



Ekonomická  
fakulta  
Faculty  
of Economics

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Ekonomická fakulta

Katedra aplikované matematiky a informatiky

Diplomová práce

# Konstrukce a řízení chytré dobíjecí stanice pro elektrokoloběžky

Vypracoval: Bc. Luděk Scholz

Vedoucí práce: Mgr. Jakub Geyer

České Budějovice 2023

# JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Ekonomická fakulta

Akademický rok: 2022/2023

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Bc. Luděk SCHOLZ  
Osobní číslo: E21050  
Studijní program: N0613A140025 Aplikovaná informatika  
Specializace: Softwarové inženýrství  
Téma práce: Konstrukce a řízení chytré dobíjecí stanice pro elektroběžky  
Zadávací katedra: \*\*\*Katedra aplikované matematiky a informatiky

### Zásady pro vypracování

Cílem práce je návrh a realizace chytré dobíjecí stanice (SCS) pro elektrokoběžky včetně systému řízení. SCS musí umožňovat fyzické uzamčení dobíjené elektrokoběžky s odemčením na základě elektronické identifikace (např. čipu). Nabíjení musí být optimalizováno úrovní nabití baterie elektrokoběžky. Vkládání elektrokoběžky do SCS a její vyjímání musí být uživatelsky jednoduché a rychlé. SCS bude umožňovat paralelní nabíjení min. čtyř elektrokoběžek. Systém bude poskytovat informace o dostupných koběžkách v SCS a logovat vyzvednutí elektrokoběžky. Řízení musí zahrnovat minimálně tyto prvky:

- Odemčení na základě čipu.
- Ukončení dobíjení při plném nabití baterie.
- Informace o dostupných elektrokoběžkách v SCS.

Součástí práce budou bezpodmínečně následující dokumenty:

- Analýza funkčních a nefunkčních požadavků
- Analýza a výběr vhodných hardwarových a softwarových prostředků
- Zdrojová data 3D modelů (CAD) a schémata elektroniky
- Dokumentace kódu
- Testovací dokumentace
- Uživatelská dokumentace

Rozsah pracovní zprávy: 50 – 60 stran  
Rozsah grafických prací: dle potřeby  
Forma zpracování diplomové práce: tištěná

Seznam doporučené literatury:

1. TAI, Y. K., LIN, Y. H., & HWU, K. I. (2020). Smart Active Battery Charger for Prototypal Electric Scooter. In *International Symposium on Computer, Consumer and Control (IS3C)*, 268-271.
2. NIKOLAEV, D., DAVYDOV, V., & CHUBUR, N. (2021). Smart parking for electric scooter. In *The 5th International Conference on Future Networks & Distributed Systems*, 532-536.

Vedoucí diplomové práce: Mgr. Jakub Geyer  
Katedra informatiky

Datum zadání diplomové práce: 22. listopadu 2022

Termín odevzdání diplomové práce: 14. dubna 2023

12   
doc. Dr. Ing. Dagmar Škodová Parmová  
děkanka

JIHOČESKÁ UNIVERZITA  
V ČESKÝCH BUDĚJOVICÍCH  
EKONOMICKÁ FAKULTA  
Studentská 13 126  
370 05 České Budějovice

  
doc. RNDr. Jana Klicnarová, Ph.D.  
vedoucí katedry

V Českých Budějovicích dne 23. listopadu 2022

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury. Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě vzniklé vypuštěním vyznačených částí archivovaných Ekonomickou fakultou a elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

Datum 30.7.2023

Podpis studenta

## **Poděkování**

Tímto bych rád poděkoval svému školiteli Mgr. Jakubu Geyerovi za odborné vedení a směřování práce, za cenné rady k tématu a také za vždy velmi rychlé reakce na mé dotazy. Dále bych rád poděkoval své rodině za trpělivost a podporu při studiu.

# Obsah

1	Úvod .....	10
1.1	Cíle práce .....	11
1.2	Funkční požadavky .....	12
1.2.1	Funkční požadavky dobíjecí stanice .....	12
1.2.2	Funkční požadavky serverové části .....	12
1.3	Nefunkční požadavky .....	13
1.3.1	Nefunkční požadavky dobíjecí stanice.....	13
1.3.2	Nefunkční požadavky serveru .....	14
2	Analýza .....	15
2.1	Stávající řešení.....	15
2.1.1	Zamykání .....	15
2.1.2	Nabíjení .....	16
2.2	Rozdělení systému.....	18
2.2.1	Stanice .....	19
2.2.2	Doky .....	19
2.2.3	Server .....	19
2.3	IS STAG.....	19
2.3.1	Webové služby IS-STAG .....	19
2.3.2	Služba webového přihlášení.....	20
3	Serverová část .....	22
3.1	Analýza a návrh .....	22
3.2	HW prostředky .....	23
3.3	Implementace .....	23
3.3.1	Prerekvizity.....	23
3.3.2	MVC a router .....	23
3.3.3	Uživatelé a správa.....	24

3.3.4	API .....	26
3.3.5	Dvě jazykové verze systému .....	27
3.3.6	Grafické rozhraní .....	29
4	Stanice.....	31
4.1	Analýza a návrh .....	31
4.2	HW prostředky .....	33
4.2.1	Raspberry Pi .....	33
4.2.2	Dotykový displej .....	34
4.2.3	Čtečka karet.....	34
4.2.4	Napájecí zdroj.....	35
4.3	Implementace .....	36
4.3.1	Instalace systému a konfigurace displeje .....	36
4.3.2	Komunikační protokol pro doky .....	39
4.3.3	Redis .....	40
4.3.4	Skripty .....	40
4.3.5	Fyzické zapojení.....	44
4.3.6	Stanice .....	45
4.3.7	Grafické rozhraní pro uživatele.....	46
5	Dobíjecí dok.....	48
5.1	Analýza a návrh .....	48
5.2	HW prostředky .....	52
5.3	Implementace .....	55
5.3.1	Řídící jednotka a elektronika .....	55
5.3.2	Mechanická část pro zamykání.....	58
5.3.3	Rozhraní.....	61
5.3.4	Koloběžka .....	62
6	Celý systém.....	63
6.1	Párování ID karty .....	63

6.2	Modularita.....	64
6.3	Sestavený systém .....	64
7	Testování .....	68
8	Závěr.....	69
8.1	Budoucí rozvoj.....	69
8.2	Výstup.....	70
9	Summary .....	71
10	Seznam zdrojů .....	73
11	Seznam obrázků .....	76
12	Seznam tabulek.....	77
13	Seznam příloh.....	78



## Vysvětlení pojmů a zkratek

Pojem / zkratka	Vysvětlení
Chytrá dobíjecí stanice	Komplexní systém sestavený z jednotlivých částí (modulů) propojených do funkčního celku jako dobíjecí stanice.
Stanice	V práci chápáno jako modul s displejem, starající se o řízení nabíjení a komunikaci s uživatelem.
Dok	V práci chápáno jako modul se zámkem a elektronikou pro připojení a uzamknutí koloběžky.
HW	Hardware – Počítačové komponenty, mikrokontrolery, diskrétní elektronické součástky a mechanické díly.
SW	Software – Operační systémy, jednotlivé programy
3D	Objekty, které jsou prostorové – trojrozměrné.
3D CAD	Z angličtiny computer-aided design, ve spojení s 3D znamená design v trojrozměrném prostoru za užití počítačových programů.
API	Jedná se o zkratku pro Application Programming Interface, což v češtině znamená aplikační programové rozhraní.
GPS	Jedná se o zkratku pro Global Positioning System, což je satelitní navigační systém.
HTTP/HTTPS	Internetový protokol určený pro komunikaci s WWW servery, HTTPS je šifrovaný.
MVC	Architektura pro návrh aplikací.
URL	Zkratka znamená Uniform Resource Locator, a slouží k identifikaci přesného umístění informací na internetu.
Framework	Jedná se o vývojovou platformu, která slouží pro podporu programování, vývoje a organizaci softwarových projektů.
DIY	Zkratka znamená Do it yourself a jedná se o druh kultury, ve které si člověk sám zhotovuje užitečné výrobky, nahrazující komerční řešení.
GPIO	Zkratka znamená general-purpose input/output, česky univerzální vstupní/výstupní pin, který slouží pro připojování periférií.
IoT	Zkratka pro Internet of Things, česky internet věcí.
UID	Zkratka pro unikátní identifikátor zařízení.
I2C	Typ komunikačního protokolu hojně využívaného u mikrokontrolerů a periférií.

# 1 Úvod

Elektrické koloběžky se v dnešní době stávají stále více oblíbeným dopravním prostředkem, který nejen že šetří životní prostředí, ale také nabízí uživatelům pohodlnou a efektivní alternativu k tradičním automobilům. S tímto vzrůstajícím trendem však přicházejí i určité problémy, omezující jejich potenciál. V některých místech je například vjezd elektrických koloběžek zakázán a je obtížné najít vhodné místo pro jejich bezpečné zamykání a dobíjení.

Cílem této práce je představit konstrukci chytré dobíjecí stanice pro elektrokoloběžky, která bude instalována v Katedře informatiky PřF a která by některé problémy řešila a poskytovala uživatelům snadné a pohodlné řešení. Jedním z hlavních přínosů navrhovaného systému je implementace jednoduchého zamykání a odemykání elektrických koloběžek pomocí čipové karty. Tato technologie díky tomu umožní uživatelům pohodlné a bezpečné parkování koloběžek.

Dalším klíčovým prvkem, na který se zaměříme v rámci této práce, je implementace systému logování užívání elektrických koloběžek. Tato funkcionality umožní získat užitečné informace o aktivních výpůjčkách a dostupnosti koloběžek v reálném čase. Díky systému logování budou uživatelé schopni předem zjistit dostupnost elektrických koloběžek a efektivně plánovat svou cestu po kampusu.

Důležitým cílem navrhovaného systému je rovněž zajištění jednoduchého a bezproblémového nabíjení elektrických koloběžek. Chytrá dobíjecí stanice bude navržena tak, aby využívala stávající originální hardware od výrobce. Celý systém musí být navržen tak aby nepotřeboval zásahy do hardwaru samotné koloběžky, což zajistí, že uživatelé nebudou narušovat záruku ani bezpečnost svých zařízení.

Práce se zaměřuje na konstrukci a design chytré dobíjecí stanice pro elektrokoloběžky, která bude umožňovat snadné zamykání, logování a bezproblémové nabíjení. Cílem je poskytnout uživatelům pohodlné řešení, které jim umožní plně využít potenciál elektrických koloběžek pro dopravu nejen v kampusu Jihočeské Univerzity.

## 1.1 Cíle práce

Hlavním cílem práce je navrhnout a sestavit řešení pro chytrou dobíjecí stanici elektrokoloběžek pro Katedru informatiky PřF Jihočeské Univerzity. Stanice musí být optimalizována pro uživatele a musí umožnit jednoduché odemknutí na základě elektronického ověření pomocí čipové karty. Stanice musí být modulární, aby byla lehce rozšiřitelná a počet možných nabíjecích doků mohl být dále rozšiřován. Systém musí obsahovat logovací server pro zaznamenávání výpůjček. Vybrané hardwarové komponenty musí být bezpečné a vhodné pro sestavení stanice. Fyzická konstrukce bude zohledňovat praktičnost, bezpečnost a prostorové požadavky.

Z hlavního cíle následně vyplývají úkoly:

1. Rešerše existujících dostupných řešení a technologií pro uzamykání a chytré nabíjení elektrokoloběžek.
2. Výběr vhodných HW prostředků pro dobíjecí stanici a systém uzamykání.
3. Návrh a realizace vlastního řešení stanice s důrazem na univerzálnost a modularitu (včetně simulace v prostředí 3D CAD).
4. Návrh a realizace softwaru řízení stanice a obslužného software pro vzdálenou správu.
5. Nasazení a otestování prototypu.

Výstupem této práce bude modulární dobíjecí stanice pro elektrokoloběžky, která bude umožňovat jednoduché rozšíření. Součástí jsou 3D modely, programy i samotný prototyp stanice.

Druhým výstupem bude dokumentace testovacích scénářů, kde bude popsáno, jaké testovací scénáře byly použity a jaké byly jejich výsledky

Dalším výstupem bude uživatelská dokumentace popisující používání jak webového rozhraní, tak i samotné stanice.

Posledním výstupem bude poté samotná dokumentace pro nasazení, která slouží jako návod pro sestavení stanice a nasazení, nebo pro rozšíření.

## 1.2 Funkční požadavky

V této podkapitole se zaměříme na funkční požadavky, což jsou specifikace, které popisují požadované chování a funkční schopnosti systému, aplikace nebo zařízení. Tyto požadavky definují, jakým způsobem by měl systém pracovat a jaké funkcionality by měl poskytovat uživatelům.

### 1.2.1 Funkční požadavky dobíjecí stanice

- 1) Dobíjecí stanice musí umožňovat fyzické uzamknutí a odemknutí dobíjené elektrokoloběžky na základě elektronické identifikace, například pomocí čipu. Uživatelé budou mít možnost snadno identifikovat sebe a elektrokoloběžku, aby mohli provést její odemknutí pro použití nebo uzamknutí pro dobíjení.
- 2) Stanice musí být schopna optimalizovat nabíjení baterie elektrokoloběžky na základě jejího aktuálního stavu nabití. Stanice musí monitorovat úroveň nabití, aby došlo k vypnutí nabíječky v momentě, kdy bude koloběžka nabitá.
- 3) Uživatelé musí mít možnost jednoduše a rychle vkládat elektrokoloběžky do dobíjecí stanice a vyjímat je. Chytrá dobíjecí stanice bude poskytovat intuitivní rozhraní, které umožní snadné fyzické umístění elektrokoloběžek do nabíjecích stanic a jejich bezpečné vyjmutí po dobíjení.
- 4) Dobíjecí stanice bude připravena na budoucí paralelní nabíjení minimálně 4 elektrokoloběžek současně. V ideálním případě omezené pouze fyzickými možnostmi místnosti, kde se stanice bude nacházet.

### 1.2.2 Funkční požadavky serverové části

- 1) Systém musí umožnit přihlášení uživatele s pomocí správných přihlašovacích údajů. Uživatelé budou mít možnost se přihlásit pomocí svého uživatelského jména a hesla nebo pomocí přihlášení za užití identity STAG.
- 2) Systém musí poskytovat aktualizované informace o volných docích, kde uživatelé mohou vrátet elektrokoloběžky, a o volných elektrokoloběžkách, které si mohou vypůjčit.
- 3) Systém musí uchovávat a zobrazovat historii výpůjček. Uživatelé budou moci získat přehled o svých předchozích výpůjčkách, administrátor poté o všech výpůjčkách, včetně uživatele, času výpůjčky, konkrétní koloběžce a místa, kde byla vypůjčena.

- 4) Systém by měl poskytovat možnost změnit jazyk uživatelského rozhraní. Uživatelé budou mít možnost vybrat si preferovaný jazyk, ve kterém chtějí používat systém.
- 5) Správci systému by měli mít možnost upravovat nastavení doků, koloběžek a uživatelů.
- 6) Systém musí poskytovat rozhraní pro komunikaci se stanicí (API).

## 1.3 Nefunkční požadavky

V této podkapitole se podíváme naopak na nefunkční požadavky, které se zaměřují na vlastnosti systému, které nejsou přímo spojené s jeho funkcí. Tyto požadavky popisují různé aspekty, jako je výkon, spolehlivost, bezpečnost, dostupnost, udržovatelnost, použitelnost, kompatibilitu a další.

### 1.3.1 Nefunkční požadavky dobíjecí stanice

- 1) Nabíjecí stanice musí být spolehlivá a dostupná ideálně 24/7. Minimalizace výpadků a chyb je klíčová pro neustálou dostupnost nabíjecí stanice pro uživatele.
- 2) Dobíjecí stanice musí poskytovat rychlou a efektivní odezvu na elektronickou identifikaci a ovládání elektrokoloběžek. Pro uživatele bude klíčová rychlá reakce po přiložení čipu nebo provedení jiné interakce s nabíjecí stanicí.
- 3) Stanice musí poskytovat bezpečné uzamknutí elektrokoloběžek a ochranu před neoprávněným přístupem.
- 4) Uživatelská přívětivost grafického rozhraní. Uživatelské rozhraní dobíjecí stanice musí být intuitivní a snadno použitelné.
- 5) Proces vkládání a vyjímání elektrokoloběžek by měl být rychlý a jednoduchý, aby se minimalizovalo zdržení uživatelů.
- 6) Škálovatelnost, systém by měl být navržen tak, aby se mohl rozšiřovat a přizpůsobovat rostoucímu počtu uživatelů a elektrokoloběžek.

### 1.3.2 Nefunkční požadavky serveru

- 1) Serverová část systému by měla být optimalizována pro rychlou odezvu.
- 2) Serverová část systému by měla poskytovat vysokou úroveň zabezpečení dat a ochranu před neoprávněným přístupem. To zahrnuje implementaci bezpečného přihlášení a ověřování uživatelů, šifrování přenosu dat a hashování citlivých dat v databázi.
- 3) Serverová část systému by měla být napsaná tak aby umožňovala dodatečné rozšiřování.
- 4) Systém by měl poskytovat líbivé uživatelské rozhraní s možností překladu.
- 5) Systém by měl poskytovat stejnou funkcionalitu pro desktop zařízení tak i pro mobilní zařízení (responzivní design)

## 2 Analýza

V této kapitole nalezneme obecnou analýzu obsahující současně dostupné řešení, a obecnou analýzu rozložení systému. Další analýza je již ke konkrétním částem systému a je obsažena v dalších kapitolách.

### 2.1 Stávající řešení

#### 2.1.1 Zamykání

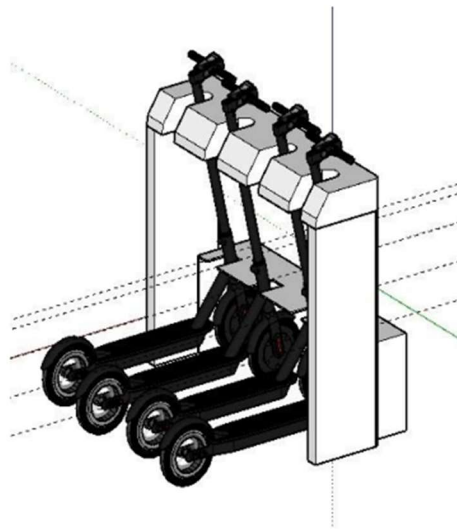
Analýza současných řešení zabezpečení elektrických koloběžek začíná zkoumáním způsobů, jak lze zabezpečit samotné zamknutí koloběžky. Existuje několik přístupů, přičemž nejzákladnějším rozdělením je mezi softwarová a fyzická opatření. Softwarově lze koloběžku zamknout tak, že je nemožné ji spustit, nebo zablokovat její elektrický motor. Toto lze realizovat pomocí informací z GPS nebo jiných senzorů. [1] Například koloběžky BOLT disponují alarmem, který se aktivuje, pokud je s koloběžkou manipulováno bez předchozího odemknutí. [2] I když je to realizováno formou zvukového varování, nedokáže zabránit samotné krádeži. Kvůli tomu vzniká značná nevýhoda softwarových řešení, protože na rozdíl od plnohodnotných dopravních prostředků, jako jsou například elektroauta, u kterých může dojít k celkovému uzamknutí, u elektrických koloběžek je možné i přes zamezení otáčení kol jednoduše koloběžku zvednout a odnést, díky její malé váze. Na trhu existuje mnoho různých typů koloběžek, jejichž hmotnost se pohybuje od lehkých modelů vážících přibližně 8 kg až po těžší varianty přesahující 20 kg. [3] [4]

Elektro-koloběžka	Váha
<b>Xiaomi Mi Electric Scooter Pro 2</b>	14,2 kg
<b>Ninebot Kick scooter MAX G30 II</b>	18 kg
<b>Xiaomi Mi Electric Scooter Pro 4 EU</b>	17 kg
<b>Sencor Scooter One S20</b>	12.5 kg
<b>Street Surfing Voltaik MGT 350</b>	13 kg

Tabulka 1 - Výběr prodávaných elektrokoloběžek s váhou (Zdroj: Autor)

Druhá varianta je fyzické zamykání koloběžky podobně jako je běžné zamykat kola. Na trhu je mnoho řešení zámků, od nejjednodušších lanových až po zámkové z kalené oceli. [5] Ovšem i když má uživatel zámek je potřeba najít vhodné místo kde koloběžku

uzamknout. K tomu lze využít ploty, lampy, nebo stojany na kola. V dnešní době jsou dostupná i komerční řešení stojanů pro elektrokoloběžky které umožňují bezpečné uzamknutí koloběžky. Ať už se jedná o konstrukční jednoduché řešení s využitím vlastních zámků, nebo složitější automatické zámky které tvoří stanici pro koloběžky. Tato stanice může být univerzální, kdy se zamyká koloběžka okolo krku, nebo specifická pro konkrétní model koloběžky, kdy toto řešení mohou využívat například poskytovatelé koloběžek na výpůjčky. To je poté řešeno speciální objímkou na krku koloběžky, která se zasouvá do zámku. [6] [7] [8]



*Obrázek 1- Schéma prototypu dobíjecí stanice z univerzity z Petrohradu*

*(Zdroj: C. K. D. Nikolaev [6])*

### 2.1.2 Nabíjení

Druhou neméně důležitou částí pro analýzu je způsob nabíjení. V současné době není nijak standardizováno nabíjecí napětí elektrokoloběžek, a tedy je možné se setkat s různými úrovněmi nabíjecího proudu. Kvůli tomu není možné mít univerzální nabíjecí stojan jako je například pro elektroauta. Každý výrobce může mít vlastní způsob nabíjení, připojení a řízení nabíjecího cyklu.



V současné době existují specifické řešení pro konkrétní typy elektrických koloběžek. Tato řešení jsou konstruována obvykle společnostmi, které provozují elektrokoloběžky jako dopravní prostředek pro vypůjčení. Jako příklad si můžeme uvést dobíjecí stanici firmy KNOT, která poskytuje možnost nabíjení jak jejich koloběžek, tak elektrokol. [7]



Obrázek 2 - Dobíjecí stanice KNOT (Zdroj: knotcity.com)

Společnost BOLT na to šla jinak a využívá pro dobíjení svých elektrokoloběžek jednoduchý mechanismus. Totiž BOLT nenabíjí elektrokoloběžky, protože jeho koloběžky mají vyjímatelné baterie což umožní rychlou výměnu baterie a následně otevírá možnost dobíjet všechny baterie centrálně. [2]



Obrázek 3 - Elektrokoloběžka BOLT5 (Zdroj: bolt.eu)

U nabíjení je také potřeba řešit způsob napájení, jestli dobíjecí stanice má přístup k elektrickým rozvodům, nebo, jestli je nutné ke stanici přivést zvlášť rozvody. Případně se nabízí možnost využití obnovitelných zdrojů, jako je například solární energie, což nejenže ulehčí životnímu prostředí, ale také sníží náklady na nabíjení elektrických dopravních prostředků. [9] [10]



Obrázek 4 - Dobíjecí stanice s využitím solární energie (zdroj: [9])

## 2.2 Rozdělení systému

Z předchozích požadavků vyplývá, že systém bude relativně komplexní a bude zahrnovat několik různých oblastí. Bude vyžadovat vývoj softwaru pro webové rozhraní, programování mikrokontrolerů pro zámky a návržení vhodné konstrukce pro celou dobíjecí stanici. V dnešní době se nejen v IT sféře stále více tlačí na rozdělování systémů do menších celků, které umožňují zvýšit modularitu systému, díky které se systémy stávají univerzálnější, lehce rozšiřitelné, nebo i snáze opravitelné. Proto by bylo ideální řešením rozdělit systém na několik menších částí což také umožní lepší modularitu celé dobíjecí stanice. [11]

Na první pohled přichází v úvahu rozdělit systém na webovou aplikaci a poté na samotný hardware, ovšem pro lepší modulární přístup bude lepší systém rozdělit na celkem tři hlavní části, na samotné doky, ve kterých bude koloběžka, na dobíjecí stanici komunikující s doky a poté na samostatný server, kde bude nasazena webová aplikace, se kterou bude stanice komunikovat. Tímto rozdělením lze dosáhnout optimalizace a efektivity celého systému.

## 2.2.1 Stanice

Hlavním prvkem celé dobíjecí stanice by měl být samotný modul stanice, který umožní plné ovládání všech doků koloběžek přiřazených k dané stanici. Ačkoliv se v současné době objevují studie ohledně lepší efektivity využití klasických tlačítek místo dotykového displeje v autech, v jiných aplikacích, primárně v chytré elektronice nebo při užití v kiosku, mají dotykové obrazovky větší smysl. A to hlavně díky své možnosti dynamicky zobrazovat ovládací prvky. Pro uživatelskou přívětivost by bylo tedy vhodné implementovat dotykový displej, který umožní snadné ovládání všech funkcí stanice a zároveň poskytne líbivý design rozhraní. [12] [13]

## 2.2.2 Doky

Na hlavní prvek, tedy na modul stanice bude nutné připojit modul, který se bude starat o zamknutí a nabíjení koloběžky. K tomu bude sloužit právě modul doku. V ideálním případě bude možné připojovat větší množství doků k jedné stanici a díky tomu docílit modulárnosti. Dok bude obsluhovat vždy jednu koloběžku.

## 2.2.3 Server

Poslední částí systému musí být samotný server, který bude sloužit pro běh webové aplikace umožňující nahlížet na aktuální výpůjčky nebo na volné koloběžky. Samotný server bude také obsahovat veškerá data o nastavení a konkrétním využití koloběžek. Důležitým prvkem také je, aby se server staral o logování dat, a tedy sjednocoval záznamy ze všech stanic, to je z bezpečnostního hlediska lepší varianta, než kdyby se logy ukládaly jen lokálně na každou konkrétní stanici. [14]

## 2.3 IS STAG

Vzhledem k požadavku, aby se mohli uživatelé systému přihlašovat i svou identitou ze systému STAG, je zapotřebí systémy nějakým způsobem propojit. STAG podporuje Rest API pro získávání dat a také poskytuje službu pro webové přihlášení pro externí aplikace.

### 2.3.1 Webové služby IS-STAG

Webové služby STAG jsou rozhraním pro komunikaci s informačním systémem STAG. Tyto služby jsou postaveny na architektuře REST a využívají standardní HTTP nebo HTTPS protokol. Jejich hlavním účelem je poskytovat rozhraní pro dotazování, vkládání, aktualizaci a odstraňování dat v systému STAG.

Pro zajištění bezpečnosti a ochrany dat jsou webové služby STAG obvykle zabezpečeny autentizací a autorizací. To znamená, že se uživatel musí před přístupem k těmto službám prokázat svou identitou a mít přidělená příslušná oprávnění. Tím je zajištěno, že pouze oprávnění uživatelé mají přístup k citlivým datům a že jsou dodržována správná práva přístupu.

Pro vývojáře je k dispozici dokumentace k API, která obsahuje informace o dostupných metodách, parametrech a odpovědích. Tato dokumentace a API slouží ke správné implementaci a integraci s informačním systémem STAG. [15]

Každá implementace systému STAG může mít vlastní úpravy, proto je zapotřebí získávat informace z lokálních zdrojů o dostupných API end pointech a datových strukturách které poskytují. V případě této práce bude zapotřebí pouze dva API end pointy a to:

```
stag-ws.jcu.cz/ws/services/rest2/help/getStagUserForActualUser
```

Který přijímá jako parametr osobní číslo a vrací údaje o aktuálně přihlášeném uživateli a poté ještě:

```
https://stag-ws.jcu.cz/ws/services/rest2/help/invalidateTicket
```

Tento API end point přijímá jako parametr ticket a poté ho deaktivuje. [16]

### 2.3.2 Služba webového přihlášení

Pro využívání API a získání osobních údajů je třeba komunikovat s API jako přihlášený uživatel, to lze pomocí služby webového přihlášení. Přihlášení je umožněno pomocí využití takzvaného uživatelského ticketu, který je generován serverem jako náhodný řetězec. Při úspěšném volání webové služby, která vyžaduje přihlášení, server vrátí speciální COOKIE s názvem WSCOOKIE obsahující uživatelský ticket. Tuto COOKIE lze předat při dalším volání serveru, čímž dojde k přihlášení.

Pokud externí aplikace chce přihlásit svého klienta a získat jeho ticket pro komunikaci s modulem webových služeb, může přesměrovat uživatele na speciální adresu serveru webových služeb a předat URL, na které bude následně uživatel přesměrován po přihlášení. V případě STAGu na jihočeské univerzitě je to adresa:

```
https://stag-ws.jcu.cz/ws/login?originalURL=
```

Server webových služeb vygeneruje uživatelský ticket a přesměruje uživatele na původní adresu, kde se do URL přidávají parametry, jako je „*stagUserTicket*“ obsahující již zmíněný uživatelský ticket a „*stagUserInfo*“ zakódované pomocí BASE64 obsahující informace o přihlášeném uživateli, včetně seznamu jeho rolí v systému IS/STAG. [15]  
[16]

## 3 Serverová část

Nyní se pojdme podrobně podívat na samotnou serverovou část. V této kapitole provedeme analýzu a následný návrh celého systému serveru, vybereme vhodný hardware a poté budeme systém implementovat.

### 3.1 Analýza a návrh

Jak bylo již v předešlé kapitole řečeno, server se musí starat o uchování a zobrazování dat a musí umožnit jejich editaci. To ovšem ani zdaleka nedefinuje rozsah celého systému, je totiž zapotřebí aby systém umožňoval jak přihlašování lokálním účtem, tak přihlašování pomocí identity z IS STAG a aby pro administrátora systému umožnil upravovat a konfigurovat prvky systému, ať už to jsou jednotliví uživatelé nebo například nastavení doků či koloběžek. Velmi důležitým prvkem serveru je i API, které bude nutné pro komunikaci samotné dobíjecí stanice se serverem.

Pro samotné naprogramování je dobré se řídit některým z návrhových vzorů, jako je například architektura MVC. Architektura MVC, která si klade za cíl oddělit logiku aplikace od zobrazení se skládá z modelu (Model), pohledu (View) a kontroleru (Controller). Model je zodpovědný za reprezentaci dat a systémovou logiku aplikace. Obsahuje datové struktury a metody pro manipulaci s daty. Pohled se stará o vizuální prezentaci dat uživateli. Zobrazuje informace získané z modelu a poskytuje uživateli grafické rozhraní, prostřednictvím kterého může interagovat s aplikací. Kontrolér řídí komunikaci mezi modelem a pohledem. Přijímá vstup od uživatele a na základě toho provádí odpovídající akce. [17]

Pro samotné napsání aplikace byl zvolen jazyk PHP jakožto velmi populární jazyk umožňující snadnou práci s databázemi, flexibilitu a jednoduchou rozšiřitelnost. Pro psaní aplikace si můžeme zvolit cestu využití hotového frameworku na kterém se může systém postavit, což může ulehčit proces psaní, protože frameworky obvykle již obsahují hlavní logiku uživatelských účtů, ověřování a routování požadavků. Mezi známé PHP frameworky můžeme zařadit například *laravel* nebo *symfony*. Existuje druhá možnost při psaní systému, a tou je vytvořit celý systém od základu. Tato možnost má svou nevýhodu v tom, že je nutné psát veškerý kód znovu, ale přináší výhodu v lehkosti kódu. Tím se systém zbavuje zbytečností, které by byly potřebné pro samotnou funkci frameworku. [18] [19]

Na začátku byla serverová část plánována pro běh na Raspberry Pi, a proto byla zvolena cesta vytvořit vlastní lehkou webovou aplikaci a nevyužívat zbytečně obalený framework.

## 3.2 HW prostředky

Serverová část systému nepotřebuje speciální hardwarové vybavení jako ostatní části systému. Hlavním požadavkem je, aby bylo možné nainstalovat webový server s PHP a databází. Je nutné, aby zařízení bylo připojené k internetu, ideálně pomocí kabelového připojení nebo stabilního bezdrátového.

Vzhledem k současným cenám elektrické energie je ideální variantou využít mikropočítač jako je například Raspberry Pi nebo jiný podobný. Na něm lze provozovat webovou aplikaci i databázi bez jakéhokoli problému. Ovšem systém je navržený tak, že je jedno na jakém hardwaru je nainstalovaný, důležité je, aby splňoval pouze základní požadavky. V současném reálném nasazení se počítá s úsporným nasazením přímo na HW dobíjecí stanice. [20]

## 3.3 Implementace

V této kapitole se podíváme na nejdůležitější části implementace. Celý kód systému lze nalézt v příloze této práce. Podíváme se na přihlašování, řešení API a také na samotné grafické rozhraní.

### 3.3.1 Prerekvizity

Aby mohla serverová část systému fungovat je zapotřebí nainstalovat prerekvizity. Byl zvolen operační systém na jádře Linuxu. Při vývoji CENTOS v nasazení poté *Raspbian OS*. Pro správné fungování celé aplikace je zapotřebí mít nainstalovaný webový server, například *Apache*, nebo *Nginx*. Vzhledem k tomu že je aplikace psaná v PHP tak je nutné do webového serveru doinstalovat modul pro PHP. Pro databázi byla zvolena *MariaDB*. Posledním krokem je povolení připojení na webserver pomocí firewallu.

### 3.3.2 MVC a router

Celá aplikace je psána pomocí návrhového vzoru, kdy se snažíme oddělit logiku dat od prezentační logiky. Veškeré úkony spojené s prací nad databází a ověřování uživatelů probíhá v jednotlivých modelech, kdy každý prvek má svůj vlastní model. Například *UserModel.php* pro uživatele, *rentalModel.php* pro výpůjčku a tak dále.

O zobrazování dat se stará pohled (View), kdy každá stránka webové aplikace má svůj vlastní pohled. Pohledy jsou ve formátu *homepage.phtml* pro úvodní stránku, *userDashboard.phtml* pro uživatelskou nástěnku a podobně. Pohledy neobsahují žádnou logiku práce s daty, pohledy pouze dostávají již zpracovaná data a jediné o co se musí postarat je správné zobrazení dat.

Pro propojení pohledů a modelů je zapotřebí kontrolér (Controller), který přijímá vstupy od uživatele a zpracovává je. Kontrolér na základě uživatelské interakce vyvolává metody modelu a získává z něj data, která následně předává pohledům pro zobrazení.

## **Router**

Ovšem jak server bude vědět jaký kontrolér použít pro interakci s uživatelem? To řeší právě router, což je speciální část aplikace, která se stará o směrování požadavků uživatele na základě url adresy. Webový server je nakonfigurován tak, aby veškeré url adresy domény směřoval na „*index.php*“ který inicializuje router a obsahuje konfiguraci cest (*routes*). Cesty jsou nakonfigurované následným formátem:

```
$router->addRoute('/login', 'loginController', 'index');
```

V tomto příkazu přidáváme routeru novou cestu, která je pro url „*/login*“, vyvolává kontrolér „*loginController*“ a konkrétně metodu „*index*“. Pro správné nakonfigurování webserveru je v kořenovém adresáři webové aplikace soubor „*htaccess*“ který upravuje konfiguraci webového serveru pro danou webovou aplikaci. Tento konfigurační soubor je součástí přílohy.

### **3.3.3 Uživatelé a správa**

System je navržen tak aby umožňoval přihlašování jak lokálním účtem, tak i účtem ověřeným z identity STAG. Samotní uživatelé také mohou mít role, které definují jejich pravomoci. V první části se podíváme na přihlašování a správu uživatelů v lokálním systému.

#### **Lokální uživatelé**

Lokální uživatel systému je takový uživatel, který má uložena všechna data o sobě v lokální databázi a jeho ověřování probíhá přímo v systému. Struktura uživatelského účtu v databázi je popsána tabulkou níže.



Tabulka uživatele	
<b>id</b>	int
<b>isExternal</b>	tinyint
<b>name</b>	varchar
<b>surname</b>	varchar
<b>username</b>	varchar
<b>email</b>	varchar
<b>password</b>	varchar
<b>role</b>	Enum(user,admin,visitor)
<b>card_id</b>	varchar
<b>scooterCount</b>	int
<b>lang</b>	varchar

*Tabulka 2 - Tabulka uživatele v DB (Zdroj: Autor)*

Při implementaci přihlašování bylo důležité zabezpečit samotnou aplikaci proti útokům zvenčí. Proto byly zavedeny opatření proti SQL injection pomocí připravených a parametrizovaných dotazů, nutnost vyplnit všechny údaje, ošetření proti CSRF ale také detail jako například, že se po nesprávném pokusu o přihlášení nevypíše, jestli je špatné uživatelské jméno nebo jen heslo. Díky tomu útočník nemůže nejprve zkusit uživatelská jména a pak když by jedno našel tak dál pomocí útoku bruteforce provést útok na heslo. Bruteforce je také další útok který byl třeba ošetřit. To se povedlo implementací mechanismu opakovaných pokusů o přihlášení, který funguje na principu pětkrát špatné přihlášení se rovná 15 minut nemožnosti se přihlásit. [21]

Lokální uživatelský účet má dvě role, buďto klasický uživatel, tedy uživatel koloběžek, který si po přihlášení může prohlížet historii pěti posledních výpůjček nebo pozměnit některá nastavení účtu, jako je vytvořit nové heslo, změnit si Jméno a Příjímání nebo přidat či odebrat ID karty.

Druhou rolí systému je administrátor, což je role uživatele, která umožňuje správu celého systému. Administrátor může přidávat nové uživatele, upravovat stávající nebo je odebírat. V případě že je v systému více druhů uživatelů, jako je lokální uživatel a uživatel ze STAGu, tak pro různý typ uživatele jsou umožněné různé úpravy. Nejdůležitější úpravou je ovšem možnost povolit půjčování koloběžek, zde je možnost určit počet koloběžek které může konkrétní uživatel mít půjčených najednou. To

umožňuje, aby si jeden vyučující mohl půjčit více koloběžek, například jednu pro sebe a druhou pro návštěvu. Administrátor dále může procházet kompletní historii výpůjček, může sledovat aktuální výpůjčky, upravovat základní parametry jednotlivých koloběžek přidávat modely koloběžek a upravovat dva parametry doků. Jsou to limit pro detekci nabití a poté časovač, který definuje zpoždění za jak dlouho po přesáhnutí limitu na nabíjení se nabíječka vypne. Časovač je v minutách. Administrátor ani žádný jiný uživatel nemůže přidat ani odebrat koloběžku ani dok, ty se přidávají automaticky při prvním připojení do stanice. Administrátor může pouze koloběžku nebo dok deaktivovat. Díky tomu se zajistí že administrátor nebude moci smazat koloběžku a tím narušit záznamy o výpůjčkách.

### **Uživatelé ze systému STAG**

Druhou variantou uživatelských účtů v systému jsou uživatelé STAG. Tento typ uživatele se vyznačuje tím, že se neověřuje na straně lokálního systému, ale vyžaduje přihlášení na stránkách „STAG – služba webového přihlášení“. Po úspěšném přihlášení je uživatel přesměrovaný zpět na web lokálního systému.

Lokální systém získá jako parametr od systému STAG token pro komunikaci s API STAG, ID uživatele a základní informace o uživateli zakódované v BASE64. Systém následně data načte a provede ještě jednou dotaz na API STAG, zda konkrétní uživatel opravdu existuje a jaké parametry má (OS číslo, email, jméno a příjmení). To je prováděno z důvodu, kdyby útočník zkusil podstrčit, nebo změnit odpověď od služby webového přihlášení STAG, která předává úvodní informace jako parametry v URL. API systému STAG, jež vrací odpovědi bezpečnější formou, a to metodou POST. Pro lepší zabezpečení je ovšem nutností využít komunikaci pomocí protokolu HTTPS. Po tom, co systém ověří že uživatel existuje v systému STAG, proběhne ověření, zda daný uživatel existuje v lokální databázi, pokud ano, přihlásí ho, pokud ne, vytvoří nový záznam a přihlásí ho.

### **3.3.4 API**

Jak již bylo v předešlém textu zmíněno, koloběžky, doky i stanice se do systému přihlašují a tím dojde k automatickému vytvoření instancí v databázi. K tomu je ale zapotřebí určité komunikační rozhraní. Proto bylo naprogramováno vlastní API, umožňující komunikaci pomocí požadavků POST od stanice. Pro správnou funkčnost API je potřeba znát API klíč, API end point a správně definovat dotaz. [22]

## API endpoint

API endpointy jsou URL adresy, které slouží k interakci s určitou funkcí nebo datovým zdrojem v rámci API. Tyto adresy umožňují specifikovat jaká data chceme získat nebo odeslat z nebo do API, a to v jasně definovaném formátu. Níže je příklad použití jednoho API end pointu pro registraci koloběžky. Ostatní API end pointy jsou popsány v příloze práce.

<b>Registrace koloběžky</b>	
Api end point slouží k vytvoření nové koloběžky, v případě již existence koloběžky v databázi dojde k její úpravě, nebo aktivaci či deaktivaci.	
<b>/api/registerScooter</b>	
<b>Metoda</b>	<b>POST</b>
<b>Parametry</b>	<ul style="list-style-type: none"><li>- <b>api_key</b> (string): API klíč</li><li>- <b>scooter_id</b> (bigInt): ID koloběžky</li><li>- <b>action</b> (string): akce -&gt; "register"</li><li>- <b>is_active</b> (bool): status koloběžky reprezentován jako 1/0</li><li>- <b>battery_level</b>: stav baterie, defaultně 0</li><li>- <b>model</b>: model koloběžky</li></ul>
<b>Odpověď</b>	- „ <b>success</b> “ pokud se podařilo koloběžku vytvořit, upravit, nebo změnit stav, jinak „ <b>fail</b> “

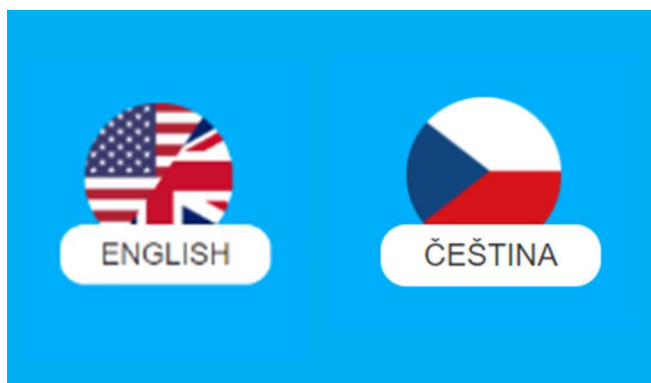
*Tabulka 3 - API endpoint pro registraci koloběžky (Zdroj: Autor)*

Díky tomuto API mohou se systémem komunikovat další stanice a tím se zlepšuje modularita systému. Lze mít jeden centrální systém – server se kterým může komunikovat více stanic.

### 3.3.5 Dvě jazykové verze systému

Původně byl systém navržen jako jednojazyčný s využitím češtiny jakožto rodného jazyka. Ovšem vzhledem k užití dobíjecí stanice v akademické sféře, kde se mohou nacházet cizojazyční uživatelé, bylo vhodné vytvořit další anglickou verzi.

Všechny texty v systému jsou ukládány zvlášť do *json* souboru a jednotlivé texty na webu jsou načítány pomocí vlastní třídy „*textManager*“ starající se o načítání správného jazyka a správného konkrétního stringu. Samotné přepínání mezi jazyky je umožněno díky jednoduchému tlačítku na každé stránce vpravo dole, které v závislosti na aktuálním jazyku nabízí změnu na jiný.



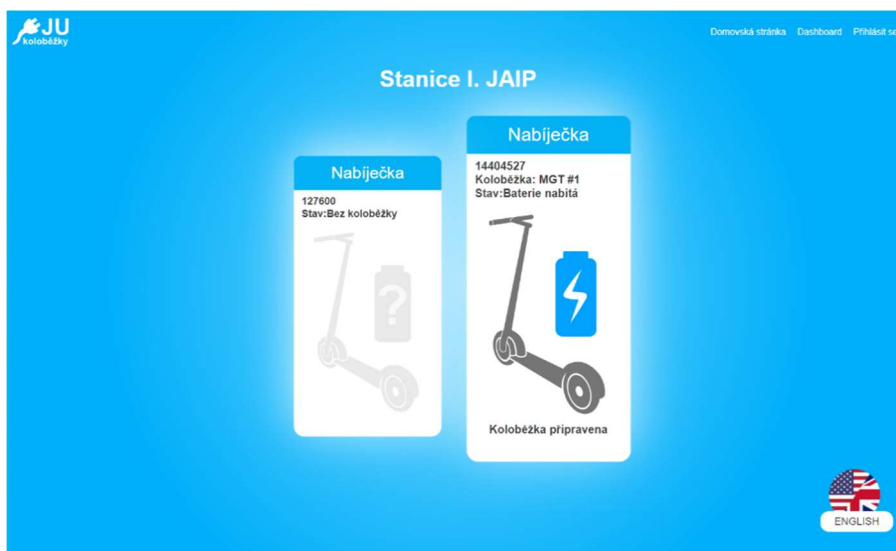
*Obrázek 5 - Obě varianty tlačítka pro přepnutí jazyka  
(Zdroj: Autor)*

Po kliknutí na tlačítko se zapíše cookies s hodnotou jazyka. Defaultně je v systému nastavená čeština. Díky tomuto principu s využitím textových souborů obsahujících veškeré texty v systému, je možné bez větších úprav přidat další textové mutace.

Systém je navržen tak, že pokud je uživatel přihlášený a změní si jazyk, dojde k zapsání nastavení do databáze a při dalším přihlášení se jazyk automaticky přenastaví na uživatelem preferovaný. Díky tomu lze informaci o preferovaném jazyku předávat pomocí API i stanicí a tím umožnit automatické přepnutí jazyka stanicí.

### 3.3.6 Grafické rozhraní

Správně fungující systém je pouze jedna část řešeného zadání. Systém bez uživatelsky přívětivého grafického rozhraní nemůže fungovat. Proto byla vypracována kompletní vizuální identita systému, kde dominantní barva je světle modrá v kombinaci s bílou barvou. Tato kombinace barev působí uklidňujícím a líbivým dojmem. Celý design je proveden v jednoduchém stylu připomínající materiál design. Všechna loga, ikony a ilustrace jsou originálním dílem autora a jsou ve vektorovém formátu. [23] Na obrázku níže můžeme vidět hlavní stránku webového rozhraní ukazující dostupné doky a koloběžky. Podrobnější popis ostatních stránek lze naléznout v příloze „Uživatelská dokumentace“ v sekci webové rozhraní.

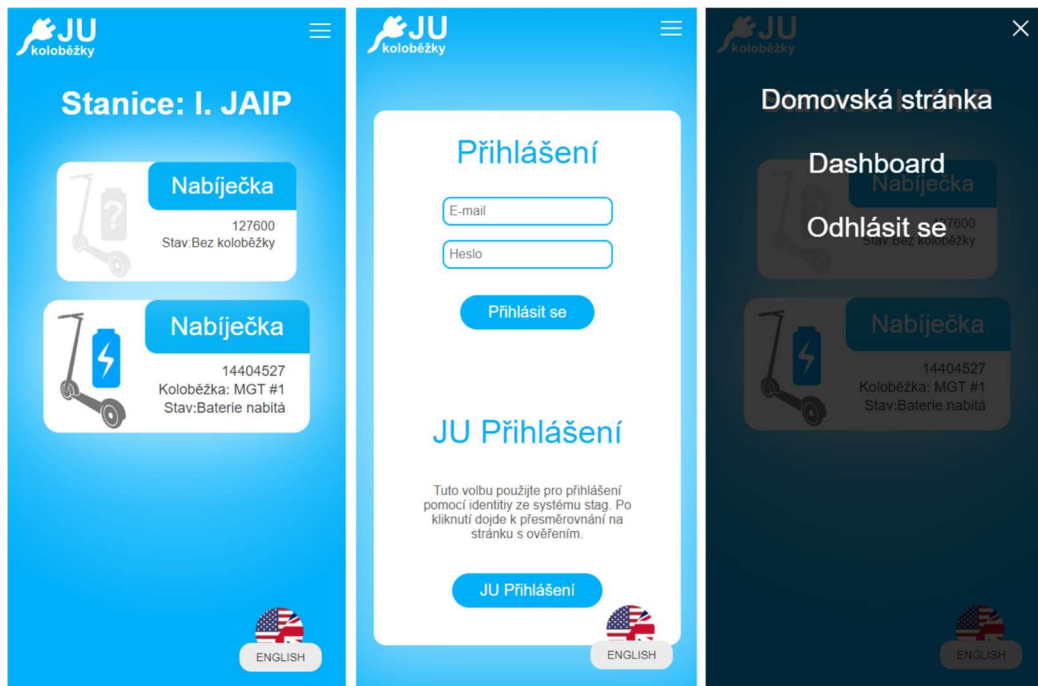


Obrázek 6 - Grafické rozhraní webové stránky, hlavní stránka  
(Zdroj: Autor)

#### Responzivní verze stránek

Celý web je také napsán s důrazem na intuitivní a přátelské prostředí pro použití s mobilním telefonem. Některé části, které jsou primárně určené pro mobilní zařízení jako je hlavní stránka, kde se počítá s nahlížením, zda je volná koloběžka, jsou upravené od základu tak, že se vizuál celkově mění oproti desktop verzi. Jiné stránky, které jsou primárně cílené na desktop verzi, jako je administrátorská správa uživatelů, kde se počítá s velkými tabulkami, je upravena stávající struktura, ale s přihlédnutím na šířku obrazovky. Na obrázku níže lze vidět hlavní stránku se zobrazením dostupných

koloběžek, poté stránku přihlášení a vpravo lze vidět menu mobilní verze které se schovává pod tzv „hamburger ikonou“.



Obrázek 7 - Mobilní verze webové stránky (Zdroj: Autor)

## 4 Stanice

Samotný server nemůže plnit všechny požadavky chytré dobíjecí stanice. Proto je zde druhá část a tou je stanice. V této kapitole se podíváme na analýzu a samotný návrh celé stanice, na hardwarové řešení a následně na celou implementaci od samotného uživatelského rozhraní až po komunikační protokoly.

### 4.1 Analýza a návrh

Stanice má plnit roli určitého prostředníka mezi jednotlivými doky, serverem a uživateli, a je klíčová pro poskytování uživatelského rozhraní. Cílem je poskytnout uživatelům přívětivé rozhraní, které by ideálně mělo být ovladatelné pomocí dotykového displeje. Důležitým prvkem je také zařízení pro čtení karet. Zároveň je nezbytné řešit komunikační protokol a zapojení jednotlivých doků ke stanici.

Pro ovládání stanice můžeme zvolit několik možností. První možností je použití dotykového displeje, pro který bude nutné naprogramovat odpovídající grafické rozhraní. Druhou možností je mít stanici bez fyzického rozhraní a pouze u každého zámku umístit čtečku karet pro odemykání a zamykání konkrétních koloběžek. Tato možnost by však výrazně omezila funkčnost celé dobíjecí stanice. Třetí možností je využití webového rozhraní, kde by uživatel u stanice načel například QR kód nebo NFC tag telefonem a byl by přesměrován na rozhraní, kde by si mohl půjčit nebo vrátit koloběžku. Tato možnost by umožnila využívat i pokročilejší funkce, avšak oddělení fyzického rozhraní od stanice by mohlo způsobit zbytečné rozdrobení systému. Proto byla zvolena první možnost, tedy použití dotykového displeje s vlastním rozhraním.

Identifikace uživatele bude probíhat za pomoci již existujících univerzitních karet. Univerzita využívá „Jednotný Identifikační Systém“, ve zkratce JIS, který je kompatibilní s čipy pracujícími na frekvenci 125 kHz a 13,56 MHz. Mezi podporované čipy patří:

- H4002 [24]
- e5560
- e5561 [25]
- MIFARE [26]
- DESFire [27]

V České republice obsahují školní čipové karty od roku 2017 dva typy čipů pro zajištění největší kompatibility, a to Mifare 1Kb a EM4102 [28]. Proto je důležité, aby čtečka u nabíjecí stanice byla kompatibilní s alespoň jedním z uvedených standardů. [29] [30]

Další částí je samotný způsob zapojení doků a způsob komunikace. Nabízí se zde několik možností. První je, že doky budou obsahovat jen nejdůležitější elektroniku pro odemknutí a budou se ovládat přímo přívodem napětí od stanice. To je jednoduché na konstrukci, ovšem vznikají dva problémy, jedním je, že jakýkoliv zvolený hardware, ať už je to Raspberry Pi nebo jiný mikropočítač, stanice bude mít omezené množství volných portů pro připojení periférií kvůli kterému bude také omezené množství možných doků. Druhým problémem je samotná jednoduchost tohoto řešení, která by umožnila potencionálnímu útočníkovi přivést pouze napětí k doku a dok by se odemkl. Lepším řešením by bylo, aby dok obsahoval nějakou chytrou elektroniku, která by komunikovala se stanicí a stanice by tedy mohla jen odesílat signály na konkrétní fyzický port. Díky tomu odpadá riziko odemknutí pouze za pomoci přivedení napětí, ale stále zůstává omezení na počet volných portů na stanici. Proto je nutné zajistit, aby komunikace mezi stanicí a dokem probíhala po jedné sběrnici a aby každý dok mohl být adresovaný. To by zajistilo možnost přidávat doky modulárně prakticky bez omezení.

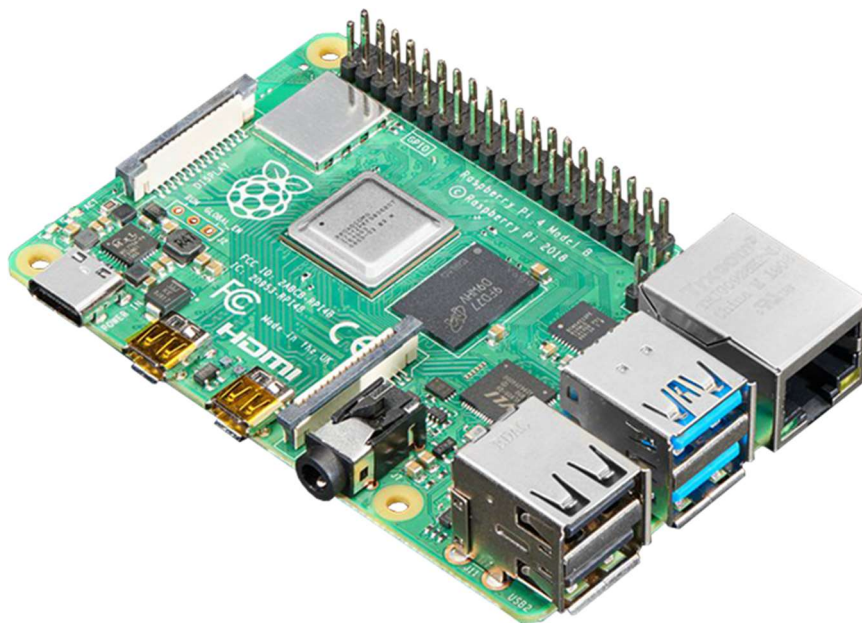


## 4.2 HW prostředky

Na rozdíl od serverové části, která nepotřebovala specifické hardwarové prostředky, stanice již potřebuje několik zásadních prvků. Je potřeba vyřešit hardware pro samotný systém stanice, hardware pro čtení karet, komunikaci se stanicemi a také samotný displej. Důležité je také vyřešit napájení celé stanice

### 4.2.1 Raspberry Pi

Jako centrální prvek celé dobíjecí stanice je nezbytné použít zařízení, které podporuje připojení dotykového displeje, čtečky karet a zároveň poskytuje uživatelsky přívětivé rozhraní pro nastavení a programování. Důležitou vlastností tohoto zařízení je také podpora hardwarových výstupů a vstupů pro připojení periférií. V tomto ohledu je



*Obrázek 8 - Raspberry Pi 4 (Zdroj: adafruit.com [38])*

ideálním kandidátem Raspberry Pi. Raspberry Pi je mikropočítač navržený s cílem poskytnout plnohodnotnou počítačovou funkcionalitu s nízkou spotřebou energie a schopností interakce se světem pomocí GPIO pinů. Raspberry Pi je široce využíváno v elektronickém DIY prostředí pro vytváření chytrých domácností a také v profesionální sféře a IoT. Jeho použití v dobíjecí stanici umožní široké možnosti programování a propojení s fyzickými perifériemi. Raspberry Pi nabízí širokou škálu možností a rozhraní, včetně Ethernetu, USB, Wi-Fi a Bluetooth, což usnadňuje komunikaci s ostatními prvky systému. Jeho otevřený a modulární charakter umožňuje snadné přizpůsobení a rozšíření funkcionality dobíjecí stanice dle specifických požadavků. [31]

## 4.2.2 Dotykový displej

Pro stanici je třeba najít dotykovou obrazovku která umožní bezproblémové propojení s Raspberry Pi. Jako ideální řešení byl nalezen displej "Waveshare 7" LCD". Tento displej disponuje kapacitní technologií dotyku a nabízí rozlišení 1024x600 pixelů. Díky svému IPS panelu zajišťuje vynikající kvalitu zobrazení s širokými pozorovacími úhly. Pro propojení s Raspberry Pi slouží HDMI rozhraní pro přenos obrazu a USB rozhraní pro přenos informací o dotyku. Tento vybraný displej poskytuje dostatečné rozlišení pro implementaci komplexnějšího grafického uživatelského rozhraní. [32]



Obrázek 9 - Dotykový displej (Zdroj: waveshare.com [32])

## 4.2.3 Čtečka karet

Na základě předchozí analýzy jsme zjistili, že potřebujeme čtečku karet, přičemž jsme identifikovali konkrétní technologii, a to MIFARE. Pro tento účel jsme vybrali generický modul čtečky RFID – RC522, který je schopen číst tyto karty. Každá karta má své unikátní identifikační číslo, tzv. UID, které umožňuje jednoznačnou identifikaci karty. Modul čtečky sám o sobě není samostatně funkčním celkem, ale je nutné ho připojit k Raspberry Pi pomocí sběrnice SPI. Modul je napájen pomocí interního 3.3V stabilizátoru v Raspberry Pi. Samotné čipové karty nepotřebují žádné vlastní napájení, protože se jedná o pasivní zařízení, které získává energii indukci při čtení od samotné čtečky.

#### 4.2.4 Napájecí zdroj

Jelikož stanice bude zajišťovat napájení nejen svého vlastního hardwaru, jako je Raspberry Pi, displej a čtečka, ale také jednotlivých doků, je klíčové vybrat zdroj s dostatečným výkonem a rezervou. Celý systém pracuje na napětí 5V, proto je nutné pořídit 5V zdroj. Pro zajištění dostatečné rezervy byl vybrán 5V 10A průmyslový zdroj, který splňuje požadované parametry.



Obrázek 10 - Průmyslový 5V 10A zdroj (Zdroj: hotair.cz [33])

## 4.3 Implementace

Nyní se pojdme podívat na samotnou implementaci stanice. V následujících podkapitolách se podíváme na fyzické řešení, instalaci displeje, nastavení režimu kiosek, implementaci čtečky a řešení některých problémů

### 4.3.1 Instalace systému a konfigurace displeje

Na Raspberry Pi je nutné nainstalovat operační systém, konkrétně byl zvolen 64bitový Raspbian Lite. Tato verze operačního systému obsahuje pouze základní knihovny bez grafického rozhraní.

Instalace začíná na počítači, kde je třeba pomocí programu Raspberry Pi Imager nainstalovat konkrétní image na SD kartu. Před instalací je možné nakonfigurovat základní parametry jako je uživatelské jméno, hostname, připojení na wifi nebo povolení SSH přístupu. Pro snadnou následnou instalaci je doporučeno vyplnit všechny údaje a aktivovat SSH pro následný přístup k systému po síti. Po provedení instalace systému na SD kartu z počítače můžeme přejít na další krok, a to vložení karty do Raspberry Pi a první boot. První start může trvat delší dobu. Po dokončení se lze přihlásit pomocí ssh a dále se systémem pracovat.

Po prvním přihlášení do systému je dobré zkontrolovat stav aktualizací pro systém, a pokud jsou aktualizace dostupné tak je stáhnout a nainstalovat. Poté lze pokračovat s instalací již potřebných součástí systému pro režim kiosku. Nejprve nainstalujeme moduly pro grafické rozhraní, protože Raspbian Lite je verze operačního systému bez grafického rozhraní. Nainstalujeme tedy XServer a jeho další součásti a utility. Pro instalaci můžeme použít následující příkazy:

```
sudo apt-get install --no-install-recommends xserver-xorg
sudo apt-get install --no-install-recommends xinit
sudo apt-get install --no-install-recommends x11-xserver-utils
```

Vzhledem k tomu že uživatelské rozhraní bude napsané jako jednoúčelová webová aplikace v PHP tak je zapotřebí nainstalovat prohlížeč, konkrétně chromium browser a také další utility umožňující spouštění prohlížeče jako kiosku. Na to slouží následující příkazy:

```
sudo apt-get install chromium-browser
```

```
sudo apt-get install matchbox-window-manager
```

```
sudo apt-get install xautomation unclutter
```

Nyní by mělo být v systému nainstalováno vše potřebné ke spuštění webového prohlížeče v režimu kiosku. Nyní je nutné vytvořit konfiguraci a upravit některé nastavení systému. Nejprve vytvoříme soubor příkazem „`nano ~/kiosk`“ do kterého vložíme následující obsah:

```
#!/bin/sh
xset dpms 0 0 20
xset s noblank
matchbox-window-manager -use_titlebar no &
unclutter &
chromium-browser --display=:0 --kiosk --incognito --window-
position=0,0 http://localhost/
```

Po vytvoření souboru je třeba mu dát spustitelná práva, tedy použít příkaz „`chmod 755 ~/kiosk`“. Následně můžeme postoupit k dalšímu kroku, a to nastavení automatického spuštění po přihlášení. To provedeme pomocí editace souboru *bashrc* pomocí příkazu „`nano ~/.bashrc`“. Na konec tohoto souboru vložíme tento řádek:  
`xinit /home/juscooters/kiosk -- vt$(fgconsole)`  
Tento příkaz platí za předpokladu, že máme uživatelský účet *juscooters*, pokud jsme přihlášení pod jiným uživatelským účtem, je třeba změnit část příkazu *juscooters* za jméno uživatele.

Nyní by již mělo spuštění prohlížeče v režimu kiosku fungovat a po přihlášení do konzole se spustí i kiosek. Ovšem pro bezproblémové fungování stanice je zapotřebí nastavit automatické přihlašování do konzole, aby se zapíнал kiosek automaticky po startu zařízení. To lze pomocí vestavěné utility „*sudo raspi-config*“

- Nejprve spustíme utilitu *raspi-config*
- V nabídce vybereme možnost "System Options"
- Poté zvolíme "Boot/Autologin"
- Vybereme možnost "Console Autologin"
- A následně potvrdíme provedení změny.

Tímto způsobem aktivujeme automatické přihlášení do konzole a po restartu Raspberry Pi se bude systém automaticky přihlašovat bez nutnosti zadávání uživatelského jména a hesla.

Posledním důležitým krokem pro plnou funkcionalitu kiosku ve stanici je instalace virtuální klávesnice, která bude umožňovat uživateli zadávat jméno a heslo při přihlašování které bude sloužit pro přiřazení karty k uživatelskému účtu. Raspberry Pi má v nabídce několik možností pro virtuální klávesnice, ale také samotný prohlížeč, postavený na jádře Chromium, poskytuje možnost instalace klávesnice jako doplňku, což ovšem bylo zavrhnuto. Důvodem byla absence české verze klávesnice a také obavy z bezpečnosti, neboť nemůžeme garantovat, že tato klávesnice neposílá data třetím stranám. Nakonec byla zvolena systémová virtuální klávesnice onboard. Tu lze nainstalovat příkazem „`sudo apt install onboard`“, pro následnou konfiguraci klávesnice můžeme zadat příkaz „`onboard-settings`“.

### **Nastavení prohlížeče**

Pro správnou konfiguraci a zajištění bezpečnosti Chromium browseru v režimu kiosku je nutné provést několik důležitých nastavení. Jedním z klíčových kroků je vypnutí automatického vyplňování a ukládání hesel, což zabraňuje nežádoucímu ukládání citlivých údajů a zvyšuje ochranu uživatelů. Tím se minimalizuje riziko, že hesla budou uložena ve stanici a náchylná k případnému zneužití útočníkem, který se pokusí přihlásit. Dalším důležitým krokem je blokování cookies třetích stran. Blokováním těchto cookies je zajištěna ochrana soukromí a minimalizace uchování údajů o předešlém užívání.

Pro ještě větší zabezpečení můžeme využít speciálního doplňku s názvem "Cookie Autodelete"<sup>1</sup>. Tento doplněk je dostupný v oficiálním obchodě s doplňky pro Chromium a umožňuje automatické mazání cookies. Po nainstalování je důležité provést dodatečné nastavení, které zajistí, že cookies budou automaticky smazány při změně domény. Tímto způsobem je zabezpečeno, že informace o uživatelově aktivitě na přihlašovací stránce STAG se po přesměrování na lokální webové rozhraní smažou a další uživatel nebude automaticky přihlášený jako předešlý uživatel.

---

<sup>1</sup> Dostupný na „<https://chrome.google.com/webstore/detail/cookie-autodelete/fhcgjolkccmbidfldomjliifgaodjagh>“

### 4.3.2 Komunikační protokol pro doky

Jak již bylo v analýze řečeno, nejlepší způsob pro komunikaci mezi stanicí a dokem bude použití sběrnice umožňující využívání jednoho nosiče dat a adresaci jednotlivých doků. Tento nápad vznikl na základě zkušenosti s používáním teplotních čidel Dallas ds18b20 v jiném mém projektu. Výhodou těchto čidel je, že využívají sběrnici „1 – Wire“, která pracuje na principu Master-Slave a díky tomu umožňuje na jeden GPIO pin připojit několik teplotních čidel. Na podobném principu pracuje i sběrnice „I2C“ která umožňuje adresaci zařízení a využívá pouze dva vodiče, jeden na takt (CLOCK) a druhý na data (DATA). Pro komunikaci mezi stanicí a doky byla tedy zvolena sběrnice I2C, kdy stanice bude Master, který se bude periodicky ptát na stav všech doků, které se budou chovat jako Slave zařízení.

Zde bylo třeba vyřešit několik možných problémů. Hlavním problémem je nekonzistence dat odesílaných doky a přijímaných stanicí. Po každé žádosti na konkrétní adresu od stanice musí dok odpovědět svým konkrétním stavem, ten se skládá z několika různých proměnných:

- DID – jedinečné ID doku
- SID – jedinečné ID právě připojené koloběžky
- S1 – Stav západky zámku
- S2 – Stav solenoidu zámku
- S3 – Aktuální stav senzoru na nabíjecí proud
- S4 – Aktuální stav senzoru na nabíjecí napětí
- DT – Typ doku

Vzhledem k možnosti větší velikosti jednotlivých ID vzniká větší riziko rušení přenosu a narušení některého bitu a tím porušení integrity dat. Vzhledem k tomu že stanice následně automaticky registruje doky a koloběžky do databáze, tak by velmi rychle mohla nastat situace, kdy se v databázi objeví koloběžky, které budou mít stejné ID lišící se pouze jedním číslem. Proto bylo zapotřebí ještě přidat další proměnnou a tou je CRC. CRC je kontrolní součet který umožňuje zkontrolovat na stanici integritu dat a pokud CRC neodpovídá jsou data zahozena a je požádáno o znovu zaslání.

Dalším problémem je občasné rušení celé sběrnice, které je vyřešené opakovaným odesláním žádosti. Zde ovšem vzniká ještě jeden další problém, který je daný samotným principem fungování sběrnice. I2C funguje na principu, že se mění napěťová úroveň na

sběrnici mezi LOW a HIGH. Pokud by ale útočník zapojil do sběrnice škodlivé zařízení, které by zaplavilo sběrnici, tak by stanice nemohla komunikovat s žádným dokem. To stejné nastane, pokud by došlo ke kritickému selhání řídicí jednotky jednoho z doků, které by umožnilo zkrat na sběrnici. Tomu bohužel nejde zamezit, ale pravděpodobnost že to nastane je minimální. Ošetřeno je zde alespoň to, že doky v tomto případě zůstanou uzamčené. Pokud by přece jen k situaci došlo, tak jednoduché odpojení poškozeného doku situaci vyřeší a sběrnice se sama vzpamatuje.

### 4.3.3 Redis

Pro usnadnění interakce a sdílení proměnných mezi různými částmi systému, zejména pak mezi PHP a Pythonem, byl zvolen server Redis. Redis je open-source databáze typu key-value, která umožňuje uchovávání proměnných a dat v paměti. Díky své rychlosti a jednoduchému použití, díky dostupným knihovnám, je vhodným nástrojem pro efektivní komunikaci a sdílení informací mezi různými programovacími jazyky. Použití Redisu výrazně zjednodušuje interakci a spolupráci mezi PHP a Pythonem, čímž usnadňuje vývoj a zvyšuje efektivitu při práci s proměnnými a daty. [34]

### 4.3.4 Skripty

Pro správné fungování celé stanice je nutné zajistit správné fungování čtečky karet, která je přímo připojená k Raspberry Pi pomocí GPIO pinů a kvůli tomu je nutné naprogramovat skript pro čtení. To samé bude v případě komunikace s doky, která bude probíhat také pomocí GPIO pinů.

#### Čtečka karet

Jak již bylo v předešlých kapitolách zmíněno, každá čipová karta má své jedinečné UID, které budeme využívat pro identifikaci konkrétní karty. Proto je zapotřebí napsat skript který bude provádět čtení UID přiložené karty a následně bude odesílat dotaz na server pomocí API end pointu „*checkCard*“. Vzhledem k tomu že modul pro čtení čipových karet komunikuje pomocí SPI je nutné aktivovat tento komunikační interface na Raspberry Pi. Použijeme tedy konfigurační utilitu *raspi-config*:

- Nejprve spustíme utilitu: `sudo raspi-config`
- V nabídce vybereme možnost "Interface Options"
- Poté zvolíme "SPI"
- A následně potvrdíme provedení změny.



Po aktivaci SPI provedeme restart. Poté můžeme pokračovat s instalací prerekvizit pro skript obsluhující čtečku karet. Nejprve, pokud již nemáme, tak nainstalujeme python a pip pomocí příkazu:

```
sudo apt install python3-dev python3-pip
```

Následně musíme nainstalovat knihovnu pro python která nám umožní spolupracovat se sběrnici SPI. To provedeme příkazem:

```
sudo pip3 install spidev
```

Nyní již můžeme pomocí python scriptu komunikovat se zařízeními připojenými k SPI interface. V pythonu je ještě navíc dostupná knihovna mfrc522, která je kompatibilní s námi zvolenou čtečkou čipových karet, a tedy lze ji použít. Nainstalujeme jí příkazem:

```
sudo pip3 install mfrc522
```

Po nainstalování této knihovny již můžeme plně komunikovat se čtečkou karet.

Celý skript se skládá z hlavní smyčky, kdy čeká na přiložení karty, v momentě přečtení karty je uloženo ID karty, které je vloženo do dotazu na server a čeká se na odpověď. Server odpovídá XML souborem s daty o uživateli. Tento skript následně uloží data do Redis proměnných a tím je předá lokálnímu grafickému rozhraní které je interpretuje. Tento skript pro čtení karty se nachází v „*./opt/JUScooters/reader.py*“. Pro správné fungování stanice je zapotřebí script spouštět okamžitě po přihlášení. Vložíme ho tedy stejně jako inicializaci kiosku do souboru „*~/ .bashrc*“.

## **Komunikace s doky**

Jak již bylo v jedné předešlé kapitole řečeno, pro komunikaci mezi stanicí a doky využíváme protokol I2C s vlastní strukturou zpráv. Proto je zapotřebí napsat skripty pro příjem dat a také pro odesílání dat. Vzhledem k tomu že skripty vyžadují přístup k I2C sběrnici na Raspberry Pi je nutné tento interface nejprve povolit pomocí utility raspiconfig:

- Nejprve spustíme utilitu: `sudo raspi-config`
- V nabídce vybereme možnost "Interface Options"
- Poté zvolíme "I2C"
- A následně potvrdíme provedení změny.

Po restartu systému můžeme pokračovat a nainstalovat potřebnou knihovnu pro python, a to smbuse pomocí příkazu:

```
pip install smbuse
```

### **Žádost o data doku a registrace**

Komunikace s doky a registrace nových doků a koloběžek se řeší v jednom skriptu dohromady. Celý skript běží ve smyčce, kdy v každé iteraci se provede nejprve oskenování dostupných adres na sběrnici I2C a následně na každou zabranou adresu zkouší odeslat výzvu o aktuální stav doku. Pokud na výzvu přijde poškozená, nebo žádná odpověď, nebo pokud je právě sběrnice zarušená, tak se opakuje pokus o přenos celkem pětkrát. Pokud ani po páté není přenos úspěšný skript se posune a zkusí další adresu v řadě. Pokud je ovšem přenos úspěšný a CRC kontrola projde, tak přichází na řadu parsování odpovědi a přiřazení do jednotlivých proměnných které se využijí v API požadavku na server. Tím se dynamicky mění nebo registrují nové doky a koloběžky. Po provedení celého cyklu žádostí na doky je provedeno zabalení všech informací doků a koloběžek do jednoho JSON souboru který je následně uložen do Redis proměnné pro zpracování grafickým rozhraním které díky tomu může dynamicky zobrazovat aktuální stavy doků. Tento skript se nachází v „*./opt/JUScooters/i2c.py*“. Pro správné fungování stanice je zapotřebí script spouštět okamžitě po přihlášení. Vložíme ho tedy stejně jako skript „*reader.py*“ do souboru „*~/ .bashrc*“.

### **Zahájení výpůjčky a odemknutí doku**

Dalším důležitým skriptem v systému je skript na vytvoření nové výpůjčky, který se zároveň musí starat o odemknutí doku. Tento skript pošle příkaz na konkrétní adresu doku, který zajistí odemknutí dokovacího zámku, poté lze koloběžku z doku vysunout. Pokud ovšem uživatel koloběžku nevysune a nechá dok odemčený, po zhruba 40 vteřinách se dok sám zamkne a pokud dojde k úspěšnému zapadnutí západky solenoidu i samotné západky zámku, výpůjčka se nevytvoří. Pokud by došlo k chybě a západka by nezapadla tak bude stanice vydávat zvukový i vizuální signál pro upozornění uživatele. Pokud dojde k úspěšnému vyjmutí koloběžky, skript odešle API příkaz na server a tím zaregistruje novou výpůjčku. Tento skript se nachází v „*./opt/JUScooters/rent.py*“. Skript je volán z grafického rozhraní, které při volání udává následující parametry:

- address – I2C adresa doku
- user – ID uživatele
- scooter – ID koloběžky

### Ukončení výpůjčky

Dalším skriptem je skript pro ukončení výpůjčky, ten musí při spuštění odemknout dok a čekat, zda uživatel vrátí koloběžku. Pokud koloběžku vrátí a je detekované ID koloběžky, tak skript dok zamkne a pokud dojde k úspěšnému zapadnutí obou západek tak se výpůjčka ukončí. Pokud by uživatel koloběžku nevrátil a například by jen zavakl prázdný dok, tak by se výpůjčka neukončila, protože dochází ke kontrole ID koloběžky připojené k doku a bez ID koloběžky nelze ani pomocí API příkazu ukončit výpůjčku. . Tento skript se nachází v „*/opt/JUScooters/return.py*“. stejně jako předešlý skript „*rent.py*“ i tento skript NESMÍ být spouštěn automaticky se startem systému. Skript je volán z grafického rozhraní, které při volání udává následující parametry:

- address – I2C adresa doku
- user – ID uživatele

### Refresher kiosku

Dalším velmi důležitým skriptem je skript pro obnovu kiosku, tedy pro ukončení aktuální instance prohlížení a znovuspuštění čistého prohlížeče. To probíhá v momentě, kdy dojde k vypnutí podsvícení displeje. Díky tomu se nemůže stát, že by zůstal kiosku zaseknutý na jedné konkrétní stránce, například kvůli nějaké chybě. K obnově kiosku dochází každých 30 vteřin. Ale lze toto nastavení upravit, protože obnova je závislá na časovači pro úsporný režim displeje. Tento skript je napsaný jako smyčka kontrolující stav displeje, je tedy nutné ho inicializovat při startu vložním, stejně jako skript „*reader.py*“, do souboru „*~/ .bashrc*“. Tento skript se nachází v „*/opt/JUScooters/monitor.py*“.

### Správce klávesnice

Posledním, ale neméně důležitým skriptem je správce klávesnice, který se stará o spuštění virtuální systémové klávesnice „*onboard*“ pokud je to od grafického rozhraní

kiosku vyžádáno. K tomu dochází pouze v případě, kdy je nutné přihlášení uživatele a k tomu dojde pouze v případě, kdy uživatel nemá spárovanou kartu s účtem. Tento skript je napsaný jako smyčka a je tedy nutné ho inicializovat při startu vložení do souboru „~/ .bashrc“. Tento skript se nachází v „/opt/JUScooters/keyboard.py“.

#### 4.3.5 Fyzické zapojení

Po úspěšné instalaci a napsání potřebných skriptů je nutné fyzické zapojení periférií do Raspberry Pi. Pro zapojení čtečky karet přes SPI je potřeba připojit čtečku k příslušným GPIO pinům. V tabulce níže je popsáno, o které piny se jedná a díky tomu lze čtečku zapojit.

Čtečka RFID	Raspberry Pi
<b>SDA</b>	PIN 24
<b>SCK</b>	PIN 23
<b>MOSI</b>	PIN 19
<b>MISO</b>	PIN 21
<b>GND</b>	PIN 6
<b>RST</b>	PIN 22
<b>3.3V</b>	PIN 1

Tabulka 4 - Zapojení RFID čtečky s Raspberry Pi (Zdroj: Autor)

Po správném připojení by se měla rozsvítit červená led dioda na čtečce karet a v systému by již měla být přístupná. [35] Dále je třeba zapojit bzučák kde GND na bzučáku se spojí s jakýmkoliv GND portem na Raspberry Pi a VCC port na bzučáku se spojí s GPIO3 (PIN15). A jako poslední zapojení je samotné I2C pro doky. Ta je na GPIO8 a GPIO9.

### 4.3.6 Stanice

Pro výrobu stanice byl využit 3D tisk. Celé rozhraní pro stanici je zapouzdřeno do jednoho celku, tedy displej, Raspberry Pi a čtečka karet. Pouzdro bylo navrženo tak, aby bylo možné Raspberry Pi chladit přídatným větrákem. Mezi hlavní prvky pouzdra patří:

- „Vše v jednom“ řešení
- Montáž na zeď
- Přesný návrh pro zasazení displeje i ostatních komponentů
- RGB signalizace ve formě loga čtečky



*Obrázek 11 - Sestavená funkční stanice (Zdroj: Autor)*

### 4.3.7 Grafické rozhraní pro uživatele

Poslední velmi důležitou částí k implementaci na stanici je grafické uživatelské rozhraní, které se zobrazuje na displeji. K tomu bylo zvoleno použití lokálního webu napsaného v PHP. Grafické rozhraní používá prvky z front endu serverového systému a obohacuje je o prvky potřebné pro ovládání stanice.

#### Funkční část

První částí grafického rozhraní je samotný backend, který přebírá data z Redis, s proměnnými dále pracuje a na základě jejich hodnot vyvolává buďto zprávy kategorie „errorMsg“, které mají za úkol upozornit uživatele nebo ho vyzvat k akci, nebo může vyvolat zobrazení informací o aktivních výpůjčkách, nebo volných koloběžkách a docích.

Dalším úkolem je vyvolávat akce, jako je aktivace klávesnice, zahájit výpůjčku nebo ukončit výpůjčku. To vše se provádí pomocí skriptů, které byly popsány v minulé kapitole. Systém buďto přímo spouští skript s parametry, nebo mění stav proměnné v Redisu, na kterou skript zareaguje.

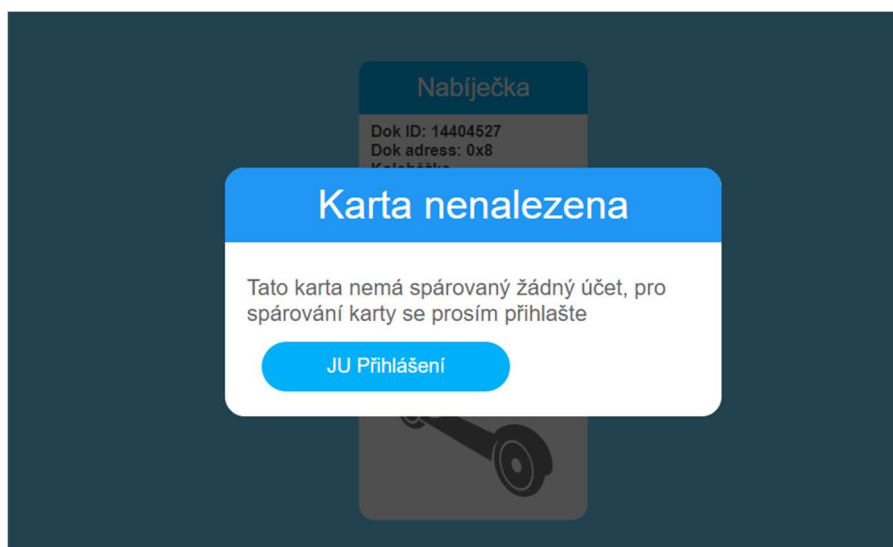
#### Grafická část

Jak již bylo řečeno, grafická část je založena na webu celého systému. Ovšem je zjednodušena, některé prvky jsou ubrány pro lepší zobrazování informací a s přihlédnutím na menší obrazovku bylo nutné zvětšit ovládací prvky a odebrat nepotřebné funkce. Většinu času se setká uživatel s úvodní obrazovkou, kde uvidí aktuální stav stanice. Na obrázku níže lze vidět rozhraní ukazující jeden aktivní dok, který obsahuje koloběžku připravenou k vypůjčení.



Obrázek 12 - Grafické rozhraní stanice (Zdroj: Autor)

Po přiložení karty mohou nastat celkem čtyři situace, první z nich nastane, když karta nebude spárována s žádným uživatelským účtem, v tom případě se objeví na obrazovce výzva pro přihlášení a tím se karta spáruje. Druhý případ nastane, pokud je karta spárována s uživatelským účtem, ale daný uživatelský účet nemá povolené půjčování. V reakci na to se zobrazí uživateli na obrazovce informace, že pro povolení musí kontaktovat správce systému emailem. Třetí situace nastane, pokud uživatel má povolené půjčování a v doku je volná koloběžka, v tom případě si ji může vypůjčit. A poslední situace nastane, pokud uživatel má koloběžku vypůjčenou, tak ji může vrátit, a pokud je ve stanici vícero kompatibilních doků s danou koloběžkou tak si může vybrat do jakého doku koloběžku vrátí.



Obrázek 13 - Grafické rozhraní stanice, chybová hláška  
(Zdroj: Autor)

## 5 Dobíjecí dok

Posledním krokem k vytvoření celého ekosystému dobíjecí stanice je sestrojení doku. To se skládá ze dvou hlavních částí, sestrojení po mechanické stránce a potom sestrojení po stránce softwarové. V následujících kapitolách si postupně projdeme celý proces konstrukce od analýzy až po samotnou realizaci.

### 5.1 Analýza a návrh

Nejdůležitějším a asi nejsložitějším úkolem bylo najít způsob jak bezpečně a jednoduše zamykat koloběžku. Původně v době před realizací této chytré dobíjecí stanice byla koloběžka zamykána cyklistickým zámekem s číselnou kombinací. To mělo několik nevýhod. První nevýhodou bylo použití číselného zámku, tedy pokud si někdo chtěl půjčit koloběžku, musel znát kombinaci a tu poté mohl sdílet dál a tím rozšiřovat okruh lidí, kteří mají přístup ke koloběžce. Druhou nevýhodou bylo samotné zabezpečení, které působilo spíše jako placebo. Koloběžka byla totiž uzamykána tak, že bylo možné zámeček vysunout z plastového držáku na koloběžce, a okolo řídítek jej napnout a koloběžku protáhnout.



Obrázek 14 - Zamykáací stanice pro elektrokoloběžky – bikeep (Zdroj: bikeep.com [39])

Jako první nápad na spolehlivé uzamykání bylo využít 3D tisk k vytisknutí držáku na celé tělo koloběžky, na samotnou stoupačí plochu, která má v sobě baterii a elektroniku. Tento mechanismus by musel mít kovové výztuhy a celkový proces zamykání by byl pro uživatele ještě složitější než v současné době. Lepším řešením je vytvoření objímky na „krk“ koloběžky, tedy na část spojující řídítka s tělem koloběžky.



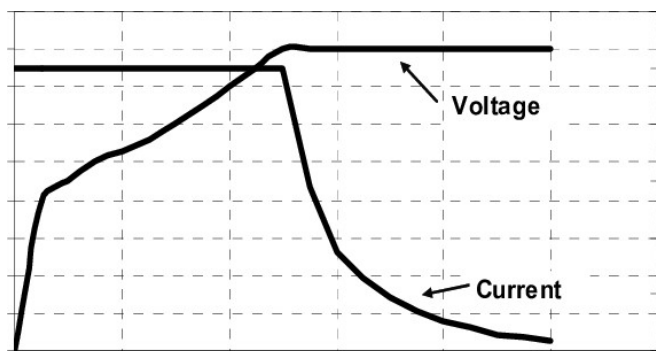
V případě, kdy použijeme velmi těsnou objímku tak ani nebude možné nijak koloběžku protáhnout. Podobným principem jsou řešené již dostupné veřejné zámky na koloběžky. Ovšem ty mají složitější mechaniku kdy se celá koloběžka musí zasunout a poté musí dojít k přihlášení a až poté může uživatel manuálně vysunout ocelový trn a tím ji zamknout.

V ideálním případě by naše dokovací stanice měla umožňovat zasunutí koloběžky a automatické uzamknutí. To by bylo možné pomocí například servomotorů, které by ovládali západky. K samotnému uzamknutí, aby nemohl útočník západku zasunout proti síle motoru by mohl sloužit solenoid který by trn zabezpečil proti nežádoucímu pohybu.

V průběhu zkoušení možných alternativ byl navržen systém, který se silně inspiroje u západky zámku u auta, tedy mechanismus západky, která se sama otočí do uzamknuté polohy tím, že se do ní vsune uzamykaný objekt. Díky tomu odpadne nutnost pro uživatele, jakkoliv manipulovat se zámekem a jediné co bude třeba, je správně koloběžku do zámku zatlačit do zapadnutí.

## Nabíjení

V řešení byl také způsob nabíjení koloběžky, kdy nejlepší způsob bude využít stávající nabíjecí adaptér s tím že ho napojíme na dok který bude schopen nabíječku podle potřeby zapnout nebo vypnout. Dok bude mít na nabíječce připojený senzor snímající buďto nabíjecí proud, nebo nabíjecí napětí, podle kterého bude schopen určit, zda je koloběžka nabita. V průběhu nabíjení se mění proud i napětí.

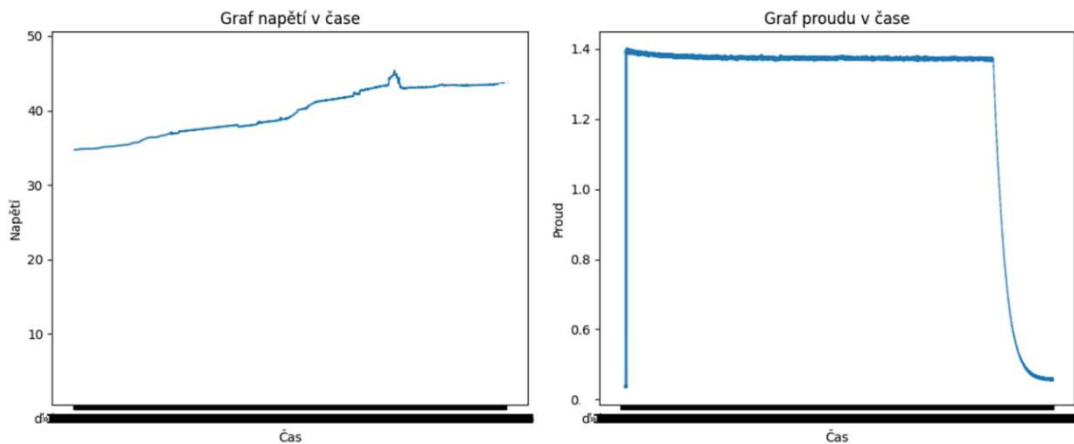


Obrázek 15 - Graf nabíjecího proudu a napětí baterií  
(zdroj: R. & D. R. & A. M. Bhuiyan [36])

Napětí se mění tak, že se začátkem nabíjení klesne na úroveň nabití baterie a postupem nabíjení roste s napětím na baterii. U proudu je to opačné, tedy ze začátku je

proud konstantní až do stavu cca 50-80% nabití a poté proud začne klesat až téměř k nule do stavu úplného nabití. [36]

Analýzou samotného nabíjení koloběžky bylo potvrzeno, že graf se téměř shoduje, s tím rozdílem, že ke klesání proudu dochází až okolo 90% nabití baterie. Naopak ke stabilizaci růstu napětí dochází dříve.



Obrázek 16 - Vlastní měření nabíjecího proudu a napětí (Zdroj: Autor)

Podle grafu z vlastního měření můžeme vidět, že nejlepší způsob pro detekci nabití baterie je měřit proud, pokud ovšem budeme poskytovat měření jak proudu, tak napětí, umožníme tím do budoucna otevřít dveře možnosti vypočítávat aktuální stav nabití, nebo odhadovat čas do úplného nabití.

### Řídicí jednotka doku

Vzhledem k tomu že dok má být připojený ke stanici tak, aby bylo možné doky za sebe řetězit a tím umožnit modulárnost celé stanice, tak je zapotřebí aby dok obsahoval sám v sobě nějakou řídicí jednotku. Zde přichází v úvahu Arduino nebo ESP8266 nebo ESP32. Vzhledem k osobní preferenci bylo zvoleno ESP8266, které se bude chovat jako Slave zařízení v I2C. Užití ESP8266 má výhodu, že by do budoucna mohlo dojít k využití bezdrátového přenosu a vytvořit například bezdrátový modul zámku, který by komunikoval se stanicí pomocí ESP-NOW. Druhou možností by bylo použít ESP32 s Bluetooth, které by otevřelo možnost komunikovat přímo s koloběžkou. To by ovšem znamenalo, že pro každý konkrétní model koloběžky, by musel být nastavený dok jinak a tím by se zhoršila jeho univerzálnost.

Během testování byl ovšem zjištěn kritický nedostatek v modulu ESP8266. ESP8266 má problémy s komunikací na sběrnici I2C v roli SLAVE. Kvůli tomu by

primární funkčnost doku, tedy komunikace se stanicí po adresované sběrnici byla velmi nespolehlivá a prakticky v reálném nasazení nemožná. Další nevýhodou je obsah pouze jednoho analogového vstupu, tedy bylo by nutné si vybrat mezi měřením proudu nebo napětí. Pro další vývoj tedy byl dok pozměněn, aby využíval Arduino Nano, které nemá problém fungovat jako I2C SLAVE a umožňuje měřit najednou jak napětí, tak i proud.

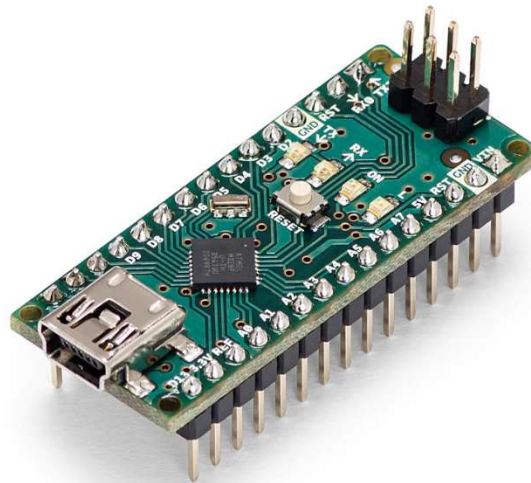
### **Koloběžka**

Pro správné fungování navrženého systému je třeba abychom mohli jednoznačně určit, zda je koloběžka opravdu zamknutá v doku. Prvním nápadem na řešení tohoto úkolu bylo použití spínače v těle koloběžky, který by zasunutím koloběžky sepnul obvod a tím by signalizoval koloběžku v doku. To je ovšem řešení, které přináší více problémů než výhod, protože bychom se spolehnali na signalizaci, že koloběžka v doku je, ale ve skutečnosti by mohl útočník pouze spojit vodičem kontakty a dok by si myslel, že došlo k vrácení koloběžky. Ideálním řešením by bylo, kdyby koloběžka obsahovala chytrou elektroniku, která by mohla komunikovat s dokem, nabízí se možnost použít ESP nebo Arduino. To by ovšem přidalo na složitosti celého systému a zbytečně zvýšilo náklady na celé řešení.

Hlavním cílem, čeho chceme dosáhnout je, aby koloběžka měla jedinečný identifikátor, který bude komunikovat s dokem. V průběhu navrhování a testování možností byl nalezen čip DS2401 který poskytuje jedinečné ID. Bohužel tento čip je v České republice poměrně nedostupný, ale existuje velmi dostupná alternativa a tou je teplotní čidlo Dallas ds18b20, které komunikuje se zařízením po sběrnici 1-Wire a každé čidlo má své jedinečné ID. Proto lze do objímky koloběžky zabudovat tento čip, který bude předávat své ID při připojení. Toto jedinečné ID bude uloženo také v databázi jako ID koloběžky. To do budoucna otevře možnosti dalšího rozvoje chytré dobíjecí stanice.

## 5.2 HW prostředky

Nyní se pojdme podívat na hardware potřebný pro realizaci jednoho doku. Jak již bylo řečeno v předešlé kapitole, je potřeba řídicí jednotka, v tomto případě byla zvolena platforma Arduino. Pro sestavení a testování prototypu byla používána vývojová deska Arduino Nano, která obsahuje několik digitálních IO portů pro připojení všech potřebných periférií a senzorů. Velikou výhodou této desky je také větší počet analogových vstupů, které umožňují zapojení jak proudového snímače, tak i odbočky s odporovým děličem umožňující snímání nabíjecího napětí. Dále disponuje I2C sběrnici pro komunikaci se stanicí. Velkým přínosem je také přítomnost USB převodníku, který umožňuje jednoduché propojení s počítačem pro programování, komunikaci a kalibraci. [37] To je důležité hlavně pokud dojde ke konstrukci dalších doků, tak bude třeba zkalibrovat výstupy jednotlivých čidel napětí a proudu pro přesný výstup z doku. Každý dok může mít použitý jiný odporový dělič, a tedy i jiné vstupní hodnoty. To je potřeba brát také v úvahu, pokud by došlo k sestavení úplně nového modelu doku pro jiné napětí, kdy při nabíjecím napětí větším, než 50V by mohlo dojít k poškození mikrokontroleru, pokud by konstruktér nepoužil adekvátní odporový dělič.



Obrázek 17 - Arduino Nano (Zdroj: Arduino.cc [37] )

## Senzory

Kromě mikrokontroleru je potřeba do doku zakomponovat dva optické spínače, které budou snímat stav západky zámku. Tento spínač na rozdíl od klasických koncových spínačů neobsahuje žádné mechanické části, proto je daleko odolnější a neklade žádný odpor při zavírání západky. Optický spínač funguje na principu snímání paprsku světla a pokud dojde k jeho přerušení tak vyšle signál. Jediná nevýhoda je nutnost aktivního napájení diody ve snímači.



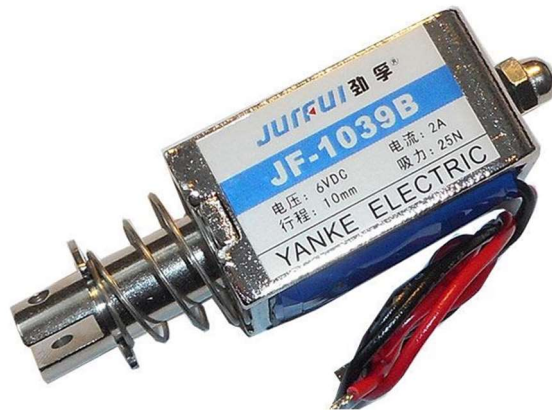
*Obrázek 18 - Optický spínač (Zdroj: ECLIPSERA s.r.o, [40])*

Dalším senzorem je senzor pro detekci úrovně nabití. Původně byl zamýšlený pouze senzor snímající úroveň proudu, ale nakonec byla zvolena i druhá metoda kterou je měření napětí přímo na jednom z ADC pinů Arduina pomocí odporového děliče, který sníží rozsah napětí, které se může objevit na nabíječce na rozsah, který dokáže bez poškození přečíst mikrokontroler. Při konstrukci odporového děliče je ovšem nutné brát v úvahu verzi mikrokontroleru, některé vývojové moduly mají již jeden odporový dělič zabudovaný a podporují větší rozsahy měření. V případě použitých mikrokontrolerů se jedná o rozsah 0-5V.

## Zámek

Konstrukce zámku je speciálně vytvořena a nelze ji tedy koupit, ale hlavním prvkem, který je nutné pořídit je solenoid který se bude chovat jako bezpečnostní západka zámku. Tento solenoid má ocelový trn, který pomocí elektromagnetů dokáže velmi rychle

přitáhnout. Vyrábějí se v různých velikostech a s různým vstupním napětím. Vzhledem k sestavení celého systému chytré dobíjecí stanice, který počítá s napájením 5V, byl vybrán solenoid s možností napájení nejbližší k této hodnotě, tedy solenoid na 6V.



konkrétně se jedná o typ „*JF-1039B 6VDC*“

### SSR a další

Vzhledem k tomu že nabíjecí dok musí umět odpojovat nabíječku při dosažení nabití, je nutné nějak nabíječku elektricky odpojit od přívodního napětí. K tomu můžeme použít klasické relé, nebo v lepším případě SSR, tedy bezkontaktní *solid state relé*, které

*Obrázek 19 - Elektromagnetický solenoid (zdroj: hadex.cz )*

nemá žádné mechanické části, které by při častém používání selhat.

Kompletní seznam na nákup pro tvorbu nového doku je v příloze.



Obrázek 20 - Bezkontaktní relé (Zdroj: hadex.cz)

## 5.3 Implementace

V této kapitole se zaměříme na implementaci samotného doku, podíváme se jak na elektroniku a řídicí jednotku, na funkčnost po stránce kódu, ale také na mechanickou část zamykání a jak je vyřešena.

### 5.3.1 Řídicí jednotka a elektronika

Hlavním mozkiem každého doku je Arduino NANO, které se stará o čtení všech senzorů, komunikaci po sběrnici I2C o odemykání a zamykání, o zapínání a vypínání nabíječky a také o identifikaci připojené koloběžky.

#### Program

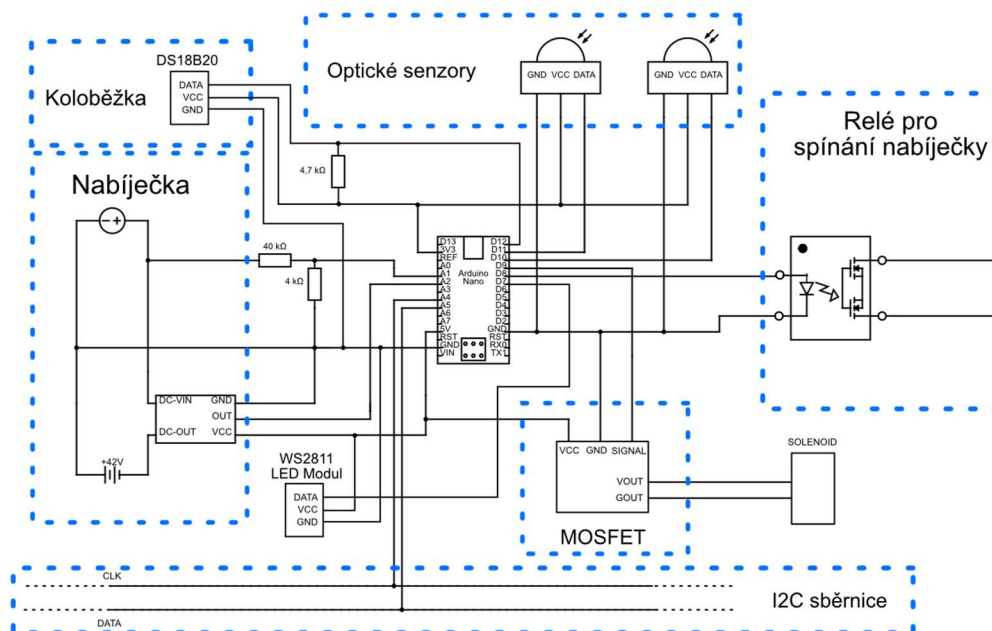
Řídicí jednotka má několik stavů, které aktualizuje a na žádost posílá doku. Program řídicí jednotky funguje v neustálém cyklu, kdy se stará o několik věcí. Za prvé každou vteřinu kontroluje připojené 1-wire zařízení a pokud dojde k připojení jakéhokoliv 1-wire zařízení, tak ho identifikuje a zaznamená do svého stavu jeho ID. V momentě, kdy dojde k odpojení tak jeho ID vymaže. Druhým úkolem je detekovat stav nabíječky, tedy nabíjecí proud. To probíhá za pomoci ADC pinu, tedy vstupu, který dokáže detekovat analogový vstup v rozmezí 0-1024. Tento údaj následně zapisuje do příslušného stavu. Dalším úkolem je řídit nabíjení, tedy vypínat nabíječku po dokončení nabíjení. To se provádí rozepnutím digitálního pinu, tedy přepnutím z HIGH na LOW.

Dalším úkolem je kontrolovat stav zámku a ovládat západku. Západka se ovládá za pomoci digitálního outputu, kterým se spíná silový MOSFET, který následně přitáhne elektromagnet. Stav západky se snímá pomocí dvou optických spínačů a jejich hodnoty se zapisují do příslušných stavů.

To byla první hlavní část o co se musí starat, ovšem je zde ještě stejně důležitá druhá část a tou je komunikace. Program se musí starat o odpovědi na vyžádání od stanice. Při každém vyžádání zabalí všechny stavy a vytvoří z nich kontrolní součet, který přibalí spolu s balíčkem stavů do těla zprávy pro stanici. V případě, kdy je s žádostí od stanice přibalen i příkaz, je vykonán na základě toho, o jaký příkaz se jedná, tedy odemknutí, zapnutí nabíjení, nebo vypnutí nabíjení. Celý kód je v příloze s kódy ve složce /dok/program/.

## Zapojení

Řídící jednotka musí být propojena se svými periferiemi. Je nutné vyřešit zapojení mikrokontroleru na napájení, i2c sběrnici, zapojení MOSFETů, senzorů a relé. Vše je propojené podle přiložených diagramů, a tak je lehké zapojení reprodukovat.



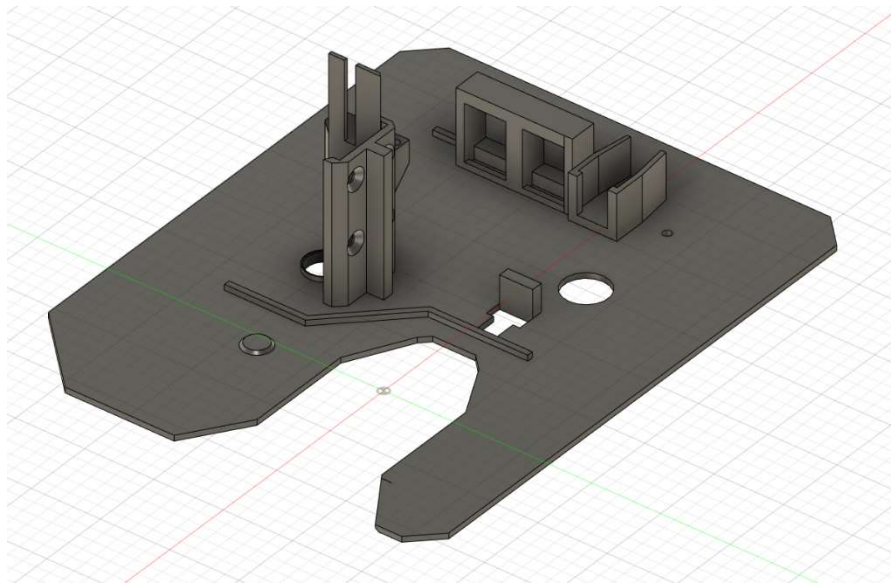
Obrázek 21 - Diagram zapojení doku (Zdroj: Autor)



Je však nutné si říct o zapojení I2C sběrnice. Vzhledem k návrhu, kdy se počítá s využitím možnosti řetězení stanic za sebou, musí být sběrnice zapojena tak aby se jednalo o průchozí vodiče, na které je pouze Arduino napojeno, díky tomu se může dále na dok připojit další a další doky.

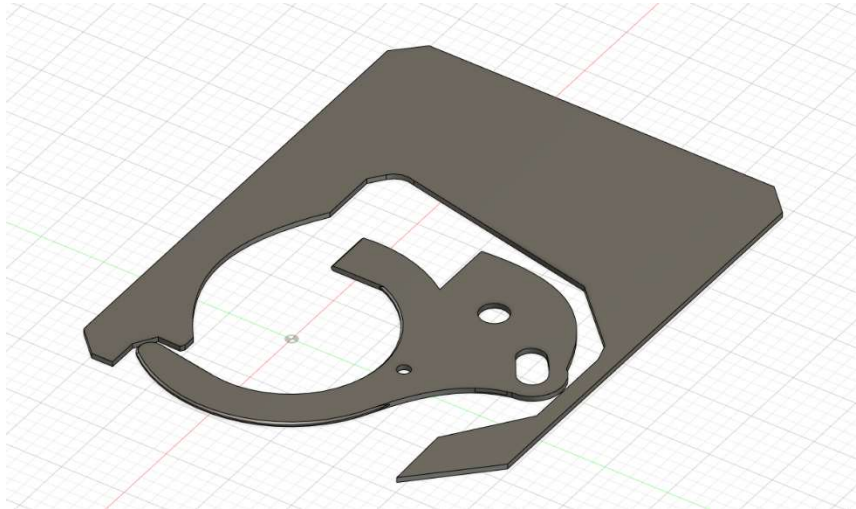
### 5.3.2 Mechanická část pro zamykání

Mechanická část pro zamykání je vyřešena formou sendviče, díky tomu lze snadno vytisknout jednotlivé vrstvy zámku na 3D tiskárně, ale také je možné jednotlivé vrstvy nahradit kovovou součástí kterou lze vyříznout z plechu a pouze vyměnit plastovou za kovovou v sendviči. Díky tomu je možné zlepšit bezpečnost zámku. Základ zámku se skládá ze tří vrstev a my si je nyní projdeme od spodní až po horní. První vrstvou je základní vrstva, která obsahuje prvky pro montáž ke kovovým L profilům a také poskytuje krytí západky a možnost skrytí vedení kabelů a uchycení elektromagnetu.



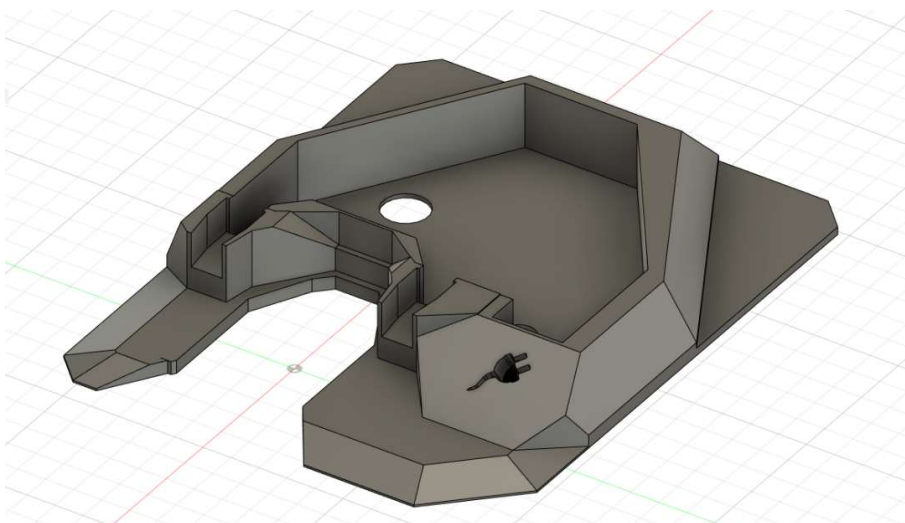
*Obrázek 22 - První vrstva doku – render modelu z pohledu ze spodní části = orientace tisku na 3D tiskárně (zdroj: Autor)*

Druhá vrstva tohoto sendviče se skládá ze dvou částí, první je pohyblivá část, a to je samotný zámek, který zamkne koloběžku, který je třeba mít zabezpečený čepem. Druhá část je statická a jedná se víceméně o výplň mezi vrstvami, která je ovšem pro funkčnost celku potřebná.



Obrázek 23- Prostřední vrstva doku – render modelu z pohledu ze shora (zdroj: Autor)

Třetí vrstva kryje samotný zámek a poskytuje pevné umístění všech komponentů. Tato část zámku je nejkompexnější, protože řeší jak uchycení jednotlivých konektorů, tak i řídicí jednotky. Tato vrstva je tedy i složitější na samotné vytvoření a je tedy předpoklad, že i kdyby došlo k vylepšení ostatních vrstev sendviče na kovové části tato by zůstala plastová. To by nevadilo, protože tuto vrstvu nejsou kladeny zvýšené požadavky na pevnost. Níže si popíšeme některé části více do detailu.

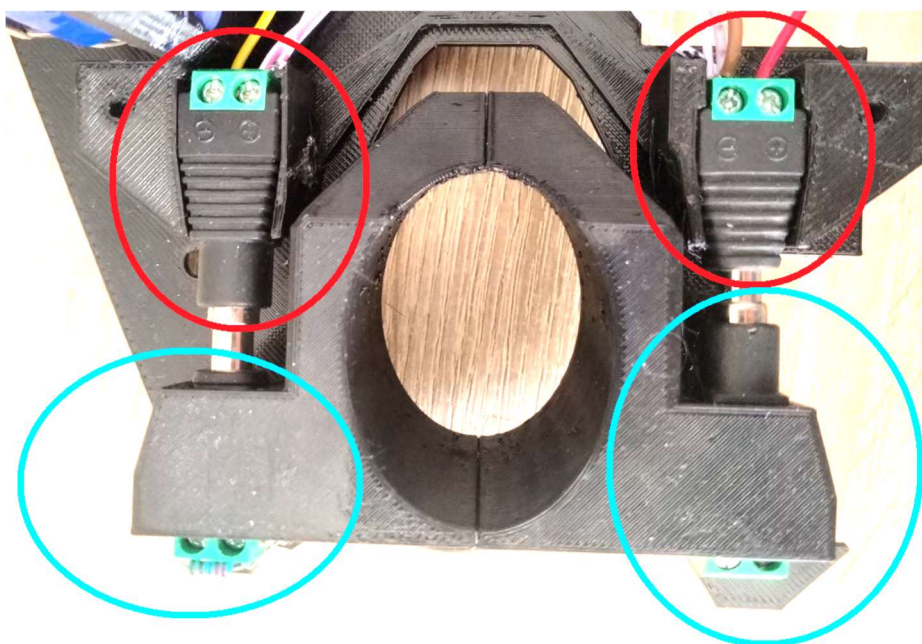


Obrázek 24- Horní vrstva doku – render modelu z pohledu ze shora = orientace 3D tisku (zdroj: Autor)

## Konektory na elektrokoloběžku

Koloběžka musí být ke stanici připojena celkem 4 vodiči, a to GND, tedy zem, +42V, +3V a DATA. Pro napájení samotné elektrokoloběžky je použit stejný konektor jako používá samotná elektrokoloběžka. Díky tomu je možné koloběžku nabíjet i mimo dobíjecí stanici. Tento typ konektoru byl použit i pro přenos +3V a dat potřebných pro identifikaci koloběžky. 3V se přenáší v konektoru zvlášť a nepoužívá se pouze odbočka ze 42v se stabilizátorem na 3V z toho důvodu že je nutné identifikovat koloběžku ještě před spuštěním napájení koloběžky.

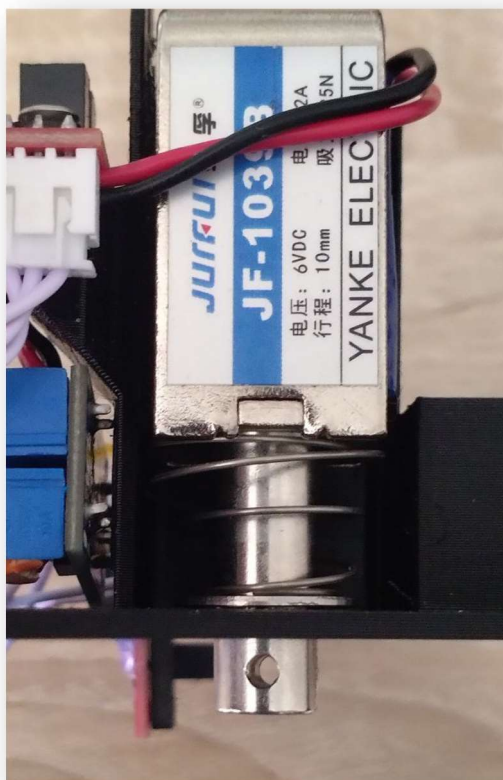
Konektory musí být umístěny relativně přesně aby došlo ke snadnému srovnání připojované části konektorů na koloběžce k připojovací části na doku. Samotné konektory ve výtisku drží velmi pevně, ale je doporučeno konektory na straně doku zabezpečit lepidlem v momentě kdy je koloběžka zadokovaná aby konektory zůstaly napevno v pozici kdy je nejsnazší koloběžku zadokovat. Na obrázku níže můžeme vidět konektory zasazené v doku v červených kroužcích a konektory zasazené v objímce koloběžky v modrých kroužcích.



*Obrázek 25 - Prototyp konektorů doku s objímkou koloběžky, modře znázorněny konektory na koloběžce, červeně znázorněny konektory na doku samotném (Zdroj: Autor)*

## Uchycení solenoidu/elektromagnetu

Zámek sám o sobě nedokáže zamezit pootočení a tedy otevření. Na to slouží západka ovládaná elektromagnetem, který vsune čep do zámku v momentě otevření, nebo zavření a tím dojde k jeho zamknutí samotný elektromagnet tedy musí být se svým čepem umístěn poměrně přesně. Na to slouží vytisknutý profil, umožňující ukotvení elektromagnetu. Pokud by v budoucnu byl využit jiný model elektromagnetu bude nutné tuto část poupravit.



*Obrázek 26 - Uchycení elektromagnetu  
(Zdroj: Autor)*

### 5.3.3 Rozhraní

Aby mohl dok upozorňovat uživatele na svůj stav, je vybaven vizuální signalizací, implementovanou v osvětleném logu. Podobně jako u stanice, kde je osvětlený symbol čtečky karet, zde signalizace slouží pro určení stavu, kdy modrá znamená, že je dok v pořádku připraven, zelená, že je připraven pro odemknutí nebo vrácení koloběžky a červená, která značí, že je ve stavu, kdy západka není zapadnutá a kvůli tomu se nenachází ve stavu otevřeno ani zavřeno a je třeba západku ručně přemístit do stavu odemknuto nebo zamknuto.



### 5.3.4 Koloběžka

Aby celý systém mohl fungovat, aby každá koloběžka měla své ID , aby šla bezpečně navést na konektor napájení, je zapotřebí aby na koloběžce byla vodící objímka, která bude obsahovat čip a konektory. Mimo to, že umožní snadné napojení na nabíječku do doku, objímka také zajistí, aby nedocházelo k poškrábání koloběžky v případě, že se dok vylepší na kovovou variantu, protože objímka kryje veškerá kontaktní místa koloběžky s dokem. Objímka na koloběžce vypadá následovně



Obrázek 27 - Koloběžka s objímkou vyznačenou červeným kolečkem (Zdroj: Autor)

V objímce je zasazen čip *Dallas ds18b20*, který vyžaduje odpor mezi DATA pinem a 3V ve velikosti ideálně 4.7 k $\Omega$ .. Čip je zapojen na levém konektoru. Aby nedošlo k záměně a nebylo připojeno k čipu 42V je konektor otočen a použit konektor typu *male*.

## 6 Celý systém

Po sestavení všech modulů, tedy serveru, stanice a doku je již možné celý systém pospojovat s pustit. V současné době je server na stejném fyzickém zařízení jako je stanice, tedy na Raspberry Pi 4. To přináší výhodu v tom, že nemusíme používat další server a komunikace nepůjde po síti, tedy bude daleko rychlejší, ale zároveň vznikají další úkony, které jsou třeba udělat. Nejprve musíme pozměnit konfigurace systému, aby mohl na jednom serveru souběžně fungovat webový server s webem pro displej tak i s webem pro webové rozhraní. Rozhraní pro display musí běžet pouze na localhostu a rozhraní pro web musí fungovat i z externích adres.

### 6.1 Párování ID karty

Vzhledem k nedostupnosti spárování ID karty s osobním číslem v systému STAG, tedy nedostupnosti potřebného API, které by poskytovalo odpověď s ID karty podle OS čísla uživatele, bylo nutné vyřešit přidělování ID karet jiným způsobem. Jako nejjednodušší způsob přicházelo v úvahu využít externí čtečku, která by přečetla ID a uživatel nebo administrátor by ID ručně do systému zadával. Tento způsob byl nakonec zachován pro lokální uživatele kdy, pokud administrátor vytvoří lokálního uživatele a povolí mu výpůjčky, tak mu může v nastavení přidělit ID karty.

Ovšem pro uživatele, přihlašující se přes systém STAG, byl vyvinut jiný způsob, který využívá samotnou dokovací stanici. Po registraci na webu se uživateli zobrazí chybové hlášení, které ho upozorňuje, že musí mít nejprve účet aktivovaný administrátorem a musí si o aktivaci zažádat emailem. Po zažádání emailem o aktivaci a po aktivaci provedené systémovým administrátorem je účet aktivovaný a může si uživatel vypůjčit koloběžku. Po přihlášení ve webovém rozhraní se mu ovšem stále zobrazí chybová hláška tentokrát o nespárované kartě. Spárování karty probíhá až u stanice při jejím prvním použití uživatelem. Pokud uživatel přiloží kartu, která není spárovaná s žádným účtem, zobrazí se mu možnost kartu spárovat. Párování probíhá tak, že je uživatel přesměrován na přihlášení pomocí identity STAG. Po přihlášení je zasláno ID načtené karty, společně s osobním číslem uživatele STAG na server, který kartu přidělí. Poté již stačí uživateli pouze znovu kartu načíst.

## 6.2 Modularita

Obrovskou výhodou tohoto řešení je jeho modularita na dvou úrovních. První úroveň je modularita spočívající v možnosti rozšíření doků. Každý dok je samostatná jednotka, která komunikuje se stanicí na sběrnici a umožňuje řetězení dalších doků na jednom společném nosném mediu dat (I2C sběrnice). To v důsledku umožňuje na jednu stanicí, do jednoho jediného konektoru, zapojit více než 100 doků. Což otevírá možnost pro v reálném užití prakticky neomezené možnosti rozšiřování stanice.

Druhou úrovní rozšiřitelnosti je úroveň samotných stanic, kdy celý systém není napsaný pouze pro jednu stanicí, ale umožňuje řízení neomezeného počtu stanic, které mezi sebou sdílejí informace o koloběžkách, uživatelích i docích. To umožňuje do budoucna postavit další dobíjecí stanici, která se napojí na server, který automaticky přidá na úvodní stránku informace o další stanici a umožní uživatelům vracet koloběžky do kompatibilních doků kdekoliv v celém ekosystému. Prohazování koloběžek mezi jednotlivými stanicemi není vše, co systém umožňuje. Další možností je bez jakéhokoliv problému odpojit dok a přepojit do jiné stanice, kde se během pár vteřin automaticky přeregistruje bez jakéhokoliv nastavování administrátorem.

## 6.3 Sestavený systém

Celý systém je zamýšlen pro nasazení v Katedře informatiky PřF JU, který se ovšem bude stěhovat do nové budovy, a proto je finální nasazení zamýšleno až v nové lokaci. Systém je otestovaný jako celek a po vzájemném propojení poskytuje očekávanou funkcionalitu. V současném stavu se jedná o prototyp, který je vytištěný na 3D tiskárně z materiálu PET-G. Ačkoliv se konstrukce ukázala jako dostatečně pevná, hlavně díky orientaci tisku, kdy celé díly jsou tisknuty na plocho, při finální instalaci se počítá s úpravou některých součástí do kovového provedení. To zvýší fyzickou odolnost celého mechanismu proti útokům.



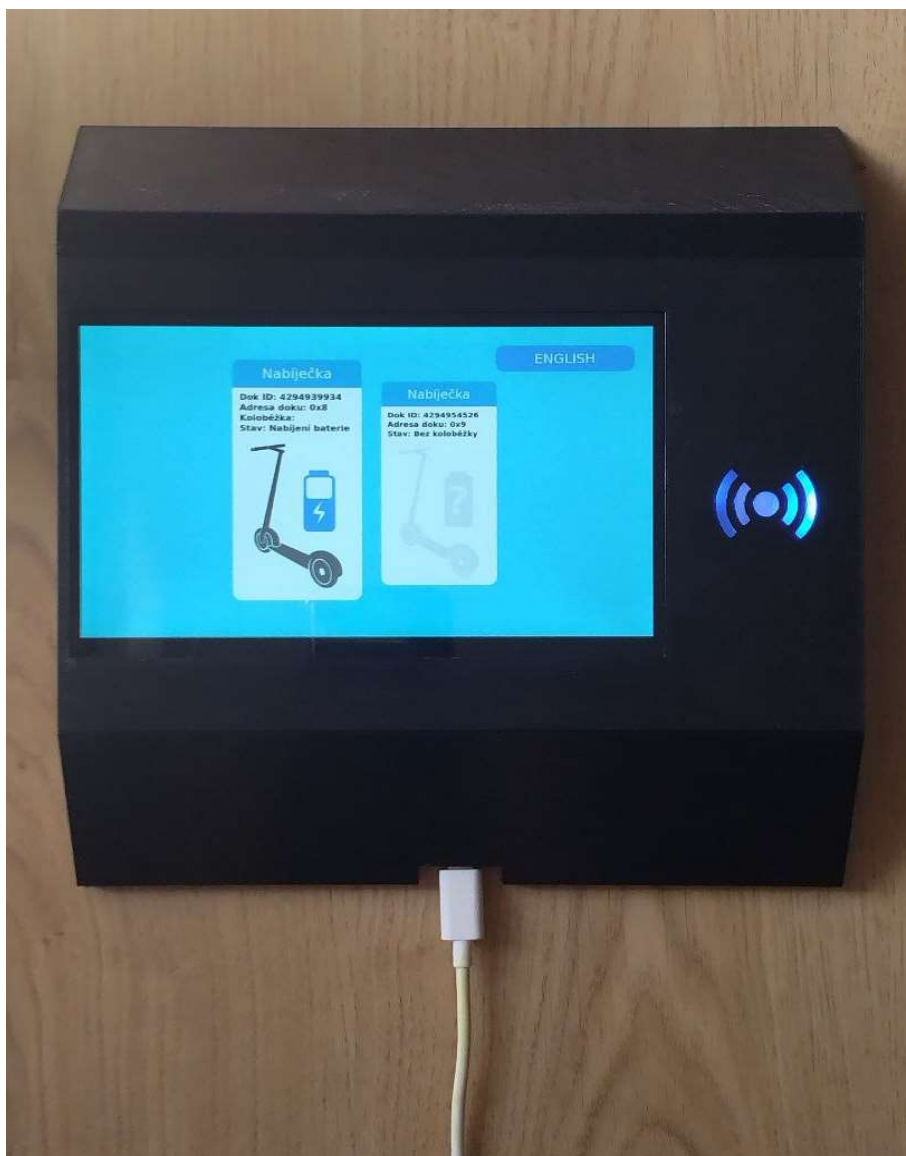
Jednotlivé doky se po připojení ke stanici zapnou a začnou pomocí RGB loga signalizovat svůj stav. Signalizace koresponduje se signalizací samotné stanice, tedy modrá barva znamená běžný stav, kdy je dok připraven a funguje dle očekávání. Zelená barva signalizuje odemknutí doku, aby uživatel věděl, jakou koloběžku si může půjčit, nebo naopak kam koloběžku vrátit. Poslední barvou je červená, která signalizuje že stanice potřebuje akci od uživatele, a to buďto plně zasunou koloběžku nebo naopak vysunout.



*Obrázek 28 - Dok pro koloběžku, nahoře otevřený, dole zavřený*

*(Zdroj: Autor)*

Celá stanice se skládá ze dvou hlavních částí, ze samotného displeje s hlavní elektronikou celé stanice a druhá část obsahuje elektroniku pracující na 230 V a také 5 V zdroj pro stanici. Díky tomu rozdělení je bezpečně oddělena pro člověka bezpečná 5 V úroveň od nebezpečného střídavého napětí. Velkou výhodou tohoto řešení také je, že od stanice může vést pouze jeden kabel, který zajišťuje jak napájení, tak komunikaci. Jednotlivé doky jsou poté zapojovány přímo k napájení. Obrázek níže **Chyba! Nenalezen zdroj odkazů.** ukazuje zapojenou stanici pomocí USB-C kabelu a na síť připojenou pomocí Wi-Fi. Stanice podporuje také kabelové připojení k internetu.



Obrázek 29 - Zapnutá stanice (Zdroj: Autor)



## 7 Testování

Tato kapitola se věnuje testování a ověřování funkcionality celého systému. Testování je důležitou součástí vývoje softwarových systémů, která pomáhá zajistit správné fungování a kvalitu výsledného produktu. V této kapitole se budeme zabývat jednotlivými testovacími scénáři, které byly navrženy a provedeny s cílem ověřit správnost jednotlivých částí systému.

Každý testovací scénář se zaměřuje na konkrétní aspekt systému a testuje se pomocí různých vstupů a očekávaných výstupů. Testování je rozděleno na dva hlavní celky. A to na první, kde je testována funkčnost serveru a lokální stanice, jelikož oba systémy pracují velmi provázaně. Druhá část se zaměřuje na testování periférií připojených ke stanici, tedy k testování doků a jejich funkcionalitě. Každý scénář obsahuje definici situace, kterou je třeba otestovat, popis testu a poté jak test proběhl a s jakými výsledky. Níže je seznam testovacích scénářů:

- Načítání karty a komunikace se serverem
- Zpracování dat pro kontrolu a povolení výpůjčky
- Přihlášení do webového rozhraní
- Změna jazyka ve webovém rozhraní
- Kontrola přístupu k omezeným stránkám
- Zpracování dat na widgetech na nástěnce
- Editace prvků v databázi přes webové rozhraní
- Testování responzivity webu a správného zobrazení prvků
- Komunikace mezi doky a stanicí pomocí I2C
- Přiřazení adres zařízením na sběrnici I2C
- Reakce doků na příkazy od stanice a provedení fyzických akcí

Všechny tyto testovací scénáře byly provedeny s pozitivními výsledky a podrobnosti o testování jsou k dispozici u každého testovacího scénáře, které lze nalézt v příloze „testovací scénáře“.

## 8 Závěr

Tato práce si kladla za cíl vytvořit chytrou dobíjecí stanici na elektrokoloběžky Katedry informatiky PřF. Tato dobíjecí stanice má za úkol umožnit intuitivní zamykání a dobíjení elektrokoloběžek s pomocí identifikace užitím univerzitní čipové karty. Výsledkem celé práce, avšak není pouze dobíjecí stanice, ale celý ekosystém.

Tento vytvořený ekosystém se skládá ze tří hlavních částí. První částí je server, který slouží jako správce dat a sjednocuje ostatní moduly. Druhou částí je stanice, která poskytuje uživatelům grafické a fyzické rozhraní pomocí dotykového displeje a čtečky karet. Poslední částí jsou moduly doků, které umožňují propojení a řetězení doků v počtu přesahující 100 na jednu stanici.

Díky tomuto vytvořenému ekosystému je toto řešení rozšiřitelné nejen přidáváním dalších doků do stanic, ale také přidáváním nových stanic do dalších budov. Celý systém umožňuje sdílení koloběžek, uživatelů a doků mezi stanicemi, které je zajištěno pomocí serveru. Tato flexibilita má výhodu v tom, že uživatelé mohou vrátit koloběžky kdekoliv v rámci systému. Také je možné měnit doky mezi stanicemi bez jakékoli konfigurace. To přináší větší pohodlí a pro uživatele a flexibilitu pro administrátory.

### 8.1 Budoucí rozvoj

V budoucnosti je možné implementovat systém, který využívá měření nabíjecího napětí pro odhad času do plného nabití koloběžky. Tento systém by na základě stavu napětí dokázal předpovědět, jak dlouho bude trvat, než se koloběžka úplně nabije na 100% kapacity baterie. Dalším vylepšením by mohla být predikce dojezdu na základě aktuálního stavu nabití, což by uživatelům umožnilo plánovat své trasy efektivněji.

Díky modularitě systému je možné do budoucna přidávat nové typy doků s vlastní konfigurací, aniž by bylo nutné zasahovat do samotného programu systému. Každá koloběžka je identifikována unikátním ID, což otevírá možnost vytvoření univerzálního dobíjecího doku. Tento dok by dokázal na základě připojené koloběžky získat informace o jejím stavu nabití ze serveru a následně spustit nabíjení s optimalizovanými parametry pro danou koloběžku. Tím by vznikl dok, který by byl kompatibilní s jakoukoli koloběžkou nebo dokonce jiným zařízením, například elektrokolem, pokud by byl adaptován zámek a držák pro kolo.

Další možností je rozšiřovat síť stanic na různých místech kampusu, které by mohly spolupracovat pod jedním centrálním systémem, protože je na to již v současné době server připraven. Tento ekosystém by umožnil uživatelům půjčovat si koloběžky na jednom konci kampusu a vrátit je na druhém konci, což by zlepšilo dostupnost a flexibilitu pro všechny uživatele systému.

## 8.2 Výstup

Výstupem této práce je sestavení a implementování chytré dobíjecí stanice pro elektrické koloběžky spolu s odpovídajícím ekosystémem, který nabízí uživatelům pohodlné řešení a administrátorům umožňuje prakticky bezúdržbovou správu stanic a doků. Součástí tohoto výstupu jsou všechny okomentované kódy, 3D modely nebo diagramy. Druhým výstupem je dokumentace testovacích scénářů, která je součástí přílohy, kde lze najít jaké testovací scénáře byly použity a jaké byly jejich výsledky. Dalším výstupem je uživatelská dokumentace popisující užívání jak webového rozhraní, tak i samotné stanice. Posledním výstupem je dokumentace pro nasazení, která slouží jako návod pro sestavení stanice a nasazení, nebo i pro pozdější rozšíření.

## 9 Summary

The aim of this thesis was to create a smart charging station for electric scooters at the Institute of Applied Informatics. This charging station aimed to enable intuitive locking and charging of electric scooters using identification through a university smart card. The result of the entire project is not just a charging station but a complete ecosystem.

This created ecosystem consists of three main components. The first component is the server, which serves as a data manager and integrates other modules. The second component is the station, which provides users with a graphical and physical interface through a touchscreen display and a card reader. The last component is the dock module, which allow the connection and chaining of docks, with the capability of exceeding 100 docks per station.

Thanks to this created ecosystem, the solution is expandable not only by adding more docks to the stations but also by adding new stations to other buildings. This system enables the sharing of scooters, users, and docks between stations, which is already supported by the server. This flexibility has the advantage that users can return scooters anywhere within the system. It is also possible to interchange docks between stations without any configuration, providing greater convenience for users and flexibility for administrators.

Due to the modularity of the system, it is possible to add new types of docks with their own configurations without the need to modify the system's core code. Each scooter is identified by a unique ID, creating the possibility of constructing a universal charging dock. This dock could retrieve information about the scooter's charging status from the server based on the connected scooter and subsequently initiate charging with optimized parameters for that particular scooter. This would result in a dock compatible with any scooter or even other devices, such as electric bikes, if equipped with a lock and holder for bikes.

Another possibility is to expand the station network to various locations on the campus, which could collaborate under a central system, as the current server is already prepared for it. This ecosystem would allow users to borrow scooters from one end of the

campus and return them at the other end, improving accessibility and flexibility for all system users.



## 10 Seznam zdrojů

- [1] K. T. I. S. Alisa Arno, „Accelerometer assisted authentication scheme for smart bicycle lock,“ *IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015.
- [2] bolt.eu, „Bolt Blog,“ 30 3 2023. [Online]. Available: <https://bolt.eu/en/blog/bolt-introduces-the-bolt-5-its-fifth-generation-scooter-model/>. [Přístup získán 1 5 2023].
- [3] V. Holub, „Nejlepší elektrické koloběžky: Jak správně vybrat a na co si dát pozor,“ *Magazín - zboží.cz*, pp. <https://www.zbozi.cz/magazin/c/nejlepsi-elektricke-kolobezky/>, 2021.
- [4] smartmag.cz, „Recenze 7 nejlepších elektrokoloběžek 2023,“ [Online]. Available: <https://www.smartmag.cz/nejlepsi-elektrokolobezky/>. [Přístup získán 18 7 2023].
- [5] riderguide.com, „Rider Guide,“ 2023. [Online]. Available: <https://riderguide.com/guides/how-to-lock-an-electric-scooter/>. [Přístup získán 15 7 2023].
- [6] C. K. D. Nikolaev, „Smart parking for an ecological type of transport - electric,“ v *Journal of Physics: Conference Series*, 2021.
- [7] KNOT, „knotcity.com,“ [Online]. Available: <https://www.knotcity.com/en/>. [Přístup získán 10 7 2023].
- [8] more4motion.com, „Smart Docking and Charging Station for Electric Scooters,“ [Online]. Available: <https://more4motion.com/products/docking-charging-station-for-electric-scooters>. [Přístup získán 10 6 2023].
- [9] P. V. D. F. G. A. J. P. B. O. I. Gautham Ram Chandra Mouli, „Sustainable E-Bike Charging Station That Enables AC, DC and Wireless Charging from Solar Energy,“ *EPE-ECCE Europe*, pp. <https://www.mdpi.com/1996-1073/13/14/3549>, 10 7 2020.
- [10] V. -A. C.-B. S. S.-C. A. Martínez-Navarro, „Photovoltaic Electric Scooter Charger Dock for the Development of Sustainable Mobility in Urban Environments,“ *IEEE Access*, vol. 8, p. <https://ieeexplore.ieee.org/abstract/document/9195537>, 14 9 2020.
- [11] Insight Editor, „Traditional VS Modular IT Infrastructure,“ *Insight*, pp. [https://ca.insight.com/en\\_CA/content-and-resources/2017/10172017-traditional-vs-modular-it-infrastructure.html](https://ca.insight.com/en_CA/content-and-resources/2017/10172017-traditional-vs-modular-it-infrastructure.html), 17 10 2017.
- [12] O. Bellwood, „Science Proves That Buttons Are Better Than Touchscreens,“ *JALOPNIK*, pp. <https://jalopnik.com/science-proves-that-buttons-are-better-than-touchscreen-1849428675>, 18 9 2022.
- [13] digitalview.com, „Buttons or Touchscreen?,“ *digitalview.com*, pp. <https://www.digitalview.com/blog/buttons-or-touchscreen/>, 8 11 2009.

- [14] OWASP, „Logging,“ [Online]. Available: [https://cheatsheetseries.owasp.org/cheatsheets/Logging\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html). [Přístup získán 1 5 2023].
- [15] CIV-SIS ZČU, „WEBOVÉ SLUŽBY NAD IS/STAG,“ [Online]. Available: [https://is-stag.zcu.cz/napoveda/web-services/ws\\_ws.html](https://is-stag.zcu.cz/napoveda/web-services/ws_ws.html). [Přístup získán 1 4 2023].
- [16] Jihočeská univerzita, „webové služby STAG,“ Jihočeská univerzita, [Online]. Available: [https://stag-ws.jcu.cz/ws/web?pp\\_locale=cs&pp\\_reqType=render&pp\\_page=title](https://stag-ws.jcu.cz/ws/web?pp_locale=cs&pp_reqType=render&pp_page=title). [Přístup získán 1 4 2023].
- [17] Codingtxt.com, „The Advantages of PHP with MVC Architecture: More Flexible, Scalable, and Secure Web Applications,“ 18 3 2023. [Online]. [Přístup získán 7 5 2023].
- [18] C. Ribeiro, „The Best PHP Frameworks for 2023,“ 3 11 2022. [Online]. Available: <https://www.sitepoint.com/best-php-frameworks/>. [Přístup získán 11 2 2023].
- [19] S. P. Pooja Rajendra Chavan, „Comparison Study Between Performance of Laravel and Other PHP Frameworks,“ *International Journal of Research in Engineering*, 10 2021.
- [20] R. PARTRIDGE, „MINI PC VS RASPBERRY PI – WHICH IS THE BETTER VALUE?,“ 4 6 2022. [Online]. Available: <https://www.techaddressed.com/reviews/mini-pc-better-value-raspberry-pi/>. [Přístup získán 20 3 2023].
- [21] Cloudflare, Inc., „What is Web Application Security?,“ [Online]. Available: <https://www.cloudflare.com/learning/security/what-is-web-application-security/>. [Přístup získán 1 6 2023].
- [22] J. Juviler, „What Is an API Endpoint? (And Why Are They So Important?),“ 27 7 2022. [Online]. Available: <https://blog.hubspot.com/website/api-endpoint>. [Přístup získán 5 3 2023].
- [23] Antalis, „Jsme v modré barvě důvěryhodnější? Studie o vlivu barev,“ [Online]. Available: <https://www.antalis.cz/domu/co-nabizime/print/inspirace/aplikace-pripadove-studie/pripadove-studie/2020/10/studie-vlivu-barev.html>. [Přístup získán 1 7 2023].
- [24] EM MICROELECTRONIC - MARIN SA, „512 bit Read/Write Contactless Identification Device - EM4469,“ 2003. [Online]. Available: <https://datasheet.datasheetarchive.com/originals/library/Datasheet-015/DSA00257981.pdf>. [Přístup získán 8 6 2023].
- [25] ATMEL, „Standard Read/Write Crypto Identification IC e5561,“ 2006. [Online]. Available: <https://www.advanide.de/wp-content/uploads/products/rfid/Atmel%20e5561.pdf>. [Přístup získán 2023 6 1].
- [26] NXP B.V, „MIFARE Classic EV1 1K - Mainstream contactless smart card,“ 23 3 2018. [Online]. Available: [https://www.nxp.com/docs/en/data-sheet/MF1S50YYX\\_V1.pdf](https://www.nxp.com/docs/en/data-sheet/MF1S50YYX_V1.pdf).
- [27] NXP B.V, „MIFARE DESFire EV1 contactless multi-application IC,“ 2015. [Online]. Available: [https://www.nxp.com/docs/en/data-sheet/MF3ICDX21\\_41\\_81\\_SDS.pdf](https://www.nxp.com/docs/en/data-sheet/MF3ICDX21_41_81_SDS.pdf).

- [28] EM MICROELECTRONIC - MARIN SA, „EM4102 Read Only Contactless Identification Device,” 2002. [Online]. Available: [https://www.digchip.com/datasheets/download\\_datasheet.php?id=296813&part-number=EM4102](https://www.digchip.com/datasheets/download_datasheet.php?id=296813&part-number=EM4102). [Přístup získán 2023 11 7].
- [29] ekotip.cz, „Používané bezpečnostní systémy,” [Online]. Available: <http://www.ekotip.cz/index.php?n=Systemy.HomePage>. [Přístup získán 1 6 2023].
- [30] <https://www.isicskolam.cz/>, „ISIC školám,” [Online]. Available: [https://www.isicskolam.cz/faq\\_category/03-cipy-a-cipove-systemy-ss/](https://www.isicskolam.cz/faq_category/03-cipy-a-cipove-systemy-ss/). [Přístup získán 1 6 2023].
- [31] „What is a Raspberry Pi?,” [Online]. Available: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. [Přístup získán 20 6 2023].
- [32] Waveshare, „<https://www.waveshare.com/>,” [Online]. Available: <https://www.waveshare.com/product/raspberry-pi/displays/7inch-hdmi-lcd.htm>. [Přístup získán 6 7 2023].
- [33] „Modulový spínaný zdroj,” hotair.cz, 2023. [Online]. Available: <https://www.hotair.cz/detail/prumyslove-zdroje/modulove-pro-vestavbu/modulovy-spinany-zdroj-wxd-50-5-5v-10a-50w.html>. [Přístup získán 1 6 2023].
- [34] Redis Ltd, „Introduction to Redis,” 2023. [Online]. Available: <https://redis.io/docs/about/>. [Přístup získán 5 6 2023].
- [35] pimylifeup.com, „How to setup a Raspberry Pi RFID RC522 Chip,” [Online]. Available: <https://pimylifeup.com/raspberry-pi-rfid-rc522/>. [Přístup získán 4 2023].
- [36] R. & D. R. & A. M. Bhuiyan, Miniature Energy Harvesting Device for Wireless Sensors in Electric Power System., 2010.
- [37] Arduino, „<https://www.arduino.cc/en/hardware/>,” 11 4 2022. [Online]. Available: <https://www.arduino.cc/en/hardware>. [Přístup získán 14 5 2023].
- [38] Adafruit, „Raspberry Pi,” [Online]. Available: <https://www.adafruit.com/category/105>. [Přístup získán 1 7 2023].
- [39] bikeep, „Personal scooter parking,” 2022. [Online]. Available: <https://bikeep.com/electric-scooter-stations/>. [Přístup získán 1 3 2023].
- [40] ECLIPSERA s.r.o, „Optický koncový spínač pro RepRap 3D tiskárny,” 2016. [Online]. Available: <https://dratek.cz/docs/produkty/0/104/1474547390.pdf>. [Přístup získán 1 2 2023].

## 11 Seznam obrázků

Obrázek 1- Schéma prototypu dobíjecí stanice z univerzity z Petrohradu .....	16
Obrázek 2 - Dobíjecí stanice KNOT .....	17
Obrázek 3 - Elektrokoloběžka BOLT5 .....	17
Obrázek 4 - Dobíjecí stanice s vyžitím solární energie .....	18
Obrázek 5 - Obě varianty tlačítka pro přepnutí jazyka .....	28
Obrázek 6 - Grafické rozhraní webové stránky, hlavní stránka .....	29
Obrázek 7 - Mobilní verze webové stránky .....	30
Obrázek 8 - Raspberry Pi 4 .....	33
Obrázek 9 - Dotykový displej .....	34
Obrázek 10 - Průmyslový 5V 10A zdroj .....	35
Obrázek 11 - Sestavená funkční stanice .....	45
Obrázek 12 - Grafické rozhraní stanice .....	46
Obrázek 13 - Grafické rozhraní stanice, chybová hláška .....	47
Obrázek 14 - Zamykací stanice pro elektrokoloběžky – bikeep .....	48
Obrázek 15 - Graf nabíjecího proudu a napětí baterií .....	49
Obrázek 16 - Vlastní měření nabíjecího proudu a napětí .....	50
Obrázek 17 - Arduino Nano .....	52
Obrázek 18 - Optický spínač .....	53
Obrázek 19 - Elektromagnetický solenoid .....	54
Obrázek 20 - Bezkontaktní relé .....	55
Obrázek 21 - Diagram zapojení doku .....	56
Obrázek 22 - První vrstva doku .....	58
Obrázek 23- Prostřední vrstva doku .....	59
Obrázek 24- Horní vrstva doku .....	59
Obrázek 25 - Prototyp konektorů doku s objímkou koloběžky .....	60
Obrázek 26 - Uchycení elektromagnetu .....	61
Obrázek 27 - Koloběžka s objímkou vyznačenou červeným kolečkem .....	62
Obrázek 28 - Dok pro koloběžku, nahoře otevřený, dole zavřený .....	65
Obrázek 29 - Zapnutá stanice .....	66

## 12 Seznam tabulek

Tabulka 1 - Výběr prodávaných elektrokoloběžek s váhou.....	15
Tabulka 2 - Tabulka uživatele v DB.....	25
Tabulka 3 - API endpoint pro registraci koloběžky .....	27
Tabulka 4 - Zapojení RFID čtečky s Raspberry Pi .....	44

## 13 Seznam příloh

### 1) Vložených v dokumentu

- Příloha 1 – Testovací scénáře
- Příloha 2 – API Endpointy
- Příloha 3 – Nasazení
- Příloha 4 – Uživatelská dokumentace

### 2) Přiložených k dokumentu

- Příloha 5 – 3D modely
- Příloha 6 – Programy
- Příloha 7 – Diagramy
- Příloha 8 – Seznam součástí

# Přílohy

# Příloha 1 - Testovací scénáře

## 1 Testování serveru a stanice

V této podkapitole se zaměřujeme na testování komunikace mezi serverem a stanicí, poté na otestování serverové části a v poslední řadě na funkcionalitu samotné stanice.

### 1.1 Načítání karty a komunikace se serverem

V rámci testování funkcionality systému byly provedeny testy načítání karty a komunikace se serverem. Tento testovací scénář se zaměřuje na různé situace a ověřuje očekávané výsledky.

#### **1. část: Odeslání požadavku na server**

Nejprve byl proveden test odeslání požadavku na server s platnými vstupními parametry, tj. správným API klíčem, akcí „check“ a číslem karty. Očekávaný výsledek testu byl úspěšná odpověď od serveru, která potvrdila úspěšné načtení karty.

#### **2. část: Zpracování odpovědi od serveru**

V další části byly provedeny testy zpracování různých typů odpovědí od serveru v závislosti na jejich kódech. Pokud server odpoví kódem "*Access denied*", očekáváme, že systém adekvátně zareaguje a vyvolá chybovou zprávu v textovém rozhraní informující o nedostatečných oprávněních způsobených neplatným API klíčem. Pokud server odpoví kódem "*Unknown command*", očekáváme, že systém vyvolá chybovou zprávu v textovém rozhraní informující o neznámé akci. Pokud server odpoví *XML souborem* obsahujícím data o kartě, očekáváme, že systém tato data správně zpracuje a předá je pro další využití.

V případě, kdy server neodpověděl nebo byl nedostupný, očekáváme, že systém adekvátně zareagoval a vyvolal chybovou zprávu v grafickém rozhraní, informující o nedostupnosti serveru.

#### **Během testování byly ověřeny následující situace:**

- Ověření správné reakce systému při použití neplatného API klíče.
- Ověření správné reakce systému při použití neznámé akce.
- Ověření správné reakce systému při nedostupnosti serveru.



## **Výsledek testu:**

Výsledky testů byly úspěšně ověřeny a odpovídají očekávaným výsledkům. Test byl označen jako 'Úspěch', protože systém správně zpracoval vstupy, reagoval na různé situace a poskytl odpovídající výstupy v grafickém a textovém rozhraní.

### **1.2 Zpracování dat pro kontrolu a povolení půjčování koloběžek**

Systém provádí několik kontrol, aby určil, zda si uživatel může půjčit koloběžku. Postup zahrnuje kontrolu, zda je ke kartě přiřazen uživatel, zda má tento uživatel povolené půjčování, zda má aktivní výpůjčku a zda je dostupný dok pro konkrétní typ koloběžky. Dále se také kontroluje, zda uživatel má právo půjčit si další koloběžku.

#### **1. část: Kontrola přiřazení karty k uživateli**

Systém zpracovává data a zjišťuje, zda je karta přiřazena k uživateli. Pokud je karta přiřazena, systém pokračuje v dalším kroku. Pokud není karta přiřazena, zobrazí se uživateli informace s výzvou, aby si kartu přiřadil přihlášením.

#### **2. část: Kontrola povolení půjčování**

Systém zjišťuje, zda má uživatel povolené půjčování. Pokud má uživatel povolené půjčování, systém pokračuje v dalším kroku. Pokud nemá povolené půjčování, systém vypíše varování a nedovolí půjčení koloběžky.

#### **3. část: Kontrola aktivní výpůjčky**

Systém zjišťuje, zda má uživatel aktivní výpůjčku. Očekávaný výsledek: Pokud má uživatel aktivní výpůjčku, systém zobrazí možnost vrácení koloběžky. Možnost vrácení je povolena pouze pokud je volný DOK pro konkrétní typ koloběžky.

#### **4. část: Kontrola práva půjčit si další koloběžku**

Systém zjišťuje, zda má uživatel právo půjčit si další koloběžku. Pokud uživatel má právo půjčit si další koloběžku, systém povolí půjčení další koloběžky.

## **Testovací podmínky:**

Očekávané vstupy:

- Data o přiřazení karty
- Data o počtu možných výpůjček

- Data o aktivní výpůjčce a dostupném doku.

Očekávané výstupy:

- Zobrazení informací o přiřazení karty
- Varování o nepovoleném půjčování
- Možnost vrácení koloběžky
- Povolení nebo zakázání půjčení další koloběžky.

### **Výstup testu:**

Testování provedeného scénáře potvrdilo správné zpracování dat a kontroly při půjčování koloběžek. Systém správně zjišťuje přiřazení karty k uživateli, povolení půjčování, aktivní výpůjčku a dostupnost konkrétních doků pro konkrétní modely koloběžek. Zobrazuje odpovídající informace a případná varování v souladu s očekávaným chováním. Test byl označen jako 'Úspěch', protože systém správně reagoval na zadaná data a poskytl odpovídající výstupy.

## **1.3 Přihlášení do webového rozhraní**

V rámci testování funkcionality přihlašování do webového rozhraní systému byly provedeny testy, které zahrnovaly ověření bezpečnostních prvků, jako je CSRF ochrana, prevence proti SQL injection a nutnost vyplnění validního emailu. Dále bylo testováno správné chování systému při zadání nesprávného uživatelského jména nebo hesla. V neposlední řadě bylo provedeno otestování správného zobrazování uživatelské nebo administrátorské nástěnky.

### **1. část: Ověření CSRF ochrany**

V první části byl proveden test ověření CSRF ochrany systému při přihlašování. Očekávaný výsledek testu je úspěšné přihlášení pouze při použití platného CSRF tokenu. Při pokusu o přihlášení s neplatným nebo chybějícím CSRF tokenem očekáváme, že systém odmítne přihlášení a zobrazí chybovou zprávu.

### **2. část: Prevence proti SQL injection**

V druhé části byly provedeny testy na prevenci proti SQL injection útokům při zadávání uživatelského jména a hesla. Očekávaný výsledek testu je, že systém efektivně zabezpečuje vstupy a nedovoluje vkládání nebezpečných SQL příkazů.

### **3. část: Nutnost vyplnění validního emailu**

Následuje test ověření, zda systém vyžaduje vyplnění emailové adresy při přihlašování. Očekávaný výsledek testu je, že systém odmítne přihlášení, pokud uživatel nevyplní povinné pole pro email a zobrazí chybovou zprávu.

#### **4. část: Zadání nesprávného uživatelského jména nebo hesla**

V poslední řadě byly provedeny testy se zadáním nesprávného uživatelského jména a hesla. Očekávaný výsledek testu je, že systém v případě chybného uživatelského jména nebo hesla vypíše stejnou obecnou chybovou zprávu, aby neumožnil odhalení, který ze zadaných údajů je nesprávný.

#### **Testovací podmínky:**

Očekávané vstupy:

- Platný CSRF token
- Bezpečná uživatelská jména a hesla
- Platná emailová adresa

Očekávané chyby:

- Neplatný CSRF token
- SQL injection pokusy
- Vynechání vyplnění povinného pole
- Zadání nesprávného uživatelského jména a hesla

#### **Výstup testu:**

Výsledky testů byly úspěšně ověřeny a odpovídají očekávaným výsledkům. Test byl označen jako 'Úspěch', protože systém správně reagoval na různé bezpečnostní scénáře a poskytl odpovídající výstupy v grafickém rozhraní.

### **1.4 Změna jazyka ve webovém rozhraní**

Změna jazyka je realizována pomocí tlačítka umístěného na spodní části stránky, které přepíná mezi češtinou a angličtinou. Dále bylo testováno, zda jsou všechny texty na stránce správně přeložené a zda se správně nastavuje jazyk na základě proměnné "language" v cookies.

#### **1. část: Změna jazyka pomocí tlačítka na stránce**

V první části je nutné otestovat funkčnosti tlačítka pro změnu jazyka na stránce. Očekávaný výsledek testu je, změna jazyka stránky na příslušný jazyk (čeština nebo angličtina) a všechny texty na stránce jsou přeloženy do daného jazyka.

## **2. část: Ověření správného překladu textů na stránce**

Další částí bylo ověření správného překladu všech částí webu. To bylo provedeno procházením celého webu a ruční kontrolou správného překladu textů. Očekávaným výsledkem jsou všechny texty na stránce přeloženy správně do aktuálně zvoleného jazyka (čeština nebo angličtina).

## **3. část: Manipulace s cookies**

Posledním testem je manipulace s cookies, kdy se pozmění hodnota nastaveného jazyka, nebo se cookies smaže. Očekávaný výsledek: Pokud proměnná "*language*" neexistuje v cookies, automaticky se nastaví jazyk na češtinu. Pokud proměnná existuje a je definován jazyk "*english*", nastaví se jazyk na angličtinu. Pokud je definován jiný jazyk nebo nastane chyba v cookies, bude aktivní jazyk vždy čeština jako výchozí nastavení.

### **Testovací podmínky:**

Očekávané vstupy:

- Kliknutí na tlačítko pro změnu jazyka
- Kontrola překladu textů
- Manipulace s cookies.

Očekávané výstupy:

- Správná změna jazyka na stránce
- Správný překlad textů
- Správné nastavení jazyka na základě proměnné "*language*" v cookies.

### **Výstup testu:**

Testování provedeného scénáře potvrdilo správnou funkčnost změny jazyka ve webovém rozhraní. Všechny texty na stránce byly správně přeloženy do příslušného jazyka. Správné nastavení jazyka na základě proměnné "*language*" v cookies bylo

ověřeno. Test byl označen jako 'Úspěch', protože systém správně reagoval na změnu jazyka a poskytl odpovídající výstupy v grafickém rozhraní.

## 1.5 Kontrola přístupu k omezeným stránkám

Bylo provedeno testování správného omezení přístupu ke stránkám, které jsou dostupné pouze přihlášeným uživatelům nebo administrátorovi. Přístupová práva jsou řízena na základě role uživatele. Testovalo se, zda se nepřihlášený uživatel nemůže dostat na omezené stránky a zda přihlášený uživatel nemá přístup na stránky určené pouze pro administrátora.

### 1. část: Pokus o přístup nepřihlášeného uživatele na omezené stránky

Provedení: Nepřihlášený uživatel se pokusí přistoupit na stránky určené pouze přihlášeným uživatelům. Očekávaný výsledek: Systém zjistí, že uživatel není přihlášený a zobrazí chybovou zprávu s oznámením "Přístup odepřen".

### 2. část: Pokus o přístup přihlášeného uživatele na stránky určené pouze administrátorovi

Provedení: Přihlášený uživatel se pokusí přistoupit na stránky určené pouze administrátorovi. Očekávaný výsledek: Systém zkontroluje roli uživatele a zjistí, že nemá dostatečné oprávnění pro přístup na dané stránky. Zobrazí se chybová zpráva s oznámením "Přístup odepřen".

#### Testovací podmínky:

Očekávané vstupy:

- Pokus o přístup nepřihlášeného uživatele na omezené stránky
- Pokus o přístup přihlášeného uživatele na stránky určené pouze administrátorovi.

Očekávané výstupy:

- Zobrazení chybové zprávy s oznámením "Přístup odepřen".

#### Výstup testu:

Testování provedeného scénáře potvrdilo správné omezení přístupu k omezeným stránkám. Nepřihlášeným uživatelům byl odepřen přístup na stránky určené pouze přihlášeným uživatelům a přihlášeným uživatelům byl odepřen přístup na stránky určené

pouze administrátorovi. Test byl označen jako 'Úspěch', protože systém správně reagoval na nedostatečná oprávnění uživatelů a poskytl odpovídající chybové zprávy.

## 1.6 Zpracování dat na widgetech na nástěnce

Bylo provedeno testování zobrazování různých dat v různých widgetech na nástěnce v systému. Testovalo se, jak se widgety chovají při úpravách dat v databázi a jak reagují na různé scénáře, jako například prázdná data nebo naopak naplněná data. Některé widgety se chovají dynamicky, když nemají co zobrazit, tak se skryjí, zatímco jiné zůstanou viditelné a nabízejí tlačítko pro konfiguraci konkrétního prvku, který je spojen s widgetem.

### 1. část: Testování zobrazení widgetů při prázdných datech

V první části jsou v databázi všechna potřebná data vymazána. Očekávaný výsledek: Widgety, které nemají dostupná data, se skryjí na nástěnce, zatímco ostatní widgety zůstanou viditelné.

### 2. část: Testování zobrazení widgetů s naplněnými daty

V této části jsou v databázi vytvořeny a naplněny relevantní záznamy. Očekávaný výsledek: Widgety, které mají dostupná data, zobrazí odpovídající informace na nástěnce.

### 3. část: Testování chování widgetů s tlačítkem pro konfiguraci

Pro vybraný widget, který musí umožňovat konfiguraci je provedena úprava dat v databázi smazáním všech prvků. Očekávaný výsledek: Widget zůstane viditelný na nástěnce a nabídne uživateli tlačítko pro konfiguraci konkrétního prvku, který je spojen s widgetem.

#### Testovací podmínky:

Očekávané vstupy:

- Prázdná data v databázi
- Naplněná data v databázi
- Úprava dat v databázi.

Očekávané výstupy:

- Skrytí nebo zobrazení widgetů na nástěnce
- Zobrazení odpovídajících informací na widgetech
- Nabídka tlačítka pro konfiguraci konkrétního prvku umožňující konfiguraci.

### **Výstup testu:**

Testování provedeného scénáře potvrdilo správné chování widgetů při zobrazování různých dat na nástěnce. Widgety správně reagovaly na prázdná data a skryly se, pokud nebylo co zobrazit. Widgety také správně zobrazovaly relevantní informace při naplněných datech v databázi. Test byl označen jako 'Úspěch', protože widgety se správně chovaly při zobrazení různých dat a poskytovaly očekávané možnosti konfigurace prvků pro uživatele.

## **1.7 Editace prvků v databázi přes webové rozhraní**

System umožňuje přidávání a úpravy uživatelů a konfigurací koloběžek a doků přes webové rozhraní. Cílem testování bylo ověřit, zda dochází k správnému načítání dat, provedení změn a zda se data skutečně aktualizují v databázi. Dále se testovalo, zda při provádění změn nedochází k jiným problémům či nekonzistencím.

### **1. část: Testování načítání dat**

Z webového rozhraní se přistupuje k různým prvkům konfigurace, jako je správa uživatelů nebo konfigurace koloběžek a doků. Očekávaný výsledek: Data se správně načtou z databáze a zobrazí se ve webovém rozhraní.

### **2. část: Testování úprav dat**

Ve webovém rozhraní se provádí úpravy různých prvků, jako je přidání nového uživatele nebo editace, konfigurace koloběžek nebo doků. Očekávaný výsledek: Po provedení úprav se změny správně zaznamenají a aktualizují v databázi.

### **Testovací podmínky:**

Očekávané vstupy:

- Načítání dat z různých prvků
- Úpravy dat různých prvků.

Očekávané výstupy:

- Správné načtení dat
- Provedení změn v databázi

### **Výstup testu:**

Testování provedeného scénáře potvrdilo správnou funkčnost editace prvků v databázi přes webové rozhraní. Data se správně načítala a zobrazovala ve webovém rozhraní. Následně prováděné úpravy se správně zaznamenávaly a aktualizovaly v databázi. Test byl označen jako 'Úspěch', protože systém fungoval správně při editaci prvků v databázi přes webové rozhraní.

## **1.8 Testování responzivity webu a správného zobrazení prvků**

Bylo provedeno testování responzivity webu s cílem zjistit, zda se prvky správně zobrazují a zda je jejich obsah čitelný na různých zařízeních a různých rozlišeních obrazovky.

### **1. část: Responzivita webu**

Web byl otevřen na různých zařízeních s různými rozměry obrazovky, včetně testování pomocí emulace. Testovalo se, zda se prvky správně přizpůsobují velikosti obrazovky a zda je jejich obsah čitelný. Očekávaný výsledek: Prvky webu by se měly správně přizpůsobit velikosti obrazovky a jejich obsah by měl být čitelný bez nutnosti přibližování nebo posouvání.

### **2. část: Překrývání prvků**

Testovalo se, zda se prvek s tlačítkem překlada správně zobrazuje a zda nezakrývá části jiných prvků stránky na různých rozměrech obrazovky. Očekávaný výsledek: Prvek s tlačítkem překlada by se neměl překrývat s jinými prvky stránky.

### **Testovací podmínky:**

Očekávané vstupy:

- Otevření webu na různých zařízeních s různými rozměry obrazovky
- Kontrola zobrazení prvku s tlačítkem překlada

Očekávané výstupy:

- Správné zobrazení prvků
- Čitelnost obsahu



- Minimalizace překrývání prvků

### Výstup testu:

Testování provedeného scénáře potvrdilo správnou responzivitu webu a správné zobrazení prvků na různých zařízeních a rozměrech obrazovky. Nicméně bylo zjištěno, že prvek s tlačítkem překladu může na některých rozměrech obrazovky překrývat části jiných prvků stránky. Tato situace vyžaduje úpravu rozložení a velikosti prvků, aby byla zajištěna jejich viditelnost a použitelnost. Test byl označen jako 'Částečný úspěch', protože většina prvků webu se zobrazuje správně, avšak bylo by třeba provést další optimalizaci týkající se překrývání prvků.

## 2 Testování doků

V této kapitole se zaměříme na otestování funkčnosti doku a funkčnosti komunikace doku se stanicí.

### 2.1 Komunikace mezi doky a stanicí pomocí I2C

Stanice (Master) periodicky zasílá výzvy dokům (Slave), aby poslali svůj stav. Při přenosu stavu se také posílá CRC (cyklický redundantní kód) a stanice musí vždy zkontrolovat, zda se CRC shoduje. V případě chyby přenosu, buď v podobě neshody CRC nebo chyby v komunikaci na sběrnici, se pokus o výzvu opakuje celkem pětkrát. Pokud ani po pěti pokusech nedojde k úspěšnému přenosu, stanice se pokusí komunikovat s dalším dokem. Chyby mohou nastat v případě zaneprázdněné sběrnice nebo přenosu s poškozením a obsahem chyby.

#### 1. část: Přenos výzvy a stavu doků

Stanice odesílá výzvy dokům a požaduje jejich stav. Doky odpovídají se svým stavem a zasílají odpovídající CRC. Stanice kontroluje CRC a ověřuje shodu s odeslaným stavem. V případě chyby přenosu se opakuje výzva k doku.

#### 2. část: Opakování přenosu v případě chyby

V této části se testuje situace, kdy přenos stavu obsahuje chybu. Stanice zaznamená chybu v přenosu nebo neshodu CRC. Pokus o výzvu k doku se opakuje maximálně pětkrát. Pokud ani po pěti pokusech nedojde k úspěšnému přenosu, stanice se pokusí komunikovat s dalším dokem.

### **3. část: Chyby přenosu a zaneprázdněná sběrnice**

V další části se testuje situace, kdy sběrnice je zaneprázdněná nebo přenos obsahuje chybu. Stanice detekuje chybu v přenosu nebo zaneprázdněnou sběrnici. Pokus o výzvu k doku se opakuje v případě zaneprázdněné sběrnice. V případě chyby přenosu se opakuje celý přenos.

#### **Testovací podmínky:**

Očekávané vstupy:

- Výzva k doku

Očekávané výstupy:

- Stav
- CRC
- Opakování výzvy k doku
- Detekce chyby přenosu nebo zaneprázdněné sběrnice

#### **Výstup testu:**

Testování provedeného scénáře potvrdilo správnou komunikaci mezi doky a stanicí pomocí I2C sběrnice. Stanice správně odesílá výzvy a zjišťuje stav doků. Kontroluje CRC a případně opakuje výzvu k doku v případě chyby nebo neshody CRC. Pokud ani po pěti pokusech nedojde k úspěšnému přenosu, stanice se pokusí komunikovat s dalším dokem. Test byl označen jako 'Úspěch', protože systém správně reagoval na přenosy dat a detekoval případné chyby v komunikaci nebo na sběrnici.

## **2.2 Přiřazení adres dokům zařízením na sběrnici I2C**

Každý dok musí mít vlastní I2C adresu, která se přiřazuje dynamicky podle dostupných adres. Po spuštění dok čeká náhodnou dobu v rozmezí 0-5 sekund a v tomto časovém okamžiku vyhledá volnou adresu a přiřadí si ji. Testovalo se, zda proces přiřazování adres funguje správně a zda je možné připojit více doků na jednu sběrnici.

### **1. část: Přiřazení adresy dokům**

Doky jsou zapojeny na sběrnici I2C a po spuštění čekají náhodnou dobu v rozmezí 0-5 sekund. Každý dok vyhledá volnou adresu na sběrnici a přiřadí si ji. Adresy doků musí být unikátní a žádný dok by neměl mít stejnou adresu jako jiný dok na stejné sběrnici.

## **2. část: Připojení více doků na jednu sběrnici**

Více doků je připojeno na jednu sběrnici I2C. Každý dok si správně přiřadí svoji vlastní unikátní adresu na sběrnici, aniž by došlo k adresačním konfliktům. Sběrnice I2C je schopna správně komunikovat s více doky.

### **Testovací podmínky:**

Očekávané vstupy:

- Zapojení doků na sběrnici I2C

Očekávané výstupy:

- Unikátní adresy doků
- Schopnost komunikace mezi stanicí a více doky na sběrnici

### **Výstup testu:**

Testování provedeného scénáře potvrdilo správné přiřazování adres doků na sběrnici I2C. Doky si správně přiřadily vlastní unikátní adresy. Připojení více doků na jednu sběrnici bylo také úspěšné, a sběrnice I2C dokázala správně komunikovat s každým dokem pomocí přiřazených adres. Test byl označen jako 'Úspěch', protože systém správně přiřazoval adresy a umožňoval připojení více doků na sběrnici.

## **2.3 Reakce doků na příkazy od stanice a provedení fyzických akcí**

Bylo provedeno testování, zda doky správně reagují na příkazy od stanice a zda doky fyzicky vykonávají příslušné akce. Testovalo se, zda se příkazy opakují v případě chyb v CRC nebo v komunikaci a zda doky provádějí požadované akce, jako je odemknutí, zapnutí nebo vypnutí nabíjení.

### **1. část: Reakce na příkazy od mastera**

Stanice odesílá příkazy dokům a očekává reakci. Testuje se správná reakce doků na příkazy a případné opakování příkazů v případě chyb v komunikaci. V optimálním

stavu doky správně reagují na příkazy od mastera. Pokud dojde k chybě v CRC nebo v komunikaci, příkazy se opakují až do jejich úspěšného provedení.

## **2. část: Vykonání fyzických akcí**

Stanice vydává příkazy pro doky k provedení fyzických akcí, například odemknutí, zapnutí nebo vypnutí nabíjení. Doky fyzicky vykonávají příkazy, které obdrží od mastera. Například při příkazu k odemknutí se doky odemknou, při příkazu k zapnutí nabíjení se nabíjení zapne, a při příkazu k vypnutí nabíjení se nabíjení vypne.

### **Testovací podmínky:**

Očekávané vstupy:

- Příkazy od mastera pro doky

Očekávané výstupy:

- Správné reakce doků na příkazy
- Provedení fyzických akcí dle příkazů

### **Výstup testu:**

Testování provedeného scénáře potvrdilo správnou reakci doků na příkazy od mastera a provedení fyzických akcí. Doky správně reagovaly na příkazy a v případě chyb v komunikaci se příkazy opakovaly až do jejich úspěšného provedení. Doky také správně vykonávaly fyzické akce podle obdržených příkazů, jako je odemknutí, zapnutí nebo vypnutí nabíjení. Test byl označen jako 'Úspěch', protože doky se chovaly podle očekávání a prováděly požadované akce.

## Příloha 2 - API endpointy

<b>Registrace koloběžky</b>	
Api end point slouží k vytvoření nové koloběžky, v případě již existence koloběžky v databázi dojde k její úpravě, nebo aktivaci či deaktivaci.	
<b>/api/registerScooter</b>	
<b>Metoda</b>	<b>POST</b>
<b>Parametry</b>	<ul style="list-style-type: none"><li>- <b>api_key</b> (string): API klíč</li><li>- <b>scooter_id</b> (bigInt): ID koloběžky</li><li>- <b>action</b> (string): akce -&gt; "register"</li><li>- <b>is_active</b> (bool): status koloběžky reprezentován jako 1/0</li><li>- <b>battery_level</b>: stav baterie, defaultně 0</li><li>- <b>model</b>: model koloběžky</li></ul>
<b>Odpověď</b>	- „ <b>success</b> “ pokud se podařilo koloběžku vytvořit, upravit, nebo změnit stav, jinak „ <b>fail</b> “

<b>Registrace doku</b>	
Api end point slouží k vytvoření nového doku, v případě již existence dojde k úpravě.	
<b>/api/registerDock</b>	
<b>Metoda</b>	<b>POST</b>
<b>Parametry</b>	<ul style="list-style-type: none"><li>- <b>api_key</b> (string): API klíč</li><li>- <b>dock_id</b> (bigInt): ID doku</li><li>- <b>station_id</b> (bigInt): ID stanice, ve které se dok nachází</li><li>- <b>scooter_id</b> (bigInt): ID koloběžky v doku (může být prázdné)</li><li>- <b>action</b> (string): akce -&gt; "register"</li><li>- <b>status</b> (string): status stanice</li><li>- <b>dock_type</b>: model doku</li></ul>
<b>Odpověď</b>	- „ <b>success</b> “ pokud se podařilo dok vytvořit nebo upravit, jinak „ <b>fail</b> “

## Registrace stanice

Api end point slouží k vytvoření nové stanice, nebo její úpravě.

**/api/registerStation**

**Metoda**                      **POST**

**Parametry**

- **api\_key** (string): API klíč
- **station\_id** (Int): ID stanice
- **name** (string): jméno stanice
- **action** (string): akce -> "register"
- **location** (string): textový popis lokace
- **is\_active**: 0/1 podle stavu

**Odpověď**

- „**success**“ pokud se podařilo stanici vytvořit nebo upravit, jinak „**fail**“

## Start výpůjčky

Api end point slouží k vytvoření nové výpůjčky.

**/api/endRental**

**Metoda**                      **POST**

**Parametry**

- **api\_key** (string): API klíč
- **user\_id** (bigInt): ID doku
- **end\_location** (str): startovací lokace
- **scooter\_id** (bigInt): ID koloběžky
- **action** (string): akce -> "start"

**Odpověď**

- „**1**“ pokud se podařilo výpůjčku vytvořit, jinak „**0**“

## Ukončení výpůjčky

Api end point sloužící k ukončení výpůjčky.

**/api/startRental**

**Metoda**                      **POST**

**Parametry**

- **api\_key** (string): API klíč
- **user\_id** (bigInt): ID doku
- **start\_location** (str): startovací lokace
- **scooter\_id** (bigInt): ID koloběžky
- **action** (string): akce -> "end"

**Odpověď**

- „1“ pokud se podařilo výpůjčku ukončit, „0“ pokud se nepodařilo výpůjčku ukončit a „-1“ pokud nebyla nalezena výpůjčka pro daného uživatele a koloběžku

## Kontrola karty

Api end point sloužící ke kontrole čísla karty.

**/api/checkCard**

**Metoda**                      **POST**

**Parametry**

- **api\_key** (string): API klíč
- **card\_id** (bigInt): ID karty
- **action** (string): akce -> "check"

**Odpověď**

- V případě nálezu vrací **XML** obsahující informace o uživateli, jeho aktuální výpůjčky a parametr kolik výpůjček celkem může daný uživatel mít.

## Příloha 3 - Nasazení

Vzhledem k tomu že se celý systém skládá z několika prvků, které je potřeba sestavit, nastavit a naprogramovat, je pro nasazení systému zapotřebí provést kroky podle daného postupu. Celý hotový systém se musí skládat minimálně z jednoho doku stanice a serveru, aby byl systém funkční. V následujících kapitolách se podíváme na nasazení a také na uživatelskou dokumentaci a řešení některých problémů které mohou nastat při používání stanice. Součástí této kapitoly je dokumentace pro nasazení, kde je popsáno, jak jednotlivé prvky systému zkonstruovat případně nainstalovat.

### 1 Server

Pro nasazení serveru můžeme zvolit dvě možnosti, buďto na samostatný server, nebo společně se stanicí na Raspberry Pi. Konfigurace je pro oba případy prakticky totožná a liší se pouze v nastavení stanice. Pro instalaci serveru je zapotřebí na Linuxu nejprve rozbalit obsah složky „00-Instalace“ do `/var/www/html/`. Následně je nutné spustit příkaz `install_server1.sh` který nainstaluje databázi a apache, nakonfiguruje databázi a vytvoří administrátorský účet.

Server musí být připojený na internet a musí být spuštěn 24/7. dál je nutné nastavit Apache aby každý požadavek směřoval do složky serveru.

### 2 Stanice

Nasazení stanice se skládá ze dvou hlavních částí, konstrukce a nastavení. Stanice je zamýšlena pro provoz na Raspberry Pi 3 nebo 4. Prvním krokem je instalace systému Raspbian Lite 64bit s uživatelským účtem `juscooters`. Po instalaci systému je zapotřebí nainstalovat a nakonfigurovat vše podle textu práce. Následně je třeba rozbalit obsah složky „02-Stanice-opt“ do složky `/opt/` a obsah složky „02-Stanice-web“ do složky `/var/www/html/display`.

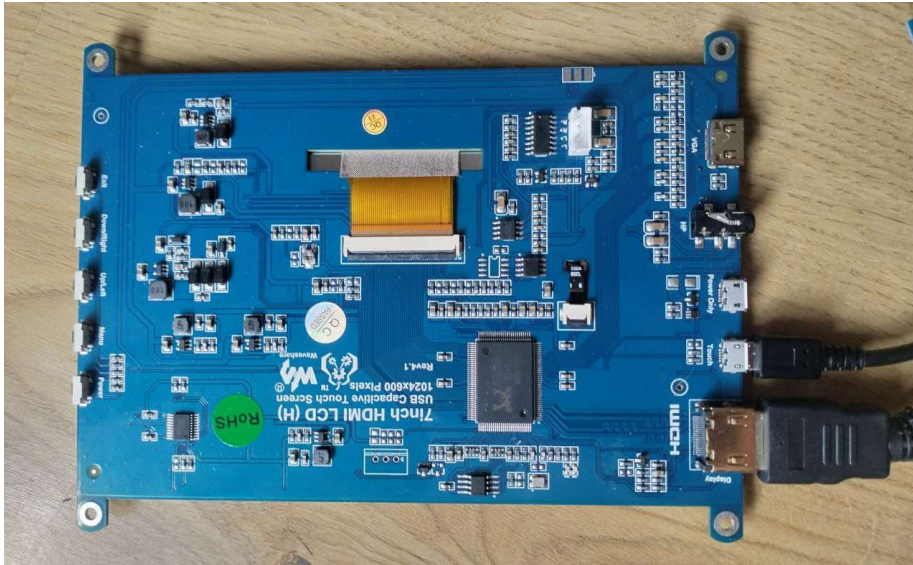
Pokud je server nasazen mimo Raspberry Pi, je nutné nakonfigurovat stanici pro komunikaci s externím serverem pomocí změny údajů v lokaci `/opt/juscooters/config.ini`. Pokud je ovšem server součástí stanice a běží na Raspberry Pi je nutné nakonfigurovat, aby lokální rozhraní bylo přístupné pouze z localhostu a server byl přístupný i zvenčí.

Pro konstrukci samotné stanice je zapotřebí zkonstruovat displej ale také box s hlavní elektronikou fungující na střídavém napětí 220 V, je tedy zapotřebí konstrukci boxu svěřit osobám k tomu oprávněným. Box lze zasadit do klasické pojistkové skříně.



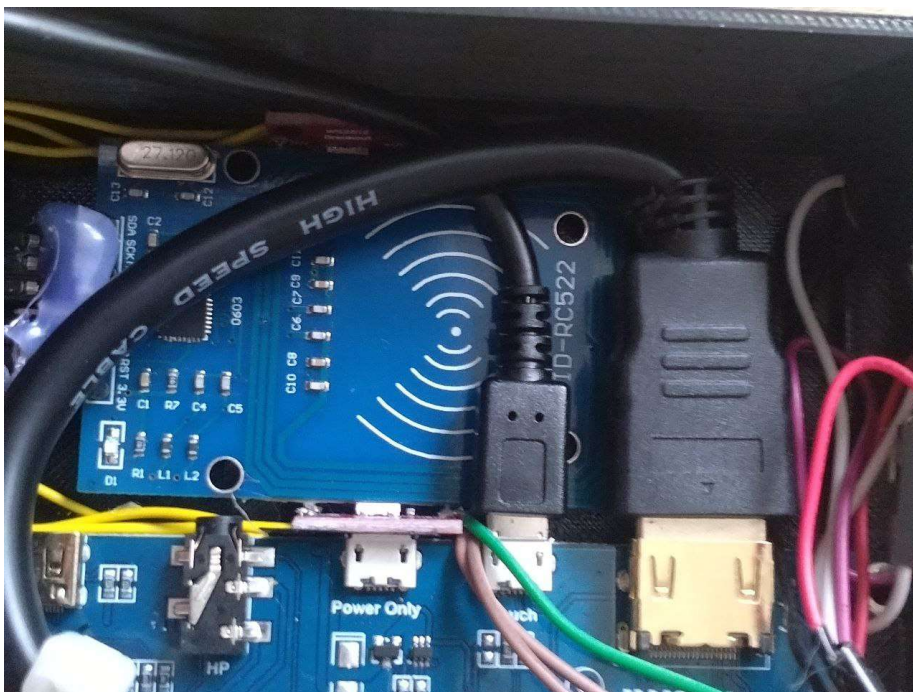
Nyní se zaměříme na konstrukci samotné stanice. Pro úspěšné dokončení je zapotřebí vytisknout všechny součástky ze složky „Příloha 5 – 3D modely“ z podsložky „01-stanice“. Následuje návod, doplněný o vyobrazení, jak stanici sestavit.

### 1. Krok – zapojení kabelů do displeje



### 2. Krok – usazení displeje do krytu

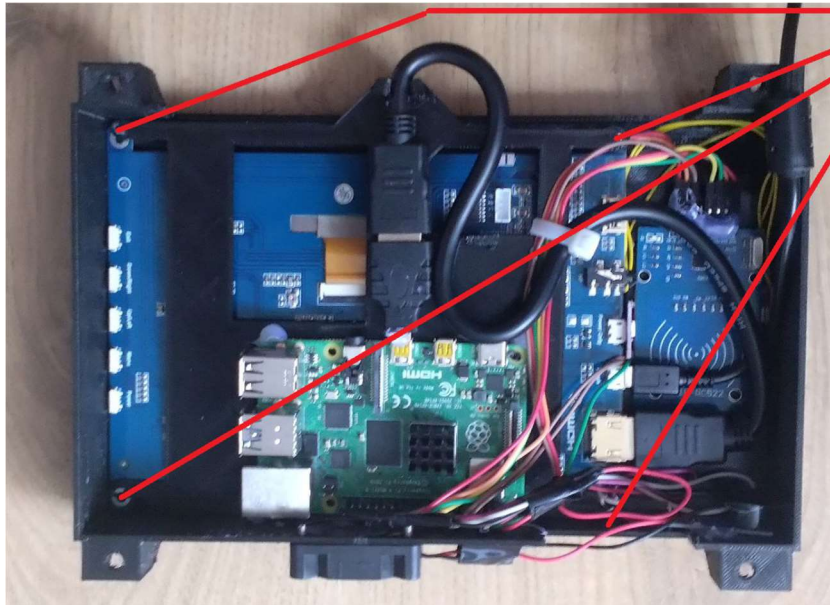
### 3. Krok – zapojení čtečky karet na Raspberry Pi



### 4. Krok – zapojení RGB adresovaných LED do Raspberry Pi

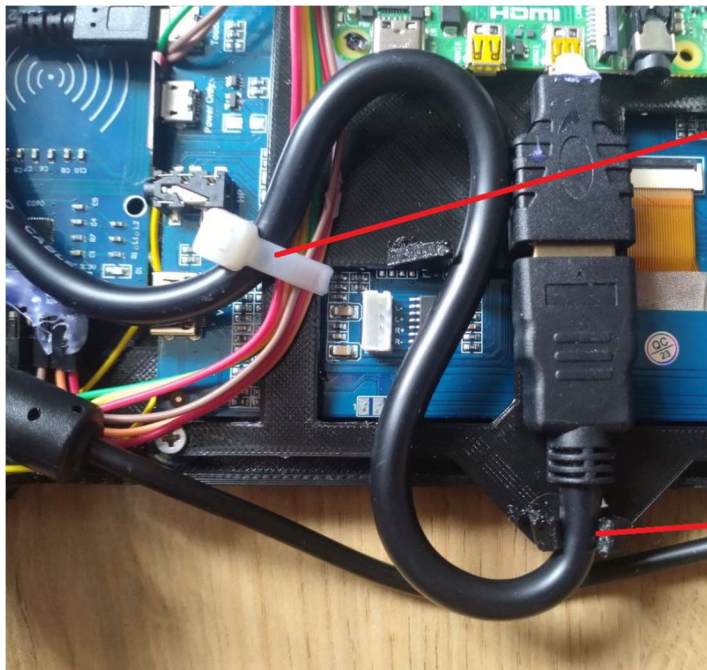
### 5. Krok – zapojení bzučáku k Raspberry Pi

**6. Krok – nasazení součástky č4 nad displej a zašroubování společně s displejem pomocí šroubů M3**



Místo pro šrouby

**7. Krok – upevnění kabelů**

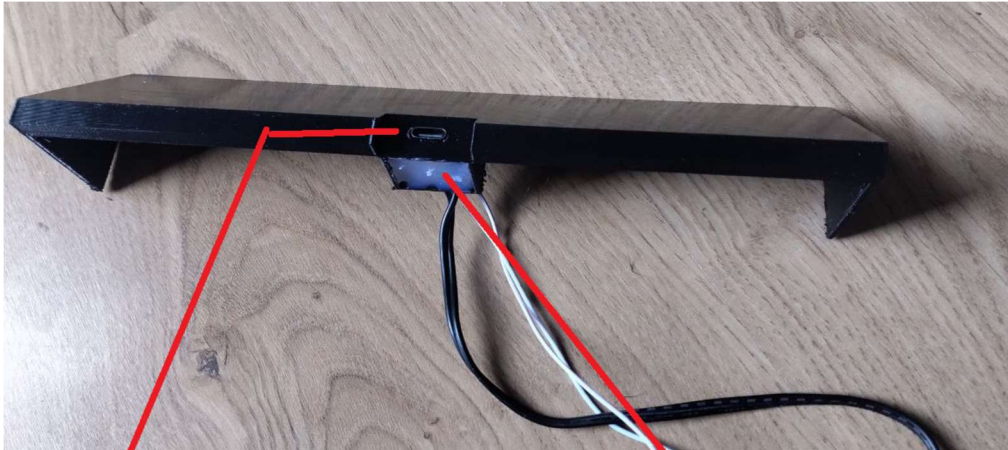


Místo pro přidělení páskou

Místo pro zacvaknutí kabelu



**8. Krok – zasazení USB konektoru do dílu č2**



port pro  
USB-C

Zalitií portu  
pro izolaci a  
bezpečnost

**9. Krok – zapojení USB konektoru a sběrnice I2C do Raspberry Pi**

**10. Krok – přišroubování hlavní konstrukce na zeď a zaslepení šroubů**



Místo pro  
přišroubování  
a následné  
zaslepení  
šroubů

## 11. Krok – zakrytování pomocí dílu č.2 a č.3 pomocí bočních šroubů M3



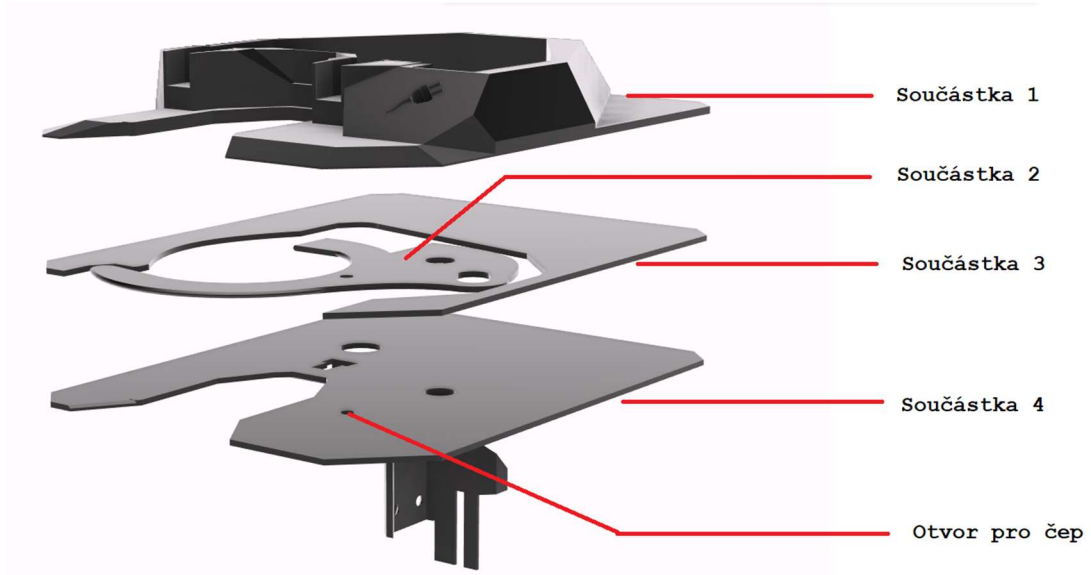
**Boční šrouby M3**

### 3 Dok

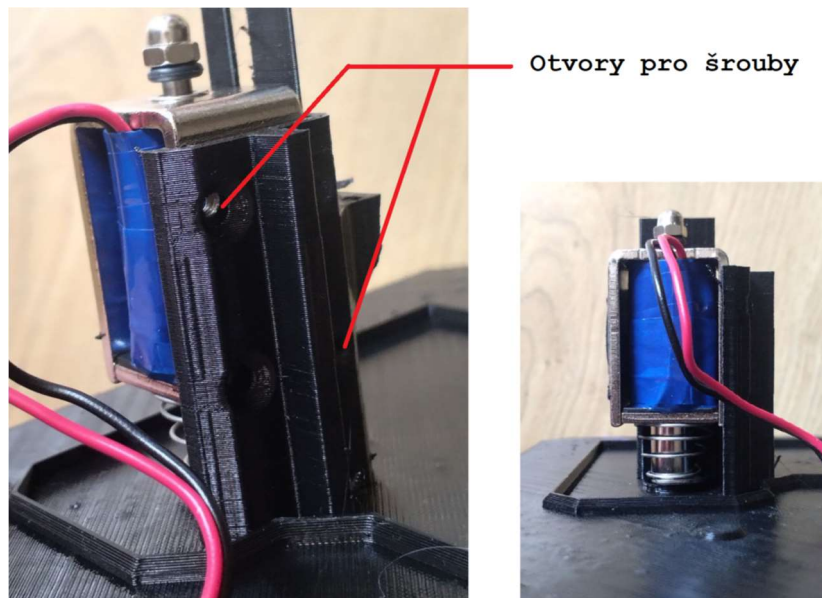
Dok je zapotřebí zkonstruovat podle přiloženého diagramu „*Zapojení doku.pdf*“. Při sestavování obvodu je třeba brát v úvahu maximální nabíjecí napětí a podle toho upravit poměr odporového děliče. Následně je nutné nahrát přiložený Arduino program *03-Dok.ino*, který lze nahrát bez úprav a dok bude fungovat. Pouze v případě, kdy se použijí komponenty jako například jiné sestavení odporového děliče, je zapotřebí tyto změny promítnout do programu, aby bylo měření přesné. Pokud sestavujeme dok pro nový typ koloběžky je zapotřebí v kódu upravit název modelu doku!

Nyní se zaměříme na konstrukci samotného doku. Pro úspěšné dokončení je zapotřebí vytisknout všechny součástky ze složky „*Příloha 5 – 3D modely*“ z podsložky „*02-dok*“. Následuje návod, obohacený o fotky, jak dok sestavit.

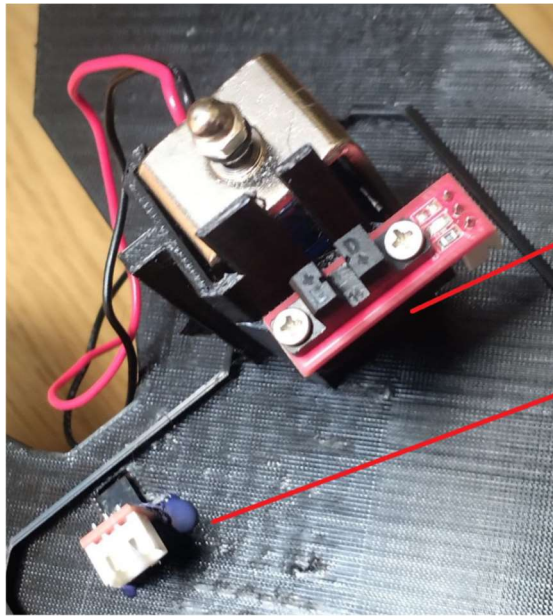
1. Krok – sestavení sendviče zámku dle přiloženého obrázku za užití dílů č.1,č.2,č.3 a č.5. V této části je možné jakýkoliv díl nahradit kovovým pro zlepšení bezpečnosti a pevnosti celé stanice je důležité vsadit kovový čep 5mmx9mm pro otáčení západky



2. Krok – nasazení solenoidu na součástku 4 a zajištění šrouby



**3. Krok – nasazení čidla č.1 a čidla č.2**



1. čidlo  
přišroubované

2. čidlo vlepené

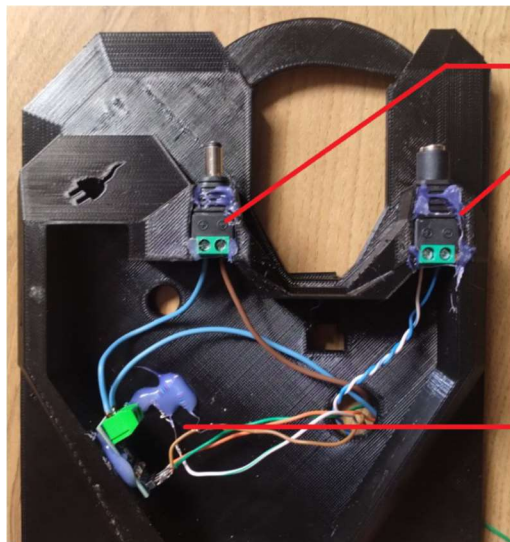
**4. Krok – usazení mikrokontroleru**

**5. Krok – usazení svorkovnice pro spínač nabíjení**

**6. Krok – usazení konektorů pro I2C**

**7. Krok – usazení konektorů pro koloběžku**

**8. Krok – zabezpečení konektorů a zakrytování doku**



Zasazení konektoru  
+ zabezpečení lepidlem

Zaliti obnažených  
součástek pro izolaci

**9. Krok – montáž doku na L profily (úhelníky) a následná montáž na zeď**

**10. Krok – zapojení I2C sběrnice**

**11. Krok – zapojení SSR (relé) pro spínání nabíjení do svorkovnice**

**12. Hotový dok**





#### 4 Koloběžka

Konstrukce objímky pro koloběžku nevyžaduje žádný software. Je zde zapotřebí pouze odpor 4,7K, čidlo Dallas DS18B20 a konektory. Celou objímku pospojujeme podle části přiloženého diagramu „Zapojení doku.pdf“. Následně připevníme objímku, tedy vytisknutý díl *L* a *R* společně pomocí M3 šroubů a vložených matek na koloběžku ve stejné výšce jako je umístěn dok. Model jsou k nalezení ve složce „Příloha 5 – 3D modely“ z podsložky „03-kolobezka“. Zde se nachází dvě verze, první již s otvorem pro krk koloběžky, druhý bez, který je připravený na úpravu v 3D CAD softwaru kdy pomocí jednoduché boolean operace lze vyříznout do objímky potřebný otvor.

# Příloha 4 - Uživatelská dokumentace

V této kapitole se uživatel dočte, jak se orientovat ve webovém rozhraní systému , jak obsluhovat stanici a jak si půjčovat koloběžky.

## 1 Webové rozhraní

Webové rozhraní poskytuje možnost rychle nahlédnout na aktuálně dostupné koloběžky ve stanicích. Pokud se uživatel chce přihlásit může pro to využít přihlašovací stránku kde je možnost se přihlásit lokálním uživatelským účtem, pro který je nutné získat přihlašovací údaje od správce systému, nebo využít přihlášení identitou STAG. Uživatel si může na webu prohlédnout své poslední výpůjčky anebo odebrat kartu či změnit některé své údaje.

### Hlavní stránka

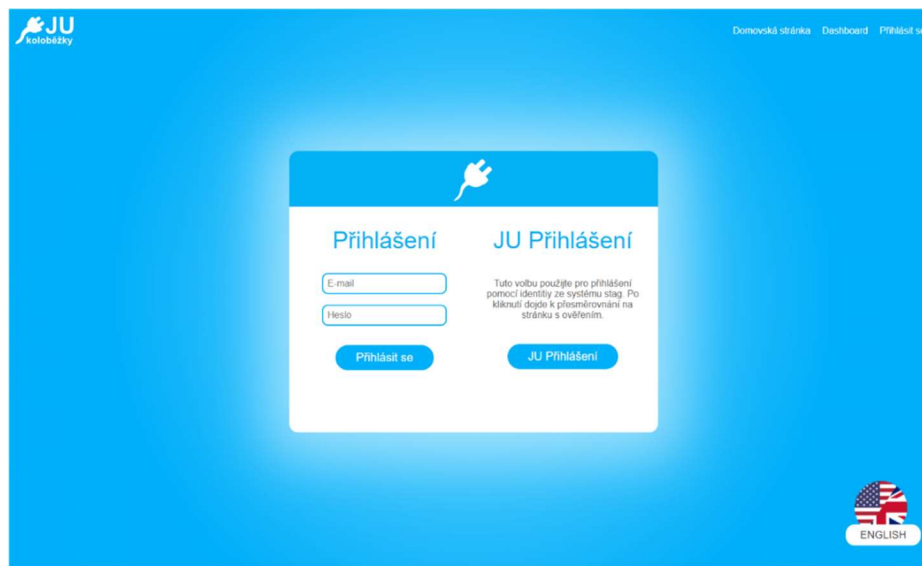
Hlavní stránka webu obsahuje zobrazení stanic s doky a dostupných koloběžek. Na obrázku níže lze vidět vlevo neobsazený dok, tedy dok bez volné koloběžky a vpravo lze vidět, jak vypadá obsazený dok s informací o koloběžce připravené na výpůjčku.





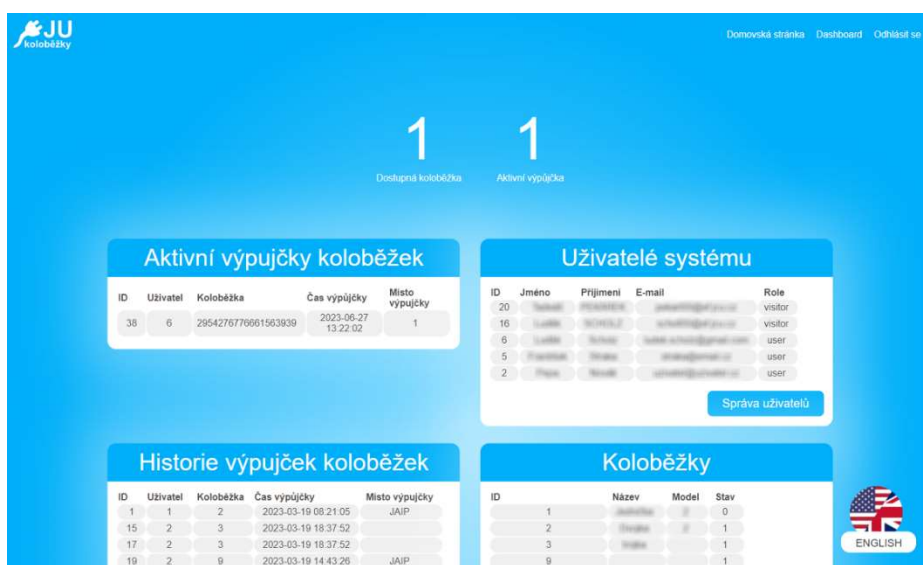
## Stránka pro přihlášení

Tato stránka slouží pro přihlášení uživatele. Uprostřed stránky je samotný přihlašovací box umožňující klasické přihlášení pomocí emailu a hesla pro lokální účet, nebo přihlášení pomocí identity STAG.



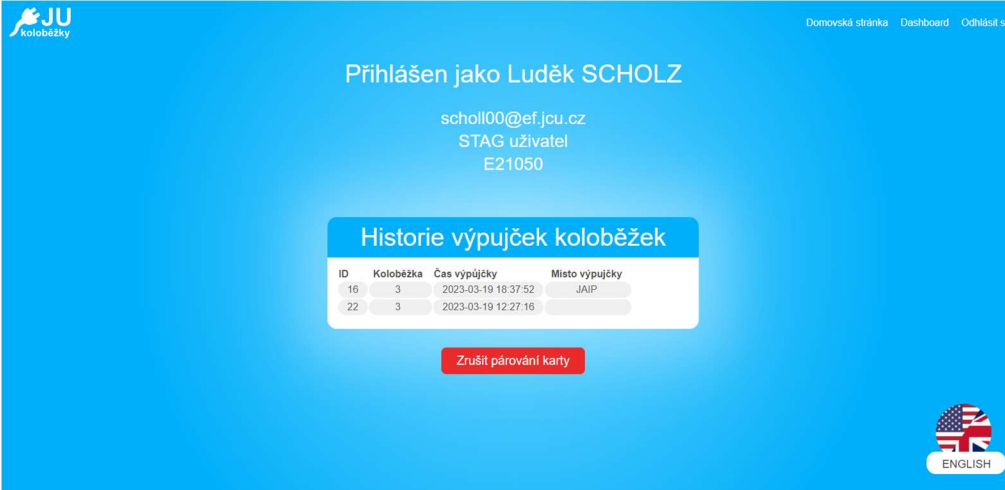
## Administrátorská nástěnka

Tato stránka se zobrazí po přihlášení administrátora. Centrálním prvkem je signalizace počtu aktuálně dostupných koloběžek a aktuálních výpůjček. Následuje plocha pro „widyety“ obsahující nejdůležitější informace o systému jako jsou konkrétní aktivní výpůjčky, uživatelé systému, koloběžky, stanice, doky nebo historie výpůjček.



## Uživatelská nástěnka

Tato stránka se zobrazí po přihlášení uživatele. V případě přihlášení uživatele STAG se zobrazí jeho informace, pět posledních výpůjček a možnost zrušit párování karty. V případě přihlášení lokálního uživatele se zobrazí stejné informace, ale k tomu ještě možnost upravit nastavení účtu, jako je změna hesla nebo dalších parametrů.

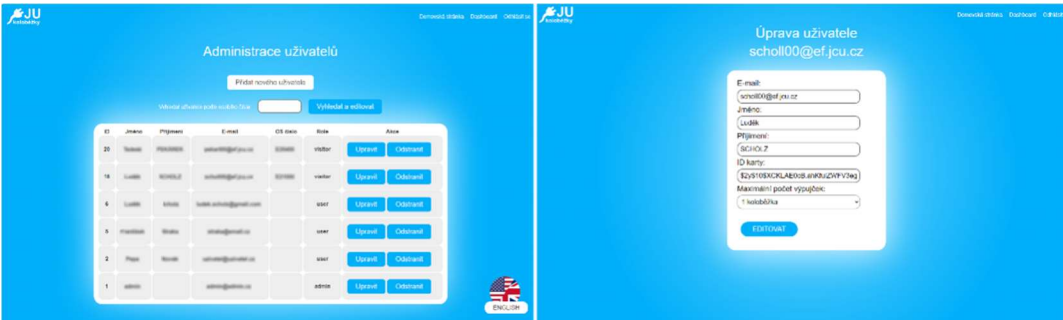


The screenshot shows a user dashboard for Luděk SCHOLZ. At the top, it says "Přihlášen jako Luděk SCHOLZ" and "scholl00@ef.jcu.cz". Below this, it identifies the user as "STAG uživatel" with ID "E21050". A section titled "Historie výpůjček koloběžek" contains a table with two rows of loan data. A red button "Zrušit párování karty" is visible below the table. In the bottom right corner, there is a language selector with a flag and the text "ENGLISH".

ID	Koloběžka	Čas výpůjčky	Místo výpůjčky
16	3	2023-03-19 18:37:52	JAIIP
22	3	2023-03-19 12:27:16	

## Editace uživatelů (Přístupné pouze pro administrátora)

Tato stránka zobrazuje všechny uživatele systému a umožňuje jejich vyhledávání a editaci. Na obrázku níže lze vidět zobrazení všech uživatelů vlevo a na pravé straně lze vidět zobrazení editace konkrétního uživatele.



The screenshot shows two parts of the user management interface. On the left, the "Administrace uživatelů" page displays a table of users with columns for ID, Jméno, Příjmení, E-mail, OS, Role, and Akce. On the right, the "Úprava uživatele" page for user "scholl00@ef.jcu.cz" shows a form with fields for E-mail, Jméno, Příjmení, OS, ID karty, and a dropdown for "Maximální počet výpůjček".

ID	Jméno	Příjmení	E-mail	OS	Role	Akce
20	Tomáš	Procházka	prochazka@ef.jcu.cz	kolobezka	uživatel	Upravit Odstranit
14	Luděk	SCHOLZ	scholl00@ef.jcu.cz	kolobezka	uživatel	Upravit Odstranit
4	Luděk	Štáhl	stahl@ef.jcu.cz	kolobezka	uživatel	Upravit Odstranit
5	Michal	Štáhl	stahl@ef.jcu.cz	kolobezka	uživatel	Upravit Odstranit
2	Michal	Štáhl	stahl@ef.jcu.cz	kolobezka	uživatel	Upravit Odstranit
1	Michal	Štáhl	stahl@ef.jcu.cz	kolobezka	uživatel	Upravit Odstranit

## Editace koloběžek, doků a modelů (Přístupné pouze pro administrátora)

Podobně jako zobrazení a editace uživatelů fungují i další stránky pro editaci koloběžek, doků a modelů. Na obrázcích níže lze vidět editaci koloběžky, doku a modelu.

**Úprava koloběžky**  
2954276776661563939  
MGT #1

Název:

Stav koloběžky:

Model koloběžky:

Limit ADC:

Časovač pro vypnutí:

Název:

Nabíjecí napětí:

Popis:

## 2 Obsluha stanice

Obsluha stanice je intuitivní a jednoduchá. Ve většině času, když přijde potencionální uživatel, bude stanice v režimu spánku se zhasnutým displejem, poté stačí přiložit na modře svítící čtečku svou identifikační kartu a tím se displej probudí. Pokud by nedošlo k probuzení je možné na displej klepnout.



Čtečka karet slouží také jako indikátor stavu stanice. Pokud svítí modře stanice je bez přihlášeného uživatele a je připravena na přiložení nové karty. Pokud svítí čtečka zeleně došlo k přihlášení známého uživatele. Pokud svítí čtečka červeně, je načtena karta, která není v systému spárována s žádným účtem. Po přiložení dojde buďto k načtení karty, v případě přiložení znovu stejné karty dojde k odhlášení, Po přiložení může nastat několik situací.



V prvním případě, pokud má uživatel spárovanou kartu a povolené výpůjčky a v doku se nachází koloběžka tak si ji může vypůjčit kliknutím na vypůjčit u konkrétní koloběžky na displeji. Následně dojde k odemknutí doku a uživatel má cca 40 vteřin na vysunutí koloběžky z doku. Zajištění zámku ve stavu otevřeno při vysouvání koloběžky zajistí aktivaci výpůjčky.

Druhý případ je, pokud má uživatel vypůjčenou koloběžku a ve stanici je volný kompatibilní dok. V tomto případě může uživatel vybrat na displeji koloběžku kterou chce vrátit a vybrat si dok který odemkne. Stanice povolí uživateli odemknout pouze dok, který je s koloběžkou kompatibilní. Po odemknutí může uživatel koloběžku do doku zasunout, dokud neuslyší cvaknutí zámku. Poté dojde k potvrzení vrácení na displeji

stanice. Je důležité vždy zkontrolovat, zda vracíme správný model koloběžky do správného doku podle štítku na displeji koloběžky!

Další situací je, pokud uživatel nemá povolené půjčování koloběžek, v tomto případě se po přihlášení objeví varování s návodem, jak si zažádat o povolení.



Pokud ovšem není karta ani spárována a je tedy pro stanici neznámá, čtečka karet začne svítit červeně a stanice vyžaduje spárování karty pomocí přihlášení za užití identity STAG.

Poslední situace, která může nastat je chyba komunikace se serverem, v tomto případě nelze stanici využít, protože není možné navázat spojení se serverem a ověřovat uživatele.

### **Řešení problémů se stanicí**

**Problém:** Na stanici je zobrazena cizí webová stránka, nebo nejde ovládat.

**Řešení:** Pokud je na stanici něco jiného než rozhraní doku, stačí vyčkat do zhasnutí displeje, cca 30 vteřin, po zhasnutí displeje dojde k obnově a vymazání původního sezení a je načteno kompletně čisté rozhraní

**Problém:** Chci vrátit koloběžku, ale dok je v zavřeném stavu.

**Řešení:** Pokud dok svítí zeleně a je připravený na vrácení koloběžky, ale je zavřený, stačí ručně vytáhnout západku do otevřeného stavu a poté již koloběžka půjde vrátit.

**Problém:** Stanice nereaguje na dotyk ani přiložení karty

**Řešení:** Můžeme zkusit vyčkat do zhasnutí displeje na obnovu, ale pokud ani to nepomůže bude nutné stanici restartovat, to lze tím že přerušíme přívod napájení do stanice.

**Problém:** Koloběžku nelze plně zasunout do doku

**Řešení:** Pokud koloběžka klade větší odpor při zasunování je nutné dbát na správné nasměrování konektorů. Dok je navržen tak, aby konektory koloběžky navedl na sebe, ale při užívání může dojít k poškození tohoto mechanismu, nebo k opotřebení které způsobí větší odpor. Poté je zapotřebí dbát na správné nasměrování konektorů.

**Problém:** Stanice nefunguje/nejde nabíjet koloběžka

**Řešení:** V případě že stanice nefunguje je nutné se obrátit na správce systému. V případě poruchy je možné zapojit koloběžku přímo na originální nabíječku, a tak ji nabíjet.