatel tři možnosti jak se vrátit zpět do „*Menu*“. První možností je plovoucí tlačítko „*return*“ umístěné vždy v pravém dolním rohu obrazovky, dále při zobrazení vyskakovacího okna tlačítkem zrušit nebo kliknutím do prostoru mimo vyskakovací okno.



Obrázek 6.5 Obrazovka mobilní aplikace – Menu

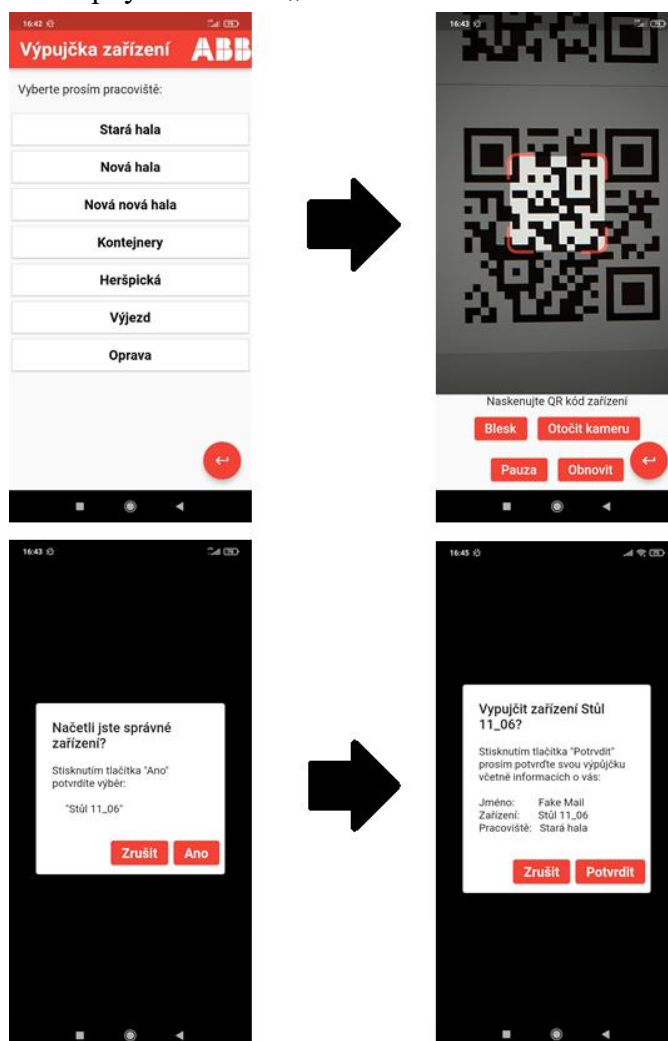
### 6.3.5 Výpůjčka

Při požadavku na výpůjčku je uživatel nejdříve vyzván, aby vybral místo, kde bude vypůjčené zařízení používat. Oproti návrhu je výběr rozšířen ze dvou na sedm možností (Stará hala, Nová hala, Nová nová hala, Kontejnery, Heršpická, Výjezd, Oprava). V nabídce jsou všechny aktuální možnosti, jež zaměstnanec má. Vybrané místo se uloží opět do globální proměnné. Následně program spustí funkci pro načítání QR kódu.

Na mobilní obrazovce je nyní vidět to, co snímá kamera. V dolní části telefonu jsou pak možnosti pro ovládání fotoaparátu. Je zde možnost zapnout blesk a posvítit si v tmavých místech na potřebný QR kód, změnit kameru, pokud telefon má více kamer, nebo přerušit a opět spustit funkci kamery. V horní části ovládacího panelu je pak výzva k akci, kterou má uživatel provést, tedy naskenovat QR kód zařízení, které si chce vypůjčit.

Při naskenování QR kódu se nejdříve ukáže uživateli vyskakovací okno s textem, který QR kód skrýval. Zde je třeba, aby uživatel potvrdil, že si naskenoval zařízení, které chtěl. Poznává to tak, že text ve vyskakovacím okně bude stejný, jako text napsaný na samolepce s QR kódem. Výběr může zrušit tlačítkem „Zrušit“ nebo potvrdit tlačítkem „Ano“. Je-li výběr potvrzen, ukáže se nové vyskakovací okno s informacemi o výpůjčce a s žádostí o potvrzení. Opět lze akci ukončit tlačítkem „Zrušit“ nebo odsouhlasit tlačítkem „Potvrdit“. Po potvrzení se ukáže poslední vyskakovací okno, které uživatele informuje o úspěšném provedení výpůjčky. Současně se uloží potřebná data do nově vzniklého dokumentu v cloudu Firestore v kolekci „borrowed“.

Dokončením výpůjčky je uživatel přesměrován zpět ke čtečce QR kódu. Nyní si může vypůjčit další zařízení, které si chce vzít na pracoviště vybrané při první výpůjčce nebo pomocí „return“ šipky se vrátí do „Menu“.



Obrázek 6.6 Obrazovky mobilní aplikace – Výpůjčka

### 6.3.6 Vrácení

Po skončení práce s vypůjčeným zařízením přichází zaměstnanec na jedno ze dvou skladišť a v „Menu“ vybírá položku „Vrácení“. Aplikace spouští funkci pro skenování QR kódu. Úkolem zaměstnance je nyní naskenovat QR kód skladu, na které přišel. Jsou jen dvě možnosti, kam lze zařízení vrátit (Sklad Stará hala nebo Sklad Nová hala).

Aplikace ověří, jestli naskenovaný kód odpovídá jedné z možností. Pokud ověření selže, ukáže se v mobilní aplikaci vyskakovací okno s varovným hlášením. Při načtení správné hodnoty je dál uživatel vyzván, aby naskenoval zařízení, které přišel vrátit. Pokud nyní naskenuje zařízení, které nemělo aktivovanou výpůjčku, vypíše se varovné hlášení a návrat nelze provést. Stejné hlášení se vypíše i za předpokladu, že uživatel naskenuje QR kód, který nepatří žádnému zařízení v databázi.

V opačném případě se ukáže vyskakovací okno s informacemi o návratu a při stisku tlačítka pro potvrzení se výpůjčka deaktivuje. V tomto okamžiku je zařízení vráceno, do databáze je zaznamenána časová značka návratu a výpůjčka je označena za neaktivní. V databázi však zůstává informace o provedení této výpůjčky. Pokud by nastala situace, že zařízení, které si půjčil první zaměstnanec, vrací druhý zaměstnanec, je v databázi výpůjček přepsána informace o tom, kdo zařízení využíval. Druhý zaměstnanec návratem převzal zodpovědnost za zařízení a je tedy na něm, aby zkontroloval, že zařízení bylo vráceno na sklad bez závad.

Zobrazením potvrzovacího okna je návrat zařízení dokončen a uživatel je přesměrován zpět na čtečku QR kódů, aby mohl případně vybrat další zařízení, které chce na daný sklad vrátit.



Obrázek 6.7 Obrazovky mobilní aplikace – Vracení

### 6.3.7 Převod

V situaci, kdy se na skladišti nenachází ani jedno požadované zařízení, je možnost převést na sebe cizí zařízení, které je odložené někde na hale. Pro tento případ nabízí vytvořená mobilní aplikace ve svém „Menu“ položku „Převod“.

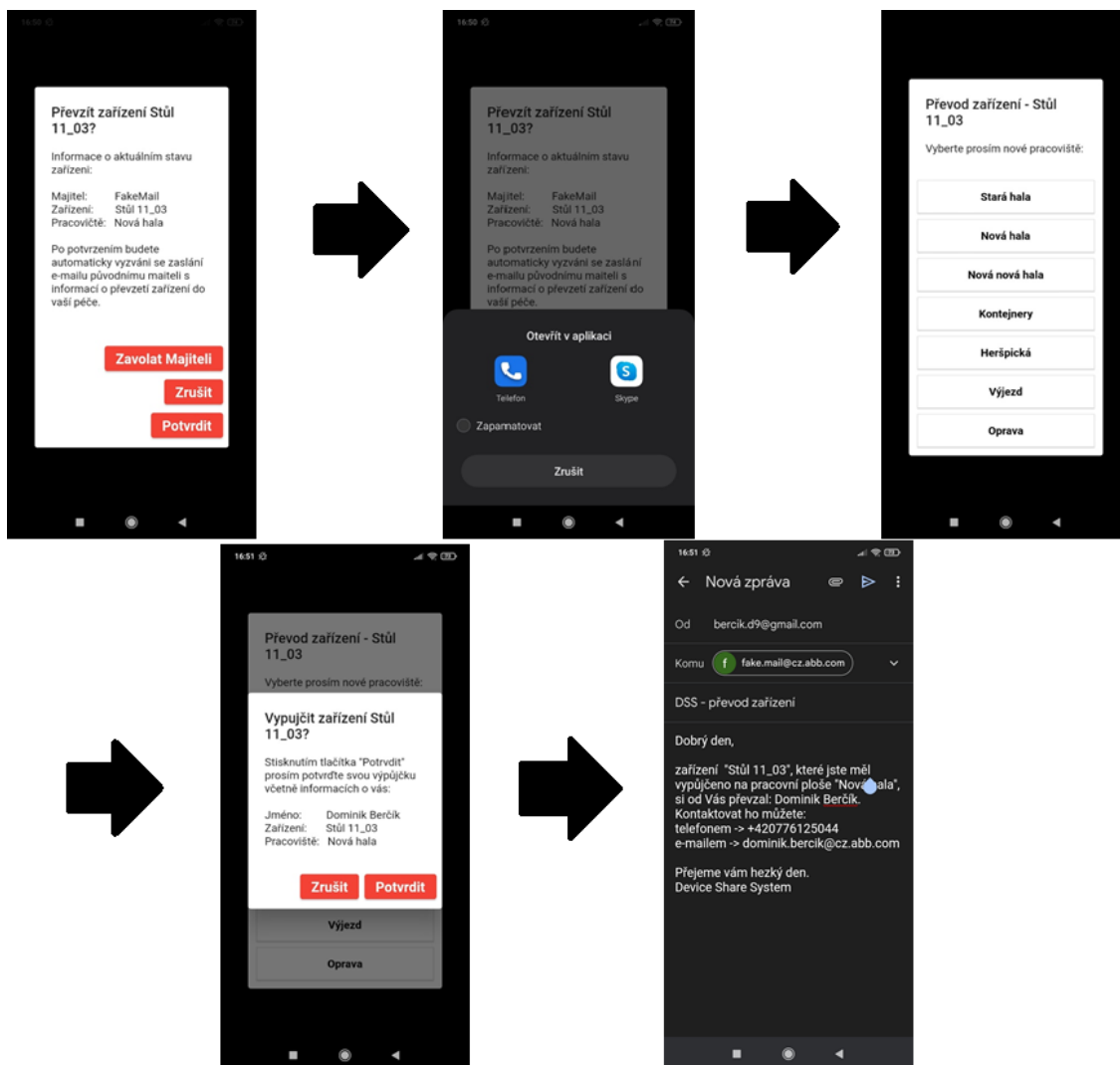
Výběrem této nabídky je uživatel opět vyzván, aby naskenoval QR kód příslušného zařízení. Pokud vybrané zařízení nemá přiřazeného žádného majitele, ukáže se vyskakovací okno s informací, že převod nelze provést a je třeba provést standardní výpůjčku. Stejná situace nastane, pokud uživatel naskenuje QR kód, který neodpovídá žádnému zařízení v databázi. Pokud uživatel naskenuje zařízení, které má již vypůjčeno, program o této skutečnosti informuje vyskakovacím oknem s příslušným hlášením.

Když skenování proběhne bez problémů, výpůjčka zařízení je v databázi nalezena, ukáže se na obrazovce telefonu okno s informacemi o výpůjčce včetně aktuálního majitele. Uživatel má nyní tři možnosti, co může udělat.

První možností je zrušit převod pomocí tlačítka „*Zrušit*“. Uživatel je pak přesměrován zpět do „*Menu*“.

Další z nabízených možností je zavolat aktuálnímu majiteli a domluvit se s ním, že uživatel si zařízení přebírá. Tuto možnost nabízí tlačítko „*Zavolat Majiteli*“, po jehož stisknutí program nabídne uživateli možnost spustit zabudovanou mobilní aplikaci pro telefonování. Po výběru se aplikace otevře s předvyplněným telefonním číslem aktuálního majitele požadovaného zařízení a stiskem na položku „*volat*“ mu zavolá a domluví se na převzetí zařízení.

V případě, že uživatel nechce s majitelem mluvit a chce si zařízení jen vzít nebo se spolu již domluvili, volí tlačítko „*Potvrdit*“. Na obrazovce se pak zobrazí výzva k výběru místa, kam si chce uživatel zařízení odvézt (Stará hala, Nová hala, Nová nová hala, Kontejnery, Heršpická, Výjezd, Oprava). Po výběru nového pracoviště se ukáže nové vyskakovací okno se sumarizací převodu. Potvrzením převodu je uživatel automaticky přesměrován do mobilního prohlížeče e-mailu nastaveného v telefonu. Zde se mu ukáže předvyplněný e-mail s informacemi pro majitele, že si přebírá jeho zařízení a s kontaktem na nového majitele. Program automaticky vyplní předmět zprávy i e-mailovou adresu původního majitele, pak už stačí pouze kliknout na odeslání zprávy. Poté, co tak učiní, je databáze aktualizována, vybrané zařízení je na něj převedeno a jeho mobilní telefon mu opět zobrazí mobilní aplikaci DSS, ve které uvidí vyskakovací okno s potvrzením, že k převodu skutečně došlo. Odsouhlasením potvrzení se program přesune zpět do „*Menu*“.



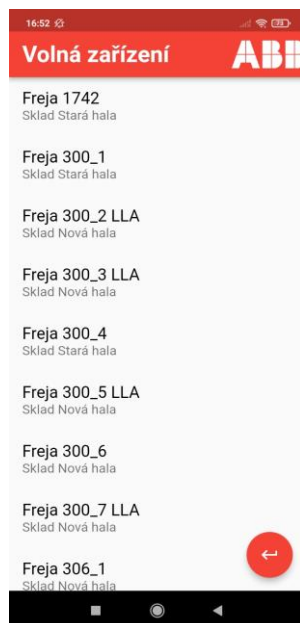
Obrázek 6.8 Obrazovky mobilní aplikace – Převod

### 6.3.8 Volná zařízení

Pro rychlé zjištění, zdali se na skladišti nachází požadované zařízení, slouží v mobilní aplikaci položka „*Volná zařízení*“. Tato funkce umožňuje pouze zjistit přítomnost zařízení na skladišti, žádné další možnosti nenabízí.

Funguje tak, že filtruje data z cloudu Firestore z kolekce „*things*“, kde vybírá jen zařízení, která mají proměnnou „*stock*“ rovnu hodnotě true. V aplikaci se pak zobrazuje seznam těchto zařízení s názvem skladu, kde se zařízení nachází.

Přidání této funkce by mělo zabránit zbytečné cestě zaměstnance na sklad, kde by pak mohl zjistit, že se zde požadované zařízení nenachází a musel by se pak vypravit na druhý sklad nebo hledat někým aktuálně nepoužívané zařízení přímo na výrobní hale.

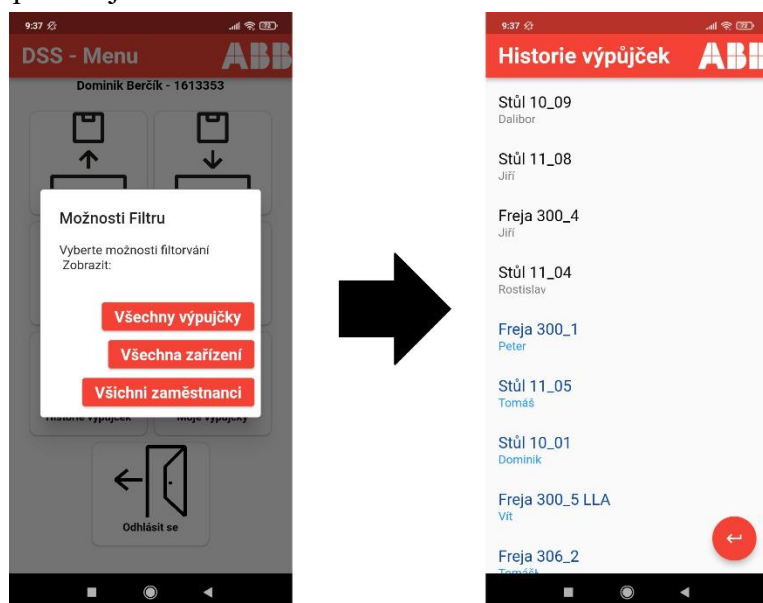


Obrázek 6.9 Obrazovka mobilní aplikace – Volná zařízení

### 6.3.9 Historie výpůjček

Jelikož je žádoucí mít přehled o tom, kdo a kdy si jaké zařízení vypůjčil, byla vytvořena funkce pro nahlížení do historie výpůjček. Funkce má sloužit hlavně pro vedoucího oddělení programátorů, který bude mít nad systémem dohled a bude řešit problémy typu poškození, ztráta, neshody mezi zaměstnanci a další.

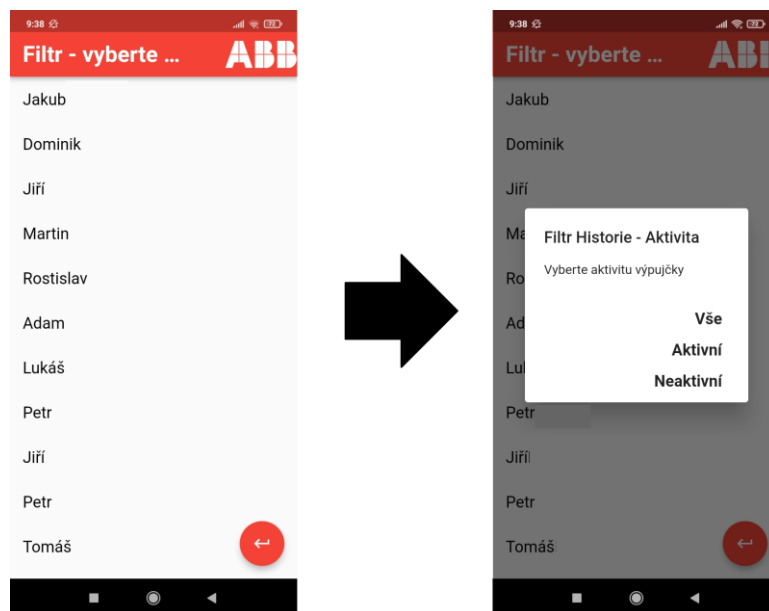
Po výběru této položky v „Menu“ se zobrazí vyskakovací okno s možností filtrování historie výpůjček. Filtrování je vytvořeno tak, aby maximálně na tři kliknutí uživatel zobrazil to, co potřebuje vidět. Filtr nabízí tři možnosti.



Obrázek 6.10 Obrazovky mobilní aplikace – Filtrování – Všechny výpůjčky

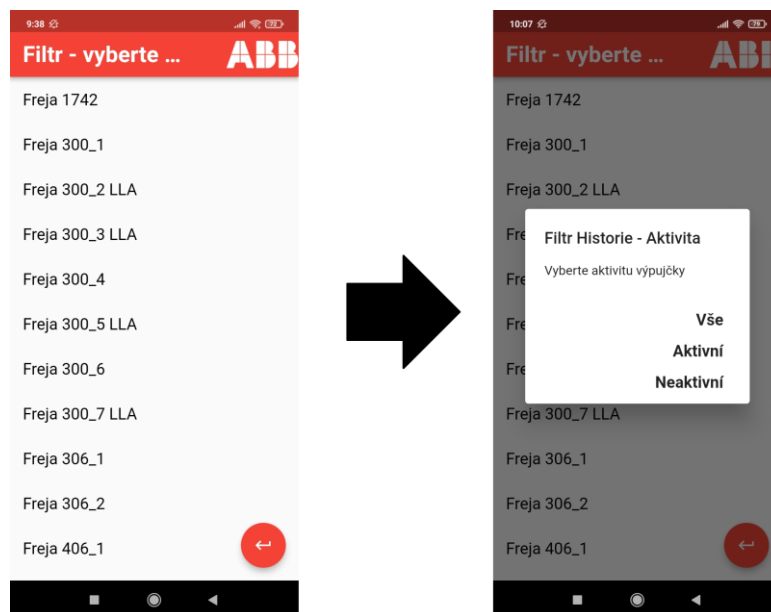
První možnost filtru je pomocí tlačítka „*Všechny výpůjčky*“. Jak název napovídá, na displeji se zobrazí seznam úplně všech provedených výpůjček uložených v databázi. Vždy je zobrazen název zařízení a hned pod ním jméno uživatele, který má zařízení vypůjčeno. Je zde barevně rozlišeno, jestli se jedná o aktivní výpůjčku nebo již o ukončenou výpůjčku. Aktivní výpůjčka má modrou barvu písma a neaktivní výpůjčka má barvu černou. Pravidla barevnosti výpůjček platí i u všech dalších možností filtrování.

Druhá možnost filtrování je pomocí tlačítka „*Všechna zařízení*“. Tato volba znamená, že uživatel chce zobrazit všechna zařízení u konkrétního člověka. Proto je uživatel vyzván, aby si vybral konkrétního člověka ze seznamu všech zaměstnanců, kteří systém DSS používají. Po výběru osoby je třeba, aby uživatel ještě vybral, zdali chce zobrazit všechny jeho aktivní výpůjčky, všechny neaktivní výpůjčky nebo úplně všechny výpůjčky bez ohledu na aktivitu. Následně se zobrazí seznam zařízení, dle volby uživatele.



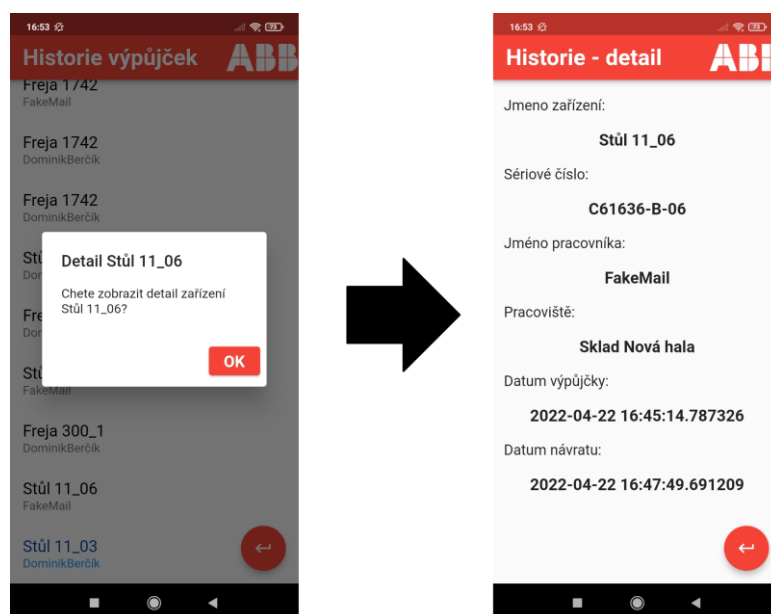
Obrázek 6.11 Obrazovky mobilní aplikace – Filtrování – Všechna zařízení

Třetí a poslední možností je filtrování pomocí tlačítka „*Všichni zaměstnanci*“. Výběr tohoto pole znamená, že uživatel chce zobrazit všechny zaměstnance, kteří měli vypůjčeno jedno konkrétní zařízení. Uživatel si zde v dalším kroku volí jedno zařízení ze seznamu všech zařízení z databáze a následně opět určuje, zdali chce zobrazit aktivní, neaktivní nebo obojí druh výpůjček. Výsledek filtrování je pak zobrazen na obrazovce.



Obrázek 6.12 Obrazovky mobilní aplikace – Filtrování – Všichni zaměstnanci

Když je zobrazena historie požadovaných výpůjček, je zde ještě možnost zobrazit detail konkrétní výpůjčky a to tak, že uživatel jednoduše klikne na pole výpůjčky. To způsobí zobrazení vyskakovacího okna s dotazem, zdali si přeje zobrazit detail výpůjčky. Po odsouhlasení se ukáže nové okno, kde je vidět kompletní záznam vybrané výpůjčky z databáze, tedy jméno zařízení, sériové číslo zařízení, jméno pracovníka, pracoviště a časová značka, kdy byla výpůjčka provedena a kdy byla případně ukončena.



Obrázek 6.13 Obrazovky mobilní aplikace – Historie výpůjček – detail

### 6.3.10 Moje výpůjčky

Aby uživatel jednoduše mohl zjistit, co všechno si již vypůjčil a kde případně zařízení nechal ležet, slouží v mobilní aplikaci funkce „*Moje výpůjčky*“. Funkce zobrazuje pouze výpůjčky, které jsou aktivní, nezobrazuje žádné zařízení, které již bylo vráceno na sklad, čímž byla výpůjčka ukončena.

Stejně jako u zobrazování historie je i zde možnost po kliknutí na konkrétní zařízení zobrazit detail výpůjčky vybraného zařízení. Oproti detailu zobrazenému v historii neukazuje detail ve funkci „*Moje výpůjčky*“ čas ukončení výpůjčky, jelikož k navrácení ještě nedošlo. Taktéž neukazuje ani jméno uživatele, který si zařízení vypůjčil, jelikož se jedná vždy přímo o uživatele.

Namísto toho je zde tlačítko „*Vrátit zařízení*“, které umožňuje ukončit výpůjčku a vyhnout se tak skenování QR kódu zařízení. Uživateli stačí pouze přinést toto zařízení do skladu a naskenovat zde QR kód skladu, aby bylo ověřeno, že zařízení je řádně uskladněno. Učiní-li tak, je vyzván k potvrzení návratu zařízení a výpůjčka je tímto ukončena a již se mu nebude zobrazovat v sekci „*Moje výpůjčky*“.



Obrázek 6.14 Obrazovka mobilní aplikace – Moje výpůjčka – detail

### 6.3.11 Odhlášení

Poslední položkou, kterou „*Menu*“ nabízí, je „*Odhlášení*“. Jak název napovídá, slouží pro odhlášení a ukončení práce uživatele v mobilní aplikaci. Po výběru této položky se ukáže vyskakovací okno s upozorněním na odhlášení a výzvou k potvrzení, že se uživatel chce skutečně odhlásit. Pokud uživatel potvrdí odhlášení, je přesměrován zpět na úvodní obrazovku, tedy na obrazovku přihlášení a pro pokračování práce v aplikaci se musí opět přihlásit.



Obrázek 6.15 Obrazovka mobilní aplikace – Odhlášení

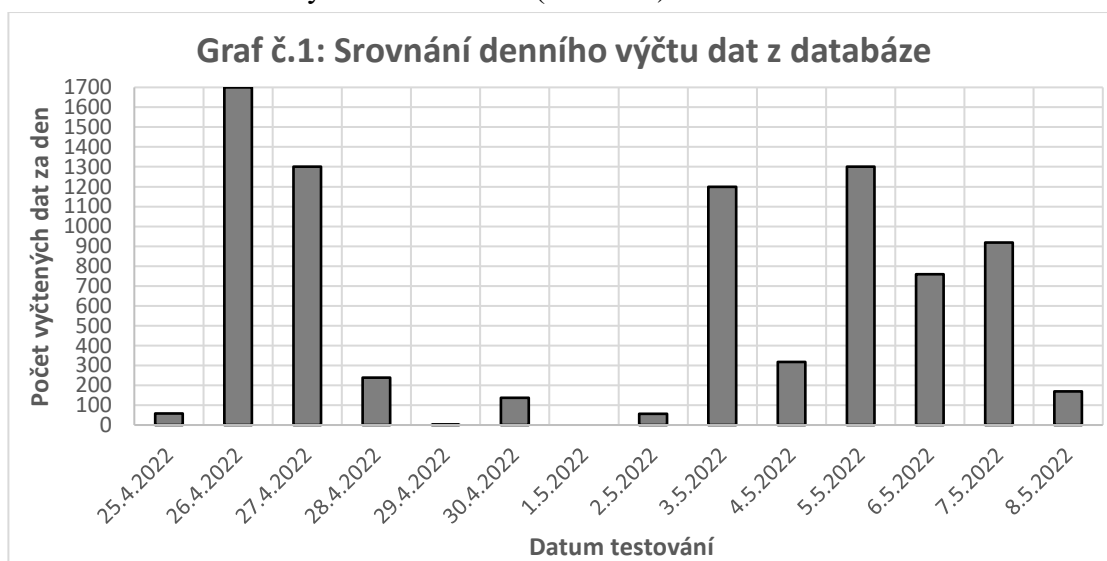
## 7. ZHODNOCENÍ PŘÍNOSU SYSTÉMU

Zprovoznění systému předcházelo vygenerování patričných QR kódů, jejich umístění na příslušná zařízení a zaznamenání potřebných informací do databáze. Rovněž proběhlo školení zaměstnanců, jak mobilní aplikaci, která slouží pro ovládání systému, používat. Pro zaměstnance byl proto vytvořen návod k použití. Rovněž jim byl předán instalační soubor s touto aplikací. Dále pak byli zaměstnanci požádáni o nahlášení jakékoliv chyby nebo možného nedostatku systému.

Vytvořený systém byl uveden do provozu v pondělí 25. 4. 2022. Následující dva týdny bylo pozorováno, zdali systém pracuje správně, zdali nevznikají zásadní chyby a zdali je systém pro firmu přínosný. Během těchto dvou týdnů byla nashromážděna data se zpětnou vazbou jak od zaměstnanců, tak i pomocí databázové platformy Firebase, která zobrazuje, jak často jsou data zaznamenána nebo čtena.

### 7.1 Výsledky z Firebase

Platforma Firebase zaznamenává počet zápisů, čtení a aktivitu připojených zařízení. Primárně tato data uchovává kvůli vyúčtování, avšak systém má využívat jen omezené množství uživatelů, tudíž by k zpoplatnění jeho služeb dojít nemělo. Výsledná data (ze kterých jsou pak vytvořeny následující grafy), jsou shrnuta v tabulce v příloze B. Graf na obrázku 7.1 ukazuje, kolik dat bylo během jednoho testovacího dne vyčteno z databáze. Jelikož školení na používání mobilní aplikace bylo naplánováno až na druhý den testovacího provozu (26.4.2022), je zde vidět velký skok, protože si v tento den větší část zaměstnanců aplikaci nainstalovala a prohlížela funkcionalitu. V následujících dnech jsou pak zaznamenána data dle předpokládaných hodnot. Z dat jde vidět i zajímavost, že někdo z uživatelů nahlížel do systému i v neděli (8.5.2022).



Obrázek 7.1 Srovnání denního výčtu dat z databáze

Záznam počtu zapsaných dat ukazuje obrázek 7.2 s grafem číslo 2. Pod zaznamenanými daty se skrývají data zapsaná při výpůjčce zařízení, při vrácení zařízení na sklad nebo při převodu zařízení mezi zaměstnanci. V prvním týdnu jde vidět, že si zaměstnanci na systém teprve zvykali. V druhém týdnu je viditelný nárůst zápisu dat a tedy i zvýšení provedených změn v databázi. Zajímavá data jsou 30.4.2022 a 7.5.2022, jedná se o soboty. Tyto dny bylo zapotřebí nejspíš dohánět práci, případně zde mohlo dojít k vypůjčení testovacích zařízení na služební cesty.



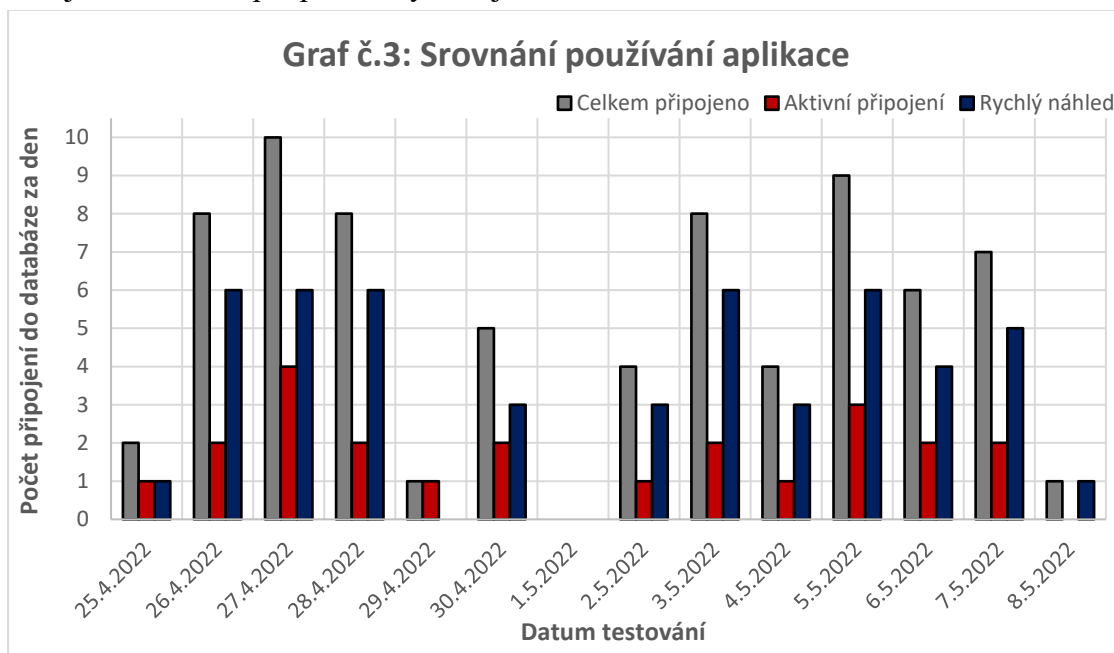
Obrázek 7.2 Srovnání denního zápisu dat do databáze

Jakým způsobem byla aplikace opravdu používána, naznačuje poslední graf číslo 3., který je na obrázku 7.3. Ukazuje tři hodnoty v každém testovacím dni. První hodnota „*Rychlý náhled*“ je značena modře. Ukazuje uživatele, kteří se daný den do aplikace přihlásili jen na krátký čas a kteří si jen prohlíželi data. Mohlo se jednat např. o prohlížení historie výpůjček nebo seznamu volných zařízení. Nevytvářeli žádné zápisy do databáze. Opakem toho je „*Aktivní připojení*“, kde je ukázán uživatel, který již vytvářel záznamy v databázi, tzn. uživatel, který vytvořil, převedl nebo ukončil výpůjčku zařízení. V grafu jsou naznačeny červenou barvou.

Součtem aktivních připojení a rychlých náhledů vznikla hodnota pro položku „*Celkem připojeno*“. Ta určuje, kolik uživatelů se za daný den připojilo. Opakované připojení uživatele do aplikace se nepočítá. Z tohoto grafu jde nejlépe poznat, jak byla aplikace používána, ale i co v ní uživatelé prováděli. Z celkových hodnot pak lze určit průměrnou hodnotu, která je přibližně 5,2, což naznačuje, že denně použije aplikaci průměrně 5 osob. Nutno podotknout, že do používání systému se ještě nezapojili všichni zaměstnanci, jelikož velká část z nich odjela na služební cestu ještě před zprovozněním systému. V přepočtu na aktuálně zapojené zaměstnance je systém denně využíván polovinou těchto zaměstnanců. Z toho vyplývá, že zařízení denně potřebuje každý druhý

zaměstnanec. Právě zisk takovýchto informací je pro zadavatele důležitý a pomáhá mu vyhodnocovat potřebu dokoupení testerů.

Zajímavý poznatek vyplývající z grafů je využívání aplikace zaměstnanci i ve svém volném čase, kdy třeba v neděli (8.5.2022) bylo do databáze jen nahlíženo. Stejně tak se dělo během sobot. Někdo doháněl projekty nebo odjížděl na služební cestu, ale také se našli jedinci, kteří opět použili systém jen k nahlížení do databáze.



Obrázek 7.3 Srovnání denního používání aplikace

## 7.2 Připomínky zaměstnanců

Při školení zaměstnanců na vytvořený systém a jeho používání byli zaměstnanci požádáni o zpětnou vazbu. Šlo především o nahlášení veškerých možných chyb nebo nedostatků, kterých si při používání mobilní aplikace všimli, aby bylo možné aplikaci dále vylepšovat. Připomínky jsou následující:

- Tlačítko operačního systému „zpět“ ruší funkcionalitu ukládání informací o výpůjčce, umožňuje vytvořit na jedno zařízení současně několik výpůjček.
- Tlačítko operačního systému „zpět“ nefunguje stejně jako aplikační tlačítko „return“ a to ve většině funkcí.
- Aplikace si nepamatuje přihlašovací údaje, opětovné zadávání zdržuje.
- Při zobrazení detailu výpůjčky v historii nebo v položce moje výpůjčka nedochází ke zmizení potvrzovacího okna. Je třeba kliknout mimo okno.
- Zlepšit filtrování ve funkci „Historie“, špatně se zde orientuje, upravit zapisované hodnoty. Úplně nahoře zobrazovat aktivní výpůjčky.
- Přidat funkci na rychlé zjištění majitele vypůjčeného zařízení, tak aby se nemuselo využívat funkce „Převod“.

Některé ze zmíněných připomínek od uživatelů se opakovaly. K těm nejvýznamnějším odhalením pak patří první dva body, kdy špatně funguje systémové tlačítko „zpět“. To se vždycky vrátí o krok zpět, tudíž naruší potvrzovací příkazy při výpůjčkách, návratech nebo při převodech. Při předávání instalačního programu mobilní aplikace zaměstnancům se vědělo, že systémové tlačítko „zpět“ nefunguje úplně správně a bylo jim doporučeno, ať jej nepoužívají. O tom, že pomocí něj jde provést duplicitní výpůjčka, se ale nevědělo.

Kromě návrhů na zlepšení a nalezených chyb byla poskytnuta i kladná zpětná vazba. Zaměstnanci chválili, že systém je rychlý a že na pár kliknutí mají vypůjčené potřebné zařízení. Dále se jim líbí možnost vytočení čísla aktuálního majitele při použití funkce pro převod zařízení. Kladně také hodnotili vzhled aplikace a její jednoduché používání.

Vytvořený systém splnil své hlavní úkoly a to urychlit sdílení zařízení a hlavně sledovat pohyb jednotlivých testovacích zařízení na výrobních halách.

### 7.3 Srovnání výsledků s požadavky

V analýze zadání se dospělo ke dvanácti hlavním bodům, které bylo dle požadavků zadavatele potřeba splnit. Od některých bodů se při konzultacích upustilo. Bylo zjištěno, že pro fungování vytvářeného systému jsou irelevantní. V následující tabulce jsou zvýrazněny modře body, od kterých se upustilo, zeleně pak body, které byly splněny dle požadavků.

Tabulka 7.1 Srovnání výsledků a požadavků ze zadání

<b>Číslo</b>	<b>Požadavek zadavatele</b>
<b>1.</b>	<i>Nalezení podobných řešení problému</i>
<b>2.</b>	<i>Zamykání zařízení na skladu pomocí elektronických zámeků</i>
<b>3.</b>	<i>Identifikace zaměstnance</i>
<b>4.</b>	<i>Navržení a sestrojení ovládacího zařízení (PLC + vhodné komponenty)</i>
<b>5.</b>	<i>Navržení a sestrojení demonstračního panelu pro testování a prezentaci</i>
<b>6.</b>	<i>Návrh řešení požadovaných funkcí</i>
<b>7.</b>	<i>Vytvoření softwaru pro demonstrační panel (ovládací panel)</i>
<b>8.</b>	<i>Ukládání dat do databáze</i>
<b>9.</b>	<i>Řešení možnosti výpadku přístupu na internet</i>
<b>10.</b>	<i>Popis webových a mobilních aplikací</i>
<b>11.</b>	<i>Nahlížení do databáze a ovládání systému pomocí mobilní a webové aplikace</i>
<b>12.</b>	<i>Zavedení systému do provozu ve firmě a testování funkčnosti</i>

Nezrealizované body číslo 4. a 5. spolu úzce souvisí. Od jejich realizace bylo upuštěno, jelikož se v rámci konzultací a přemýšlení, jak vytvořit co nejlevnější a nejefektivnější systém, přišlo na to, že požadavky, které zadavatel má na ovládací a

demonstrační panel, velmi snadno pokryje mobilní aplikace. Největší výhodou je, že mobilní aplikace nevyžaduje žádný další hardware, který by po čase přinesl omezení systému, ale nabízí nenákladné rozšíření například o nové funkce, a navíc firemní mobilní telefon patří k běžným vybavením zaměstnancům zadavatelské firmy.

Tím, že se rozhodlo, že demonstrační panel nebude vytvořen, se rozhodlo i o tom, že se upustí i od použití elektronických zámek na jednotlivé zařízení (bod požadavků číslo 2.). Místo toho je vložena důvěra do zaměstnanců firmy, že budou řádně používat mobilní aplikaci a pravidla sdílení zařízení. Pokud by nastala potřeba zabezpečit zařízení na skladě před odcizováním, je systém předpřipraven na připojení bezpečnostních bran, které by zamezili vypůjčení bez zadání do systému. Ve vytvořené databázi je připravená boolean proměnná, pomocí které by tato kontrola byla provedena.

Bod požadavku číslo sedm nebyl řešen ve znění, jak je zadán. Místo vytváření softwaru pro demonstrační panel, jehož realizace byla zamítnuta, byl jako ovládací prvek systému vyvinuta software pro mobilní aplikaci.

Důležité je podotknout, že bod číslo devět byl řešen v rámci konzultací, na kterých se zadavatel zaručil, že v prostorách firmy bude internetové připojení možné vždy. A to buď přes Wi-Fi připojení nebo pomocí mobilních dat, která má každý zaměstnanec předplacená pro používání na svém pracovním mobilním telefonu.

Pro úplné srovnání je třeba ještě zmínit funkce, které zadavatel po systému požadoval a které nakonec byly použity. Pro toto srovnání je opět vytvořena barevně rozlišená tabulka, kde modrý text značí funkce, které po konzultaci se zadavatelem vytvořeny nebyly, zelený text značí funkce vytvořené.

Tabulka 7.2 Srovnání použitých a nepoužitých funkcí systému

<b>Číslo</b>	<b>Požadovaná funkce systému</b>
<b>1.</b>	<i>Vypůjčka zařízení</i>
<b>2.</b>	<i>Návrat zařízení</i>
<b>3.</b>	<i>Nahlížení do databáze systému</i>
<b>4.</b>	<i>Převod zařízení na jiného zaměstnance</i>
<b>5.</b>	<i>Rezervace zařízení</i>

Původně požadovaná funkce „Rezervace zařízení“ (bod číslo 5.) nebyla nakonec vytvořena, jelikož se během konzultací dospělo k názoru, že bez uzamykání zařízení nebo užití jiných bezpečnostních prvků na skladu, nemůže být zaručeno, že bude rezervační systém dodržován. Stejně tak by zde byla možnost, že by někteří zaměstnanci měli jedno zařízení neustále rezervováno, čímž by bylo znemožněno jeho sdílení.

Ve výsledném provedení je třetí funkce („Nahlížení do databáze systému“) rozdělena do tří individuálních funkcí pro rychlejší orientaci a nalezení potřebných dat. Patří sem funkce „Historie vypůjček“, „Moje vypůjčky“ a „Volná zařízení“.

## 8. ZÁVĚR

Prvním úkolem této práce bylo analyzovat požadavky zadavatele. Bylo zjištěno, že zadavatel požaduje vytvoření systému, který bude levný a uživatelsky přívětivý. Systém má umožňovat vypůjčku zařízení a následně pak návrat tohoto zařízení zpět na sklad. Dále byla požadována funkce na převod již vypůjčeného zařízení od jednoho uživatele na druhého. Kromě toho má systém umožňovat nahlížení do databáze zařízení, kde bude jednoduše zjistitelné, jaké zařízení se nachází na skladišti, nebo kdo má či měl požadované zařízení vypůjčeno. Poslední požadovanou funkcí je umožnit rezervaci zařízení. Tato funkce byla však v pozdější části vývoje z požadavků odstraněna, jelikož v rámci možnosti lehkého zneužití neustálým opakováním rezervace na jedno zařízení od jednoho uživatele, by se zabránilo původní myšlence systému pro sdílení zařízení. Zadavatel také vyžadoval udělat průzkum již existujících systémů podobného typu pro inspiraci a poučení z chyb.

Kromě funkčních požadavků na systém zadavatel požadoval i mechanické vlastnosti systému, jako jsou použití elektronických zámků, tvorba demonstračního panelu a další. Od všech fyzických prvků, které měl vytvářený systém obsahovat, se rovněž při tvorbě návrhu upustilo, jelikož se přišlo na to, že jejich použití není potřebné a vše podstatné obstará pouze jediný ovládací prvek, kterým je mobilní aplikace.

Při návrhu se tedy upřednostnila jednoduchá mobilní aplikace s tím, že v ní bude, pomocí skeneru QR kódu v telefonu a samolepek s QR kódem nalepených na jednotlivých zařízeních, velmi jednoduché všechna zařízení jednoznačně identifikovat. Původní potřeba zamykat zařízení na skladišti byla také zrušena, jelikož sem mají přístup jen pověřeni zaměstnanci. V případě potřeby vyššího zabezpečení zařízení je zde možnost rozšířit systém o bezpečnostní brány, které by hlídaly odcizení zařízení a to kontrolou jediné předpřipravené boolean proměnné z databáze.

Při návrhu systému se počítalo i s vytvořením webové aplikace, která měla sloužit hlavně pro správu databáze a tvorbu rezervací. Problém však nastal při pokusech připojit se na již existující firemní SQL databázi. Z bezpečnostních důvodů nebyl umožněn přístup do této databáze, a proto se využilo služeb platformy Firebase. Firebase nabízí mnoho možností jak uchovávat a zpracovávat data. Díky němu již nebylo třeba vytvářet nové webové prostředí pro správu aplikace, jelikož Firebase požadovanou správu databáze v uživatelsky přívětivém prostředí již umožňuje. Přístup do této databáze má jen administrátor systému, ten může upravovat data bez jakéhokoliv omezení. Velká nevýhoda systému Firebase je, že po překročení limitů pro čtení nebo zápis dat je služba zpoplatněna a to za vysoké částky. Proto byl proveden průzkum a bylo zjištěno, že pro požadavky zadavatele, jsou limity více než dostatečné a tudíž by ke zpoplatnění dojít nemělo.

V rámci volby databáze, která jednoduše umožňuje komunikovat s mobilní aplikací, je Firebase jedna z nejlepších možností. Je přímo určená ke komunikaci s mobilní

aplikací a rovněž i pro komunikaci s webovou aplikací. Jsou v ní uloženy tři tabulky, tak jak ukazuje datový model v příloze A. Ty obsahují nejnútnejší informace o uživateli, zařízeních a provedených výpůjčkách. Samotná komunikace je pak nastavena dle konfiguračních instrukcí, které poskytuje přímo platforma Firebase. Obnáší přidání jednoho inicializačního souboru do projektu vyvíjené aplikace. Pro správné fungování je třeba v ní definovat i potřebné ovladače.

Mobilní aplikace byla vyvinuta v prostředí Android studia za pomoci uživatelského prostředí Flutter, kde je kód psán pomocí jazyka Dart. Kombinace použitých prostředků dovoluje vytvářet mobilní aplikace jak pro operační systém Android tak i pro iOS. Výsledná aplikace je však vyvinuta pouze na operační systém Android, jelikož pro využití na iOS je třeba aplikaci vyvíjet pod operačním systémem od společnosti Apple. Tento požadavek však nebyl splněn a tak je projekt pouze připraven na budoucí rozšíření. Při možnosti dopracování projektu na požadovaném operačním systému by mělo stačit umístit inicializační soubor z Firebase do složky, která je z jiných operačních systémů nedosažitelná a tudíž ani do ní nelze uložit požadovaný soubor.

Veškerá funkcionálna a posuny ve vývoji systému byly neustále konzultovány se zadavatelem. Od toho se odvíjí i fakt, že výsledná aplikace nefunguje přesně, tak jak ukazuje návrh. Některé položky jednotlivých funkcí z návrhu byly odstraněny, například potvrzení předání zařízení ve funkci pro „Převod“. Zde vyšlo najevo, že čekání na souhlas by proces převodu značně zpomalilo a tak nastal předpoklad, že by se tato funkce přestala využívat.

Zavedení systému do provozu firmy předcházelo připravení všech potřebných QR kódu a jejich následné umístění na patřičné zařízení. Zaměstnancům byl poskytnut instalační soubor mobilní aplikace. Rovněž proběhlo školení, kde bylo vysvětleno, jak systém funguje a jak se má používat. Systém byl v testovacím provozu dva týdny a z výsledků vyplynulo, že průměrně využije systém za jeden den polovina z možných zaměstnanců. Zúčastnění zaměstnanci podali i zpětnou vazbu s náměty na vylepšení a odhalení chyb. Nejzávažnější chyba je, že systémové tlačítko „zpět“ při vytváření výpůjčky umožňuje požadovanou výpůjčku duplikovat. To znamená, že na jednoho uživatele a jedno zařízení se v databázi uloží několik stejných výpůjček. Tato chyba bude v nejbližší době opravena. Ostatní nahlášené nedostatky budou také opraveny, ale nemají zásadní vliv na fungování systému. Zaměstnanci kladně hodnotili vzhled aplikace, její jednoduchost a rychlost.

Cílem této diplomové práce bylo navrhnout a vytvořit systém, který detekuje vybrané měřicí zařízení a přidělí jej v databázi k zaměstnanci, který zařízení vybral. Cíl práce byl společně se zadáním splněn ve všech požadovaných bodech. Vytvořený systém je plně funkční a v zadavatelské firmě denně používán. Výsledek je odlišný od původních požadavků zadavatele, ale všechny tyto odlišnosti jím byly schváleny a funkčnost systému nijak neomezují.

## LITERATURA

- [1] BERČÍK, Dominik. *Návrh a realizace systému pro výpůjčku zkušebních zařízení* [online]. Brno, 2022 [cit. 2022-04-25]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/138337>. Semestrální práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Petr Fiedler.
- [2] MM PRŮMYSLOVÉ SPEKTRUM. *Systém pro automatický výdej nástrojů*. [online]. 2012, 12.července 2012, (7,8) [cit. 2021-10-24]. 120737. Dostupné z: <https://www.mmspektrum.com/clanek/system-pro-automaticky-vydej-nastroju>
- [3] ČERNOŠEK, Martin. *Konstrukční návrh prodejního automatu na energetické tyčinky* [online]. Brno, 2011 [cit. 2021-10-24]. Dostupné z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=41155](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=41155). Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Vedoucí práce Ing. Michal Vaverka, Ph.D.
- [4] WÜRTH. *Výdejní automaty* [online]. 2018 [cit. 2021-10-24]. Dostupné z: [https://www.wuerth.cz/cs/wuerth\\_cz/produkty/vydejni\\_automaty/orsy\\_mat.php](https://www.wuerth.cz/cs/wuerth_cz/produkty/vydejni_automaty/orsy_mat.php)
- [5] KONOPKA, Petr. *Zařízení na zavážení pilin do kotle* [online]. Ostrava, 2011 [cit. 2021-10-24]. Dostupné z: <https://core.ac.uk/download/pdf/8990998.pdf>. Bakalářská práce. VŠB – Technická univerzita Ostrava, Fakulta strojní. Vedoucí práce Ing. Martin Janečka.
- [6] KLATT. *Spirálový výdejový automat* [online]. Kolín [cit. 2021-10-26]. Dostupné z: <https://vydejniautomat.eu/wa-maxi-spiralovy-vydejovy-automat>
- [7] ALIEXPRESS. *Mini Cargo Conveyor Belt Machine Vending Machine Grocery Pickup Contactless Table Top Belt Conveyor* [online]. China, 2021 [cit. 2021-10-26]. Dostupné z: <https://www.aliexpress.com/item/10000248527273.html>
- [8] PATENTDOCS. *Dipensing mechanism of vending* [online]. Shenzhen, 2014 [cit. 2021-10-26]. Dostupné z: <https://www.patentsencyclopedia.com/app/20140183217>
- [9] SMARTMAN. *Jak funguje výdejní box Z-BOX vs. AlzaBox a WE/DO Box?* [online]. ByznysWeb.cz, 2021 [cit. 2021-10-26]. Dostupné z: <https://www.smartmen.cz/n/jak-funguji-vydejni-boxy-z-box-vs-alzabox-a-wedo-box>
- [10] MACHINERY. *Tool vending systems provide production* [online]. Ireland, 2021 [cit. 2021-10-26]. Dostupné z: <https://www.machinery.co.uk/machinery-features/tool-vending-systems-provide-production-intelligence-tool-usage-data-drives-improvement>
- [11] CRIB MASTER. *Expres toolbox* [online]. 2021 [cit. 2021-10-26]. Dostupné z: <https://www.cribmaster.com/products/express-toolbox>
- [12] PHAM, Le. *Mobile application: Definition, technology types and examples 2022*. Magenest [online]. 2021, 31.listopadu 2021 [cit. 2022-04-27]. Dostupné z: <https://magenest.com/en/mobile-application/>

- [13] ADOBE. *Informace o webových aplikacích* [online]. 2021, 19.5.2021 [cit. 2021-11-03]. Dostupné z: [https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html#how\\_a\\_web\\_application\\_works](https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html#how_a_web_application_works)
- [14] ORCALE. *What Is a Database?* [online]. 2021 [cit. 2021-11-03]. Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>
- [15] PETERSON, Richar. *What is a Database? Definition, Meaning, Types, Example. GURU99* [online]. 2021, 7.října 2021 [cit. 2021-11-03]. Dostupné z: <https://www.guru99.com/introduction-to-database-sql.html>
- [16] NAVRÁTIL, Jiří. *BEZPEČNÝ PŘENOS DAT MEZI MULTIPLATFORMNÍMI DATABÁZOVÝMI SERVERY* [online]. Brno, 2008 [cit. 2021-11-08]. Dostupné z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=9226](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=9226). Balakářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Ing. Ivo Lattenberg, Ph.D.
- [17] KOSEK, Jiří. *HTML5 Zpřístupnění stránek světu: Webový server* [online]. 2014 [cit. 2021-11-08]. Dostupné z: <http://htmlguru.cz/vystaveni-webovy-server.html>
- [18] *Webový server. Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021, 2015 [cit. 2021-11-08]. Dostupné z: [https://cs.wikipedia.org/wiki/Webov%C3%BD\\_server](https://cs.wikipedia.org/wiki/Webov%C3%BD_server)
- [19] FIREBASE. *Products and solutions you can rely on through your app's journey* [online]. Kalifornie (USA), 2022 [cit. 2022-05-02]. Dostupné z: <https://firebase.google.com>
- [20] EDPRESSO TEAM. *What is Firebase?, Educative: Learn something new. Everyday.* [online]. 2022 [cit. 2022-05-02]. Dostupné z: <https://www.educative.io/edpresso/what-is-firebase>
- [21] CLOUD FIRESTORE. *Firestore: Dokumentace Firestore* [online]. Kalifornie (USA), 2022, 28.dubna 2022 [cit. 2022-05-02]. Dostupné z: <https://firebase.google.com/docs/firestore>
- [22] FIREBASE AUTHENTICATION. *Firestore: Dokumentace Firestore* [online]. Kalifornie (USA), 2022, 19.dubna 2022 [cit. 2022-05-02]. Dostupné z: <https://firebase.google.com/docs/auth>
- [23] ANDROID STUDIO. *Developers* [online]. 2022 [cit. 2022-04-27]. Dostupné z: <https://developer.android.com/studio>
- [24] *IntelliJ IDEA: Why IntelliJ IDEA* [online]. Praha: JetBrains, 2020 [cit. 2022-04-27]. Dostupné z: <https://www.jetbrains.com/idea/>
- [25] *Gradle Build Tool* [online]. San Francisco: Gradle\_Inc., 2022 [cit. 2022-04-27]. Dostupné z: <https://gradle.org>
- [26] ANDROID STUDIO. *Run apps on the Android Emulator* [online]. 2022 [cit. 2022-04-27]. Dostupné z: <https://developer.android.com/studio/run/emulator>
- [27] Third Rock Techkno. *12 New Android Frameworks for App Development in 2022* [online]. San Jose, 2022, 3.července 2020 [cit. 2022-04-27]. Dostupné z: <https://www.thirdrocktechkno.com/blog/12-best-frameworks-for-android-application-development-in-2020/>

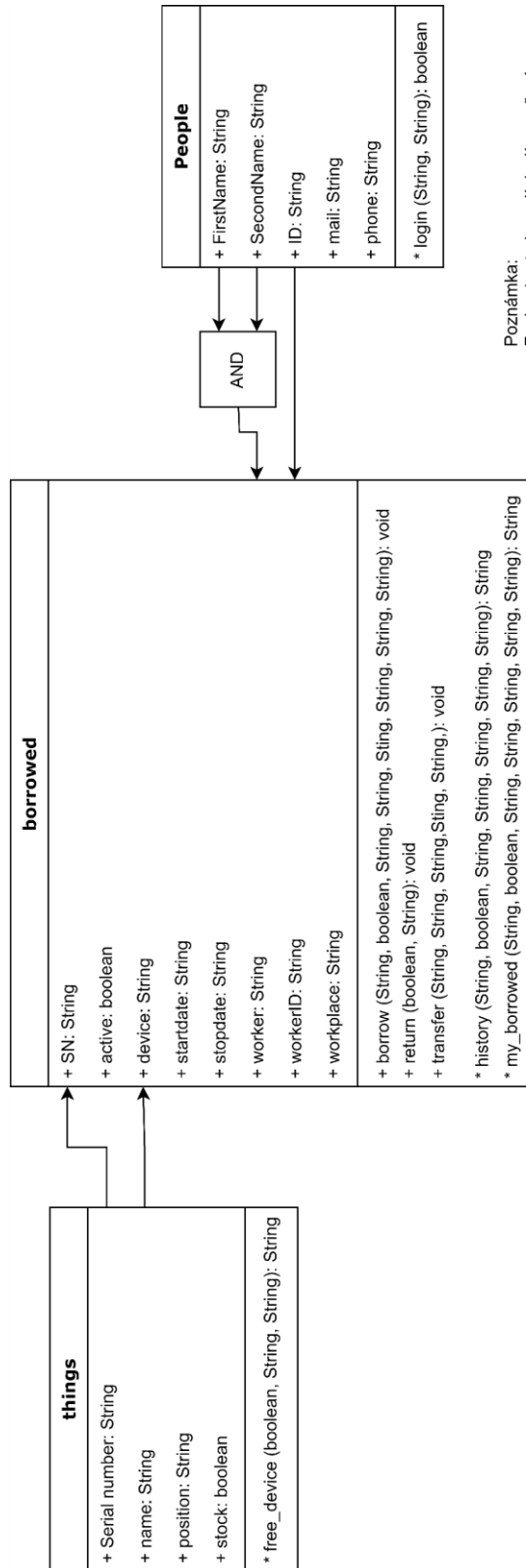
- [28] FLUTTER: *Build apps for any screen* [online]. Kalifornie (USA), 2022 [cit. 2022-05-02]. Dostupné z: <https://flutter.dev>
- [29] GAËL, Thomas. *What is Flutter and Why You Should Learn it in 2020*. FreeCodeCamp.org [online]. 2019, 12.prosince 2019 [cit. 2022-05-02]. Dostupné z: <https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>
- [30] JAISWAL, Sonoo. *What is Dart Programming*. Java T point [online]. 2021 [cit. 2022-05-02]. Dostupné z: <https://www.javatpoint.com/flutter-dart-programming>

## SEZNAM PŘÍLOH

- A. Datový model*
- B. Tabulka dat z Firebase*
- C. Projekt z Android Studio*
- D. Instalační soubor mobilní aplikace Device Share System*
- E. Návod na použití pro zaměstnance*

Poznámka: Přílohy C, D a E se nacházejí na přiloženém CD.

# PŘÍLOHA A - DATOVÝ MODEL



Poznámka:  
 Funkce/metody mají dvojití označení a  
 význam: "+-" mění data  
 "\*" čte data

## PŘÍLOHA B - TABAULKA DAT Z FIREBASE

Datum	Vyčtená data z databáze	Zapsaná data do databáze	Rychlý náhled	Aktivní připojení	Celkem připojeno
25.4.2022	58	2	1	1	2
26.4.2022	1700	8	6	2	8
27.4.2022	1300	15	6	4	10
28.4.2022	238	4	6	2	8
29.4.2022	2	4	0	1	1
30.4.2022	137	6	3	2	5
1.5.2022	0	0	0	0	0
2.5.2022	56	0	3	1	4
3.5.2022	1200	12	6	2	8
4.5.2022	318	18	3	1	4
5.5.2022	1300	16	6	3	9
6.5.2022	759	24	4	2	6
7.5.2022	918	10	5	2	7
8.5.2022	169	0	1	0	1

Průměrné denní připojení: 5,21