



**Fakulta elektrotechnická**  
**Katedra radioelektroniky**

**Bakalářská práce**  
**Asistenční pomůcka pro seniory**  
**Matěj Foukal**

**Bakalářský program:**  
**Elektronika a komunikace**

**Vedoucí práce:**  
**Doc. Ing. Stanislav Vitek, PhD.**  
**V Praze, Květen 2023**

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Foukal** Jméno: **Matěj** Osobní číslo: **499199**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra radioelektroniky**  
Studijní program: **Elektronika a komunikace**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Asistenční pomůcka pro seniory**

Název bakalářské práce anglicky:

**Assistive Aid for the Elderly**

Pokyny pro vypracování:

Navrhnete a realizujete pomůcku pro seniory ve formě IoT systému, který v rámci chytré domácnosti zlepší možnosti orientace osob v čase noci. Při vypracování se řiďte následujícími pokyny:

- 1) systém umožňuje detekovat, že osoba opouští lůžko
- 2) po opuštění lůžka je automaticky spuštěna světelná navigace, která osobě usnadní orientaci v prostoru
- 3) spuštění navigačního systému je zaznamenáno v databázi
- 4) k databázi a nastavení systému bude přístup pomocí webového rozhraní se zabezpečením přístupu
- 5) vlastnosti systému otestujete na prototypu

Seznam doporučené literatury:

- [1] COHEN-MCFARLANE, Madison, et al. Bed-Based Health monitoring using pressure sensitive technology: A review. IEEE Instrumentation & Measurement Magazine, 2021, 24.2: 13-23.  
[2] WALSH, Lorcan, et al. Noncontact pressure-based sleep/wake discrimination. IEEE Transactions on Biomedical Engineering, 2016, 64.8: 1750-1760.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**doc. Ing. Stanislav Vítek, Ph.D. katedra radioelektroniky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **31.01.2023** Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

\_\_\_\_\_  
doc. Ing. Stanislav Vítek, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
doc. Ing. Stanislav Vítek, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## **ANOTACE**

Tato bakalářská práce se zabývá návrhem a implementací IoT systému pro pomoc seniorům v samostatném fungování v domácnosti. Cílem práce je navrhnout systém IoT, který bude snadno rozšiřitelný o nová zařízení a postavení prototypu jednoho takového zařízení, které se systémem bude komunikovat.

## **KLÍČOVÁ SLOVA**

internet věcí; chytrá domácnost; asistenční systém; uživatelské rozhraní; stáří; chytrá postel; stárnutí v domácnosti



## **ABSTRACT**

This bachelor thesis focuses on the design and implementation of an IoT system aimed at assisting seniors in maintaining their independence within their households. The objective of this study is to propose an easily expandable IoT system capable of accommodating new devices, while also developing a prototype of one such device that communicates with the system.

## **KEYWORDS**

Internet of Things; smart home; assistance system; old age; smart bed; aging in place



## **PODĚKOVÁNÍ**

Rád bych poděkoval doc. Ing. Stanislavu Vítkovi, Ph.D. za rady a odborný dohled nad touto prací. Děkuji, že jste si vždy, když jsem potřeboval, našel čas podělit se o své zkušenosti. Děkuji také své rodině za poskytnuté zázemí, bez kterého bych nebyl schopen studovat. Mé poděkování také patří mé babičce Boženě Kolářkové za konzultace ohledně tohoto projektu.





## PROHLÁŠENÍ

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

## HOW TO CITE

FOUKAL, Matěj. *Asistenční pomůcka pro seniory*: bakalářská práce. Praha: České Vysoké Učení Technické v Praze, Fakulta elektrotechnická, Department of Radioelectronics, 2023. 31 s. Vedoucí práce byl Doc. Ing. Stanislav Vítek, PhD.



# Obsah

Seznam obrázků	13
Seznam kódů	15
<b>1 Úvod</b>	<b>1</b>
<b>2 Internet věcí, soukromí a bezpečnost</b>	<b>3</b>
2.1 Soukromí . . . . .	3
2.2 Bezpečnost . . . . .	3
<b>3 Stáří a domácí asistenti</b>	<b>5</b>
3.1 Problémy spojené se stárnutím . . . . .	5
3.1.1 Riziko pádů . . . . .	5
3.1.2 Problémy s pamětí . . . . .	5
3.1.3 Problémy s orientací . . . . .	5
3.2 Příklady IoT asistentů . . . . .	6
3.2.1 Detekce pádu . . . . .	6
3.2.2 Asistenti pro podporu paměti . . . . .	6
3.2.3 Automatické osvětlení pro pomoc s orientací . . . . .	6
<b>4 Koncept návrhu a jeho cíle</b>	<b>7</b>
4.1 Konceptuální návrh . . . . .	7
4.2 Hlavní cíle . . . . .	8
4.3 Vedlejší cíle . . . . .	8
<b>5 Komunikační protokoly IoT</b>	<b>9</b>
5.1 ZigBee . . . . .	9
5.2 HTTP a HTTPS . . . . .	9
5.3 MQTT . . . . .	10
<b>6 Vývoj webové aplikace</b>	<b>11</b>
6.1 Frontend . . . . .	11
6.1.1 HTML . . . . .	11
6.1.2 CSS . . . . .	12
6.1.3 JavaScript . . . . .	13
6.1.4 Bootstrap . . . . .	13
6.2 Backend . . . . .	14
6.2.1 Flask . . . . .	14
6.2.2 Flask-login . . . . .	14
6.2.3 SQLite3 . . . . .	15
6.2.4 Eclipse Paho . . . . .	15
<b>7 Technické řešení</b>	<b>17</b>
7.1 Server . . . . .	17

7.1.1	Zigbee2MQTT . . . . .	17
7.1.2	Eclipse Mosquitto . . . . .	17
7.1.3	Webová aplikace . . . . .	18
7.2	Senzorová část . . . . .	18
7.2.1	Výběr senzoru . . . . .	18
7.2.2	Lokální mikrokontrolér . . . . .	20
7.3	Aktuátorová část . . . . .	20
<b>8</b>	<b>Výsledek práce</b>	<b>21</b>
8.1	Uživatelské rozhraní . . . . .	21
8.1.1	Přihlašovací formulář . . . . .	21
8.1.2	Navigační menu . . . . .	22
8.1.3	Vykreslování grafu spánku . . . . .	22
8.1.4	Nákupní seznam . . . . .	23
8.1.5	Ovládání zařízení . . . . .	24
8.2	Reakce systému na události . . . . .	25
8.3	Postup implementace nového zařízení . . . . .	26
<b>9</b>	<b>Závěr</b>	<b>27</b>

## Seznam obrázků

4.1	Obecný diagram projektu . . . . .	7
4.2	Požadavky na systém vyplívající ze zadání . . . . .	8
6.1	Ukázka stránky pouze s HTML . . . . .	12
6.2	CSS syntax . . . . .	12
6.3	Ukázka stránky s HTML a CSS . . . . .	13
7.1	Váhový senzor 50kg [17] . . . . .	18
7.2	Senzor ohybu [18] . . . . .	19
7.3	Senzor tlakové síly [19] . . . . .	19
7.4	Raspberry Pi Pico W [20] . . . . .	20
8.1	Přihlašovací formulář . . . . .	21
8.2	Navigační menu . . . . .	22
8.3	Detaily stránky <i>spánek</i> . . . . .	23
8.4	Nákupní seznam . . . . .	23
8.5	Ovládací panel zařízení . . . . .	24
8.6	Ukázka funkčního prototypu . . . . .	25



## Seznam kódů

6.1	Ukázka HTML . . . . .	12
6.2	Základní aplikace Flasku . . . . .	14
8.1	Ukázka spouštěče a akce . . . . .	26





# 1 Úvod

Stárnutí populace, jak potvrdilo mimo jiné i sčítání lidu roku 2021 [1], je aktuálním problémem a to nejen v České republice. Ruku v ruce s touto problematikou přichází také problém nedostatečné kapacity zařízení, která zajišťují péči o seniory [2]. částečnou náplastí na tento problém jsou systémy chytré domácnosti, které jsou navrženy pro pomoc osobám se sníženou samostatností.

Takový systém může například detekovat, zda uživatel neupadl a případně zavolat pomoc. Může pomoci s přesným dávkováním léků. Notifikovat uživatele, kdyby na léky zapomněl, nebo ho naopak varovat, kdyby se pokusil překročit doporučenou dávku.

Hlavním cílem této bakalářské práce je navrhnout systém, za použití technologií internetu věcí, který by umožňoval jednoduchou implementaci nových zařízení pro pomoc seniorům. A umožňoval uživatelům jednoduchý přístup k informacím ze senzorů a ovládání jednotlivých prvků systému pomocí uživatelského rozhraní.

Druhý hlavní cíl je postavit prototyp jednoho zařízení se senzory, které by bylo součástí zmíněného systému a poskytovalo by mu data ze svých senzorů pro řízení aktuátorů. Konkrétně se jedná o ovládání osvětlení pomocí senzorů tlaku v posteli. To, jak takové zařízení může pomoci je podrobněji rozebráno v kapitole 3.

V kapitole 2 tohoto textu je popsáno, co je to internet věcí a rizika jeho implementace. Kapitola 3 se věnuje problémům spojeným se stářím a tomu, jak lidem, co jimi trpí může internet věcí pomoci. Následující kapitoly obsahují koncept tohoto projektu a popis nástrojů použitých k vývoji. Poslední dvě kapitoly, kapitola 7 a kapitola 8, objasňují funkčnost jednotlivých komponentů tohoto projektu.



## 2 Internet věcí, soukromí a bezpečnost

Internet věcí, časteji pouze jako zkratka anglického názvu Internet of Things – IoT, je termín, který není možné v dnešní době přehlédnout. Tento pojem zavedl už v roce 1999 technologický průkopník Kevin Ashton, aby popsal systém v němž je internet propojený s fyzickým světem všudepřítomnými senzory [3]. To, co vlastně IoT je, ale není úplně zřejmé. Exaktní definice, na které by se shodli všichni neexistuje. Nejčastěji přijmaná definice však popisuje IoT jako síť fyzických zařízení, které jsou vybaveny senzory, jsou připojeny ke společné síti a sbírají data, které si následně mezi sebou vyměňují. Data mohou dále zpracovávat lokálně, či je pro zpracování zaslat do lokálních serverů, nebo cloudových aplikací.

IoT technologie mohou značně zvýšit efektivitu průmyslu, či nám usnadnit život v mnoha dalších ohledech. Je ale extrémně důležité zvážit při implementaci takového systému i rizika, která IoT přináší. Protože cílem této bakalářské práce je právě návrh systému IoT, je na místě zmínit hrozby, které mohou nastat a to převážně u internetu věcí v domácnostech, kterým se práce věnuje.

### 2.1 Soukromí

Právě sběr dat, který je jedním z hlavních prvků IoT technologií, představuje unikátní výzvu v oblasti ochrany soukromí. Sbíraná data mohou obsahovat citlivé informace o uživatelích, jejich chování, zvyklostech nebo dokonce jejich fyzickém zdraví. V případě, že by došlo k úniku dat, mohl by útočník zjistit, kdy uživatel není doma a tuto informaci zneužít.

Při návrhu IoT systému je potřeba počítat i s nejhorší variantou - kdy je systém napaden a útočník má přístup ke všem datům. Při návrhu systému je tedy potřeba zvážit, jaká data budou sbírána a jak budou v síti informace přenášeny. Například využití kamer je v IoT velmi užitečné, je ale potřeba vyhodnotit riziko jejich zneužití.

### 2.2 Bezpečnost

Protože jsou IoT zařízení součástí sítě, může jejich nedostatečné zabezpečení oslabit bezpečnost celé sítě. Příkladem útoku na síť přes IoT zařízení může být pokus vědců z Check Point Software Technologies Ltd., kteří pomocí chyby v ZigBee protokolu nahráli na chytrou žárovku malware, který se po opětovném připojení chytré žárovky do sítě rozšířil do ostatních zařízení v síti. Podrobněji je tento útok popsán na jejich stránkách [4].

V případě zavedení IoT systému je tedy potřeba jednat s určitou ostražitostí. Dobrým řešením je rozdělit lokální síť pomocí subsítě a tak oddělit IoT zařízení od ostatních zařízení.

## 3 Stáří a domácí asistenti

Stáří je neoddelitelnou součástí lidského života. S přibývajícím věkem přibývají i problémy týkající se fyzického i psychického zdraví. Při správném použití internetu věcí jako domácích asistentů je možné dopad upadajícího zdraví na kvalitu života zmírnit. V první části této kapitoly si nejdříve uvedeme nejčastější problémy a výzvy, se kterými se starší lidé nejčastěji potýkají. Druhá část se bude věnovat existujícím návrhům a existujícím řešením těchto problémů.

### 3.1 Problémy spojené se stárnutím

#### 3.1.1 Riziko pádů

Podle Světové zdravotnické organizace (WHO) jsou pády druhou nejčastější příčinou úmrtí v důsledku neúmyslného zranění, nejrizikovější jsou pády právě pro osoby starší 60 let [5]. Tento problém je navíc prohlouben tím, že spousta starších lidí žije o samotě a v případě pádu si nemusí být schopna zavolat o pomoc.

#### 3.1.2 Problémy s pamětí

Velmi běžným problémem mezi seniory je také zhoršení paměti. To může být způsobeno například demencí nebo Alzheimerovou chorobou. Člověk, který trpí symptomy těchto nemocí může zapomínat vzít si své léky, dodržovat pitný režim, nebo také mohou mít problém pamatovat si, zda dokončili běžné domácí úkony.

#### 3.1.3 Problémy s orientací

Dalším problémem, který se může vyskytnout u starších lidí, je ztráta orientace a snížená pohyblivost. To může vést k problémům s navigací v domácnosti, zejména v noci, kdy je viditelnost omezená. Taková situace může zvýšit pravděpodobnost pádu a podobných nehod.

Častým problémem, který na to navazuje, je potřeba noční návštěvy toalety. Dostatečné osvětlení cesty je zásadní pro bezproblémové splnění této aktivity. Ve starší zástavbě je navíc zcela běžné, že vypínač osvětlení místnosti je umístěn u dveří, tj. zpravidla ve zcela jiné části místnosti, než je umístěna postel. Navíc je třeba počítat s tím, že cesta může vést i několika místnostmi. Špatné osvětlení, či problém s orientací a najitím vypínače může představovat

zvýšené riziko úrazu. Špatně zvolená teplotní chromatičnost osvětlení by mohla narušit cyklický rytmus uživatele, který by poté mohl mít problémy s pokračováním v dalším spánku. Nedostatek spánku by se mohl poté negativně projevit na psychickém i fyzickém zdraví.

## 3.2 Příklady IoT asistentů

### 3.2.1 Detekce pádu

Skvělým příkladem domácího asistenta je systém, který by zaznamenal pád uživatele, informaci o pádu přeposlal serveru, server by informaci zpracoval a kontaktoval tísňovou linku, ošetřovatele, či příbuzenstvo. Okamžitou reakcí systému na tuto situaci by byla rizika zmíněná v 3.1.1 snížena.

Detekce může být provedena například zvukovými senzory připevněnými k podlaze [6], nebo senzory, které by uživatel nosil u sebe [7].

### 3.2.2 Asistenti pro podporu paměti

V případě problémů s pamětí zmíněných v 3.1.2 by mohla pomoci i velmi jednoduchá aplikace, která by uživateli připomněla, kdy si má vzít léky a dodržování pitné režimu. Také by mohlo být vytvořeno zařízení pro dávkování léků, které by zamezilo, aby uživatel překročil doporučenou dávku. Správné dávkování by mohlo být nastaveno ve webové aplikaci IoT systému, nebo by mohlo být například načteno na základě e-receptu.

Také by mohl tento IoT asistent používat umělou inteligenci k rozpoznání rutin uživatele a v případě, že by došlo k odchylce, upozornil by uživatele, pro případ, že by něco zapomněl.

Příkladem existujícího IoT řešení pro podporu paměti je třeba tato práce [8], která pomáhá lidem s Alzheimerem rozpoznávat obličeje lidí, které znají.

### 3.2.3 Automatické osvětlení pro pomoc s orientací

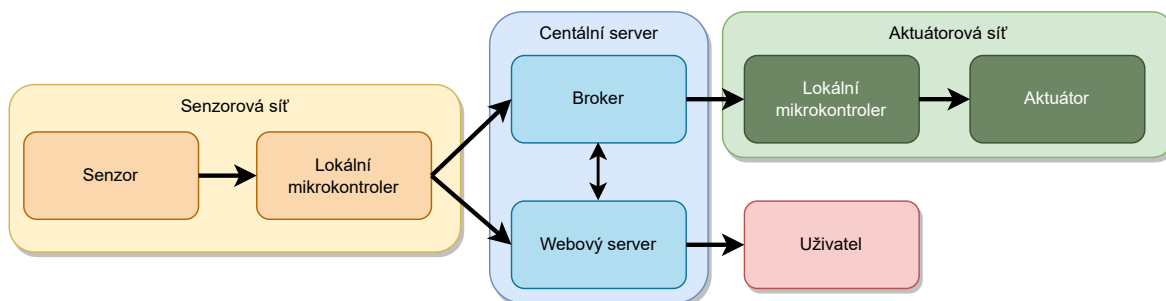
Řešení problému s orientací zmíněném v kapitole 3.1.3 je součástí této bakalářské práce. Tento systém obsahuje tlakové senzory, které detekují, zda uživatel leží v posteli a to komunikují serveru, který podle těchto dat reguluje osvětlení. Server také volí správnou teplotu chromatičnosti podle aktuálního času. Návrh tohoto systému je blíže popsán v následujících kapitolách.

## 4 Koncept návrhu a jeho cíle

Hlavním cílem této bakalářské práce je navrhnout robustní systém IoT, který bude snadno rozšiřitelný o další funkce, jako například zmíněné příklady IoT asistentů v kapitole 3.2. Pro ukázání funkčnosti tohoto systému je také navrhnout jeden domácí asistent, konkrétně Automatické osvětlení pro pomoc s orientací, zmíněné v 3.2.3. Tato kapitola se věnuje konceptuálnímu návrhu celého projektu a také upřesnění cílů této práce, detaily o tom, jak budou tyto cíle dosaženy, budou podrobněji popsány v následujících kapitolách.

### 4.1 Konceptuální návrh

Obecný návrh IoT systému je zobrazen na diagramu 4.1. Tak jak je vyobrazen, plní několik funkcí. Nejdříve mikrokontrolér v sensorové síti sbírá data, která posílá serveru, přes některé z jeho rozhraní. Ten data vyhodnotí a v závislosti na nich pošle příkaz aktuátorové síti. Součástí je také webový server, na kterém si může uživatel zobrazit data nebo ovládat samotné prvky systému. Je možné, aby nějaká zařízení měla lokální mikrokontrolér, který je zároveň v sensorové a aktuátorové síti a také, aby sensorová síť komunikovala přímo s aktuátorovou.



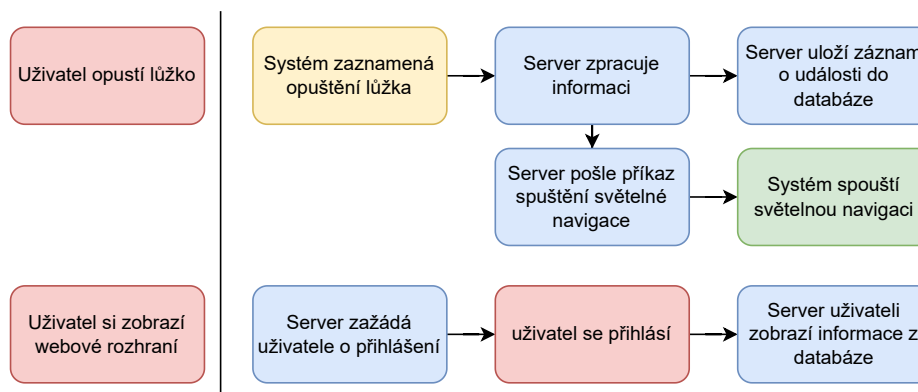
Obr. 4.1: Obecný diagram projektu

## 4.2 Hlavní cíle

Hlavní cíle této práce jsou dány zadáním a jsou následující:

- *Detekce opuštění lůžka:* Systém musí být schopen poznat, kdy si uživatel lehne a kdy své lůžko opustí. Tato funkčnost je klíčová, pro správně spuštění světelné navigace.
- *Automatické spuštění světelné navigace:* Po detekci opuštění lůžka by měl systém automaticky aktivovat světelnou navigaci, která pomůže uživateli se orientovat v prostoru.
- *Databáze událostí:* spuštění světelné navigace musí být zaznamenáno v databázi.
- *Webové rozhraní pro přístup k databázi:* databáze musí být přístupná z webového rozhraní. Webové rozhraní musí mít zabezpečený přístup.

Přehled hlavních cílů je zobrazen v diagramu reakcí systému na události podle zadání 4.2. Barvy korespondují s částmi systému podle diagramu 4.1



Obr. 4.2: Požadavky na systém vyplývající ze zadání

## 4.3 Vedlejší cíle

Vedlejší cíle této práce jsou stanoveny tak, aby výsledný systém co nejlépe vyhovoval potřebám cílové skupiny a prostředí, ve kterém bude implementován.

- *Rozšiřitelnost:* Systém by měl být navržen s ohledem na možnost budoucího rozšíření. Nové funkce a zařízení by mělo být možné přidávat bez nutnosti zásadních změn v základní struktuře systému. To zahrnuje modulární design a využití otevřených standardů.
- *Jednoduchost instalace a údržby:* S ohledem na cílovou skupinu, systém by měl být jednoduchý na instalaci a nevyžadovat složitou, lépe žádnou údržbu.
- *Bezpečnost a soukromí:* Systém by měl být navržen tak, aby byl bezpečný a chránil soukromí uživatelů.
- *Finanční dostupnost:* Projekt by měl být pro cílovou skupinu ekonomicky přijatelný.



## 5 Komunikační protokoly IoT

Komunikace mezi jednotlivými zařízeními je klíčová charakteristika IoT systémů. To, jak mezi sebou zařízení komunikují, definují komunikační protokoly. V této kapitole se podíváme na tři základní protokoly, které jsou v tomto návrhu použity: ZigBee, HTTP a MQTT. Stručně popíšeme, jak fungují a čím se liší od ostatních.

### 5.1 ZigBee

ZigBee je bezdrátový obousměrný komunikační standard. Architekturu ZigBee hierarchie tvoří čtyři vrstvy. Každá vrstva poskytuje sadu služeb pro vrstvu nad sebou. Vlastnosti těchto vrstev jsou podrobně popsány v dokumentaci ZigBee [9]. Spodní dvě vrstvy jsou definované normou IEEE 802.15.4. [10]. Od nejvyšší po nejnižší lze vrstvy pospat takto:

1. *Aplikační vrstva (APL)*: Tato vrstva funguje jako most mezi koncovými aplikacemi a sítí ZigBee. Přijímá data od aplikací, připravuje je pro přenos pomocí ostatních vrstev a naopak, přijímá data přicházející z sítě a poskytuje je aplikacím.
2. *Síťová vrstva (NWK)*: Je zodpovědná za řízení síťových operací, jako je přidávání nových uzlů do sítě, směrování zpráv mezi uzly a udržování bezpečnosti sítě. NWK podporuje topologii hvězdy, stromu i mesh. V hvězdicové topologii ovládá celou síť právě jedno zařízení, ZigBee koordinátor, který se stará o iniciaci a údržbu koncových zařízení v síti. Koncová zařízení mezi sebou komunikují skrze ZigBee koordinátor. V mesh a stromové topologii je ZigBee koordinátor zodpovědný za prvotní nastavení sítě a nastavení jejich klíčových parametrů.
3. *Vrstva řízení přístupu k médiu (MAC)*: Vrstva řízení přístupu k médiu (MAC) koordinuje přístup k rádiovým kanálům, aby se zabránilo kolizím a zajistilo spolehlivé doručení dat. Také se stará o synchronizaci mezi zařízeními a vysílání řídicích rámců.
4. *Fyzická vrstva (PHY)*: Tato vrstva se zabývá přímým odesláním a přijímáním dat pomocí rádiových vln.

### 5.2 HTTP a HTTPS

HTTP (HyperText Transfer Protocol) je *request/response* protokol aplikační vrstvy, který umožňuje flexibilní interakci s HyperTextovými webovými servery.

Komunikace začíná požadavkem klienta na server, tzv. HTTP *request*. HTTP *requesty* mají několik různých metod, které určují, co přesně klient požaduje od serveru. Například

metoda *GET* říká serveru, aby poslal data webové stránky. Metoda *POST* umožňuje vašemu počítači poslat data na server, například když vyplňujete formulář na webové stránce. Metoda *DELETE* naopak říká serveru, aby určitá data odstranil.

Komunikace končí odpovědí serveru. Server odpovídá na požadavek klienta pomocí tzv. HTTP *response*, která se skládá ze stavového kódu, hlaviček a těla zprávy. Stavový kód je tříciferné číslo, které označuje, zda byl požadavek úspěšný a jaký druh chyby nastal, pokud k nějaké došlo. Hlavičky obsahují dodatečné informace, jako je typ obsahu nebo datum. Tělo zprávy obsahuje samotná data - například HTML kód webové stránky.

HTTPS (HyperText Transfer Protocol Secure) je rozšíření HTTP, které přidává vrstvu šifrování pro zvýšení bezpečnosti komunikace. HTTPS je zásadní pro ochranu citlivých informací, které mohou být přenášeny mezi klientem a serverem.

## 5.3 MQTT

MQTT (Message Queuing Telemetry Transport) je komunikační protokol založený na modelu publikování (publish) a odběru (subscribe). V tomto modelu spolu nekomunikují koncová zařízení (v terminologii MQTT - klienti) přímo mezi sebou, ale místo toho komunikaci mezi nimi zprostředkovává centrální prvek, který se nazývá broker. Každý klient může posílat zprávy na konkrétní témata, broker poté tyto zprávy přepośle klientům, které téma odebírají.

## 6 Vývoj webové aplikace

Webová aplikace je klíčovým prvkem jakéhokoli systému založeném na Internetu věcí. Slouží jako uživatelské rozhraní, které umožňuje interakci mezi uživateli a IoT zařízeními. Webové aplikace mohou poskytnout přehled o aktuálním stavu systému, umožňují sledování a analýzu dat z IoT zařízení a mohou také poskytnout rozhraní pro ovládání těchto zařízení.

Tato kapitola se věnuje stručnému popisu nástrojů, které byli při vývoji webové aplikace použity. Nejprve se zaměříme na nástroje a technologie použité pro vývoj frontendu, včetně HTML, CSS, JavaScript a Bootstrap. Poté se podíváme na backend vývoj, kde se zaměříme na použití programovacího jazyka Python a jeho knihoven, jako je Flask pro vývoj webového serveru, Flask-login pro správu uživatelských relací, SQLite3 pro správu databáze a Eclipse Paho pro komunikaci s MQTT sítí.

### 6.1 Frontend

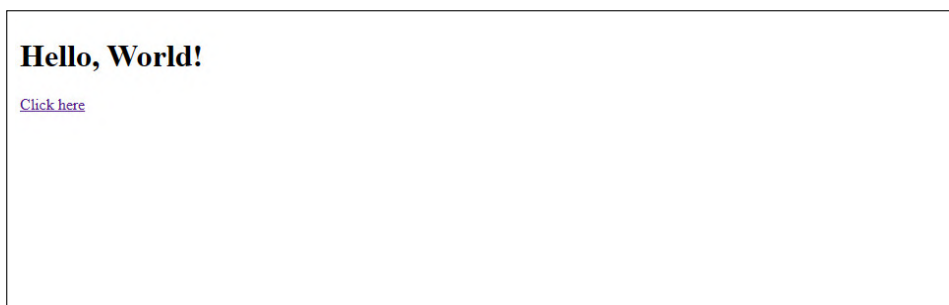
Frontend vývoj je vývoj grafického uživatelského rozhraní. Hlavním cílem dobrého frontendu je intuitivní a přehledné uživatelské rozhraní, které je schopné přizpůsobit se rozlišení, které uživatel používá. V následujících kapitolách jsou popsány nástroje, které se ve vývoji frontendu používají nejčastěji a které jsou zároveň použity v tomto projektu.

#### 6.1.1 HTML

HTML (HyperText Markup Language) je mezi značkovacími jazyky dodnes nejpoužívanější nástroj pro tvorbu webových stránek, podle serveru W3Tech jich HTML používá 94,9 % [11]. Značkovací jazyky jsou systém kódování textu pomocí vložených symbolů do textu za účelem řízení struktury a formátování samotného textu [12]. Takto kódovaný text poté prohlížeč zobrazí do renderovatelného obsahu.

HTML používá párové a nepárové značky (častěji označovány anglickým termínem tags). Ty jsou ve formě textu ohraničeným lomenými závorkami. Tagy mohou mít různé vlastnosti. Mohou upravovat velikost fontu jako například hlavičkové tagy `<h1>` až `<h6>`. Pomocí tagu `<a href="url">` můžeme například tvořit odkazy. Velmi časté jsou také tagy `<div>`, které mohou sloužit ke seskupení více HTML prvků dohromady.

V kombinaci s tagy pro označení typu dokumentu, kořenového prvku, těla a hlavičky vypadá stránka pouze s HTML jako na obrázku 6.1.



Obr. 6.1: Ukázka stránky pouze s HTML

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <div>
7       <h1>Hello , World!</h1>
8       <a href="https://www.cvut.cz">Click here</a>
9     </div>
10  </body>
11 </html>

```

Kód. 6.1: Ukázka HTML

## 6.1.2 CSS

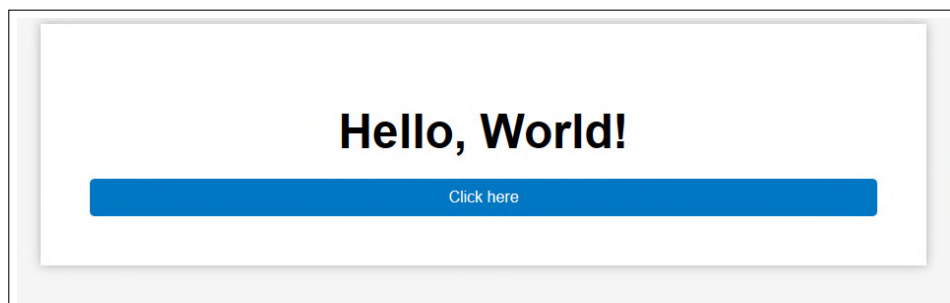
CSS (Cascade style sheets) doplňuje HTML o další způsoby, jak upravit text. Může například změnit barvu nebo font textu, celkové uspořádání stránky, nebo také přizpůsobit stránku uživateli v závislosti na rozlišení jeho zařízení. Syntaxe CSS se skládá ze dvou částí, selektoru a bloku deklarací. Selektor ukazuje na prvek, který chceme stylovat. Může to být obecný název tagu, například `div`. V případě, že bychom chtěli upravit nějaký konkrétní prvek, můžeme mu přidat atribut `class` a ten později použít jako selektor. Blok deklarací může obsahovat jednu nebo více deklarací, ty se skládají z vlastnosti a hodnoty, například vlastnost `barva` s hodnotou `červená`.

CSS syntax	selektor	blok deklarací	
		vlastnost	hodnota
výběr podle prvku	<code>div</code>	<code>{color: red;}</code>	
výběr podle třídy	<code>.class</code>	<code>{color: red;}</code>	

Obr. 6.2: CSS syntax

CSS potom můžeme s HTML propojit několika způsoby. Můžeme styl přidat přímo do tagu, který chceme upravit ve formě atributu `<div style="color:red">`, uvnitř html souboru

pomocí tagu `<style> div color:red </style>` nebo můžeme použít externí soubor CSS a propojit ho s HTML pomocí tagu `<link rel=stylesheet href=„odkaz.css“>`, kde odkaz.css je cesta k CSS souboru. Přidání CSS do HTML z předchozí kapitoly by mohlo vypadat jako na obrázku 6.3.



Obr. 6.3: Ukázka stránky s HTML a CSS

### ■ 6.1.3 JavaScript

JavaScript je objektově orientovaný programovací jazyk. Jedním z klíčových rozdílů mezi JavaScriptem a ostatními kódovacími jazyky je jeho schopnost komunikovat s aplikacemi webových stránek na straně klienta. JavaScript se primárně používá jako skriptovací jazyk na straně klienta, což znamená, že se spouští ve webovém prohlížeči uživatele, nikoli na straně serveru. To umožňuje vytvářet dynamické a interaktivní webové stránky, které mohou reagovat na vstupy uživatele, aniž by se stránka musela znovu načítat. JavaScript může být použit na frontendu pro webové animace nebo vizualizace dat. Může být ale použit i na backendu. V tomto projektu je však použit pouze pro vykreslování interaktivních grafů pomocí knihovny Chart.js.

### ■ 6.1.4 Bootstrap

Bootstrap je open-source toolkit, který využívá sadu předdefinovaných tříd pro HTML prvky. Předdefinované třídy mají už nastavené CSS, případně jsou upraveny JavaScriptem. Bootstrap tak umožní velmi rychlé kódování webových rozhraní. Výhodou je, že všechny třídy mají responzivní chování, tzn. přizpůsobí se rozlišení zařízení uživatele. Všechny třídy Bootstrap je zároveň možné upravovat podle CSS rozšíření Sass. Mimo jiné Bootstrap také poskytuje svůj vlastní systém mřížkového rozmístění, který je velmi flexibilní a dobře se s ním pracuje.

## 6.2 Backend

Backend je část webové aplikace, která není přímo viditelná uživatelům. Zajišťuje funkčnost celé aplikace, stará se o práci s daty a autentizaci uživatelů. V tomto projektu je převážná část backendu napsaná v pythonu a několika jeho knihovnách.

### 6.2.1 Flask

Flask je webový aplikační framework pro programovací jazyk Python. Umožňuje základní funkce jako je například routování URL, obsluhu HTTP požadavků a šablonovací systém. Nejzákladnější aplikace Flasku vypadá následovně:

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route("/")
6 def hello_world():
7     return "<p>Hello, World!</p>"
```

**Kód. 6.2:** Základní aplikace Flasku

Nejdříve importujeme třídu Flask, vytvoříme její instanci app a potom pomocí dekorátoru route() určíme, jaké URL spustí naši funkci. V uvedeném příkladu server při dotazu na úvodní stranu vrátí HTML dokument se zprávou “Hello World”. Tento příklad je dostupný v oficiální dokumentaci Flasku [13].

### 6.2.2 Flask-login

Díky popularitě flasku a open source komunitě má Flask k dispozici spoustu modulů. Jedním z nich je právě flask-login. Toto rozšíření nám poskytuje možnost spravovat uživatelské relace. Umožní vytvořit systém pro přihlášení, odhlášení uživatele a omezit zobrazování obsahu, který není určen pro nepřihlášené uživatele. Flask-login také umí udržovat session token, díky němu nemusí uživatel v rámci své relace zadávat své údaje při každém otevření stránky.

### ■ 6.2.3 SQLite3

SQLite3 je relační databázový systém. Narozdíl od jiných databázových systémů, jako je například MySQL nebo PostgreSQL, se liší tím, že funguje pouze v rámci jednoho souboru a nepotřebuje běžet na samostatném serveru.

SQLite3 má také výbornou podporu pro Python. Příklady integrace této databáze s pythonem jsou dobře popsány v tomto tutoriálu [14].

### ■ 6.2.4 Eclipse Paho

Paho je sada klientských knihoven pro MQTT. Díky Paho se můžeme se zařízením, na kterém kód běží, připojit do MQTT sítě jako klient, odebírat vybraná témata a také do nich sdílet data. Paho podporuje Python i Micropython.





## 7 Technické řešení

Tato kapitola se věnuje detailnímu popisu technických aspektů a komponentů, které tvoří základ celého projektu. Obsahuje podrobný popis serverové části a také detailní popis senzorové a aktuátorové části. Dále je zde popsána implementace a využití různých technologií, jako je Zigbee2MQTT, Eclipse Mosquitto a webová aplikace pro uživatelské rozhraní. V kapitole je také podrobně rozebrána volba a použití konkrétních hardwarových komponent, jako jsou různé druhy senzorů a výběr konkrétního mikrokontroléru.

### 7.1 Server

Server je jádrem celého projektu, obsluhující řadu služeb, které budou podrobněji popsány v následujících kapitolách. Prototyp serveru byl vytvořen a testován na počítači s operačním systémem Windows 11, ale veškerý použitý software je kompatibilní i s operačním systémem Linux a není hardwarově náročný. V praxi by mohl být jako server použit například jednodeskový počítač Raspberry Pi, což by snížilo jak náklady, tak spotřebu elektřiny.

#### 7.1.1 Zigbee2MQTT

ZigBee2MQTT je aplikace fungující jako brána mezi ZigBee a MQTT sítěmi. Na straně ZigBee se chová jako koordinátor v hvězdicové topologii (viz. kapitola ZigBee). Přijímá zprávy od MQTT Brokeru a předává je příslušným zařízením v ZigBee síti, které odebírají topic zprávy. Pro použití ZigBee2MQTT je nutné mít ZigBee adaptér s ZigBee2MQTT firmwarem. V tomto projektu byl použit adaptér SONOFF ZIGBEE 3.0 USB DONGLE PLUS. Kompletní seznam podporovaných adaptérů včetně návodu na instalaci firmwaru je uveden v dokumentaci [15].

#### 7.1.2 Eclipse Mosquitto

Eclipse Mosquitto je open-source MQTT broker, který nabízí robustní a efektivní řešení pro řízení komunikace mezi jednotlivými zařízeními v síti. Mosquitto je navržen tak, aby měl co nejnižší nároky na hardware, což umožňuje jeho nasazení na různých platformách, včetně jednodeskových počítačů, jako je Raspberry Pi [16].

### ■ 7.1.3 Webová aplikace

Server hostuje také webovou aplikaci, která poskytuje uživatelské rozhraní. Uživatelé umožňuje prohlížet si data z databáze ve formě grafu. Dále umožňuje zobrazení a ovládání stavu jednotlivých zařízení. Webová aplikace je postavena na technologiích, které jsou popsány v kapitole 6. Nejdůležitější z nich je backendový framework Flask. Webová aplikace je zároveň připojena jako klient sítě MQTT, pomocí klientské knihovny Paho, a může tedy komunikovat po MQTT protokolu s ostatními zařízeními.

## 7.2 Senzorová část

### ■ 7.2.1 Výběr senzoru

Senzor v posteli má za úkol detekovat, zda někdo v posteli leží. V průběhu vývoje byly využity tři různé druhy senzorů, ale jen jeden z nich se prokázal jako vhodný. Jako první byly použity váhové senzory s tenzometrickým můstkem.



Obr. 7.1: Váhový senzor 50kg [17]

Senzory byly sice levné, ale velmi nízká kvalita součástek tohoto typu byla až příliš kritická. U některých z nich byl signál nerozeznatelný od šumu. Hlavní důvod proč ale tento senzor nebyl použit byla potřeba speciálního kotvení. To sice samo o sobě nebylo problematické navrhnout a vytvořit, ale komplikace se nacházela v tom, že taková konstrukce by musela být uzpůsobená na míru podle rámu postele. Takové řešení by tedy nebylo univerzální a vyžadovalo by složitější instalaci.

Místo toho se přešlo na návrh v podobě senzorové podložky do postele. Instalace takové senzorové podložky spočívá pouze v položení podložky na postel, toto řešení je kompatibilní s každou postelí. Vzhledem k tomu, že bude senzor přímo mezi ležícím uživatelem a matrací, hrozí při použití nevhodných materiálů snížení pohodlí. V původních plánech se počítalo s tím, že bude senzorová podložka uložena v kapse, která bude ušita na spodní straně chrániče

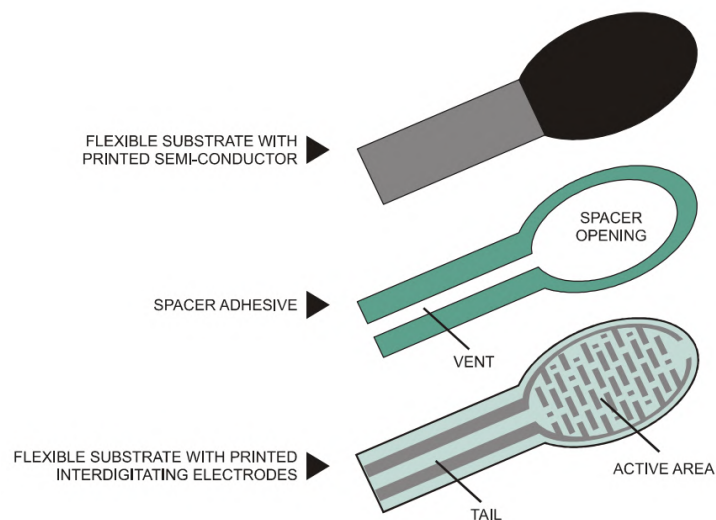
matrace. Chránič matrace je další vrstva, která se přikládá na matraci, aby se prodloužila její životnost. Chránič matrace je sám o sobě polstrovaný a může tak zvýšit komfort napříč sensorům pod ním. Během testování se ale došlo k závěru, že podložka i bez použití chrániče na matraci nijak neubírá na pohodlí. Systém byl testovaný i s chráničem matrace, ten může být použit ke snížení opotřebení sensorové podložky.

Jako první byl v této konfiguraci použit senzor ohybu od společnosti spectrasymbol. Tento senzor byl vybrán, protože zbyl z některého z předchozích projektů. Ohyb, který byl potřeba pro detekci změny je ale příliš velký na to, aby tento senzor mohl být pro tuto práci použit. V případě, že by uživatel používal tvrdší matraci, nemusel by senzor zaznamenat nic.



Obr. 7.2: Senzor ohybu [18]

Jako poslední senzor byl vybrán odporový senzor tlakové síly FSR-402. Senzor se skládá ze dvou vodičů, které jsou odděleny nevodivou stlačitelnou vrstvou. Při působení síly na senzor se vodiče přiblíží a sníží se tedy odpor celého senzoru.

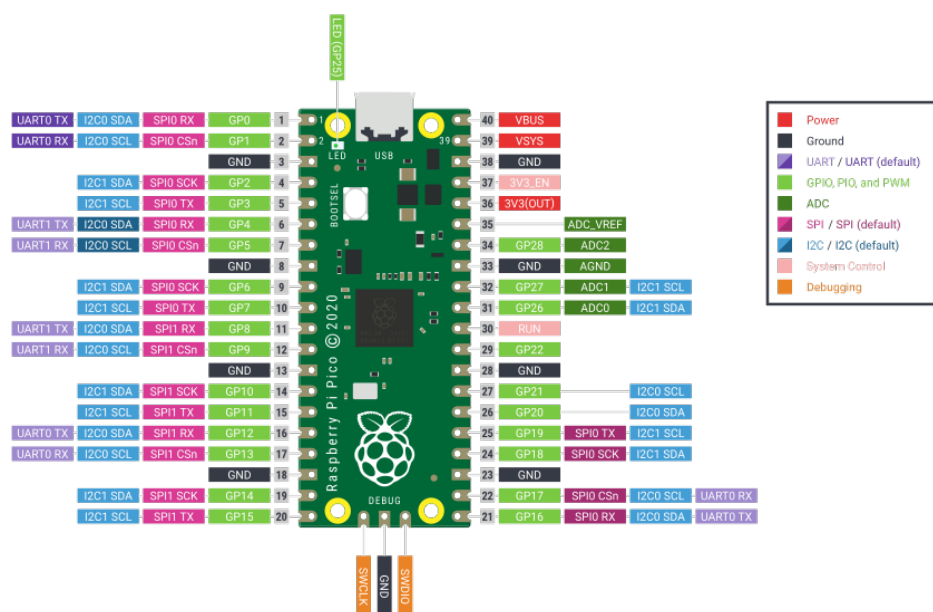


Obr. 7.3: Senzor tlakové síly [19]

Tuto změnu můžeme detekovat pomocí zabudovaného AD převodníku mikrokontroléru Pico W. Při testování tohoto senzoru byl oproti předchozím senzorům poměr signálu ku šumu mnohem větší.

## 7.2.2 Lokální mikrokontrolér

Pro účely zpracování dat z senzorů a následné komunikace s centrálním serverem byl vybrán mikrokontrolér Pico W. Tento kompaktní a cenově dostupný mikrokontrolér poskytuje dostatečný výpočetní výkon pro potřeby tohoto projektu a zároveň podporu WiFi konektivity.



Obr. 7.4: Raspberry Pi Pico W [20]

Komunikace se serverem probíhá pomocí HTTPS popsaného v sekci 5.2. Pro komunikaci je možné používat i MQTT, který je efektivnější. Výhodou HTTPS je, že mikrokontrolér nemusí být ve stejné lokální síti jako server a nemůže tedy být zneužit jako vstupní bod pro útok.

## 7.3 Aktuátorová část

Jako osvětlení, která v tomto projektu používáme jsme vybrali LED TRÅDFRI od IKEA. Jsou cenově dostupná a zároveň je pro ně podpora zigbee2mqtt. V případě, že by pro zařízení, které bychom chtěli použít podpora nebyla, je možné ji jednoduše přidat podle [21].

Pro komunikaci s osvětlením TRÅDFRI v našem systému je použit software zigbee2mqtt. Tento software umožňuje zařízením, která používají protokol ZigBee, komunikovat s MQTT brokerem. Díky tomu můžeme ovládat světla prostřednictvím MQTT zpráv.

## 8 Výsledek práce

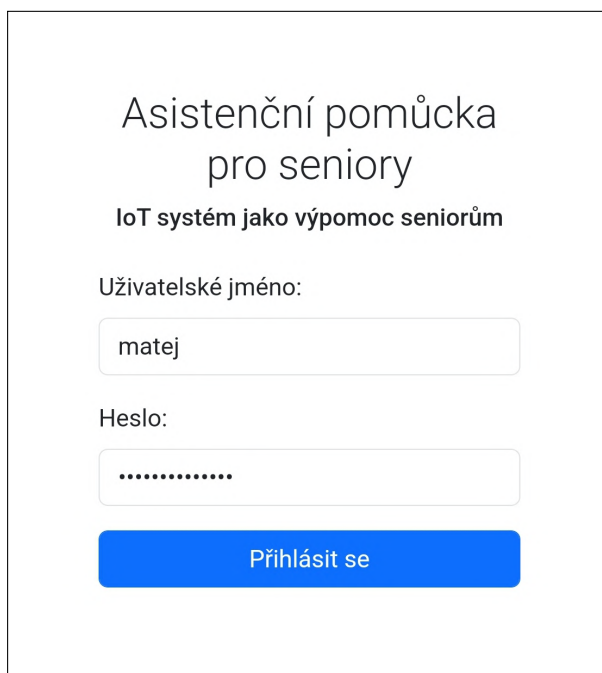
Tato kapitola slouží k ukázce výstupů práce. První část se věnuje uživatelskému rozhraní, součástí jsou snímky obrazovky a popis funkčnosti. Druhá část popisuje, jak systém reaguje na jednotlivé události. Třetí část má za úkol ukázat postup implementace nového zařízení do systému.

### 8.1 Uživatelské rozhraní

Uživatelské rozhraní bylo navrženo tak, aby bylo co nejjednodušší, snadno ovládatelné a plně se přizpůsobilo rozlišení obrazovky uživatele.

#### 8.1.1 Přihlašovací formulář

První část, kterou uživatel po otevření webové stránky uvidí, je právě přihlašovací formulář. Je to také stránka, na které uživatel skončí, pokud se dotáže na jakoukoliv jinou část webu, bez toho aby byl přihlášen. Tato funkčnost je implementována pomocí knihovny *flask-login*.



Asistenční pomůcka  
pro seniory

IoT systém jako výpomoc seniorům

Uživatelské jméno:

Heslo:

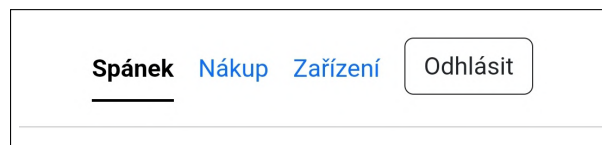
Přihlásit se

Obr. 8.1: Přihlašovací formulář

V případě, že uživatel zmáčkne tlačítko *Přihlásit se* spustí se skript, který porovná vložené heslo se zahashovaným heslem v databázi uživatelů. Hashování a porovnávání hesel je implementováno knihovnou *werkzeug.security*. Pokud jsou hesla shodná, je uživatel přihlášen.

### ■ 8.1.2 Navigační menu

Po přihlášení se uživatel dostane na úvodní stranu. V současné době je to strana *spánek*, ale v případě rozšíření systému by to mohlo být krátké shrnutí všech informací z každé strany, nebo stručný návod ovládání aplikace. Co se zobrazí jako první je navigační menu této webové aplikace. To je vygenerováno z python slovníku, kde klíč slovníku je adresa stránky a hodnota slovníku je text, který se zobrazí. Pro přidání stránky tedy stačí pouze přidat záznam do tohoto slovníku a vytvořit příslušný soubor HTML. To, jak v pythonu fungují slovníky, je možné si přečíst v oficiální python dokumentaci [22]. Navigační menu obsahuje odkazy na jednotlivé stránky a tlačítko pro odhlášení. Aktuální stránka je zvýrazněna změnou barvy fontu a podtržením.



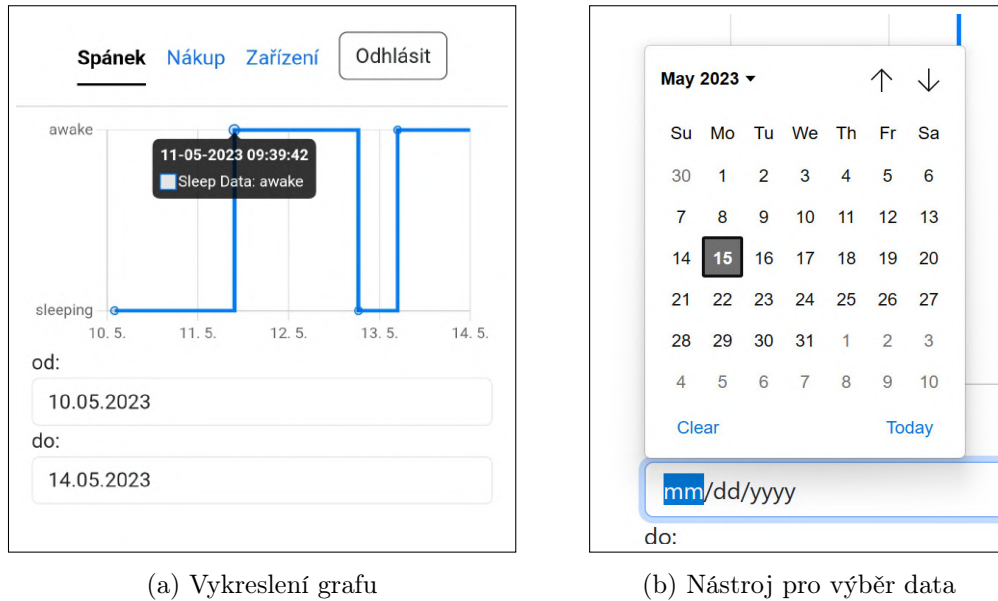
Obr. 8.2: Navigační menu

Za předpokladu, že by bylo přidáno více nových stránek, byla by potřeba změnit uspořádání odkazů. Menu by se mohlo upravit do tzv. *hamburger menu*. *Hamburger menu* má všechny odkazy skryté, většinou obsahuje pouze název stránky a piktogram tří horizontálních čar, které značí, že se jedná o rozklikávací menu. Po kliknutí se objeví všechny odkazy pod sebou. Takové menu má ale nevýhodu, že člověk který se na internetu běžně nepohybuje, rozklikávací menu nerozpozná. Animace takového menu, nebo to, jak menu funguje, může starší lidi poplést. Proto jsou v současném designu použity jen jednoduché statické odkazy.

### ■ 8.1.3 Vykreslování grafu spánku

Při načtení stránky *spánek* se načtou data z databáze, která na server zaslal mikrokontrolér se senzorem, a zobrazí se uživateli interaktivní graf vygenerovaný pomocí Javascriptové knihovny *Chart.js*. Uživatel může kliknout na jednotlivé body grafu, aby si zobrazil přesný čas události.

Součástí této stránky jsou také dva uživatelské vstupy, které určují limity časové osy grafu. Při kliknutí na vstup se uživateli otevře nástroj pro výběr počátečního, či koncového data. Po výběru data se pomocí Javascriptu aktualizují hodnoty grafu a graf se znovu načte. Protože se jedná o Javascript, není potřeba znovu načítat celou stránku.

Obr. 8.3: Detaily stránky *spánek*

#### 8.1.4 Nákupní seznam

Sdílený nákupní seznam se může například hodit v modelové situaci, kdy vnuk jede za prarodiči. Po cestě se podívá do nákupního seznamu a nakoupí, co je potřeba. Při nákupu může nakoupené položky odstranit. Když při návštěvě zjistí, že prarodičům něco chybí, můžou to ještě do seznamu přidat. U přidané položky se automaticky zobrazí *username* uživatele. Nákupní seznam může potom vypadat podobně jako na obrázku 8.4.

ID	položka	počet	zadavatel
x	<input type="text"/>	<input type="text"/>	<input type="button" value="přidat"/>
8	chleba	1	matej ✕
9	máslo	1	matej ✕
10	mléko	2	matej ✕
11	tablety do záchoda	1	matej ✕
12	pivo	6	babička ✕

Obr. 8.4: Nákupní seznam

### 8.1.5 Ovládání zařízení

Poslední strana pod názvem *Zařízení* slouží jako ovládací panel zařízení. Zařízení jsou organizované do tzv. karet. Jako součást prototypu byla navržena pouze karta pro ovládání osvětlení. Ovládací prvky jsou zabaleny do HTML tagu form s metodou post. Po stisknutí tlačítka *Přepnout* se odešle celý formulář včetně hodnot jasu a teplotní chromatičnosti. Backend přijme hodnoty a publikuje pomocí Paho knihovny MQTT zprávu brokeru se zmíněnými hodnotami. Backend předá zprávu MQTT Brokeru, ten zprávu dále pošle Zigbee2MQTT, které ovládá ZigBee prvky. Stav osvětlení se změní podle vložených hodnot.



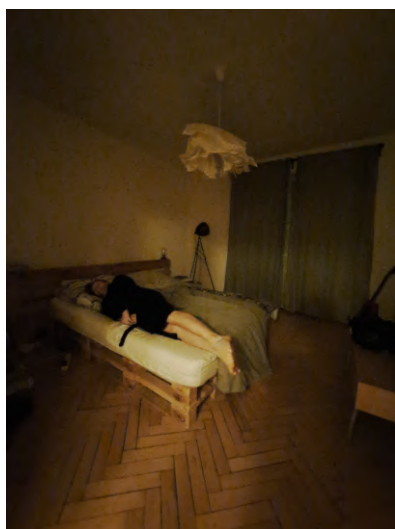
Obr. 8.5: Ovládací panel zařízení



## 8.2 Reakce systému na události

Uživatelské rozhraní není jediný způsob interakce s IoT systémem. Ten reaguje také na změny stavu senzorů a zařízení. Tato kapitola stručně popíše reakce systému na některé události, které mohou nastat.

*Uživatel ulehá do postele.* Mikrokontrolér zaznamená změnu stavu senzoru a předává informaci serveru pomocí HTTPS protokolu. Server přijme informaci. Ukládá záznam do databáze společně s časovou značkou. Posílá pomocí Paho knihovny MQTT zprávu MQTT Brokeru Mosquitto. Broker předává zprávu Zigbee2MQTT, které následně postupně zhasíná osvětlení se správnými parametry.



(a) uživatel je v posteli  
navigační systém vypnut



(b) uživatel vstal z postele  
navigační systém zapnut

**Obr. 8.6:** Ukázka funkčního prototypu

*Uživatel vstává z postele.* Tato interakce probíhá stejně jako předchozí, pouze se pomalu rozsvítí osvětlení. Zvolená teplotní chromatičnost napodobuje slunce, kdy v pravé poledne je barva světla nejstudenější a o půlnoci nejteplejší.

*Uživatel zapíná osvětlení pomocí vypínače.* Pokud zapne uživatel světlo pomocí vypínače, nastaví se do posledního stavu, ve kterém bylo. To může být problém například v případě, kdy uživatel vypnul osvětlení pomocí ovládacího panelu - osvětlení by se po stisknutí vypínače sice připojilo do sítě, ale jas by byl stále nastaven na nulu, osvětlení by nesvítilo. Řešení tohoto problému využívá toho, že po připojení osvětlení do sítě Zigbee2MQTT posílá zprávu do MQTT sítě o připojení nového zařízení. Server může tuto zprávu zachytit a osvětlení zapnout ve správné tepelné chromatičnosti.

## 8.3 Postup implementace nového zařízení

Projekt byl navržen tak, aby implementace byla co nejjednodušší. Dá se udělat v pár krocích. Pokud nové zařízení potřebuje svojí html stránku, je třeba přidat do slovníku stránek adresu a název stránky, viz. kapitola 8.1.2, a vytvořit soubor, který bude mít stejné jméno jako adresa stránky s příponou html. Například stránka "budicek": "Budíček" by měla soubor *budicek.html*. V souboru je třeba definovat pouze obsah stránky, navigační menu včetně nového odkazu se přidá samo.

Pokud chceme přidat funkci serveru. Je potřeba mít nějaký *trigger*, neboli spouštěč akce. Může to být například HTTP request, nebo MQTT zpráva. Například v následujícím zjednodušeném kódu je *trigger* dekorátor `@app.route()` s metodou GET.

```
1 @app.route('/nakup/', methods=['GET'])
2 def nakup():
3     shopping_list = load_data_shopping_list(database_name)
4     return render_template('index.html', nav_list=nav_list, page='nakup',
        shopping_list=shopping_list)
```

**Kód. 8.1:** Ukázka spouštěče a akce

Samotná akce je poté definována ve funkci pod dekorátorem. Tato funkce nejdříve načte databázi do proměnné *shoppinglist* a následně vrátí HTML stránku. Pomocí argumentů funkce *render\_template* předá backend proměnné frontendu. První proměnná *navlist* určuje odkazy navigačního menu, argument *page* určuje aktuální stránku a *shoppinglist* je nákupní seznam načtený z databáze v předchozím kroku.

## 9 Závěr

Tato bakalářská práce se věnovala návrhu IoT systému. Nejdříve diskutovala, jak může internet věcí v oblasti zdraví seniorů pomoci, zmínila rizika implementace IoT systému a popsala technologie, které byly při vývoji použity.

Praktická část této práce splnila cíle stanovené v úvodu. Povedlo se vytvořit snadno rozšiřitelný IoT systém, včetně implementace zařízení pro řízení osvětlení. Pro tento systém bylo také vytvořeno uživatelské rozhraní, ve kterém může uživatel zobrazit senzory získaná data, monitorovat stav zařízení či zařízení ovládat.

Výstup této bakalářské práce by mohl být v budoucnosti použit jako robustní základ pro přidání nových zařízení. Například použité senzory tlakové síly by mohly být vyměněny za senzor z optických kabelů popsany v této práci [23]. Konstrukce tohoto senzoru nám umožňuje nejen sledovat, zda uživatel v posteli leží, ale i měřit jeho tepovou frekvenci. Systém je také, mimo přidávání nových zařízení, možné rozšířit i přidáním nových způsobů, jak zpracovávat data.



## Seznam použité literatury

1. VACHUŠKA, Jakub. Česká populace zestárá: Sčítání lidu, domů a bytů 2021 potvrdilo pokračující demografické stárnutí populace Česka. [B.r.]. Dostupné také z: <https://www.statistikaamy.cz/2022/07/12/ceska-populace-zestrala>.
2. JUNA, Petr. Stárnoucímu Česku nestačí kapacity domovů pro seniory. Čekací doba je až rok. *Seznam Zprávy*. 2022. Dostupné také z: <https://www.seznamzpravy.cz/clanek/fakta-starnoucimu-cesku-nestaci-kapacity-domovu-pro-seniory-cekaci-doba-je-az-rok-188756>.
3. *Interview with Kevin Ashton – inventor of IoT: Is driven by the users*. [B.r.]. Dostupné také z: <https://www.avnet.com/wps/portal/silica/resources/article/interview-with-iot-inventor-kevin-ashton-iot-is-driven-by-the-users/>.
4. *The Dark Side of Smart Lighting: Check Point Research Shows How Business and Home Networks Can Be Hacked from a Lightbulb*. [B.r.]. Dostupné také z: <https://blog.checkpoint.com/security/the-dark-side-of-smart-lighting-check-point-research-shows-how-business-and-home-networks-can-be-hacked-from-a-lightbulb/>.
5. *Falls*. [B.r.]. Dostupné také z: <https://www.who.int/news-room/fact-sheets/detail/falls>.
6. ZIGEL, Yaniv; LITVAK, Dima; GANNOT, Israel. A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound-Proof of Concept on Human Mimicking Doll Falls. *IEEE transactions on biomedical engineering*. 2009, roč. 56, č. 12, s. 2858–2867. ISBN 0018-9294.
7. SCHWICKERT, L.; BECKER, Clemens; LINDEMANN, U.; CONSORTIUM, FARSEEING; FARSEEING META DATABASE CONSENSUS GROUP, the; FARSEEING CONSORTIUM, for the; FARSEEING META DATABASE CONSENSUS GROUP, the. Fall detection with body-worn sensors: a systematic review. *Zeitschrift für Gerontologie und Geriatrie*. 2013, roč. 46, č. 8, s. 706–719. ISBN 0948-6704.
8. ROOPAEL, Mehdi; RAD, Paul; PREVOST, John J.; IEEE. A Wearable IoT with Complex Artificial Perception Embedding for Alzheimer Patients. In: NEW YORK: TSI Press, 2018, sv. 2018-, s. 1–6. ISBN 2154-4824.
9. ALLIANCE, zigbee. ZigBee Specification: Revision 22.1.0. In: [b.r.], s. 22–428. Dostupné také z: <https://csa-iot.org/wp-content/uploads/2022/01/docs-05-3474-22-0csg-zigbee-specification-1.pdf>.
10. *IEEE 802.15.4-2020: IEEE Standard for Low-Rate Wireless Networks*. 23.7.2020. vyd. IEEE SA, 2020.
11. *Usage statistics of markup languages for websites*. Q-success: W3Techs, 2023. Dostupné také z: [https://w3techs.com/technologies/overview/markup\\_language](https://w3techs.com/technologies/overview/markup_language).
12. *Markup language*. Encyclopaedia Britannica, inc, [b.r.]. Dostupné také z: <https://www.britannica.com/technology/markup-language>.
13. *Flask documentation: minimal application*. [B.r.]. Dostupné také z: [flask.palletsprojects.com/en/2.3.x/quickstart/#a-minimal-application](https://flask.palletsprojects.com/en/2.3.x/quickstart/#a-minimal-application).

14. *SQLite LEARN SQLITE: SQLite Python* [online]. [cit. 2023-05-11]. Dostupné z: [https://www.tutorialspoint.com/sqlite/sqlite\\_python.htm](https://www.tutorialspoint.com/sqlite/sqlite_python.htm).
15. *ZigBee2MQTT: Supported adapters*. [B.r.]. Dostupné také z: <https://www.zigbee2mqtt.io/guide/adapters/%5C#experimental>.
16. *Eclipse Mosquitto™: An open source MQTT broker*. [B.r.]. Dostupné také z: <https://mosquitto.org/>.
17. *Váhový senzor - 50 kg: Vnitřní 1000 Ohm poloviční tenzometrický můstek s rozsahem až 50 kg*. [B.r.]. Dostupné také z: <https://dratek.cz/photos/produkty/f/2/2202.jpg?m=1516085370>.
18. *Flex Sensor by Spectra Symbol*. [B.r.]. Dostupné také z: [https://store.spectrasymbol.com/products/flex-sensor?gclid=CjwKCAjwjYKjBhB5EiwAiFdSflCh5y1BeY14Hp-xUwrqka7m65noWuMArloMktr4c0Miv\\_NEbQwWIXoC-j4QAvD\\_BwE](https://store.spectrasymbol.com/products/flex-sensor?gclid=CjwKCAjwjYKjBhB5EiwAiFdSflCh5y1BeY14Hp-xUwrqka7m65noWuMArloMktr4c0Miv_NEbQwWIXoC-j4QAvD_BwE).
19. *Force Sensing Resistors: An Overview of the Technology*. INTERLINK ELECTRONICS, [b.r.]. Dostupné také z: [https://makeabilitylab.github.io/physcomp/assets/datasheets/ForceSensitiveResistor\\_Interlink\\_IntegrationGuide.pdf](https://makeabilitylab.github.io/physcomp/assets/datasheets/ForceSensitiveResistor_Interlink_IntegrationGuide.pdf).
20. *Raspberry Pi Pico W: Pinout and design files*. Raspberry Pi, [b.r.]. Dostupné také z: <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html>.
21. *Zigbee2MQTT: Support new devices*. [B.r.]. Dostupné také z: [https://www.zigbee2mqtt.io/advanced/support-new-devices/01\\_support\\_new\\_devices.html](https://www.zigbee2mqtt.io/advanced/support-new-devices/01_support_new_devices.html).
22. *Python: Data Structures*. [B.r.]. Dostupné také z: <https://docs.python.org/3/tutorial/datastructures.html>.
23. SARTIANO, Demetrio; SALES, Salvador. Low Cost Plastic Optical Fiber Pressure Sensor Embedded in Mattress for Vital Signal Monitoring. *Sensors (Basel, Switzerland)*. 2017, roč. 17, č. 12, s. 2900. ISBN 1424-8220.

## Příloha

Součástí této práce jsou soubory s kódy webové aplikace a mikrokontroléru. Hierarchie složky a obsah složky je následující:

```
/
├── assistiveaid/ .....root složka webové aplikace
│   ├── app.py .....soubor obsahující Flask framework
│   ├── bed.py ..... funkce pro manipulaci databáze s daty spánku
│   ├── db_functions.py ..... funkce pro manipulaci databáze s daty uživatelů
│   ├── forms.py ..... třídy formulářů
│   ├── lights.py .....ovládání osvětlení
│   ├── login.py .....funkce a třídy související s přihlášením
│   ├── README.md ..... konfigurace a spuštění projektu
│   └── templates/ ..... složka obsahující html stránky
│       ├── resources/ ..... složka pro grafické soubory
│       ├── index.html ..... html soubor s hlavičkou
│       ├── login.html .....html soubor s přihlašovacím formulářem
│       ├── nakup.html .....html soubor pro stránku s nákupem
│       ├── spanek.html .....html soubor pro graf spánku
│       └── zarizeni.html .....html soubor pro karty zařízení
└── picow/ .....root složka mikrokontroléru
    └── main.py ..... kód mikrokontroléru
```