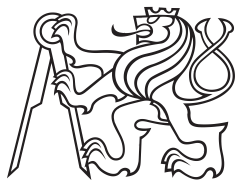


Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra telekomunikační techniky

System pro automatické polohování nemocničního lůžka

Bc. Michal Pícek

Vedoucí: Ing. Bc. Marek Neruda, Ph.D.
Obor: Elektronika a komunikace
Studijní program: Komunikační sítě a internet
Květen 2020

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Picek** Jméno: **Michal** Osobní číslo: **457079**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra telekomunikační techniky**
Studijní program: **Elektronika a komunikace**
Specializace: **Komunikační sítě a internet**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Systém pro automatické polohování nemocničního lůžka

Název diplomové práce anglicky:

System for Automatic Positioning of Hospital Bed

Pokyny pro vypracování:

Navrhněte a zrealizujte systém pro automatické polohování nemocničního lůžka. Uvažujte senzorovou síť textilních senzorů zatížení, případně RFID tagů, centrální výpočetní jednotku v podobě komunikační platformy vytvořenou na katedře telekomunikační techniky, FEL, ČVUT v Praze, proprietární komunikační protokol pro polohování nemocničního lůžka a vhodný způsob vizualizace dat získaných ze senzorové sítě. Navržený systém otestujte.

Seznam doporučené literatury:

[1] VLAŠIMSKÝ, Martin. Velkoplošný textilní senzor pro snímání plochy zatížené lidským tělem. Praha, 2018. Bakalářská práce. České vysoké učení technické v Praze. [2] ŠUBČÍK, Zdeněk. Ovládání nemocničního lůžka pomocí mobilní aplikace. Praha, 2016. Diplomová práce. České vysoké učení technické v Praze.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Marek Neruda, Ph.D., katedra telekomunikační techniky FEL

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

Datum zadání diplomové práce: **08.01.2020**

Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce: **30.09.2021**

Ing. Marek Neruda, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat svému vedoucímu diplomové práce Ing. Bc. Marku Nerudovi, Ph.D., Doc. Ing. Lukáši Vojtěchovi, Ph.D. za rady a pomoc. Také děkuji Ing. Pavlu Hnykovi, který vytvořil vizualizaci pro textilní senzory. Dále díky patří mé rodině a přítelkyni za velkou podporu.

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.

V Praze, 22. května 2020

Abstrakt

Cílem této diplomové práce je navrhnout systém pro automatické polohování nemocničního lůžka značky PROMA REHA. V systému je třeba uvažovat textilní senzory sítí zatížení a centrální jednotku. Následně navrhnout a vytvořit propojení textilní senzorové sítě skrze proprietární komunikační protokol pro polohování nemocničního lůžka s vhodnou vizualizací dat. Nakonec je nutné celý systém otestovat.

Klíčová slova: nemocniční lůžko, node-red, python, senzory

Vedoucí: Ing. Bc. Marek Neruda, Ph.D. FEL, katedra telekomunikační techniky

Abstract

Goal of these master thesis is to found system for automatic movements of medical bed PROMA REHA. It is needed to think about textile sensors and central controller. Next goal is to create connection between textile sensor's site through proprietary communication protocol for positioning medical bed and to found appropriate visualization of data. After that it is needed to test whole system.

Keywords: medical bed, node-red, python, sensors

Title translation: System for Automatic Positioning of Hospital Bed

Obsah

1 Úvod 1

Část I Popis fyzických komponent

2 Nemocniční lůžko PROMA REHA 5

2.1 Základní popis 5

2.2 Sesterský panel 7

2.2.1 Nastavení polohy 7

3 Textilní senzory zatížení 9

3.1 Vlastnosti textilního senzoru 9

4 Kontrolní jednotka textilního senzoru zatížení 13

4.1 Základní popis 13

4.2 Obsluha pomocí programovacích jazyků 14

4.3 Ovládání Kontrolní jednotky ... 14

5 Centrální výpočetní jednotka 17

5.1 Základní popis 17

5.2 Vlastnosti 18

Část II Popis softwarových komponent

6 Programovací jazyk Python 21

6.1 Vlastnosti 21

6.2 Historie 22

6.2.1 Verze 22

7 Node-red 25

7.1 Vlastnosti 25

7.2 Historie 26

8 PostgreSQL 27

8.1 Vlastnosti 27

8.2 Historie 28

8.2.1 Verze 28

Část III Praktická část

9 Návrh komunikačního systému 31

9.1 Motivace 31

9.2 Návrh řešení	32	12.3 Tabulka "Users"	53
10 Excelový soubor	35	13 Webová aplikace	55
10.1 Ochrana proti proleženinám . . .	36	13.1 Ovládací části	55
10.2 Nastavené polohy lůžka	37	13.2 Vizualizace	63
11 Komunikační protokol	39	14 Testování	65
11.1 Třída "serial_communication".	39	14.1 Komunikace mezi KJ a Centrální výpočetní jednotkou	66
11.2 Třída "data_database"	41	14.2 Napojení Centrální výpočetní jednotky na Databázi	68
11.3 Třída "bed_automat"	43	14.3 Komunikace Centrální výpočetní jednotky s nemocničním lůžkem skrze Bluetooth	69
11.4 Třída "main_database"	45	14.4 Ovládání Excelovým souborem	70
11.5 Třída "main_bedmoves"	46	14.5 Funkčnost webové aplikace	71
11.6 Třída "main_bedsore"	46	14.5.1 Sekce "Aktualizace senz. hodnot"	71
11.7 Třída "main_bed_starting_position" . . .	47	14.5.2 Sekce "Nastavené polohy lůžka"	72
11.8 Třída "main_create_user"	47	14.5.3 Sekce "Prevence proleženin"	73
11.9 Třída "main_login_user"	47	14.5.4 Sekce "Nastavit výchozí pozici"	73
12 Databáze	49	14.6 Správnost zobrazení vizualiace .	74
12.1 Tabulka "Sensors"	50		
12.2 Tabulka "Station"	52		

15 Závěr	75
Literatura	77
Seznam zkratek	79

Obrázky

2.1 PROMA REHA s.r.o. logo.	5	10.1 List pro kontrolu proleženin. . .	36
2.2 Nemocniční lůžko PROMA REHA. 6		10.2 List poloh pro provedení.	37
2.3 Sesterský panel PROMA REHA [1].	7	12.1 Tabulka Sensors.	51
3.1 Textilní senzor	9	12.2 Tabulka Station.	52
3.2 Textilní senzor připojení ke KJ..	10	12.3 Tabulka Users.	53
3.3 Velkoplošný textilní senzor [2]...	11	13.1 Seznam záložek ver. 1.0.	56
4.1 Kontrolní jednotka textilního senzoru zatizení.	15	13.2 Seznam záložek ver. 2.0.	56
5.1 LPWAN IoT Router [3].	17	13.3 Registrace uživatele.	57
6.1 Rozdíly mezi verzemi 2 a 3 [4], upraveno.	23	13.4 Přihlašovací obrazovka.	57
7.1 Node-red prostředí pro vývoj. . .	26	13.5 Ovládání nemocničního lůžka. .	58
7.2 Node-red příklad aplikace [5]. . .	26	13.6 Sekce uživatelé v administrátorském rozhraní.	59
8.1 PgAdmin 4 ukázka.	28	13.7 Ovládací část Aktualizace senz. hodnot.	60
9.1 Návrh komunikačního diagramu. 33		13.8 Ovládací část Nastavené polohy lůžka.	61
		13.9 Ovládací část Prevence proleženin.	62
		13.10 Ovládací část Nastavit výchozí pozici.	63

13.11 Ukázka Real-time zobrazení 1.	64
13.12 Ukázka Real-time zobrazení 2.	64

Tabulky

2.1 Ovládací funkce sesterského panelu [1].....	8
5.1 Vlastnosti Raspberry Pi.....	18

Kapitola 1

Úvod

Moderní svět se snaží vše zrychlit, zautomatizovat, vylepšit. Bez automatizace některá odvětví, jako je například automobilový průmysl, již nedokáží efektivně fungovat. Staví se více budov, které se nazývají tzv. „inteligentní budovy“. Ty stojí na využití senzorů a automatizace. Tyto senzory se používají například pro určení intenzity osvětlení na základě světla mimo budovu, předání hodnot centrální jednotce, která řídí motory zatahování, rozevírání žaluzií apod.

Cílem této práce je navrhnout a realizovat automatizovaný systém pro polohování nemocničního lůžka od firmy PROMA REHA, s.r.o., který na základě sítě textilních senzorů zatížení monitoruje rozložení hmotnosti člověka na lůžku. Tento systém bude sloužit k zabránění zdravotních komplikací (například proleženin, zahleňování a podobně). Ovládání zajistí proprietární komunikační protokol. K bezdrátové komunikaci s lůžkem je využito Bluetooth zařízení. Dále je nutné vytvořit vhodnou vizualizaci dat získaných ze sensorové sítě pomocí Kontrolní jednotky textilního senzoru zatížení a následně systém otestovat.

Jako základ pro tuto práci byla využita diplomová práce Zdeňka Šubčíka [1] a bakalářská práce Martina Vlašimského [2].

Práce Zdeňka Šubčíka pojednává o sestrojení Bluetooth modulu pro ovládání nemocničního lůžka a následném ovládání pomocí aplikace naprogramované pro platformu Android. Z této práce bylo převzato Bluetooth ovládání sesterského panelu nemocničního lůžka a samotný panel. Z práce Martina Vlašimského byly převzaty textilní senzory, pomocí kterých je určováno rozložení hmotnosti těla pacienta ležícího na lůžku.



Část I

Popis fyzických komponent

Kapitola 2

Nemocniční lůžko PROMA REHA

2.1 Základní popis

Polohovatelné nemocniční lůžko má velký význam pro zlepšení životního komfortu osob, kteří na něm musí ležet ze zdravotních důvodů. Toto lůžko umožňuje pacientům, bez nutnosti zásahu ošetřovatele, si nastavit polohu, jakou požadují. Oproti běžnému lůžku je zde možné nastavit více poloh, které usnadňují některé životně důležité úkoly (např. resuscitační poloha pro případnou resuscitaci pacienta). Další kladnou vlastností lůžka je, že šetří čas personálu. Čas je tak na efektivně využit [1].



Obrázek 2.1: PROMA REHA s.r.o. logo.

Společnost PROMA REHA, s.r.o. vytvořila takzvané "inteligentní" nemocniční lůžko. Obsahuje nastavení více poloh a má velké množství ovládaní. Mezi prvky ovládaní se řadí funkce zdvihu celého lůžka a dále například laterální

2. Nemocniční lůžko PROMA REHA

náklon, který umožňuje pacientovi ležet na boku dle potřeby. Je vhodné pro pacienty, kteří musí na lůžku setrvat delší dobu a z toho důvodu by jim mohly hrozit proleženiny.

K základnímu vybavení lůžka patří hrazda s hrazdičkou. Ta snese zatížení až 75 kg a umožňuje pacientovi si pomoci při sedání rukama. Dále je zde držák na kyslíkové lahve, teleskopický výsuvný infúzní stojan a držáky pro příslušenství na každém rohu lůžka. Dalším možným rozšířením je například stolek na přístroje s aretací nebo trakčním systémem určený pro montáž na čelo lůžka.

Větší podrobnosti je možné nalézt v diplomové práci Zdeňka Šubčíka [1].



Obrázek 2.2: Nemocniční lůžko PROMA REHA.

2.2 Sesterský panel

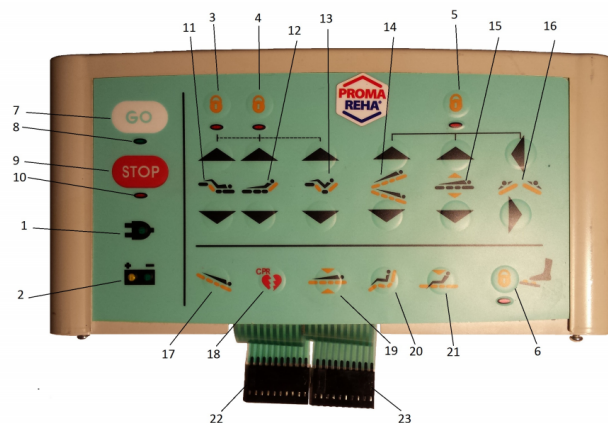
Sesterský panel umožňuje větší množství nastavení polohy. Pro bezpečnost pacienta je možné dočasně nebo trvale zamknout většinu funkcí ostatních ovladačů. Dočasného uzamknutí je možné docílit pomocí tlačítka "STOP". Pro trvalé zamknutí je potřeba zmáčknout tlačítko se symbolem zámku. Uzamčení je značeno svítící LED diodou pod příslušným zamykacím tlačítkem [1].

Zdeněk Šubčík v rámci diplomové práce implementoval do sesterského panelu Bluetooth modul. Ten slouží pro ovládání nemocničního lůžka zařízeními s operačním systémem Android. V této diplomové práci bude Bluetooth modul využit pro komunikaci s centrální výpočetní jednotkou, která obsahuje operační systém Raspbian. Využití Bluetooth je výhodné díky bezdrátové komunikaci. Systém Raspbian je vhodné využít z hlediska open-source licence a nenáročnosti systému na hardwarové komponenty.

2.2.1 Nastavení polohy

Pro nastavení polohy, kterou pacient požaduje je třeba nejprve zmáčknout tlačítko "GO" a následně si navolit polohu. Tlačítko je třeba držet do doby, dokud se lůžko nedostane do polohy, která je pacientem požadována. Po uvolnění tlačítka dojde k automatické aretaci v nastavené poloze [1].

Polohy, které je možné na sesterském panelu nastavit lze nalézt na obrázku 2.3. Popis jednotlivých tlačítek najdete v tabulce 2.1.



Obrázek 2.3: Sesterský panel PROMA REHA [1].

Číslo	Popis funkce
1	Indikátor napájení lůžka
2	Dvojice diod signalizujících stav akumulátoru lůžka: - Svítící zelená dioda - nabitý záložní zdroj - Přeblikávání zelená a oranžové diody - vybití - Svítící oranžová dioda - nízký stav záložního zdroje
3	Zámek polohování stehenního dílu
4	Zámek polohování zádového dílu
5	Zámek nastavení náklonu lůžka a zdvihu ložné plochy
6	Zámek nožního ovladače
7	Aktivační tlačítko GO
8	Indikátor aktivace funkcí sesterského panelu
9	Tlačítko STOP pro dočasné uzamčení funkcí ostatních panelů (deaktivovaný režim)
10	Indikátor deaktivovaného režimu
11	Polohování stehenního dílu
12	Polohování zádového dílu
13	Polohování stehenního a zádového dílu současně (autokontura)
14	Nastavení polohy Trendeleburg nebo Antitrendelenburg
15	Nastavení zdvihu ložné plochy
16	Nastavení laterálního náklonu (doleva nebo doprava)
17	Nastavení polohy Trendelenburg
18	Nastavení resuscitační polohy (CPR)
19	Nastavení vyšetřovací polohy
20	Nastavení polohy Kardiacké křeslo
21	Nastavení polohy pro snadné vstávání z lůžka
22	Konektor klávesnice
23	Konektor stavových LED diod

Tabulka 2.1: Ovládací funkce sesterského panelu [1].

Kapitola 3

Textilní senzory zatížení

3.1 Vlastnosti textilního senzoru

Textilní senzor zatížení je vytvořen jako kompozitní materiál složený ze tří vrstev. Vnější jsou tvořené vodivou vrstvou, která je buď z vodivé látky nebo z vodivého vlákna. Prostřední je z piezorezistivní plastové folie (Velostat). Jedná se o plastovou hmotu, která je 4 mm tenká a má el. odpor ($<500 \Omega/\text{cm}^2$). Na základě odporu lze určit, kde je váhové zatížení největší. Ukázka senzoru je zobrazena na obrázku 3.2.



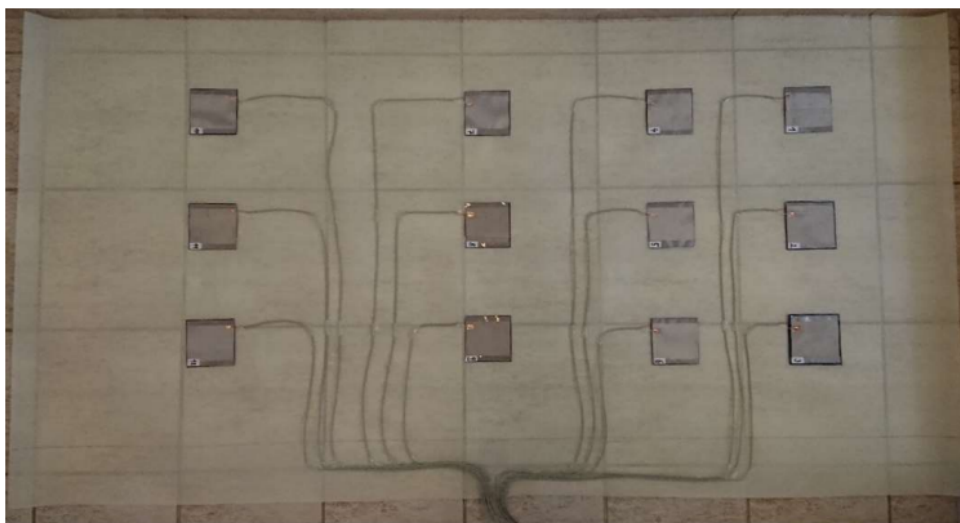
Obrázek 3.1: Textilní senzor

Vyvinutým zatížením se mění odpor. Vnější vodivé vrstvy se chovají jako elektrody. Pro obalení senzoru je použit izolační materiál (např. lepicí páska). Izolační materiál je zde použit k zamezení zkratu mezi dvěma vodivými vlákny. [2]. Více detailní popis senzorů, velkoplošného senzoru (obrázek 3.3) a jejich vývoj je možné nalézt v bakalářské práci Martina Vlašimského [2].

Textilní senzory jsou kabelově napojené na Kontrolní jednotku textilního senzoru zatížení (4), která získává hodnoty ze všech senzorů. Ukázka senzoru připojeného ke KJ je zobrazena na obrázku 3.2.



Obrázek 3.2: Textilní senzor připojení ke KJ.



Obrázek 3.3: Velkoplošný textilní senzor [2].

Kapitola 4

Kontrolní jednotka textilního senzoru zatížení

4.1 Základní popis

Kontrolní jednotka textilního senzoru zatížení (KJ) je komponenta velkoplošného textilního senzoru zatížení 3.3 s primární funkcí správy vyčítání dat z jednotlivých textilních senzorů zatížení. Vyčítání dat je umožněno pomocí jednoduchých příkazů. KJ je napájena pomocí micro-USB (zapojeným konektorem USB-A do výpočetní jednotky), pomocí kterého lze také vyčítat data. Tím je minimalizována potřebná kabeláž k provozu senzoru. Tato jednotka vznikla v rámci projektu c. *LTE117005 Technologie Auto-ID a Internetu věcí pro zvýšení kvality zdravotnických služeb programu Inter-Eureka* za podpory RVVI [6], která sdružuje projekty ČR. Kontrolní jednotka je názorně zobrazena na fotografii 4.1 včetně připojeného Bluetooth modulu.

4.2 Obsluha pomocí programovacích jazyků

KJ je možné obsluhovat různými programovacími jazyky pomocí sériové komunikace. Defaultní nastavení:

- baud rate 9600 bps
- 8 bits
- no parity
- 1 Stop bit
- no flow control

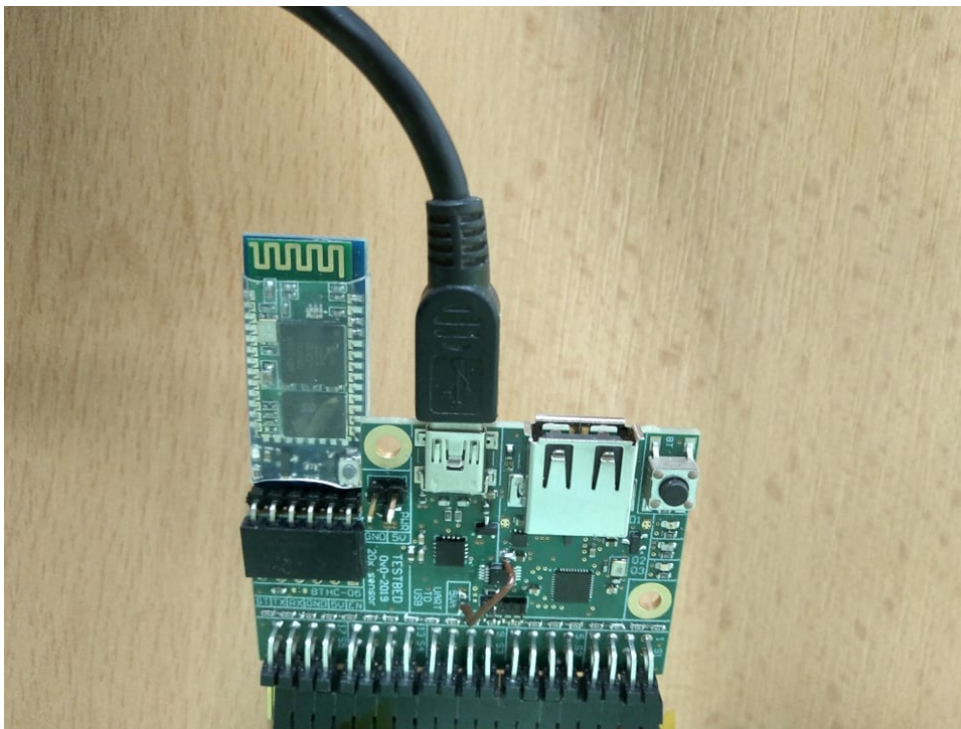
Je velmi důležité nastavit terminál, aby všechny příkazy byly ukončeny sekvencí <CR><LF>.

4.3 Ovládání Kontrolní jednotky

Po připojení napájem lze nápovědu získat pomocí příkazu help. Zobrazí se následující příkazy:

- read - p
 - Periodické vyčítání dat ze senzorů.
- read - a,b
 - Vyčítání dat ze senzorů a výpis ve formátu rozmístění senzorů zatížení "BED VIEW" (0 - 1023).
- read - a,l
 - Vyčítání dat ze senzorů a výpis v jednom řádku.
- read - n
 - Výpis dat n-tého senzoru.

- set - mini
 - Nastavení komunikace pomocí micro-USB.
- set - bt
 - Nastavení komunikace pomocí Bluetooth.
- AT - mode
 - Přepnutí do AT režimu, ze kterého lze posílat AT příkazy přímo do Bluetooth modulu.
- sleep
 - Přepnutí do režimu IDLE SLEEP.



Obrázek 4.1: Kontrolní jednotka textilního senzoru zatížení.

Kapitola 5

Centrální výpočetní jednotka

5.1 Základní popis

IoT LPWAN Router je kompaktní mobilní router s ethernetem, metalickým a bezdrátovým rozhraním pro připojení komponent. Zařízení bylo navrženo kolektivem pracovníků na fakultě elektrotechnické ČVUT v Praze pro provoz IoT služeb a pro aplikace používající technologie bezdrátových a senzorových sítí v rámci řešení projektu *TH02010568/Vícekanálová komunikační platforma pro Internet věcí (IoT) s finanční podporou TA ČR za podpory RVVI [6]*, která sdružuje projekty ČR. Jako jádro je zde umístěn modul Raspberry Pi s možnou kompatibilitou různých HW verzí.

Při instalaci vhodných modulů může sdružovat různé komunikační technologie a to s využitím metalických, případně optických vedení [3]. Jednotka je zobrazena na obrázku 5.1.



Obrázek 5.1: LPWAN IoT Router [3].

Raspberry Pi je název malého jednodeskového počítače s deskou plošných spojů ve velikosti platební karty. Raspberry Pi bylo designováno a vytvořeno nadací Raspberry Pi Foundation, což je nezisková organizace se zameřením na počítače a programové instrukce dostupné širokému okruhu lidí, s cílem podpořit výuku informatiky ve školách. Zařízení bylo přijato širším okruhem, než pouze studenty. Spousta firem začala využívat platformu pro různé projekty, jako jsou kontroléry pro roboty, pro domácí využití nebo do chytré domácnosti [7].

5.2 Vlastnosti

V tabulce 5.1 lze nalézt hardwarové vlastnosti použitého Raspberry Pi (3B+).

Vlastnosti
1.2 Ghz ARM processor Systems-On-a-Chip (SoC)
integrovaná paměť 4 GB RAM
1x HDMI port
1x 3.5mm jack (video i audio)
4x USB 2.0 porty pro periferie
1x microSD čtečka karet
1x Ethernet LAN port
1x WIFI/Bluetooth radio anténa
1x microUSB pro napájení
1x GPIO (General Purpose Input/Output) rozhraní

Tabulka 5.1: Vlastnosti Raspberry Pi.

Pro testování a vývoj je využita výpočetní jednotka Raspberry Pi 3, model B. Ta je kompatibilní s IoT LPWAN Routerem. Má stejné vlastnosti, potřebné rozhraní a konektory pro práci. Raspberry Pi je zde centrální výpočetní jednotka zajišťující hlavní chod webové aplikace, připojení k databázi a spouštění komunikačního protokolu.

Zařízení je napájeno skrze micro-USB. Pomocí Bluetooth modulu, který v sobě Raspberry Pi obsahuje, je připojené k upravenému sestřickému panelu nemocničního lůžka PROMA REHA (kapitola 2.2). Pro získávání hodnot ze senzorů je k Raspberry Pi připojeno pomocí USB-A ke Kontrolní jednotce textilního senzoru zatížení (kapitola 4), která je ovládána pomocí příkazů (kapitola 4.3).

Část II

Popis softwarových komponent



Kapitola 6

Programovací jazyk Python



6.1 Vlastnosti

Python je interpretovací, objektově orientovaný, vysokoúrovňový programovací jazyk s dynamickou sémantikou. Jeho vysokoúrovňová stavba v datových strukturách, kombinovaná s dynamickým psaním je velice atraktivní pro rychlé psaní. Python je jednoduchý pro naučení syntaxe a má velice dobrou čitelnost. Z toho důvodu je velice dobrý pro rychlé opravy a následnou údržbu.

Python využívají například firmy jako je Google, RedHat, IBM nebo Seznam.

Jazyk podporuje velké množství modulů a knihoven, které ho dělají ještě více atraktivním a využitelným v mnoha směrech. Externí knihovny jsou dostupné zdarma pro veškeré hlavní systémové platformy a mohou být nadále volně distribuovány [8].

■ 6.2 Historie

Jazyk Python byl vytvořen v letech 1990-1991 *Guido van Rossumem* v Matematickém centru Stichting v Nizozemsku jako následník jazyku ABC. Jeho tvůrce nadále zůstává hlavním autorem, i když je zde mnoho dalších přispěvatelů.

Název Python pochází z Monty Python's Flying Circus. Guido van Rossum uvedl, že v době, kdy hledal jméno pro svůj nový jazyk, běžel v televizi tento seriál a přišlo mu to vhodné a zábavné [9].

■ 6.2.1 Verze

Pro tuto práci byla zvolena verze 3.7.5 z důvodu podpory a knihoven (aktuální verze 3.8.2 neměla podporu všech potřebných knihoven). Verze a roky jejich vydání. [10]

- únor 1991 - publikována první verze 0.9.0
- leden 1997 - verze 1.0
- 19. říjen 2000 - verze 2.0
- 3. prosinec 2008 - verze 3.0
- 15. prosince 2019 - verze 3.7.5 (použitá)
- 24. února 2020 - verze 3.8.2 (aktuální)

Relativně velké rozdíly vznikly mezi verzí Python 2 a Python 3. Ty jsou znázorněny na obrázku 6.1.



Obrázek 6.1: Rozdíly mezi verzemi 2 a 3 [4], upraveno.

Kapitola 7

Node-red

7.1 Vlastnosti

Node-RED je flow-based programovací software vytvořený týmem IBM's Emerging Technology Services, který je aktuálně částí JS Foundation. Flow-based znamená řízení toku, tedy příkazy postupně "protékají" jako řeka.

Node-RED sestavuje aplikaci pomocí sítě "Nodů". Každý Node má definovaný význam. Pro příklad: získání dat, úprava dat, konvertování dat do požadovaného formátu atd. Síť, kterou jsou data připojené zodpovídá za přenášení dat mezi Nodes.

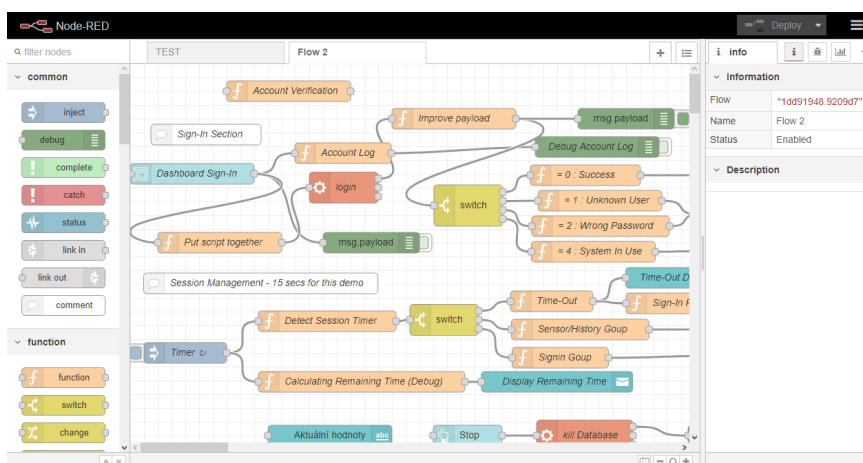
Pro vytvoření webové aplikace v Node-red není potřeba velkých programovacích zkušeností. Tento model je velice dobře reprezentovaný vizuálně a je vhodný pro většinu uživatelů. Pokud se vyskytne chyba, uživatel je schopný si pomocí flow a debug Nodes zjistit, kde se stala chyba [5].

Rozšíření programovacího nástroje Node-RED proběhlo, protože umožňuje rychle a jednoduše software propojit s několika komponentami. Těmi jsou zejména senzory a aktuátory s řídicí jednotkou v podobě Embedded počítače (primárně určené pro Raspberry Pi a jejich operační systém (OS) Linux Raspbian. Požadavek přišel s rozvojem IoT. Node-RED je stavěn jako aplikace pro řízení toku dat z IoT zařízení.

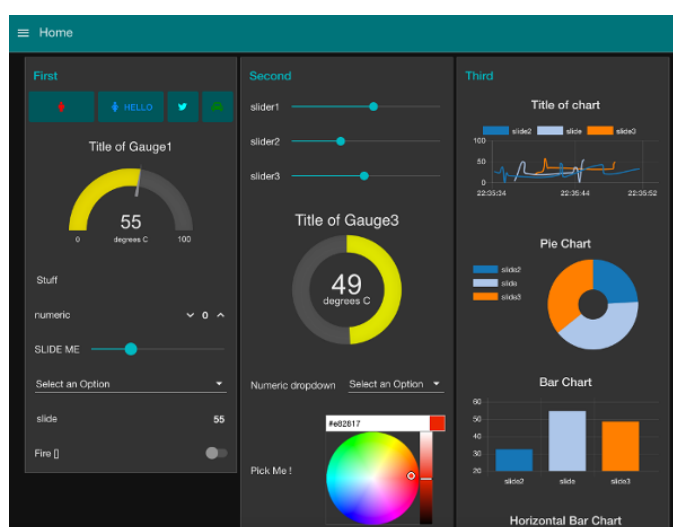
7.2 Historie

Node-RED vznikl začátkem roku 2013 jako vedlejší projekt Nick O'Learyho a Dave Conway-Jonese z IBM's Emerging Technology Services group.

Ze začátku koncept sloužil pro vizualizaci a manipulační mapování mezi přenosy vzdálených měření pomocí front zpráv (MQTT). Tyto zprávy představují způsob nenáročného přenosu malého množství dat prostřednictvím TCP/IP internetové sítě. Brzy se vytvořil více všeobecný nástroj pro rozšíření. V září 2013 byl nástroj poskytnut a vyvíjen pod open-sourced licencí. Pro vývoj byl vytvořen nový projekt firmy JS Foundation v říjnu 2016 [5].



Obrázek 7.1: Node-red prostředí pro vývoj.



Obrázek 7.2: Node-red příklad aplikace [5].

Kapitola 8

PostgreSQL

8.1 Vlastnosti

Structured Query Language (SQL) je strukturovaný dotazovací jazyk, který se používá pro práci s daty v relační databázi.

PostgreSQL je open-source objektově relační databázový systém, který používá a rozšiřuje jazyk SQL. Rozšiřuje o další možnosti a úpravy v bezpečném skladování a komplikovaných datových workloadech.

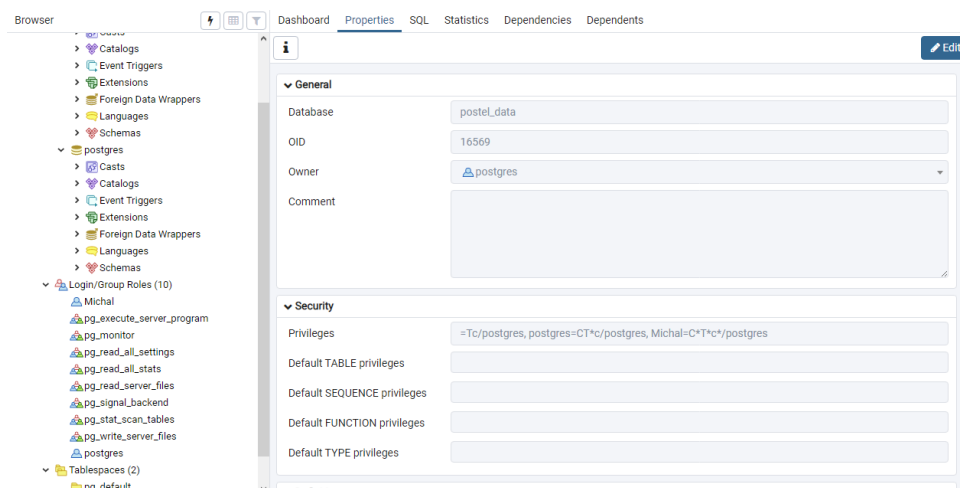
Systém PostgreSQL získal svoji reputaci díky své architektuře, spolehlivosti, datové integritě, robustnosti setů, možnosti rozšíření. Databázový systém funguje na většině operačních systémech. Systém přichází se spoustou změn, které jsou mířeny na podporu vývojářů aplikací, administrátorů na ochranu datové integrity, dále pro vytvoření robustního prostředí a správu velkého množství dat. Má řadu rozšíření: lze si nadefinovat vlastní datový typ, vytvořit vlastní funkce, napsat kód v jiném programovacím jazyce bez rekompilací databáze [11].

8.2 Historie

Základ PostgreSQL je datován do roku 1986 jako část projektu nazývaný POSTGRES na Univerzitě Berkeleyho v Californii. Jádro platformy je aktivně vyvíjeno více než 30 let [11].

8.2.1 Verze

V této práci je využita verze 11.5. (datum vydání: 8. srpna 2019) a pro administraci databáze byl zvolen software PgAdmin 4, který je přehledný pro správu dat.



Obrázek 8.1: PgAdmin 4 ukázka.



Část III

Praktická část



Kapitola 9

Návrh komunikačního systému



9.1 Motivace

Motivace pro vytvoření této práce je zlepšení podmínek pro ošetřovatelky, sestry i doktory v nemocnicích, domovech důchodců a jiných zařízeních s péčí o pacienty, kteří jsou upoutaní na lůžko. Automatizace by mohla například pomoci: při polohování pacientů a tím zabránění proleženin, nebo sestřám na nočních službách. Sestry nebo ošetřovatelky musí všechny pacienty obejít a manuálně jim změnit polohu. Toto řešení dává větší prostor pro vykonávání jiných povinností a zlepšení péče.

9.2 Návrh řešení

Textilní senzory jsou fyzicky spojené s KJ (kapitola 4). Rozhraní micro-USB/USB-A bude umožňovat KJ připojit k IoT LPWAN Routeru (kapitola 5), který bude zastoupený v kompatibilním zařízení Raspberry Pi ve verzi (3B+). Tímto zařízením bude možné KJ napájet a zároveň přes to samé rozhraní lze číst data ze senzorů. Další vlastností centrální řídicí jednotky je možnost komunikovat pomocí komunikačního protokolu se zařízením, na kterém bude nainstalována databáze PostgreSQL

Komunikační protokol je vhodné psát v Python jazyce (kapitola 6) z důvodu výše uvedených. Komunikační protokol bude obsahovat řady skriptů pro komunikaci mezi všemi zařízeními. Mezi tyto skripty budou patřit následující:

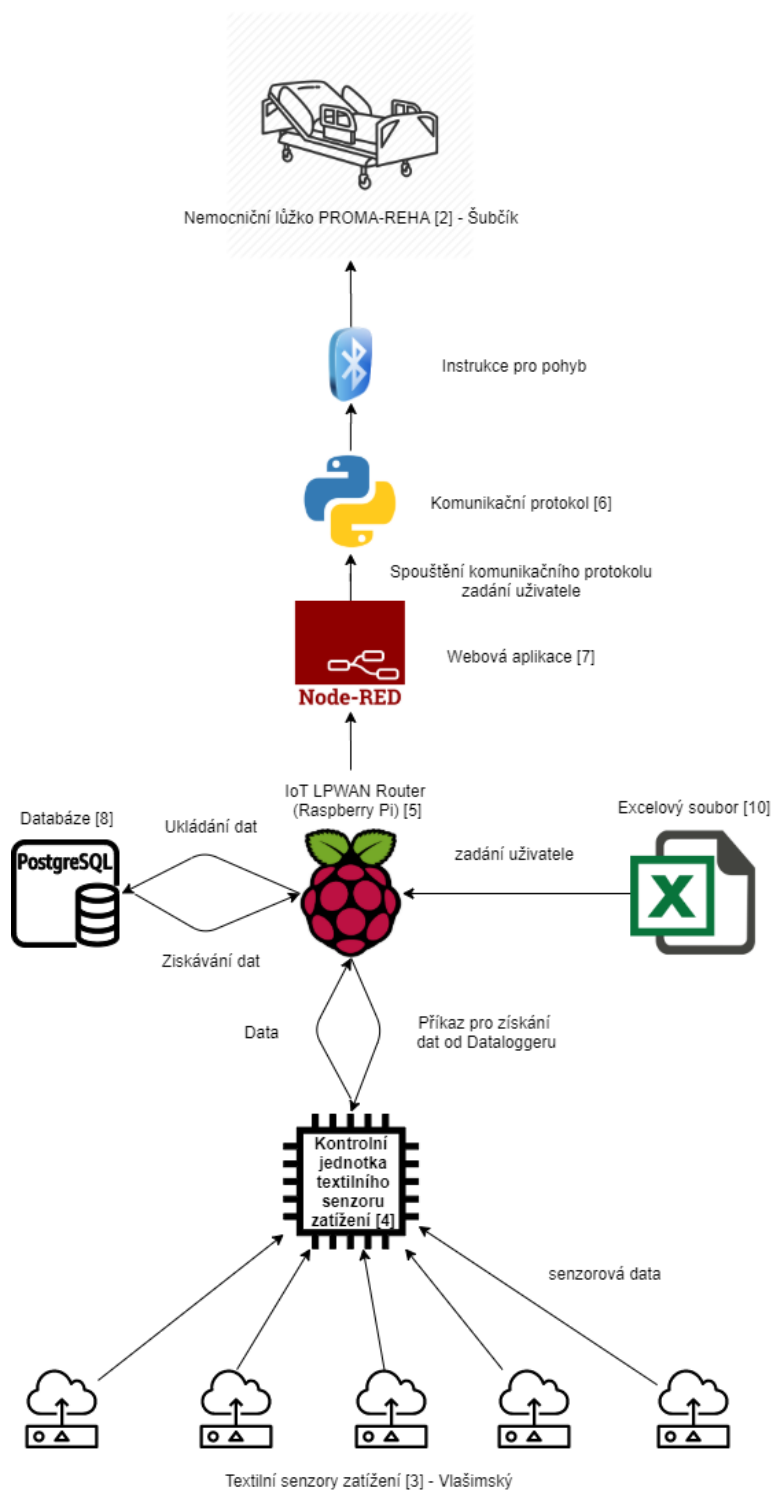
- obsluhu Kontrolní jednotky textilního senzoru
- ukládání a čtení databáze
- připojení lůžka k LPWAN routeru pomocí Bluetooth
- obsluhu lůžka dle pokynů zadávajícího
- časová kontrola proti proleženinám
- uvedení do základní polohy
- přihlašování do webové aplikace
- registrace do webové aplikace

Pomocí aplikace a excelového souboru bude možné do skriptů, určených pro ovládání a ochranu proleženin, vkládat nastavení polohy lůžka.

K databázi se IoT LPWAN Router přihlásí pomocí vlastního profilu. Data, která se přijmou ze senzorů se následně uloží do tabulky databázi s ID nemocničního lůžka a časovou značkou. Kontroly polohy lůžka bude vhodné ukládat do další tabulky v databázi.

IoT LPWAN Router je brána jako centrální řídicí jednotka, na které bude nainstalována webová aplikace. Ta bude napsána v Node-red (kapitola 7), z důvodů již výše uvedených v kapitole 7. Díky webové aplikaci bude mít uživatel možnost přidávat nastavení. Následně IoT LPWAN Router bude číst

data ze senzorů, spouštět a ukončovat komunikační protokol.
Blokové schéma lze nalézt na obrázku 9.1.



Obrázek 9.1: Návrh komunikačního diagramu.

Kapitola 10

Excelový soubor

Z důvodu jednoduššího ovládání bude vytvořen excelový soubor, do kterého může uživatel, například doktor, připravit polohy, které budou s lůžkem provedeny. Nastavení nemusí být pouze pro jedno lůžko, ale pro všechny lůžka, která lze připojit k centrální výpočetní jednotce (kapitola 5). Komunikační protokol (kapitola 11) si odliší podle ID lůžka, jaké y musí provést.

Pro nastavení poloh je vytvořen seznam všech poloh, které budou podporovány Komunikačním protokolem.

Seznam poloh	Pohyb nahoru	202
	Pohyb dolů	206
	Náklon lůžka doprava	218
	Náklon lůžka doleva	222
	Polohování hlavové podpěry nahoru	186
	Polohování hlavové podpěry dolů	190
	Polohování podpěry noh nahoru	170
	Polohování podpěry noh dolů	174
	Celé lůžko hlavou nahoru	138
	Celé lůžko hlavou dolů	178

Soubor je rozdělen na 2 listy, kde první list (*Ochrana proti proleženinám*) se váže k ochraně proti proleženinám a druhý list (*Nastavené polohy lůžka*) je pro nastavení posloupnosti poloh.

Kapitola 11

Komunikační protokol

11.1 Třída "serial_communication"

Obsahuje třídu *SerialInitCommands*, která má za úkol udržovat komunikaci mezi KJ (kapitola 4) a PostgreSQL databází (kapitola 8). Pro komunikaci s databází je zde využita třída *DataDatabase*. Vstupní parametry *SerialInitCommands* jsou:

1. **ID** - ID zařízení
2. **port** - USB, na kterém je kontrolní jednotka připojena.
3. **speed** - rychlost přenosu pro USB
4. **bt_mac** - MAC adresa Bluetooth kontrolní jednotky.

Komunikace může probíhat pomocí USB nebo při vhodné implementaci je možné využít Bluetooth. Pro USB komunikaci je použita knihovna *serial* a pro Bluetooth je nejlepší knihovna *PyBluez*. Kontrolní jednotka textilního senzoru zatížení používá vlastní příkazy pro výpis hodnot senzorů. Pro každý příkaz je napsaná vlastní funkce.

■ Funkce "clean_data"

Komunikační protokol pomocí funkce *clean_data*, která má jeden číselný parametr, přebírá od KJ (kapitola 4) data ze senzorů, které čistí od přebytečných znaků. Poté je ukládá do požadovaného výstupu podle zadaného parametru.

■ Funkce "print_one"

Print_one funkce má jeden číselný parametr. Pomocí parametru určíme senzor, jehož hodnota se má vypsát. Ta se vypíše zavoláním příkazu *read - "číslo z parametru"*. Tato zpráva je kódována pomocí *utf-8* a poté zaslána do KJ (kapitola 4). Následovně je zavolána funkce *clean_data* s hodnotou parametru 2. Návratovou hodnotou této funkce je hodnota senzoru.

■ Funkce "print_all_periodic"

Funkce, která zavolá příkaz *read - p*. Zpráva je kódována pomocí *utf-8* a poslána do KJ (kapitola 4). Funkce dále zavolá *clean_data* s hodnotou parametru 3. KJ bude periodicky vypisovat hodnoty všech senzorů na konzoli, dokud mu nebude poslán příkaz znovu.

■ Funkce "print_all"

Print_all zavolá příkaz *read -a,b*, který je opět kódován pomocí *utf-8*. Zavolá se funkce *clean_data* s hodnotou parametru 1, ta vrátí očištěná data. Návratovou hodnotou funkce *print_all* je pole hodnot všech senzorů.

■ Funkce "helper"

Helper je funkce, která zavolá příkaz *help* zakódovaný pomocí *utf-8*. Následně je zavolán *clean_data* s hodnotou parametru 4.

11.2 Třída "data_database"

Třída *DataDatabase* má za úkol připojit se do databáze a provést import a export pro dané části programu. Nemá žádné vstupní parametry.

V inicializaci se třída připojí do databáze (kapitola 8). Ta byla rozdělena na 3 tabulky. První tabulka se nazývá *Sensors* (podkapitola 12.1). Uchovává hodnoty vyčtené ze senzorů, ID lůžka a časovou značku. Druhou tabulkou je *Station* (podkapitola 12.2). Ta obsahuje informace o pozicích lůžka, ID lůžka a časovou značku. V třetí tabulce, která se nazývá *Users* (podkapitola 12.3) se uchovávají přihlašovací údaje uživatele (přihlašovací jméno a zašifrované heslo).

Pro připojení do databáze, zapisování a získávání hodnot využívá tato třída funkce *execute* z knihovny *psycopg2*. Pro časového odlišení se zde používá časovou značku (timestamp) z knihovny *datetime*.

Funkce "import_to_sensors"

Funkce, pomocí které program do databáze zapisuje nově naměřené hodnoty ze senzorů, ID nemocničního lůžka (kapitola 2) a časovou značku.

Funkce "export_from_sensors"

Využití této funkce je použito pro vypisování deseti nejaktuálnějších měření na základě parametru *bed_id*. Tento parametr znamená ID daného nemocničního lůžka v databázi. Návrátovou hodnotou je pole polí s hodnotami deseti posledních měření.

Funkce "import_to_station"

Touto funkcí se do tabulky *Station*(podkapitola 12.2) v databázi zapisuje ID nemocničního lůžka, jeho hodnoty náklonů jednotlivých stran a časovou značku.

■ Funkce "export_from_station"

Funkce, která má parametr *bed_id*, značící ID nemocničního lůžka. Před každou novou pohybovou operací lůžko získává nejaktuálnější data o náklonech. Návrátovou hodnotou je pole s aktuální pozicí lůžka.

■ Funkce "create_login"

Create_login obsahuje parametry *nickname* a *password*. První parametr označuje uživatelské jméno. Druhý parametr je heslo, které je při vstupu bez zašifrování. Pomocí této funkce zašifrujeme heslo pomocí knihovny *bcrypt*. Tato knihovna využívá symetrickou blokovou šifru s názvem Blowfish. Uživatelské jméno společně se zašifrovaným heslem uložíme do databáze do tabulky *Users* (podkapitola 12.3) a tím vytvoříme uživatelský přístupový profil. Při úspěšném vytvoření uživatelského profilu je návratová hodnota číslo 0. Neúspěšný pokus vrací hodnotu 1.

■ Funkce "check_login"

Tuto funkci využívá webová aplikace k ověřování uživatelského jména a hesla. Má stejné parametry, jako funkce *create_login* - tedy *nickname* a *password*. Při přihlašování je heslo nejprve zašifrováno stejnou šifrou, jako při vytváření uživatele. Následně se porovnává se zašifrovaným heslem v databázi. Pokud je heslo a uživatelské jméno správně a jestli je uživatel aktivován administrátorem, tak ještě probíhá kontrola administrátorských oprávnění funkcí *check_admin*. Při potvrzení administrátorských oprávnění vrací funkce jako návratovou hodnotu číslo 3, jinak je návratovou hodnotou funkce číslo 0. Pokud uživatel neexistuje, je návratovou hodnotou číslo 1. Neúspěšný pokus o přihlášení vrací návratovou hodnotu s číslem 2. Webová aplikace na základě návratové hodnoty dokončí nebo nedokončí přihlašování.

■ Funkce "check_admin"

Pro ověření, jestli je uživatel administrátor se používá funkce *check_admin*. Vstupním parametrem je uživatelské jméno, podle kterého funkce vyhledá v databázi, zda má uživatel administrátorská oprávnění. Pokud je obsahuje, pak je návratovou hodnotou číslo 0.

■ 11.3 Třída "bed_automat"

Obsahuje třídu *Automatization*. Třída řeší navázání spojení mezi Rapsberry Pi a nemocničním lůžkem pomocí rozhraní Bluetooth. Jejím jediným parametrem je id, který představuje ID nemocničního lůžka. Parametr se využívá pro získávání předchozích náklonů lůžka a pro zapisování nových hodnot do databáze. V inicializační fázi vytvoří klientský soket pro navázání spojení mezi komponentami. Pomocí funkce *connect_to_bed* je navázáno spojení. Následně program pomocí definovaných funkcí provádí operace. Mezi operace se řadí polování, kontrola náklonů, získání náklonů lůžka z předchozího stavu, aktualizace náklonů v databázi, získání dat ze souboru s příponou xls a vybrání poolohy pro prevenci proleženin. Pro využití Bluetooth komunikace je zde využita knihovna *PyBluez* a pro komunikaci s databází se využívá třída *DataDatabase*.

■ Funkce "connect_to_bed"

Funkce slouží pro připojení *centrální řídicí jednotky* (kapitola 5) k *sesterskému panelu nemocničního lůžka* (podkapitola 2.2). Obsahuje adresu pro připojení.

■ Funkce "disconnect_from_bed"

Funkcí přerušíme spojení mezi *centrální řídicí jednotkou* (kapitola 5) a *sesterským panelem na nemocničním lůžku* (podkapitola 2.2).

■ Funkce "bed_data"

Komunikační protokol funkcí získává hodnoty ze sesterského panelu nemocničního lůžka (podkapitola 2.2). Pro získávání hodnot se využívá Bluetooth funkce *recv* z knihovny *PyBluez*.

■ Funkce "get_latest_data_from_db"

Funkce pro získání nejaktuálnějších náklonů nemocničního lůžka z databáze. Využívá funkci *export_from_station* z třídy *DataDatabase*.

■ Funkce "check_positioning"

Funkce kontroluje maximální možné náklony a zda je pozice validní. Tyto hodnoty určuje na základě vstupních parametrů *number_of_update* a *move*, kde první parametr je index v poli náklonů a hodnot a druhý parametr značí aktuálně zadaná poloha, která bude použita pro nemocniční lůžko. Návrátovou hodnotou této funkce je *True* nebo *False*.

■ Funkce "move_to_update"

Pomocná statická funkce, která má za úkol přiřadit správný index podle vstupního parametru *number_of_move*. Využívá se pro zaktualizování správné hodnoty náklonu nebo polohy. Tato hodnota je vrácena touto funkcí.

■ Funkce "get_data_from_xls"

Funkce získává data z excelového souboru. Vstupním parametrem (*excel_list*) je zde list excelové souboru. Pro získání dat je využita knihovna *pandas*. Pomocí funkce *read_excel*, která se nachází v knihovně čteme konkrétní list a konkrétní sloupce. Návrátovou hodnotou je zde list poloh pro nemocniční lůžko kapitola (2).

■ Funkce "get_bedsore_prevention_move"

V této funkci získáváme konkrétní polohu pro efektivní přenesení váhy pacienta a tím preventujeme proleženiny. Vstupními parametry je zde seznam senzorů s polohou, který reaguje na jeho přetížení a číslo senzoru, který má největší odporové zatížení během posledních 10 měření.

Návratovou hodnotou je zde poloha s délkou trvání.

■ Funkce "send"

Nejdůležitější funkce celé této třídy. Její vstupní parametry jsou *number_of_move* a *num_of_ticks*. První parametr uvádí konkrétní ID příkazu polohy pro lůžko a druhý parametr rozsah polohy. Tato funkce komunikuje pomocí Bluetooth rozhraní s nemocničním lůžkem a posílá mu pomocí funkce *send* z *PyBluez* knihovny příkaz.

Jako první se zasílá příkaz pro aktivaci tlačítka GO, které je ochranným tlačítkem proti nechtěnému pohybu. Funkce získá nejaktuálnější hodnoty náklonu lůžka pomocí funkce *get_latest_data_from_db*. Provede se kontrola, zda následující poloha může být provedena z hlediska bezpečnosti člověka ležícího na lůžku. Následně se ve smyčce zasílá požadovaný příkaz polohy a neustále dochází ke kontrole, zda následující poloha je v požadované normě hodnot podle *check_positioning*.

■ 11.4 Třída "main_database"

Spouštěcí skript pro kontrolu a zápis nových hodnot ze senzorů do tabulky *Sensors* (podkapitola 12.1) v databázi.

Skript nejdříve načte ID lůžka, které je vstupním parametrem. Ten je zadán uživatelem ve webové aplikaci. Dojde k navázání spojení s KJ (podkapitola 4) a databází inicializací třídy *SerialInitCommands* ze skriptu *serial_communication* (podkapitola 11.1). Ověří se, zda je databáze přístupná a vloží *ID lůžka*, zadané jako vstupní parametr, pro správný zápis hodnot.

V dalším kroku je nekonečná smyčka se zápisem hodnot do databáze, které se zapisují jednou za 5 minut.

11.5 Třída "main_bedmoves"

Spouštěcí skript pro komunikaci s nemocničním lůžkem.

První část skriptu načte vstupní parametry z webové aplikace (kapitola 13) - *ID lůžka* a *Počet cyklů*. Inicializuje třídu *Automation* ze skriptu *bed_automat*, které předá ID parametr lůžka a spojí se s lůžkem funkcí *connect_to_bed*. Po připojení načte data z excelového souboru (kapitola 10) a data budou čtena po řádcích.

Data obsahují:

- **ID lůžka** - Získ hodnot pouze pro dané lůžko.
- **Prodleva mezi polohami v minutách**
- **Rozsah polohy**
- **Konkrétní poloha**

Po úspěšném dočtení všech řádků funkce zkontroluje vstupní parametr pro počet cyklů. Pokud bude počet cyklů větší než 1, tak všechny polohy z excelového souboru pro dané lůžko zopakuje. Po dokončení všech cyklů dojde k odpojení od lůžka pomocí funkce *disconnect_from_bed* z skriptu *bed_automat* (podkapitola 11.3).

11.6 Třída "main_bedsore"

Tento spouštěcí skript provádí kontrolu zatíženosti senzorů a následně reaguje. Vstupními parametry z webové aplikace jsou *ID lůžka* a *Čas*. Doba, po které bude provedena kontrola zatíženosti. Po zpracování parametrů jsou načtená data z listu *Ochrana proti proleženinám* (podkapitola 10.1) uloženého v excelovém souboru (kapitola 10), navázáno spojení s databází. Následuje nekonečná smyčka, kdy čekáme časový interval, který byl zadán ve vstupních parametrech. Pro tento krok byla využita funkce *sleep* z knihovny *time*. Po uplynutí

určitého času jsou načteny nejaktuálnější hodnoty z měření na senzorech funkcí 11.2 z třídy *Data_Database* (podkapitola 11.2). Tyto nejnovější hodnoty jsou hodnoceny podle nejstaršího z měření. Pokud hodnota některého ze senzorů překročila rozhodovací úroveň, je zapsána do seznamu. Po vyhodnocení všech měření se vybírá nejvíce zatížený senzor ze seznamu. Podle čísla vybraného senzoru je funkcí *get_bedsore_prevention_move* (podkapitola 11.3) získána poloha z *Ochrana proti proleženinám* (podkapitola 10.1). Navazuje test připojení k nemocničnímu lůžku pomocí funkce *connect_to_bed* (podkapitola 11.3). Když je připojení úspěšné, je poslána funkcí *send* (podkapitola 11.3) poloha s jejím rozsahem. Na závěr dojde k odpojení od lůžka a smyčka se vrátí do režimu „čekání“.

11.7 Třída "main_bed_starting_position"

Tímto spouštěcím skriptem uvedeme lůžko do základní pozice. Vstupními parametr je zde ID lůžka, se kterým bude pohybováno. Po načtení parametrů je proveden test připojení k lůžku. Když je test připojení úspěšný, tak je lůžko pomocí funkce *send* (podkapitola 11.3) uvedeno do základní pozice a poté k odpojení od lůžka.

11.8 Třída "main_create_user"

Pomocí tohoto spouštěcího skriptu je přístup pro uživatele do webové aplikace. Vstupními parametry z webové aplikace jsou uživatelské jméno a heslo. V dalším kroku je zavolána funkce *create_login* (podkapitola 11.2), která provede šifrování a vložení do databáze.

11.9 Třída "main_login_user"

Skript provádí kontrolu uživatele, který se snaží přihlásit. Vstupními parametry z webové aplikace jsou uživatelské jméno a heslo. Po načtení parametrů je využita funkce *check_login* (podkapitola 11.2), která provede kontrolu uživatelského jména a hesla z databáze.



Kapitola 12

Databáze

Jako databáze byla vybrána PostgreSQL (kapitola 8). Databáze má přehlednou webovou aplikaci pro ovládání a kontrolu dat. Pro Python existuje knihovna *psycopg2*, která velice dobře pracuje s databází. Centrální řídicí jednotka (kapitola 5) se do databáze přihlašuje pomocí vlastního profilu, který zná Komunikační protokol (kapitola 11). Je tak možné získat požadované data.

Pro tuto práci byly vytvořeny následující tabulky:

- Sensors 12.1
- Station 12.2
- Users 12.3

■ 12.1 Tabulka "Sensors"

Tabulka Sensors obsahuje data jednotlivých senzorů, které jsou kontinuálně zapisovány pomocí skriptu *main_database*.

Obsahuje tyto sloupce:

- **id** - Primární klíč tabulky.
- **resist_1, resist_2, resist_3, resist_4, resist_5, resist_6, resist_7, resist_8, resist_9, resist_10, resist_11, resist_12, resist_13, resist_14, resist_15, resist_16, resist_17, resist_18, resist_19, resist_20** - Hodnoty jednotlivých senzorů.
- **time** - Časová značka pro každý zápis (timestamp).
- **id_equipment** - ID lůžka pro filtrování a kontrolování hodnot senzorů.

Sensors	
id	int
resist_1	numeric
resist_2	numeric
resist_3	numeric
resist_4	numeric
resist_5	numeric
resist_6	numeric
resist_7	numeric
resist_8	numeric
resist_9	numeric
resist_10	numeric
resist_11	numeric
resist_12	numeric
resist_13	numeric
resist_14	numeric
resist_15	numeric
resist_16	numeric
resist_17	numeric
resist_18	numeric
resist_19	numeric
resist_20	numeric
time	timestamp
id_equipment	int

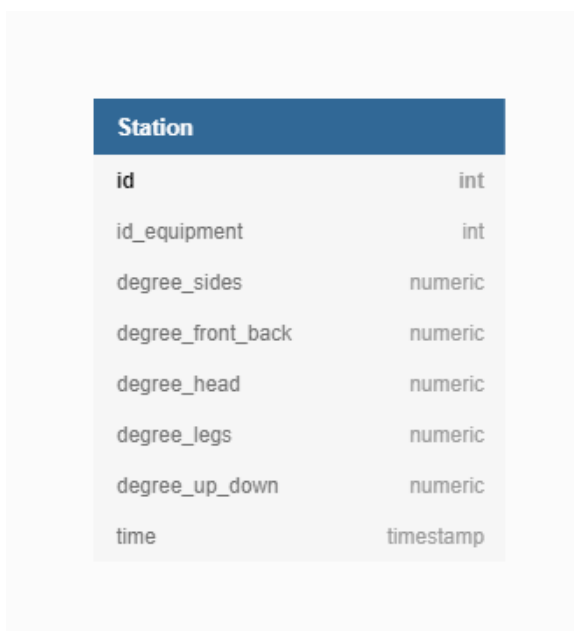
Obrázek 12.1: Tabulka Sensors.

12.2 Tabulka "Station"

Tabulka *Station* obsahuje informace o náklonech a polohách lůžka. Do této tabulky se zapisuje pomocí *import_to_station*. Data se získávají pomocí *export_from_station* z třídy *DataDatabase*.

Obsahuje tyto sloupce:

- **id** - Primární klíč tabulky.
- **id_equipment** - ID lůžka pro filtrování a získávání naposled zapsaných dat z databáze.
- **degree_sides** - Hodnota náklonu do stran lůžka.
- **degree_front_back** - Náklon lůžka hlavou nahoru nebo hlavou dolů.
- **degree_head** - Náklon hlavové podpěry.
- **degree_legs** - Náklon podpěry pro nohy.
- **degree_up_down** - Výška lůžka.
- **time** - Časová značka pro každý zápis (timestamp).



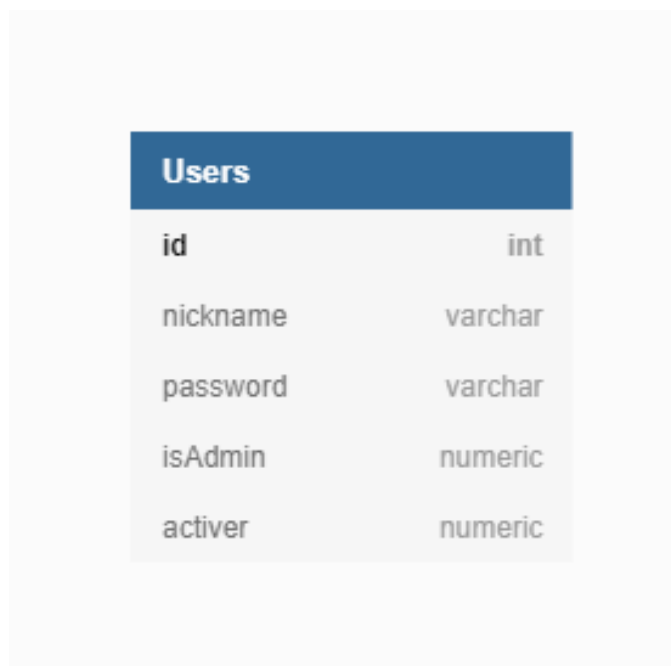
Station	
id	int
id_equipment	int
degree_sides	numeric
degree_front_back	numeric
degree_head	numeric
degree_legs	numeric
degree_up_down	numeric
time	timestamp

Obrázek 12.2: Tabulka Station.

12.3 Tabulka "Users"

Tabulka users obsahuje uživatelské jména a zašifrované hesla pomocí metody Blowfish (podkapitola 11.2) pro přihlašování do webové aplikace. Sloupce:

- **id** - Unikátní primární klíč tabulky.
- **nickname** - Uživatelské jméno.
- **password** - Zašifrované heslo pomocí šifry Blowfish.
- **isAdmin** - Kontrola administrátorských práv.
- **active** - Zda je uživatel aktivován administrátorem.



Users	
id	int
nickname	varchar
password	varchar
isAdmin	numeric
active	numeric

Obrázek 12.3: Tabulka Users.



Kapitola 13

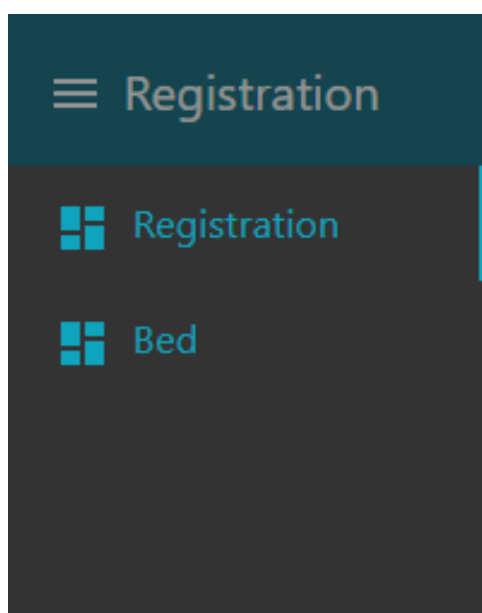
Webová aplikace



13.1 Ovládací části

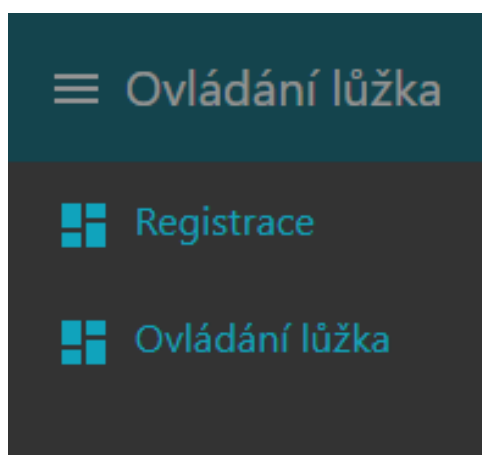
Z důvodů, které byly popsány v kapitole 7, byl vybrán softwarový nástroj Node-RED k vytvoření webové aplikace.

V části této kapitoly je možné vidět aktuální zobrazení webové aplikace, která je již ve verzi 2.0. Po vytvoření verze 1.0 se přemýšlelo nad chováním uživatele a nad interaktivností. Následně byly upraveny nadpisy, funkčnost a zobrazení tak, aby interakce uživatele s aplikací co nejpřívětivější a nejjednodušší. Pro příklad ve verzi 2.0 je přeložen seznam záložek (viz. obrázek 13.1).

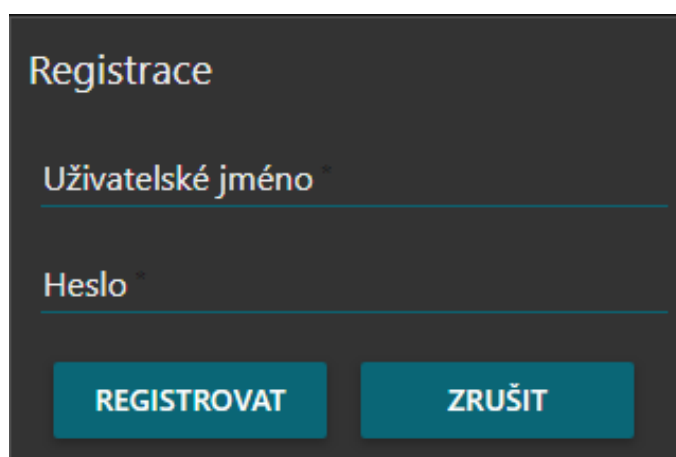


Obrázek 13.1: Seznam záložek ver. 1.0.

Do webové aplikace se lze přihlásit po zhotovení profilu uživatele. Pro vytvoření je připravena záložka *Registrace* (obrázek 13.2), kde najdeme formulář pro vyplnění. Ve formuláři je povinné vyplnit uživatelské jméno a heslo. Po potvrzení formuláře (obrázek 13.3) pomocí tlačítka "REGISTROVAT", dojde k uložení, které probíhá za pomoci funkce (podkapitola 11.8) z Komunikačního protokolu. Informace se vkládají do tabulky *Users* (podkapitola 12.3) v databázi. Po registraci je nutné počkat na aktivaci uživatelského profilu administrátorem.



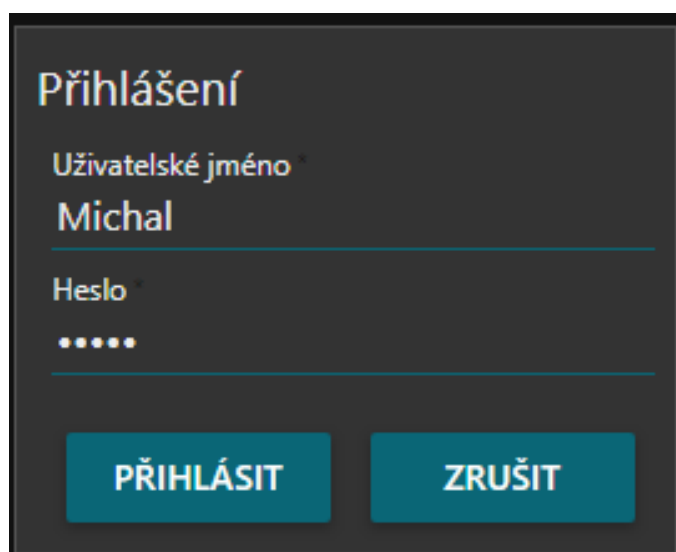
Obrázek 13.2: Seznam záložek ver. 2.0.



The image shows a registration form titled "Registrace" on a dark background. It features two input fields: "Uživatelské jméno" (Username) and "Heslo" (Password). Below the fields are two buttons: "REGISTROVAT" (Register) and "ZRUŠIT" (Cancel).

Obrázek 13.3: Registrace uživatele.

V případě přihlášení uživatel zvolí záložku *Ovládání lůžka* (obrázek 13.4). Následně uživatel vyplní formulář s jeho uživatelským jménem a heslem, který si navolil již při registraci a stiskne tlačítko "PŘIHLÁSIT". Funkce *check_login* (podkapitola 11.2), které je součástí funkce 11.9 v Komunikačním protokolu ověří, zda jsou údaje zadané správně a jestli byl uživatel aktivován administrátorem. Pokud ano, uživatele přihlásí.



The image shows a login form titled "Přihlášení" on a dark background. It features two input fields: "Uživatelské jméno" (Username) with the value "Michal" and "Heslo" (Password) with five dots. Below the fields are two buttons: "PŘIHLÁSIT" (Login) and "ZRUŠIT" (Cancel).

Obrázek 13.4: Přihlašovací obrazovka.

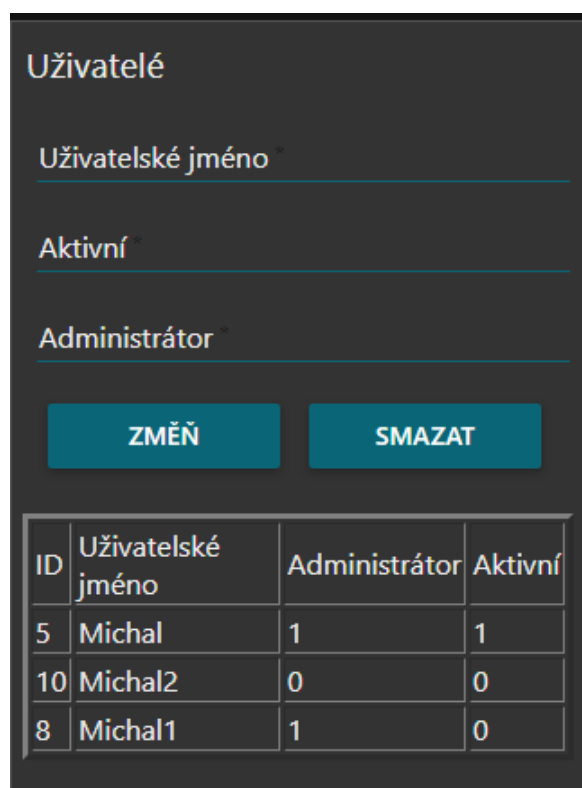
Po přihlášení se zobrazí hlavní panel (obrázek 13.6) pro řízení nemocničního lůžka. Panel se skládá ze čtyř sekcí:

- Aktualizace senz. hodnot
- Nastavené polohy lůžka
- Prevence proleženin
- Nastavit výchozí pozici

Aktualizace senz. hodnot	Nastavené polohy lůžka	Prevence proleženin	Nastavit výchozí pozici
Zadej ID lůžka	Zadej ID lůžka	Zadej ID lůžka	Zadej ID lůžka
<input type="button" value="SPUSTIT"/> <input type="button" value="SMAZAT"/>	Zadej počet cyklů	Zadej čas kontroly [min]	<input type="button" value="SPUSTIT"/> <input type="button" value="SMAZAT"/>
	<input type="button" value="SPUSTIT"/> <input type="button" value="SMAZAT"/>	<input type="button" value="SPUSTIT"/> <input type="button" value="SMAZAT"/>	
Aktuální hodnoty	Aktuální hodnoty	Aktuální hodnoty	
ID lůžka:	ID lůžka:	ID lůžka:	
	Počet cyklů:	Čas kontroly [min]:	
ZASTAVIT AKTUALIZACE SENZ. HODNOT	ZASTAVIT NASTAVENÉ POLOHY LŮŽKA	ZASTAVIT PREVENCI PROLEŽENIN	

Obrázek 13.5: Ovládání nemocničního lůžka.

V případě administrátorského rozhraní obsahuje hlavní panel navíc sekci *Uživatelé*, kde administrátor má možnost správy uživatelů.



Uživatelé

Uživatelské jméno

Aktivní

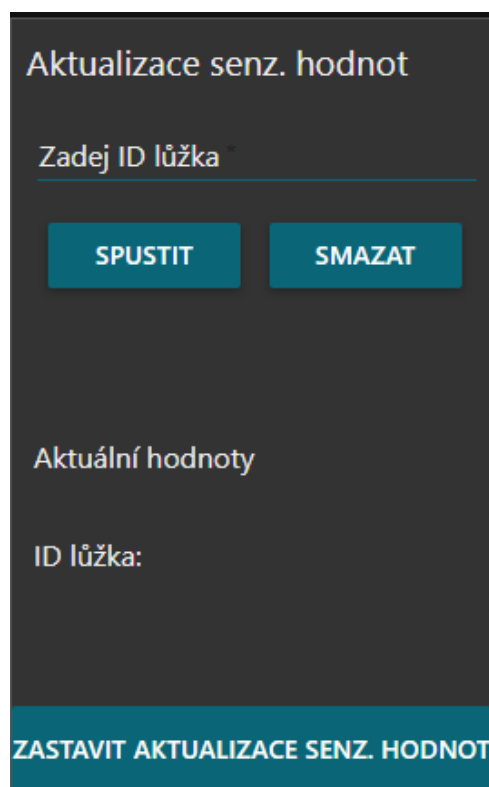
Administrátor

ZMĚŇ SMAZAT

ID	Uživatelské jméno	Administrátor	Aktivní
5	Michal	1	1
10	Michal2	0	0
8	Michal1	1	0

Obrázek 13.6: Sekce uživatelé v administrátorském rozhraní.

Část webové aplikace *Aktualizace senz. hodnot* slouží k ovládní ukládání sensorových hodnot přímo do databáze. V Database je nutné zadat parametr *ID lůžka*. Po stisku tlačítka "SPUSTIT" je spuštěna funkce Komunikačního protokolu, konkrétně funkce 11.4. Parametr je vkládán do databáze spolu s hodnotami sensorů, které slouží pro správné detekování další části Komunikačního protokolu (kapitola 11). Pro ukončení snímání sensorů je tlačítko "ZASTAVIT AKTUALIZACE SENZ. HODNOT", které ukončí tuto část.



Aktualizace senz. hodnot

Zadej ID lůžka

SPUSTIT SMAZAT

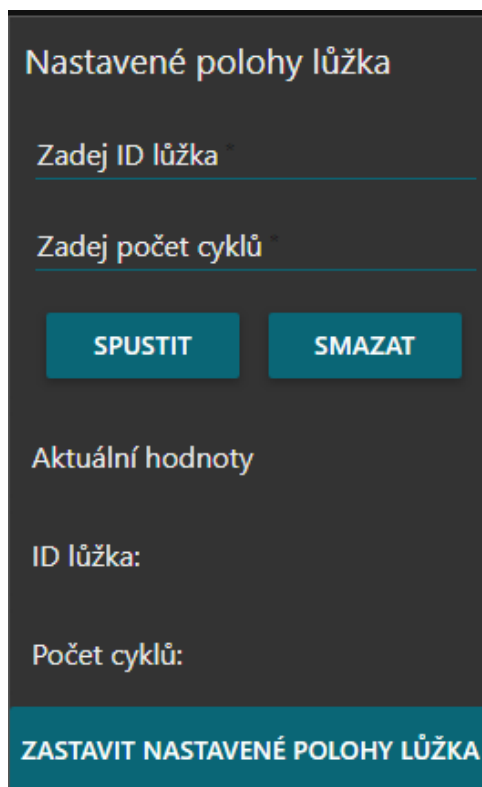
Aktuální hodnoty

ID lůžka:

ZASTAVIT AKTUALIZACE SENZ. HODNOT

Obrázek 13.7: Ovládací část Aktualizace senz. hodnot.

Nastavené polohy lůžka je další možnost ovládnání Komunikačního protokolu přes webovou aplikaci. Tato část umožňuje uživateli spustit automatické polohování pro konkrétní lůžko. Polohy, které budou provedené, se nastavují pomocí listu *Nastavené polohy lůžka* (podkapitola 10.2) v excelovém souboru (kapitola 10). Následně uživatel vyplní parametr *ID lůžka* a *počet cyklů*. Po vyplnění parametrů a stisku tlačítka "SPUSTIT" dojde ke spuštění Komunikačního protokolu, konkrétně části 11.6. Pro zastavení automatického polohování je tlačítko "ZASTAVIT NASTAVENÉ POLOHY LŮŽKA", které zastaví proces.



Nastavené polohy lůžka

Zadej ID lůžka

Zadej počet cyklů

SPUSTIT SMAZAT

Aktuální hodnoty

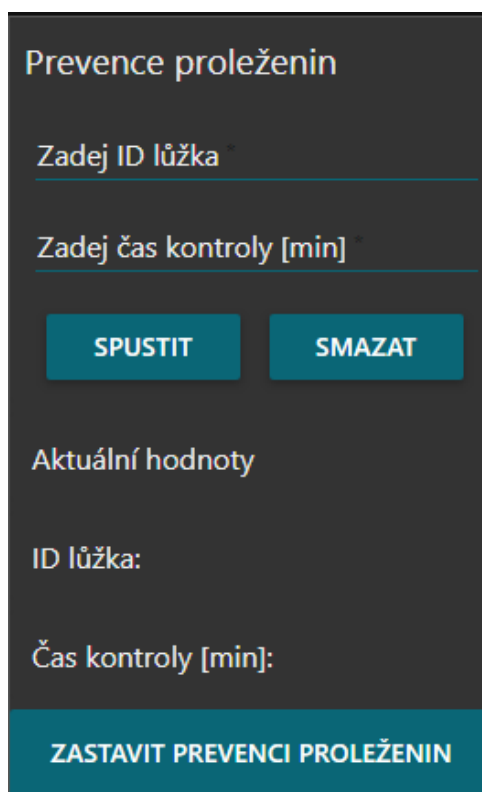
ID lůžka:

Počet cyklů:

ZASTAVIT NASTAVENÉ POLOHY LŮŽKA

Obrázek 13.8: Ovládací část Nastavené polohy lůžka.

Jako možnost ochrany proti proleženinám je zde ovládání *Prevence proleženin*. V této části se Komunikační protokol zaměřuje na detekci dlouhodobého zatížení textilních senzorů. Zde je třeba vyplnit parametry *ID lůžka* a *Čas kontroly [min]*. Pro spuštění je třeba stisknout tlačítko "SPUSTIT", které spouští část 11.6 v Komunikačním protokolu. Pro vypnutí ochrany proti proleženinám je tlačítko "ZASTAVIT PREVENCI PROLEŽENIN", které zastaví Komunikační protokol.



Prevence proleženin

Zadej ID lůžka

Zadej čas kontroly [min]

SPUSTIT SMAZAT

Aktuální hodnoty

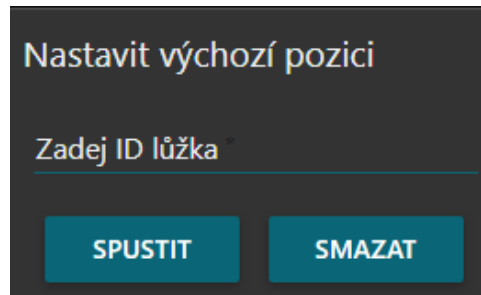
ID lůžka:

Čas kontroly [min]:

ZASTAVIT PREVENCI PROLEŽENIN

Obrázek 13.9: Ovládací část Prevence proleženin.

Z důvodu nemožnosti detekce manuálního ovládání zde byla přidána aplikační část pro uvedení do základní polohy s názvem *Nastavit výchozí pozici*. Zde je nutné vyplnit parametr *ID lůžka*, které udává konkrétní lůžko. Po kliknutí na tlačítko "SPUSTIT" provede pomocí části 11.7 v Komunikačním protokolu (kapitola 11). Tlačítko "SMAZAT" zde vymaže hodnotu zadanou v poli *ID lůžka*.



Obrázek 13.10: Ovládací část Nastavit výchozí pozici.

13.2 Vizualizace

Pro vizualizaci zatížení textilních senzorů je využita platforma QT. Nezobrazuje pouze zatížení text. senzorů, ale také polohu člověka na lůžku pomocí kamery, která je umístěna nad ním. Zobrazení a software pro zobrazení na platformě QT vytvořil Ing. Pavel Hnyk v rámci řešení projektu *LTE117005 Technologie Auto-ID a Internetu věcí pro zvýšení kvality zdravotnických služeb programu Inter-Eureka* za podpory RVVI [6], která sdružuje projekty ČR.

Software je aktuálně nainstalován na operačním systému (OS) Windows 10. Z důvodu problémů při převodu vizualizace na OS Raspbian bylo toto řešení ponecháno z časových důvodů. V budoucnu by bylo vhodné, aby proběhl přesun na OS Raspbian.

Ukázka vizualizace je zobrazena na obrázcích 13.11 a 13.12.



Obrázek 13.11: Ukázka Real-time zobrazení 1.



Obrázek 13.12: Ukázka Real-time zobrazení 2.

Kapitola 14

Testování

Testování probíhalo postupným navrstvováním jednotlivých částí. První část testování je zaměřeno na propojení a funkčnost hardwaru. Druhá část je testování softwaru, která byla rozdělena na následující podčásti:

- Komunikace mezi KJ a Centrální výpočetní jednotkou
- Napojení Centrální výpočetní jednotky na Databázi
- Komunikace Centrální výpočetní jednotky s nemocničním lůžkem skrze Bluetooth
- Ovládání Excelovým souborem
- Funkčnost webové aplikace
- Spouštění komunikačního protokolu pomocí webové aplikace
- Správnost zobrazení vizualizace

Byla vytvořena legenda závažností problému:

1. na problému závisí funkčnost aplikace
2. problém omezuje funkčnost aplikace
3. problém zásadně neomezuje funkčnost aplikace

14.1 Komunikace mezi KJ a Centrální výpočetní jednotkou

Testování 1.0 proběhlo dne 17. 12. 2019.

ID: 1

TEST: Python - spuštění KJ

Konkretizace: První start

Výsledek testu: V průběhu prvního startu se po připojení napájení objeví chybová hláška.

Poznámka: Problém závažnosti 2.

ID: 2

TEST: Python - USB

Konkretizace: Připojení KJ skrze USB.

Výsledek testu: Bez chyb.

ID: 3

TEST: Python - Bluetooth

Konkretizace: Připojení KJ pomocí Bluetooth

Výsledek testu: Nelze se v Pythonu připojit na datalogger senzoru.

Poznámka: Problém závažnosti 2

ID: 4

TEST: Python - Výpisy

Konkretizace: Výpisy z KJ

Výsledek testu: Výpis se nadbytečným počtem znaků.

Poznámka: Problém závažnosti 3.

Testování 1.1 proběhlo dne 13. 2. 2020.

ID: 5

TEST: Python - spuštění KJ

Konkretizace: První start

Výsledek testu: Test proběhl bez problémů.

Poznámka: Chyba z testování 1.0 byla opravena.

ID: 6

TEST: Python - Bluetooth

Konkretizace: Připojení KJ pomocí Bluetooth.

Výsledek testu: Nelze se v Pythonu připojit na datalogger senzoru.

Poznámka: Problém závažnosti 2, tento problém nevyřešen.

ID: 7

TEST: Python - Výpisy

Konkretizace: Opětovné vyzkoušení výpisu z KJ.

Výsledek testu: Test proběhl bez problému.

14.2 Napojení Centrální výpočetní jednotky na Databázi

Testování 1.0 proběhlo dne 5. 3. 2020.

ID: 8

TEST: Přístup do DB softwarem PgAdmin4

Konkretizace: Připojení pomocí hlavního přístupového heslo do DB.

Výsledek testu: Test proběhl bez problému.

ID: 9

TEST: Python - připojení do DB

Konkretizace: Připojení do databáze skrze knihovnu Psycopg2 v Python.

Výsledek testu: Připojení proběhlo bez problémů pomocí uživatelského účtu.

ID: 10

TEST: Výpis z tabulek DB

Konkretizace: Vypisování hodnot z DB skrze Python.

Výsledek testu: Test proběhl bez problému.

ID: 11

TEST: Zápis do tabulek DB

Konkretizace: Zapisování hodnot do DB skrze Python.

Výsledek testu: Test proběhl bez problému.

■ 14.3 Komunikace Centrální výpočetní jednotky s nemocničním lůžkem skrze Bluetooth

Testování 1.0 proběhlo dne 5. 3. 2020.

ID: 12

TEST: Připojení k lůžku

Konkretizace: Vytvoření propojení Centrální výpočetní jednotky (kapitola 5) a Nemocničního lůžka (kapitola 2) skrze Bluetooth soket knihovnou PyBluez v Pythonu.

Výsledek testu: Zařízení úspěšně propojena a soket vytvořen.

ID: 13

TEST: Zaslání příkazu

Konkretizace: Zaslání příkazu Nemocničnímu lůžku knihovnou PyBluez skrze již vytvořený soket v Pythonu.

Výsledek testu: Zařízení úspěšně dostalo pokyn - testováno na tlačítku "GO".

ID: 14

TEST: Čtení příkazů

Konkretizace: Čtení aktualně použitých příkazů hodnot z lůžka pomocí Bluetooth soketu v Pythonu.

Výsledek testu: Nelze číst aktuálně použité příkazy spouštěné na lůžku. Problém závažnosti 2.

Poznámka: Tento problém nevyřešen.

Řešení Tento problém vyřešen naprogramováním výchozí pozice.

14.4 Ovládání Excelovým souborem

Testování 1.0 proběhlo dne 5. 3. 2020.

ID: 15

TEST: Načtení excelové souboru

Konkretizace: Načtení souboru knihovnou pandas v Pythonu.

Výsledek testu: Načtení proběhlo úspěšně.

ID: 16

TEST: Čtení příkazů z excelu

Konkretizace: Čtení příkazů řádků z excelu knihovnou pandas v Pythonu.

Výsledek testu: Čtení proběhlo úspěšně.

ID: 17

TEST: Kontrola parametrů

Konkretizace: Porovnání správnosti parametrů a vybírání pouze řádků, které souvisejí s daným *ID lůžka*.

Výsledek testu: Porovnání funguje správně.

14.5 Funkčnost webové aplikace

Testování 1.1 proběhlo dne 7. 5. 2020.

ID: 18

TEST: Registrace

Konkretizace: Po zadání uživatelského jména a hesla se vytvoří uživatelský profil.

Výsledek testu: Vytvoření proběhlo úspěšně a uživatel byl přidán do DB.

ID: 19

TEST: Přihlášení

Konkretizace: Uživatel se uživ. jménem a heslem přihlásí do aplikace.

Výsledek testu: Testo proběhl úspěšně. Po zadání správných přihlašovacích údajů byl uživatel přihlášen. Při špatném jménu nebo heslu je zobrazena chybová hláška.

ID: 20

TEST: Kontrola parametrů

Konkretizace: Porovnání správnosti parametrů a vybírání pouze řádků, které souvisejí s daným *ID lůžka*.

Výsledek testu: Porovnání funguje správně.

14.5.1 Sekce "Aktualizace senz. hodnot"

ID: 21

TEST: Spuštění sekce

Konkretizace: Spuštění části Komunikačního protokolu pro sekci Aktualizace senz. hodnot.

Výsledek testu: Test proběhl úspěšně. Sekce, po zadání parametrů, spustí část Komunikačního protokolu pro ukládání dat do DB pomocí tlačítka "SPUSTIT".

ID: 22

TEST: Zobrazení aktuálních hodnot

Konkretizace: Po startu sekce se mají zobrazit aktuálně zadané parametrické hodnoty.

Výsledek testu: Test proběhl úspěšně. Hodnoty po zmáčknutí tlačítka "SPUSTIT" zobrazeny.

ID: 23

TEST: Okamžité ukončení sekce

Konkretizace: Okamžité ukončení části Komunikačního protokolu pro sekci Aktualizace senz. hodnot.

Výsledek testu: Test proběhl úspěšně. Proces ukládání hodnot do DB byl ukončen pomocí tlačítka "ZASTAVIT AKTUALIZACE SENZ. HODNOT".

■ 14.5.2 Sekce "Nastavené polohy lůžka"

ID: 24

TEST: Spuštění sekce

Konkretizace: Spuštění části Komunikačního protokolu pro sekci Nastavené polohy lůžka.

Výsledek testu: Test proběhl úspěšně. Sekce, po zadání parametrů se kliknutím na tlačítko "SPUSTIT" spustí část Komunikačního protokolu pro získávání poloh z excelové souboru a začne polohovat lůžko.

ID: 25

TEST: Zobrazení aktuálních hodnot

Konkretizace: Po startu sekce se mají zobrazit aktuálně zadané parametrické hodnoty.

Výsledek testu: Test proběhl úspěšně. Hodnoty po zmáčknutí tlačítka "SPUSTIT" zobrazeny.

ID: 26

TEST: Okamžité ukončení sekce

Konkretizace: Okamžité ukončení části Komunikačního protokolu pro sekci Nastavené polohy lůžka.

Výsledek testu: Test proběhl úspěšně. Proces polohování byl ukončen pomocí tlačítka "ZASTAVIT NASTAVENÉ POLOHY LŮŽKA".

■ 14.5.3 Sekce "Prevence proleženin"

ID: 27

TEST: Spuštění sekce

Konkretizace: Spuštění části Komunikačního protokolu pro sekci Prevence proleženin.

Výsledek testu: Test proběhl úspěšně. Sekce, po zadání parametrů, spustí část Komunikačního protokolu pro kontrolu zatížení senzorů pomocí tlačítka "SPUSTIT".

ID: 28

TEST: Zobrazení aktuálních hodnot

Konkretizace: Po startu sekce se mají zobrazit aktuálně zadané parametrické hodnoty.

Výsledek testu: Test proběhl úspěšně. Hodnoty po zmáčknutí tlačítka "SPUSTIT"zobrazeny.

ID: 29

TEST: Okamžité ukončení sekce

Konkretizace: Okamžité ukončení části Komunikačního protokolu pro sekci Prevence proleženin.

Výsledek testu: Test proběhl úspěšně. Proces kontroly zatížení senzorů byl ukončen pomocí tlačítka "ZASTAVIT KONTROLU PROLEŽENIN".

■ 14.5.4 Sekce "Nastavit výchozí pozici"

ID: 30

TEST: Spuštění sekce

Konkretizace: Spuštění části Komunikačního protokolu pro sekci Nastavit výchozí pozici.

Výsledek testu: Test proběhl úspěšně. Sekce, po zadání parametru, spustí část Komunikačního protokolu pro nastavení výchozí pozice pomocí tlačítka "SPUSTIT".

■ 14.6 Správnost zobrazení vizualizace

Testování 1.1 proběhlo dne 14. 5. 2020.

ID: 31

TEST: Spuštění vizualizace

Konkretizace: Spuštění real-time vizualizace na QT platformě.

Výsledek testu: Test proběhl bez problémů.

ID: 32

TEST: Zastavení vizualizace

Konkretizace: Zastavení zobrazení vizualizace.

Výsledek testu: Test proběhl bez problémů.

ID: 33

TEST: Zobrazení zatíženosti senzoru

Konkretizace: Test pro zobrazení zatíženosti senzoru v aktuálním čase.

Výsledek testu: Po připojení senzorů k platformě test proběhl úspěšně.

ID: 34

TEST: Zobrazení pacienta na kameře

Konkretizace: Test pro nastavení kamery snímající pacienta.

Výsledek testu: Test proběhl úspěšně.

Kapitola 15

Závěr

Cílem této práce bylo navrhnout a zrealizovat systém pro automatické polohování nemocničního lůžka na základě dat získaných ze zatížení textilní sensorové sítě. Ta monitoruje rozložení hmotnosti pacienta ležícího na lůžku. Tímto řešením je možné zabránit zdravotním komplikacím (například proleženin, zahleňování apod.).

Prvně bylo třeba se seznámit s hardwarem. Dále následoval výběr vhodných prostředků pro tvorbu SW na základě kritérií. Poté vytvořit návrh řešení a schéma propojení.

Textilní senzory jsou s Kontrolní jednotkou textilního senzoru zatížení (KJ) propojeny metalickým vedením. KJ jednotka je propojená s centrální výpočetní jednotkou kabelem micro-USB-USB-A, kterým je KJ napájena a probíhá komunikace mezi jednotkami. Centrální výpočetní jednotka je napájena ze sítě a data, které získá z KJ ukládá Komunikačním protokolem (kapitola 11) do databáze.

Další její funkcí je komunikace s nemocničním lůžkem při využití Bluetooth rozhraní. Na centrální výpočetní jednotce je uložený soubor s příponou xls pro uživatelské vstupy. Je zde také implementována webová aplikace, která ovládá komunikaci skrze Komunikační protokol.

Pro propojení komponent v Komunikačním protokolu bylo nejlepší využít programovacího jazyku Python. Daný protokol zajišťuje komunikaci mezi jednotlivými částmi návrhu (např. řízení nemocničního lůžka centrální vý-

početní jednotkou). K vytvoření databáze byl využit PostgreSQL (kapitola 12). V softwaru Node-RED byla navržena webová aplikace (kapitola 13). Real-time vizualizace zatížení textilních senzorů byla vytvořena Ing. Pavlem Hnykem na QT platformě. Tato aplikace bohužel aktuálně běží pouze na Windows 10, tím pádem není spuštěna na centrální výpočetní jednotce jako například webová aplikace, kvůli problému s konvertováním na OS Raspbian. Zmíněný problém lze v budoucnu vyřešit.

Návrh řešení byl v průběhu i po implementaci testován (kapitola 14) a upravován v případě nalezení vhodnějšího východiska. Testované nejdříve bylo metalické propojení a následně byla testována softwarová sekce, ve které je Komunikační protokol, databáze, soubor s příponou xls, webová aplikace a vizualizace.

V této práci jsou i části, které by bylo vhodné v budoucnu upravit. Mezi ně patří například již zmiňovaný přesun QT aplikace na Centrální výpočetní jednotku a také úprava webová aplikace.



Literatura

- [1] Zdeněk Šubčík. Ovládání nemocničního lůžka pomocí mobilní aplikace, 2016.
- [2] Martin Vlašimský. Velkoplošný textilní senzor pro snímání plochy zatížené lidským tělem, 2018.
- [3] Pavel Hnyk, Lukáš Vojtěch, and Marek Neruda. LPWAN IOT router prototyp, 2020.
- [4] Python 2 vs 3. Citováno dne: [3. 4. 2020], <https://pythonfundu.blogspot.com/2019/02/python-2-vs-3.html>, 2019.
- [5] Node-RED. Citováno dne: [3. 4. 2020], <https://nodered.org>.
- [6] Informační systém výzkumu, experimentálního vývoje a inovací. Citováno dne: [13. 5. 2020], www.rvvi.cz.
- [7] How to Geek. Everything You Need to Know About Getting Started with the Raspberry Pi. Citováno dne: [4. 4. 2020], <https://www.howtogeek.com/138281/the-htg-guide-to-getting-started-with-raspberry-pi/>.
- [8] Official Python web. Citováno dne: [3. 4. 2020], <https://www.python.org/doc/essays/blurb/>.
- [9] Pavel Kosina and Zdeněk Lehocký. Python - popis jazyka. Citováno dne: [3. 4. 2020], <http://programujte.com/clanek/1970010106-python-popis-jazyka/>, 2005.

- [10] Python History and Versions. Citováno dne: [3. 4. 2020],
<https://www.javatpoint.com/python-history>.
- [11] Postgresql about. Citováno dne: [8. 5. 2020],
<https://www.postgresql.org/about/>.



Seznam zkratek

Zkratka	Celý název
CPR	Cardiopulmonary resuscitation
DB	Databáze
HW	Hardware
ID	Identification
IoT	Internet of Things
KJ	Kontrolní jednotka textilního senzoru zatížení
LPWAN	Low Power Wide Area Network
MQTT	Message Queuing Telemetry Transport
OS	Operační systém
RVVI	Rada pro výzkum, vývoj a inovace
SoC	Systems On a Chip
SQL	Structured Query Language
SW	Software
TCP/IP	Transmission Control Protocol/Internet Protocol
USB	Universal Serial Bus