**Ovládací software pro nezávislé ovládání pěti pneumatických svalů v robotické struktuře**

**Control software for independent control of five pneumatic muscles in a robotic structure**

Bakalářská práce

Studijní program: Informatika a kybernetika ve zdravotnictví

Studijní obor: Biomedicínská informatika

Autor bakalářské práce: Marek Darsa

Vedoucí bakalářské práce: Ing. Martin Bejtic

**Zadání práce**

Místo této stránky umístěte zadání bakalářské práce, které si stáhnete z databáze Projects ve formátu PDF. Zadání je nezbytnou součástí nerozebíratelné vazby závěrečné práce. Na zadání je uvedena platnost – relevantní jsou pouze platná zadání závěrečných prací. Při hodnocení práce je posuzována míra splnění zadání.

**PROHLÁŠENÍ**

Prohlašuji, že jsem bakalářskou práci s názvem „Ovládací software pro nezávislé ovládání pěti pneumatických svalů v robotické struktuře“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci. Prohlašuji, že tato práce je přesnou kopií odevzdané elektronické verze.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona   
č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně datum …...….………...………………...

Marek Darsa

**PODĚKOVÁNÍ**

Rád/a bych poděkoval/a…

Poděkování je nepovinné, ale obvyklé. Vedoucímu práce se zpravidla děkuje, oponentovi zásadně ne. Poraďte se s vedoucím práce, zda by nebylo vhodné uvést v poděkování číslo grantu, ze kterého byla práce podpořena.

**ABSTRAKT**

**Ovládací software pro nezávislé ovládání pěti pneumatických svalů v robotické struktuře**

Tato bakalářská práce se zabývá vývojem ovládacího softwaru pro pneumatickou soustavu s pěti svaly. Cílem práce bylo vytvořit sofistikovaný a uživatelsky přívětivý software umožňující ovládání a kalibraci svalů s důrazem na přesnost, bezpečnost a adaptabilitu systému. Software byl vyvinut v programovacím jazyce Python a disponuje moduly pro správu uživatelských rolí, nastavení hodnot na technickou nulu a integraci bezpečnostních omezení, jako je validace vstupů a omezení maximálního tlaku v pneumatických svalech. Funkce softwaru zahrnují přihlašování uživatelů, individuální ovládání svalů, nastavení bezpečnostních limitů a transformaci požadovaných pohybů na odpovídající tlaky v různých jednotkách. Mezi tyto jednotky patří, jednotky krokových motorů kroky, tlakové jednotky mbar a také mV. Vedle implementace softwaru je klíčovou součástí práce také tvorba podrobné dokumentace, která usnadní uživatelům orientaci v systému a jeho možnostech. Softwarové řešení představuje kombinaci přívětivého uživatelského rozhraní a robustních funkcí, umožňující efektivní ovládání pneumatické soustavy v robotických aplikacích.

**Klíčová slova**

Pneumatické svaly

Ovládací software

Robotická struktura

Uživatelská dokumentace

**ABSTRACT**

**Control software for independent control of five pneumatic muscles in a robotic structure.**

This bachelor thesis deals with the development of control software for a pneumatic system with five muscles. The aim of the thesis was to develop sophisticated and user-friendly software that allows control and calibration of the muscles with emphasis on accuracy, safety, and adaptability of the system. The software was developed in the Python programming language and has modules for managing user roles, setting values to technical zero and integrating safety constraints such as input validation and maximum pressure limitations in the pneumatic muscles. Software features include user login, individual muscle control, setting safety limits, and transforming desired movements into corresponding pressures in different units. These units include stepper motor step units, mbar pressure units and mV. In addition to the software implementation, a key part of the work is the creation of detailed documentation to help users navigate the system and its capabilities. The software solution is a combination of a user-friendly interface and robust features, enabling efficient control of the pneumatic system in robotic applications.

Translated with DeepL.com (free version).

**Keywords**

Pneumatic muscles

Control software

Robotic structure

User documentation

Obsah

[Seznam symbolů a zkratek 5](#_Toc162001577)

[1 Úvod 6](#_Toc162001578)

[2 Přehled současného stavu 7](#_Toc162001579)

[2.1 Motivace pro vznik projektu 7](#_Toc162001580)

[2.1.1 Základní principy intubačních přístrojů 7](#_Toc162001581)

[2.1.2 Principy magnetické rezonance 7](#_Toc162001582)

[2.2 Současná pneumatická soustava a její fungování 7](#_Toc162001583)

[2.2.1 Komponenty pneumatické soustavy 7](#_Toc162001584)

[2.2.2 Operační systém pro řízení soustavy 7](#_Toc162001585)

[2.3 Pneumatické svaly a jejich typy 7](#_Toc162001586)

[2.3.1 Popis různých typů pneumatických svalů 7](#_Toc162001587)

[2.3.2 Principy práce pneumatických svalů 7](#_Toc162001588)

[2.4 Softwarové a robotické aspekty 7](#_Toc162001589)

[2.4.1 Vývoj softwaru pro ovládání pneumatických svalů 7](#_Toc162001590)

[2.4.2 Integrace pneumatických svalů do robotických struktur 7](#_Toc162001591)

[2.5 Python 7](#_Toc162001592)

[2.5.1 Výhody a nevýhody oproti ostatním programovacím jazykům 7](#_Toc162001593)

[2.5.2 Proč jsem si ho vůbec vybral a proč se předešlá část soustavy taky programovala v pythonu 7](#_Toc162001594)

[2.6 SQL 8](#_Toc162001595)

[2.7 UML 8](#_Toc162001596)

[3 Cíle práce 9](#_Toc162001597)

[4 Návrh aplikace 11](#_Toc162001598)

[4.1 Funkční specifikace 11](#_Toc162001599)

[4.1.1 Knihovna pro GUI 11](#_Toc162001600)

[4.1.2 Databázová knihovna 15](#_Toc162001601)

[5 Implementace 21](#_Toc162001602)

[5.1 Pneumatická soustava 21](#_Toc162001603)

[5.2 DeviceLoader 21](#_Toc162001604)

[5.2.1 getBoardx() 22](#_Toc162001605)

[5.3 Main 22](#_Toc162001606)

[5.4 AdminWindow 25](#_Toc162001607)

[5.5 UserWindow 30](#_Toc162001608)

[6 Uživatelská dokumentace 34](#_Toc162001609)

[7 Výsledky (Testování) 35](#_Toc162001610)

[8 Diskuse 36](#_Toc162001611)

[9 Závěr 37](#_Toc162001612)

[Seznam použité literatury 38](#_Toc162001613)

[Příloha A: Požadavky na formátování práce 40](#_Toc162001614)

[Příloha B: Základní typografické zásady 40](#_Toc162001615)

[Příloha C: Další doporučení pro přehlednost textu 40](#_Toc162001616)

[Příloha D: Obsah přiloženého CD 40](#_Toc162001617)

# Seznam symbolů a zkratek

#### Seznam symbolů

|  |  |  |
| --- | --- | --- |
| Symbol | Jednotka | Význam |
| *CDP*end | cmH2O | Střední distenzní tlak po ustálení přechodového děje |
|  |
|  |  |  |

#### Seznam zkratek

|  |  |
| --- | --- |
| Zkratka | Význam |
| SW | Software (Program) |
| GUI  DB  SQL  CMD  MVCC  CLI | Graphical User Interface (Grafické uživatelské rozhraní)  Database (Databáze)  Structured Query Language (Strukturovaný dotazovací jazyk)  Command Prompt (Příkazový řádek)  Multi-Version Concurrency Control (Kontrola konkurence více verzí)  Command Line Interface (Rozhraní příkazového řádku) |

# Úvod

Tato práce vychází z rozsáhlejšího projektu, který se soustředí na zabezpečení poskytování kyslíkové terapie pacientům trpícím respiračními obtížemi nebo pacientům v kómatu, u nichž je vyšetřování prováděno zařízeními, která vyžadují absenci feromagnetických částic. Aby nedošlo k blokaci dýchacích cest a případnému udušení v důsledku nedostatečného zásobování kyslíkem během této procedury je nutnost připojení pacienta k nějakému typu přístroje, který je schopen dodávat pacientovy kyslík a zároveň zajistí že nedojde k blokaci jeho dýchacích cest. Vzhledem k přítomnosti feromagnetických částic v tradičních intubačních zařízeních je jejich použití v prostředí magnetické rezonance nebezpečné a nevhodné. Proto jsou pacienti v takových případech připojeni pouze kyslíkovou terapií pomocí nosních kanyl, což však nese riziko obstrukce dýchacích cest. V současné době je problém řešen manuální manipulací hlavy pacienta s cílem dosáhnout optimální polohy pro zajištění průchodnosti dýchacích cest. Toto řešení, představuje významné bezpečnostní riziko, zejména z dlouhodobého hlediska v prostředí magnetické rezonance, což přináší nebezpečí spojené s dlouhodobým vystavením tomuto záření.

Projekt, na který navazuje tato bakalářská práce, se věnuje primárně mechanické stránce problému, avšak postrádá softwarovou implementaci. Předchozí pneumatický systém, se kterým tato práce pracuje, umožňuje pouze otáčení krokového motoru na základě zadané funkce a počtu kroků. Tímto způsobem se motor posouvá v určeném směru a interaguje s ventilem, který reguluje tlak v systému. Takovéto řešení ovšem není dostatečně přesné, neboť nastavení tlaku v systému je možné pouze prostřednictvím neintuitivních jednotek krokového motoru, což může být pro obsluhujícího lékaře obtížné a neefektivní. Navíc by bylo nutné ovládání softwaru provádět přímo z vývojového prostředí, což by znamenalo velkou možnost chyby. Nedostatek pokročilých funkcí v softwaru by omezoval efektivitu systému, neboť uživatel by musel provádět všechny činnosti ručně a opakovaně.

Zde prezentovaná práce představuje kompletní softwarové řešení pro eliminaci nedostatků současného systému. Zahrnuje vytvoření grafického uživatelského rozhraní s využitím moderních knihoven, které umožňuje ovládání až pěti pneumatických svalů současně, a to s možností volby různých jednotek pro ovládání. Kromě toho je implementována funkce kalibrace soustavy a tvorba nových převodních vzorců. Důležitou součástí je také přístup do databáze obsahující veškeré potřebné informace pro správný provoz systému, včetně uvedených převodních vzorců a informací o aktuálně nastaveném výchozím převodním vzorci.

# Přehled současného stavu

## Motivace pro vznik projektu

### Základní principy intubačních přístrojů

### Principy magnetické rezonance

#### Vliv magnetické rezonance na lidské tělo

#### Bezpečnostní opatření při práci s magnetickou rezonancí

## Současná pneumatická soustava a její fungování

### Komponenty pneumatické soustavy

#### Popis jednotlivých komponent soustavy

### Operační systém pro řízení soustavy

## Pneumatické svaly a jejich typy

### Popis různých typů pneumatických svalů

### Principy práce pneumatických svalů

## Softwarové a robotické aspekty

### Vývoj softwaru pro ovládání pneumatických svalů

### Integrace pneumatických svalů do robotických struktur

## Python

### Výhody a nevýhody oproti ostatním programovacím jazykům

### Proč jsem si ho vůbec vybral a proč se předešlá část soustavy taky programovala v pythonu

## SQL

## UML

# Cíle práce

Hlavním cílem této práce bude vytvořit sofistikovaný a uživatelsky přívětivý software, který bude sloužit k ovládání a kalibraci pneumatických svalů, přičemž důraz bude kladen na přesnost, bezpečnost a adaptabilitu systému. Tento jeden hlavní cíl se skládal z mnoha malých podcílů.

Vyhledat knihovnu pro tvorbu grafického uživatelského rozhraní:

* Primárním kritériem je moderní estetika. Knihovna by měla nabízet moderně vypadající prvky a možnost konfigurace barevného schématu.
* Preferovaná knihovna by měla být jednoduchá a nenadbytečně komplexní. Měla by se jednat spíše o kolekci widgetů a nástrojů pro rychlé vytvoření uživatelského rozhraní než o rozsáhlý framework optimalizovaný pro vytváření grafiky pro hry.

Vyhledat vhodnou databázi pro ukládání dat:

* Lokálnost je primárním požadavkem na databázi. Systém by měl být schopen fungovat i bez přístupu k internetu, což vyžaduje zvážení databázových systémů pracujících na principu klient-server. Proto je vhodné zvolit databázi, která ukládá své záznamy lokálně.
* Databáze musí být schopna provádět základní operace pomocí SQL jazyka nebo jeho nadstaveb. Vybraný databázový systém by neměl být příliš komplexní a měl by být schopen základního odkazování mezi tabulkami pomocí vedlejších klíčů.

Implementovat GUI aplikaci pomocí vybrané knihovny:

* Navržení jednoduchého a intuitivního rozhraní komponent pro snadnou práci uživatele.
* Implementace mechanismu automatické validace vstupů na základě definovaných pravidel.
* Přidání možnosti potvrzování vstupů pomocí klávesy "Enter" na vhodných místech uživatelského rozhraní.
* Rozšíření funkcionalit tak, aby uživatel mohl interagovat s aplikací nejen prostřednictvím tlačítek, ale také klávesovými zkratkami.

Implementovat autentizaci a oprávnění uživatelů:

* Uživatelé se dostali do uživatelské části po zadání uživatelského hesla a administrátoři se dostali do administrátorské části po zadání administrátorského hesla.
* Implementovat do systému autentizace hashovací funkci, která zabezpečení integritu dat v databázi v případě hackerského útoku.

Kalibrace pneumatických svalů:

* Navrhnout a implementovat funkce pro kalibraci pneumatických svalů pro administrátory.
* Zabezpečit ukládání výsledků kalibrace, ve formátu .xlsx na domovskou obrazovku, pro další analýzu.

GUI pro správu databáze a převodních vzorců:

* Navrhnout uživatelské rozhraní pro správu databáze, které zobrazí hodnoty převodních vzorců pro každý sval.
* Implementovat možnost úpravy převodních vzorců a volbu aktuálního vzorce pro použití v aplikaci.

# Návrh aplikace

## Funkční specifikace

### Knihovna pro GUI

Programovací jazyk Python obsahuje jednu integrovanou knihovnou pro grafické uživatelské rozhraní ta je rovněž i nejpoužívanější a jedná se o knihovnu tkinter[1]. Dalšími velice populárními knihovnami jsou Kivy a PyQt. Tyto knihovny nalézají uplatnění zejména u uživatelů, kteří preferují modernější vzhled jejich uživatelského rozhraní. Nicméně, Tkinter zůstává stále nejpopulárnější volbou. Tento fakt je zčásti důsledkem skutečnosti, že uživatelé, kteří usilují o vytvoření výkonné a moderní okenní aplikace, často volí jiný programovací jazyk než Python. Proto se v Pythonu často využívá Tkinter pro jednoduché formy grafických uživatelských rozhraní.

#### Kivy

Kivy je knihovna pro programovací jazyk python. Tato knihovna je pod licencí open-source. ,,Jako open source se označují programy, jejichž zdrojový kód je volně přístupný široké veřejnosti‘‘[2]. Hlavní výhodou Kivy je její multiplatformnost, umožňující spouštění aplikací na různých operačních systémech včetně Windows, Linuxu, macOS, a také na mobilních zařízeních jako jsou Android a iOS. Kivy vyniká zejména v oblasti vývoje 2D her, kde poskytuje pokročilé možnosti tvorby simulací a grafiky. Samotná knihovna je napsaná v jazyce Python.[3]

Knihovna Kivy představuje mocný nástroj, vhodný pro mnohé účely, nicméně v případě tohoto konkrétního projektu je knihovna Kivi příliš komplexní. Ukázka GUI pomocí knihovny Kivy Obrázek 4.1.

A screenshot of a computer

Description automatically generated

Obrázek 4.1 Zdroj: <https://i.stack.imgur.com/y6Hmq.png>

GUI vytvořené pomocí knihovny Kivy

#### PyQt

PyQt představuje další významnou knihovnu pro tvorbu grafických uživatelských rozhraní v jazyce Python. Tato knihovna umožnuje využívat Qt aplikační framework. ,,Framework (Programovací rámec) jedná se o sadu hotových bloků. Programátorovy ulehčí práci především v tom, že nemusí své programy začínat vždy od nuly‘‘[4]. Qt framework je implementován v jazyce C++ a umožňuje psaní multiplatformních aplikací, které jsou kompatibilní s hlavními operačními systémy pro počítače i mobilní zařízení.

PyQt knihovna poskytuje výhodu v tom, že není nutné využívat celý Qt framework pro konkrétní projekt. Místo toho se může využít pouze grafické vlastnosti frameworku a případně kombinovat s jinými frameworky nebo pracovat bez nich.[5]

Knihovna PyQt je velice komplexní, a ještě komplexnější je, když se využívá přímo framework Qt. Ovšem v tomto projektu není takto komplexní framework potřeba proto se ani tato knihovna nevyužila. Ukázka grafického uživatelského rozhraní v pythonu pomocí knihovny PyQt Obrázek 4.2.

A screenshot of a computer

Description automatically generated

Obrázek 4.2 Zdroj: <https://i.imgur.com/QN8YUZW.png>

GUI vytvořené pomocí knihovny PyQt.

#### Tkinter a CustomTkinter

Tkinter je knihovna pro tvorbu grafických uživatelských rozhraní, která je zabudována přímo do distribuce programovacího jazyka Python. Tím pádem je tato knihovna automaticky dostupná při stažení Pythonu z oficiálních zdrojů a není nutné ji instalovat zvlášť. Pro programátora stačí pouze inicializovat Tkinter a může s ním ihned začít pracovat. Díky těmto vlastnostem je Tkinter zvláště populární mezi začátečníky a programátory, kteří upřednostňují jednoduchost před moderním vzhledem grafických uživatelských rozhraní, které nabízejí ostatní knihovny jako Kivy nebo PyQt.[6]. Ukázka GUI vytvořeného pomocí knihovny tkinter Obrázek 4.3.



Obrázek 4.3 Zdroj: <https://static.wixstatic.com/media/90b6f2_3041927f0a11481a995dde96b3e2b01d~mv2.jpg/v1/fill/w_703,h_547,al_c,q_90/90b6f2_3041927f0a11481a995dde96b3e2b01d~mv2.jpg>

GUI vyhotovené pomocí knihovny tkinter

Knihovna CustomTkinter je uživatelská nadstavba pro knihovnu tkinter. Tato knihovna byla vytvořena pod licencí MIT – jedná se o open-source knihovnu. Tuto knihovnu vytvořil programátor jménem Tom Schimansky. Tato knihovno vznikla v závislosti na tkinteru přebírá některé její vlastnosti a upravuje je do moderního vzhledu. Knihovny spolu mohou pracovat tudíž je možnost pracovat s oběma knihovnami zároveň a z každé si vzít nějaké její vlastnosti, což se ovšem i doporučuje v oficiální dokumentaci knihovny custumtkinter. Hlavními přednostmi této knihovny je například možnost vybrání si vlastního režimu ('světlý', 'tmavý'). Tento režim se dá nastavit v závislosti nastavení vašeho počítače nebo se dá i ručně měnit.[7] [8]

Pro všechny pozitivní vlastnosti se vybrala tato knihovna jako knihovna, pomocí které se bude vytvářet grafické uživatelské rozhraní a její nedostatky budou doplněny knihovnou tkinter. Ukázka moderního grafického uživatelské rozhraní vytvořeného pomocí knihovny CustomTkinter Obrázek 4.4.



Obrázek 4.1 Zdroj: <https://raw.githubusercontent.com/TomSchimansky/CustomTkinter/master/documentation_images/complex_example_dark_Windows.png>

GUI vyhotovené pomocí knihovny customtkinter

### Databázová knihovna

,,Databáze je organizovaný soubor strukturovaných informací neboli dat, které se obvykle ukládají v elektronické podobě v počítačovém systému‘‘[9].

Existuje velké množství různých databází a každá tato databáze má svoje vlastní vhodné využití jedním z nejkomplexnějších typem databázových systémů jsou distribuované databáze.

#### Distribuované databáze

Distribuované databáze představují významný typ databázových systémů, který je vhodný pro velké databázové systémy. Jednou z hlavních vlastností distribuovaných databází je možnost distribuce dat mezi více serverů umístěných na různých místech. Tato schopnost umožňuje efektivní zpracování a správu velkého objemu dat prostřednictvím víceuživatelské sítě, kde každý uzel může provádět operace na své části databáze.[9]

V rámci bakalářské práce nebyl použitý databázový systém z důvodu jeho značné komplexity a schopnosti pracovat s velkým objemem dat distribuovaných na více serverech. Požadavky na databázový systém v daném projektu se soustředily především na lokální databázi s omezeným počtem tabulek.

#### Databáze typu NoSQL

NoSQL databáze představují moderní přístup k ukládání a manipulaci s daty, který se odlišuje od tradičních relačních databází. Zkratka NoSQL znamená "Not only SQL", což naznačuje, že tyto databáze nejsou omezeny pouze na strukturovaný SQL jazyk. Jedním z hlavních rysů NoSQL databází je schopnost pracovat s nestrukturovanými daty a umožňovat ukládání různých typů informací, včetně binární reprezentace multimediálního obsahu, jako jsou obrázky, videa nebo zvukové stopy.

Díky této flexibilitě jsou NoSQL databáze ideální pro aplikace, které vyžadují škálovatelnost a možnost pracovat s různými typy obsahu. Například pro sociální sítě, e‑commerce platformy nebo systémy pro správu obsahu, kde je potřeba efektivně pracovat s velkým objemem dat a zároveň udržovat vysoký výkon systému.[9]

Tento databázový systém poskytuje optimální řešení pro aplikace vyžadující ukládání rozmanitých typů dat, včetně textových, číselných a multimediálních souborů, jako jsou obrázky nebo celé soubory. Nicméně, v rámci tohoto konkrétního projektu není nutné implementovat takovou rozsáhlou funkcionalitu. Z daných požadavků vyplývá, že databáze bude využívána primárně pro uchovávání textových a číselných dat.

#### Relační databáze:

Relační databáze představují jeden z nejrozšířenějších typů databází, který se opírá o strukturu tabulek s řádky a sloupci. Jejich klíčovým prvkem jsou primární klíče, které jednoznačně identifikují každý záznam v tabulce. Tento typ databáze je ideální pro ukládání strukturovaných dat s definovanými vztahy mezi nimi, což umožňuje efektivní správu informací v rámci organizace.

Jednou z klíčových výhod relačních databází je možnost propojování tabulek pomocí vedlejších klíčů. To znamená, že v jiné tabulce může být uložen například úplný profil pacienta včetně adresy, a na tuto adresu se může odkazovat pomocí vedlejšího klíče v tabulce, kde jsou uloženy informace o pacientech. Tento přístup umožňuje efektivní a flexibilní správu dat, jelikož adresu může mít jak pacient, tak i lékař, nebo může být sdílena mezi různými tabulkami v rámci databáze.[9]

Další výhodou relačních databází je schopnost jedné tabulky sloužit pro více účelů. Například, místo trvalého bydliště pacienta může být definováno jako trvalé bydliště, ale také jako přechodné bydliště, a tato informace může být využita ve více kontextech, což zvyšuje flexibilitu a použitelnost dat v databázi.

Tento databázový systém byl vybrán pro účely této bakalářské práce s přihlédnutím k jeho jednoduchému uživatelskému rozhraní a dostatečnému pokrytí funkcionalit. Specifikace požadavků explicitně stanovují nutnost ukládání výhradně textových a číselných dat.

#### MySQL

Rozhodnutí o typu databáze pro implementaci v bakalářské práci nyní přechází k specifikaci konkrétního relačního databázového systému. Jedním z nejvyužívanějších a dobře etablovaných systémů v této oblasti je MySQL.

MySQL je open-source databázový systém postavený na jazyce SQL, což jej činí vhodným pro široké spektrum aplikací. Jeho architektura byla navržena tak, aby nebyla omezena na žádnou konkrétní platformu, což znamená, že je schopný provozu na různých operačních systémech. Primárním cílem jeho vývoje bylo vytvoření databázového systému optimalizovaného pro webové aplikace.[9]

MySQL se vyznačuje vysokou úrovní výkonu a škálovatelnosti, což mu umožňuje efektivně zpracovávat velké množství dotazů a obsluhovat tisíce paralelních procesů. Tyto vlastnosti jsou klíčové zejména pro webové obchody, kde je nezbytné zajistit rychlé a spolehlivé odezvy databázových operací při zpracování transakcí a udržování stability systému. Z tohoto důvodu je MySQL často preferovanou volbou pro implementaci databázového řešení v prostředí webových aplikací.

MySQL databáze, s ohledem na svou vysokou komplexitu a využitelnost v oblastech vyžadujících online přístup k datům, se vyznačuje prominentním postavením na poli relačních databázových systémů. Nicméně, v kontextu našeho specifického projektu, který postrádá potřebu pro tuto charakteristickou vlastnost, byla tato platforma záměrně vyloučena z uvažovaných variant.

#### ProstegreSQL

PostgreSQL je relační databázový systém, který je vyvíjen komunitou, nikoli velkou korporací, což přispívá k jeho popularitě. Jedná se o klient-server architekturu, kde veškerá data jsou uchovávána na serveru a klienti se k němu připojují, často prostřednictvím příkazové řádky (CLI). PostgreSQL disponuje širokou škálou pokročilých funkcí, které převyšují možnosti MySQL, avšak na druhou stranu není tak rychlý jako zmíněný konkurenční systém.

Hlavní výhodou PostgreSQL je jeho využití metody MVCC (Multi-Version Concurrency Control). Tato technika umožňuje databázi umožnit přístup k datům více uživatelům současně, aniž by docházelo k blokaci. To znamená, že čtenáři neblokují zapisovatele a zapisovatelé neblokují čtenáře. Tento efekt je dosažen tím, že databáze udržuje informace o tom, která data byla poskytnuta klientům a tyto informace udržuje až do konce transakce. Tímto způsobem se minimalizují konflikty a umožňuje se efektivní manipulace s daty.[10]

Vzhledem k vysoké komplexitě a rozmanitosti možností, které relační databázový systém PostgreSQL nabízí, není vhodnou volbou pro tento projekt. Hlavními kritérii výběru databázového systému jsou jednoduchost a lokalita, což vylučuje nasazení PostgreSQL pro tuto konkrétní aplikaci.

#### SQLite3

SQLite3 je relační databázový systém, který se odlišuje od tradičních server-klient modelů tím, že veškerá data jsou uložena lokálně na disku v jediném souboru, což představuje podobnost s běžnými soubory, jako jsou například textové soubory ve svém formátu .doxc, které můžeme otevírat a následně s nimi pracovat například pomocí aplikace ,,Word‘‘. Databázové systémy mají také svůj vlastní formát .db, s tímto formátem se dá pracovat například v aplikaci ,, SQLite Database Browser‘‘. SQLite3 je schopen ukládat a manipulovat se strukturovanými daty, což jej činí mocným nástrojem pro práci s databázemi.[11]

Tento typ databáze je zahrnut v základní distribuci jazyka Python. To znamená, že programátoři, kteří používají Python, nemusí instalovat žádné dodatečné knihovny pro práci s SQLite3, stačí pouze inicializovat tuto knihovnu a mohou začít vytvářet vlastní databáze.

Díky své jednoduchosti je SQLite3 vhodný pro širokou škálu uživatelů, od začátečníků, kteří se teprve seznamují s práci s databázemi, až po zkušené programátory hledající výkonnou a jednoduchou databázi pro lokální projekty.

Je však třeba poznamenat, že kvůli své povaze lokální databáze a absenci serverového modelu, SQLite3 neposkytuje pokročilé mechanismy jako je například MVCC (Multiversion Concurrency Control). Avšak, tato jednoduchost neznamená, že SQLite3 je omezený pouze na základní operace ukládání a načítání dat. Databázový systém je také schopen pracovat se složitějšími procesy, včetně triggerů, což umožňuje vytvářet dynamické a komplexní aplikace. ,, Některé aplikace mohou používat SQLite pro interní ukládání dat. Je také možné vytvořit prototyp pomocí SQLite a pak portovat kód do větší databáze, jako je PostgreSQL nebo Oracle‘‘[11].

Pro účely této bakalářské práce byl jako databázový systém zvolen SQLite3. Tato volba byla motivována především její jednoduchostí a vysokou rychlostí. Vzhledem k tomu, že soustava, na níž je tato bakalářská práce vypracovávána, nenabízí přístup k internetu, bylo nevhodné zvolit databázový systém založený na klient-server architektuře. Proto byla vlastnost SQLite3 umožňující lokální uložení dat na disku považována za vhodnou volbu.

#### Hashovací knihovna

Hashovací knihovna je nedílnou součástí této bakalářské práce, kvůli správné autentizaci uživatelů. V kontextu zabezpečení je běžnou praxí ukládat hesla do databází ve formě zahashovaného textu. Tento proces slouží k ochraně hesel uložených v databázi před hackerským útokem. Hash je řetězec o pevné délce, který vzniká z libovolně dlouhého vstupního řetězce. Důležitou vlastností této procedury je, že je jednosměrná – z hashové hodnoty není možné zpětně získat původní heslo. Dokonce i prázdný řetězec má svou jedinečnou hashovou hodnotu, "e3b0c44298fc1c149afbf4c8996fb92427ae41e 4649b934ca495991b7852b855". Tento fakt výrazně komplikuje útoky na hesla, neboť útočník nemá jinou možnost, než zkoušet náhodná hesla a provádět bruteforce útoky.

Tato knihovna nabízí spoustu různých hashovacích algoritmů jako jsou ,,sha384, whirlpool, sha256, atd.‘‘. V tomto projektu se využívá hashovací funkce sha256. ,,Důvodem pro použití SHA-256 je, že se jedná o jeden z nejznámějších a nejbezpečnějších hashovacích algoritmů, které se v současné době používají, a zároveň nabízí méně času potřebného k výpočtu hashe‘‘[12].

#### Knihovna time

Tato knihovna je nástrojem primárně určeným pro manipulaci s časem v programovacím prostředí. Poskytuje rozsáhlou funkcionalitu pro práci s časem, včetně získání aktuálního času, manipulace s kompletními daty ve formátu "%Y-%m-%d %H:%M:%S", a umožňuje řízení časových prodlev pomocí funkce `time.sleep(X)`, kde programátor může nastavit hodnotu proměnné X na požadovanou dobu spánku programu v milisekundách.[13]

V tomto projektu byla knihovna ,,time‘‘ využita zejména pro regulaci časové odezvy na změny tlaku v pneumatických svalech. Pomocí funkce ,,time.sleep(X)‘‘ bylo možné definovat dobu čekání programu mezi jednotlivými čtecími časy, což umožnilo citlivě reagovat na dynamiku změn tlaku v daném systému.

#### Knihovna openpyxl

Vzhledem k tomu, že tato knihovna není součástí základního balíčku Pythonu, je nutné ji doinstalovat, aby s ní mohl uživatel pracovat. Tato knihovna je navržena k manipulaci se soubory s příponami „.xlsx“, „.xlsm“, „.xltx“ a „.xltm“, což jsou typy souborů spojené s programem Excel od společnosti Microsoft. Nicméně tato označení jsou poněkud zavádějící, neboť „Excel“ je pouze název softwaru vyvíjeného společností Microsoft, který je schopen pracovat s těmito soubory.[14]

Je třeba si uvědomit, že mnoho uživatelů nemusí používat program Excel, zejména kvůli nemožnosti pracovat off-line bez zakoupení drahé licence. Existují však alternativy, jako například „Google Sheets“ od společnosti Google, které poskytují podobné funkce a jsou zdarma k dispozici online, nebo z celkového balíčku LibreOffice aplikace ,,Calc‘‘ tento sw je dostupný zdarma i pro off-line práci.

Vzhledem k této problematice je vhodné použít obecnější termín pro označení tohoto typu softwaru, a to „Tabulkový editor“. Tento termín lépe odráží povahu programů, které umožňují vytvářet, editovat a analyzovat tabulková data bez ohledu na konkrétní platformu nebo poskytovatele softwaru.

Tato knihovna představuje významný nástroj díky svým mnohostranným funkcím, které umožňují nejen základní manipulaci s daty, jako je jejich vkládání na specifická místa a následné načítání, ale také jejich formátování. Jednou z významných a pokročilých vlastností této knihovny je schopnost vytvářet grafy z poskytnutých dat.

#### Knihovna pathlib

Tato knihovna je součástí standardní distribuce programovacího jazyka Python a zaujímá klíčové postavení při manipulaci s cestami v Python programování. Její využití je obzvláště efektivní díky široké škále předdefinovaných cest, jako je například funkce "Path.home()", která umožňuje snadný přístup k domovské složce uživatele a ukládání souborů na tuto lokaci. Tato knihovna se stává mocným nástrojem v programování díky své schopnosti poskytnout uživatelům flexibilitu při určování umístění uložení souborů. Výhodou je, že každý uživatel má možnost volby umístění uložení souborů dle vlastní preference, což by nebylo možné s pevně stanovenou cestou, která by neodpovídala individuálním potřebám uživatelů. [15]

#### Knihovna re

Jedná se o další knihovnu ze základní distribuce programovacího jazyka Python. Tato knihovna slouží především pro práci s regulárními výrazy. ,, Regulární výrazy, známé také jako regex nebo regexp, jsou nástroji pro práci s textem, které umožňují vyhledávání, nahrazování a manipulaci s textovými řetězci na základě definovaného vzoru (patternu)‘‘[16]. V rámci této bakalářské práce byla knihovna použita k validaci vstupních dat, kde byl specifikován regulární výraz, akceptující pouze čísla a znaménko mínus. Veškeré jiné znaky byly automaticky označeny jako neplatné a tímto způsobem byla implementována efektivní ochrana proti nevalidním vstupům.[17]

#### Knihovna PneumoCVUTFBMI.DeviceLoader

Tato knihovna byla speciálně vytvořena se soustavou, pro niž je v rámci této práce navržen grafický uživatelský rozhraní (GUI). Účelem této knihovny je usnadnit komunikaci s dalšími knihovnami, jako je sériová sběrnice či krokové motory, v rámci dané mechanické soustavy.

Tato knihovna představuje klíčový spojovací prvek mezi GUI a jednotlivými mechanickými součástmi daného systému. Její návrh a implementace jsou klíčovými kroky pro efektivní a intuitivní ovládání a monitorování soustavy pomocí uživatelsky přívětivého rozhraní.

# Implementace

Tento projekt byl rozdělen do jednotlivých komponent (složek). Tento přístup se zvolil především z důvodu přehlednosti výsledného projektu. Na tomto projektu se podílelo již více lidí vůbec před začátkem této bakalářské práce a po skončení této bakalářské práce by měl být software neustále udržován a zlepšován techniky kteří tuto soustavu dostanou na starosti. Proto je důležité, aby projekt byl správně strukturovaný pro snadnou následnou práci. Tento projekt se rozdělil do 6 hlavních komponent Obrázek 5.1.



Obrázek 5.1 Zdroj: Vlastní

Komponent diagram soustavy této bakalářské práce vyhotoven v jazyce UML.

## Pneumatická soustava

Pneumatická soustava nebyla součástí této bakalářské práce. Jedná se o soustavu, která je schopna ovládat nezávisle na sobě 5 krokových motorů které kontrolují ventily a pomocí těchto ventilů se mění tlak na jednotlivých vývodech. K těmto vývodům můžou být připojeny pneumatické svaly.

## DeviceLoader

Jedná se o komponentu, která je schopna komunikovat se soustavou a pomocí toho ovládat jednotlivé krokové motory. S touto třídou může komunikovat zbytek sw, ale nemusí být v této komponentě zahrnut pouze jí využívá jako prostředníka pro komunikaci se soustavou. Z této komponenty se využívá jedna základní funkce

### getBoardx()

x - > číslo desky. Příklad getBoard1() nám dovolí přistupovat k jednotlivým funkcím ovládací desky. Mezi tyto funkce patří:

* on() – tato funkce musí být deklarována na začátku každého programu se používá každá tato soustava, slouží k tomu aby se jednotlivá deska zapnula a stala se aktivní pro další operace
* readA0() – slouží k odečtení hodnoty v mV z integrovaného voltmetru soustavy. Každá deska má svůj vlastní volt metr, proto je nutné pro každé měření nutné specifikovat pro jakou desku toto měření chcete provést.
* go\_forward(speed, steps) – jedná se o základní pracovní funkci pomocí které je schopný sw nastavovat požadovaný posun. Tato funkce má dvě proměnné speed (rychlost) – která určuje za jak dlouho se jednotlivá akce provede. Například pokud bude speed = 20 tak to znamená že se krokový motor bude točit rychlostí 20 kroků za vteřinu. Druhou proměnou v této proměnné jsou steps (kroky), jedná se o jednotku o kolik se má jednotlivý krokový motor posunou. Pokud tedy bude funkce nastavena na go\_forward(20,100) – tak se tato funkce bude provádět 5 vteřin.
* go\_backward(speed, steps) – jedná se o analogii předchozí funkce pouze s tím rozdílem že tato funkci provádí jednotlivé kroky pozpátku. Obě funkce ovšem disponují možností zadávat do nich záporné hodnoty, proto při specifickém použití je jedna funkce redundantní.

## Main

A screenshot of a computer

Description automatically generatedJedná se o první komponent vytvořený v rámci této bakalářské práce. Tento komponent je základní kamenem celého grafického uživatelského rozhraní, neboť se v něm toto GUI inicializuje a vytváří se v něm. Tato komponenta komunikuje se třemi dalšími komponentami a těmi jsou AdminWindow, UserWindow a databáze. V této komonentě je jedna základní třída App. Jedná se o třídu ve které se inicializuje okenní aplikace Obrázek 5.2.

Obrázek 5.2 Zdroj: Vlastní

Základní okno grafického uživatelské rozhraní

První oknem, které se objeví při spuštění aplikace je zároveň také autentizačním oknem. Je vytvořeno pomocí grafické knihovny cutomtkinter. Pozice tohoto okna stejně všech ostatních oken je vypočítávána dynamicky Kód 5.1. Parametry width (šířka) a height (výška) jsou parametry pro nastavení velikosti okna.

Kód 5.1: Výpočet pozice okna

# Zjištění šířky okna

screen\_width = self.winfo\_screenwidth()

# Zjištění výšky okna

screen\_height = self.winfo\_screenheight()

# Vypočet prostředku obrazovky

screen\_x = (screen\_width - width) // 2

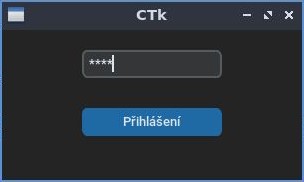
screen\_y = (screen\_height - height) // 2

# Nastavení jednotlivých parametrů

self.geometry(f"{width}x{height}+{screen\_x}+{screen\_y}")

Na obrázku 5.2. v pravém horním rohu se nachází trojice základních tlačítek pro práci s oknem jako je minimalizovat na lištu, přizpůsobit obrazovce případně zmenšit a zavřít okno. Tyto funkce jsou zde automaticky vytvořeny proto se v kódu nikde nenachází žádný kód s nimi spojený.

Oknu obrázek 5.2. dominují dvě hlavní widgety (prvky).

První z nich je CTkEntry pole. Jedná se o widgetu do které je možné zapisovat znaky a následně pomocí funkce get() tyto znaky jako string získat a následně s nimi v rámci projektu pracovat. Tato komponenta má dva parametry. Prvním z nich je placeholder\_text jedná se o parametr který je uvnitř entery pole a napovídá uživatele co má daného entery pole zadat obrázek 5.2. Teto place holder se smaže poté co do něj uživatel začne vyplňovat hodnoty. Druhým parametrem je parametr show. Tento parametr je především vhodný v případě, kdy je nutné aby nebylo vidět co daný uživatel do entery pole zadává stejně jako v tomto případě kdy zadává heslo Obrázek 5.3.

Obrázek 5.3 Zdroj: Vlastní

GUI v tmavém režimu

Druhým widgetem je CTkButton jedná se o tlačítko, pomocí které má dva parametry. Prvním parametrem je text – jedná se o možnost napsat do tlačítka jakýkoliv text, druhým parametrem je command (příkaz) – tento parametr slouží k tomu, aby se přiřadil specifický příkaz k tlačítku. Například v tomto případě slouží tlačítko k tomu, aby se spustila série akcí, které autentizují uživatele. Tato funkce je zároveň také přiřazena speciální bindovací funkci, pomocí které si můžeme vybrat jakou klávesnici k tomu úkonu využijeme. V tomto kódu se rozhodlo pro standartní Enter Kód 5.2.

Kód 5.2: přiřazení funkce login() fyzické klávese enter.

self.bind("<Return>", lambda event: self.login())

Autentizační mechanizmus byl navrhnut jednoduše pokud pracovník zadá uživatelského heslo dostane se na uživatelské okno, pokud zadá heslo pro admina dostane se do administrátorské části. Toto heslo se ověřuje v databázi a pokud najde databáze shodné heslo tak uživatele vpustí dál na základě informace kterou mu databáze vrátí. Pro bezpečí uložených hesel je heslo v databázi hashováno a proto když uživatel potvrdí svoje heslo tak se nejprve převede na hash a až poté se posílá na ověření do databáze. Pokud uživatel zadá špatné přístupové heslo tak mu bude vrácena hláška Obrázek 5.4.

A screenshot of a computer

Description automatically generated

Obrázek 5.4 Zdroj Vlastní

GUI v případě že uživateli bude odepřen přístup

Pokud uživatel zadá správné heslo program vytvoří nové toplevel okno bud administrátorské nebo uživatelské části. A současné okno bude automaticky zavřeno.

CtkToplevel je speciální druh okna které v sobě uchovává informaci o svém hlavním okně, a proto je schopno se na něj zase zpětně v případě potřeby vrátit bez nutnosti vytvoření nového okna.

## AdminWindow

A screenshot of a computer

Description automatically generatedKomponenta AdminWindow je přístupná pouze po zadání pouze správného administrátorského hesla. Administrátor by měla být pouze osoba dostatečně zodpovědná, aby se jí dali svěřit náročnější úkony bez takové kontroly chyby jako u normálního uživatele. S tou to premisou se pracovalo i v tomto projektu, proto je administrátor schopen změny. První obrazovku, kterou administrátor uvidí poté co se přihlásí pomocí svého hesla je okno kde si bude moci vybrat mezi dvěma úkony Obrázek 5.5.

Obrázek 5.5 Zdroj: Vlastní

GUI pro administrátora pro výběr jeho činnosti

V nejspodnější části okna se nachází tlačítko s názvem Hlavní Stránka. Jedná se o tlačítko, které disponují funkcí návratu zpět na autentizační obrazovku. Toto tlačítko je vhodné především pro administrátory, kteří si chtějí například otestovat nový převodní vzorec po jeho nastavení. A toto tlačítko jim slouží k tomu, aby nemuseli celou aplikaci vypínat a znovu spouštět, po stisknutí tlačítka se aktivuje funkce, která zavře aktuální okno a znovu otevře autentizační.

A screenshot of a computer

Description automatically generatedV prostřední části se nachází 5 tlačítek. Pomocí těchto tlačítek se je administrátor schopen prokliknout do databázového okna Obrázek 5.6. Byla zvolena metoda vytvoření tlačítka pro každý sval z důvodu, že si jméno tohoto tlačítka předáváme do dalšího okna Kód 5.3. a následně s těmito hodnotami pracovat.

Obrázek 5.6 Zdroj: Vlastní

Grafické uživatelské rozhraní pro databázi

command=lambda: self.databaseWindow("sval1")

Kód 5.3: Nastavení parametr tlačítka command na lambda funkci která je schopná předávat string

Hodnota svalu se následně využívá k vybrání požadovaných tabulek z databáze Kód 5.4. V pythonu je funkce, když dáte před string f tak jsem schopni pak do toho stringu pomocí složených závorek {} vložit přímo proměnou. Kód se tím stává přehlednější oproti klasickému zápisu, kdy se musí jednotlivé strigy a proměnné spojovat logickým operátorem plus +. V tomto případě se to provedlo pro všechny 4 převodní vzorce.

Kód 5.4: Zavolání funkce pro načtení požadované tabulky z databáze

self.load\_table(f"{sval}\_mm", 0)

Toto okno obrázek 5.6 je rozděleno do 5 hlavních částí. První část je tlačítko pro návrat na hlavní administrátorskou obrazovku jde si bude moci uživatel vybrat jiný sval pro práci s jeho databázi nebo jinou akci. Dále jsou zde čtyři téměř totožné části. Jediný jejich rozdíl je, z jaké tabulky v databázi načítají data. Význam jednotlivých převodních vzorců je vysvětlen v kapitole 5.5 UserWindow.

Každá část má tři prvky. Prvním prvkem jsou entery pole kde se nachází sklon, posun a popis funkce. Tento prvek byl záměrně zvolen jako entery pole, aby administrátor nemusel složitě spouštět editování jednotlivých vzorců takhle pouze stačí změnit jakoukoliv část převodního vzorce na požadovanou hodnotu a pak využít druhý prvek kterým je tlačítko upravit.

Toto tlačítko slouží k uložení změn do databáze. Toto tlačítko zároveň disponuje funkcí, pomocí které se volí aktuální převodní vzorec. Soustava například po dlouhé době může obsahovat spoustu převodních vzorců například pro různě těžké hlavy pacientů. A proto je potřeba rychle mezi těmito vzorci měnit. K tomu pouze stačí zmáčknout tlačítko upravit a tím se nastaví nový současně využívaný převodní vzorec.

První dvě entery pole pro sklon a posun disponují validační funkční která vyhodí varovnou hlášku a nedovolí provést v databázi změny, pokud bude entery pole prázdné nebo se nebude jednat o číslo.

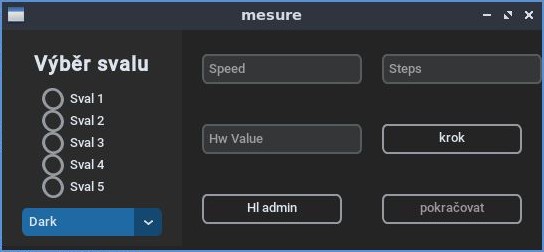
A screenshot of a computer

Description automatically generatedPosledním prvek je tlačítko nový. Pomocí tohoto tlačítka je schopen administrátor vytvářet nové převodní vzorce. Po stisknutí tlačítka se objeví nové entery pole s tlačítkem uložit Obrázek 5.7. Zde uživatel musí vyplnit entery pole příslušnými hodnotami a poté co bude hotov stiskne tlačítko uložit a tím se vzorec uloží do databáze.

Obrázek 5.7 Zdroj: Vlastní

GUI databáze, když administrátor vytváří nový převodní vzorec.

Druhé pracovní okno administrátorské části je okno pro vytváření měření pro následnou práci na vyhotovení nových převodních vzorců Obrázek 5.8.

Toto okno je rozděleno do dvou framu. Pomocí těchto framu byla možnost vytvoření přehledného uživatelského prostředí s jasně rozdělenými prvků. V levé části okna se nachází CtkRadioButton jedná se o společný druh tlačítka i když spíše je podobný chechboxu – který se vyznačuje svojí možností vybrat více možností a následně se všemi těmito možnostmi pracovat. Ovšem komponenta se liší od již zmíněného chechboxu tím že lze mít vždy vybranou pouze jednu možnost. V tomto případě to bylo velice vhodné, neboť bylo nutné zajistit aby uživatel mohl ovládat vždy pouze jenom jeden sval.

Obrázek 5.8 Zdroj: Vlastní

Grafické uživatelské rozhraní pro administrátora pro vytváření měření.

Ve spodní části tohoto framu se nachází CtkOptionMenu. Tato komponenta je rozbalovací prvek, který zobrazuje pouze aktivní prvek. Když se tato komponenta rozklikne tak ukáže všechny možnosti nastavení ,,Dark, Light, System‘‘. Jedná se o barevné schéma aplikace. Custom tkinter disponuje nastavením tmavého a světlého modelu. Toto nastavení může být ruční nebo podle systému. V tomto projektu se jako automatický barevný model vybral Systém, což umožní se programu přizpůsobit nastavení soustavy.

Druhý fragment je hlavní pracovní částí tohoto okna. Nacházejí se zde tři entery pole, do kterých uživatel musí zadat číselnou hodnotu. Prvním entery polem je pole speed do tohoto pole musí administrátor zadat požadovanou rychlost kolik kroků chce, aby soustava vykonala. Druhým entery polem je pole steps do tohoto pole administrátor musí zadat kolik kroků má soustava vykonat. Posledním entery polem s place holder textem ,,HW Value‘‘ je entery pole do kterého administrátor musí zadávat ručně hodnoty z externího tlakoměru v jednotkách mbar. Tyto entery pole také disponují validačním mechanismem, který zamezuje zadání jiného vstupu než je číslo.

Tlačítko krok je schopno celý tento proces potvrdit a provést měření tím že se provede požadovaný postup na krokovém motoru.

S přechozím tlačítkem ,,krok‘‘ souvisí tlačítko pokračovat jedná se o tlačítko které umožní administrátorovy změnu svalů bez ztráty původních dat.

Proces měření dat je navrhnut způsobem kdy administrátor má jeden externí tlakoměr, pomocí kterého je schopen měřit tlak v jednotlivých svalech soustavy. A proto soustava potřebuje nějaké jištění, že Administrátor nezačal provádět měření na svalu když například ještě nemá přepojený tlakoměr na nový sval. Proto toto tlačítko slouží jako pojistka, když se administrátor rozhodně měřit jiný sval soustava v programu skočí do podmínky, že současný sval je jiný než minulý sval a tím se spustí série akcí. Soustava aktivuje tlačítko pro potvrzení, že vše je, jak má být a dokud nebude toto tlačítko zmáčknuto soustava administrátorovy nedovolá pokračovat v práci na soustavě.

Další akcí, která se spustí je automatické sjetí původního měřeného svalu na technickou nulu. Program si pamatuje, kolik kroků od začátku udělal a poté následně co uživatel zvolí jiný sval program začne automaticky sjíždět na technickou nulu a až poté co soustava sjede na technickou nulu bude administrátory zpřístupněna možnost zmáčknout tlačítko pokračovat a následně provést měření.

Hlavní funkcí tohoto měření je ovšem zápis jednotlivých hodnot měření do souboru .xlsx. To je schopen pomocí knihovny openpyxl. Program do tohoto souboru ukládá Počet kroků motoru od začátku měření, aktuální hodnota interního voltmetru v mV, Hodnotu fyzického tlakoměru v mbar a rychlost jakou se v tu danou chvíli krokový motor pohyboval. Všechny tyto hodnoty se ukládají do té doby, dokud administrátor nezmění sval poté soustava začne provádět všechny kroky spojené s tímto úkonem a také uloží tento soubor na plochu počítače pro snadné následné nalezení pro další práci.

Tento program je schopen pracovat pro 5 jednotlivých svalů ale každý funguje úplně stejně jediné v čem se liší je jejich název, a proto zde hrozí vážná refundace kódu která by byla při nesprávném přístupu 5násobná oproti normálu. Proto se v tomto programu vytvořil slovník, kde byly definovány jednotlivé desky, které komunikují s jednotlivý m svalem a následně se ze slovníku vybírala ta správná deska podle čísla svalu Kód 5.5.

Kód 5.4: Využití slovníku pro omezení redundance kódu

b\_objects = {1: b1, 2: b2, 3: b3, 4: b4, 5: b5}

b\_objects[Minuly\_Sval].go\_backward(Speed, Steps)

V kódu 5.4 je vidět že se vybírá ten sval který odpovídá hodnotě minulého svalu. Tento kód byl vyjmut z části programu kde program pohyboval minulým svalem na technickou nulu.

## UserWindow

A screenshot of a computer

Description automatically generatedUserWindow je poslední částí této bakalářské práce. Jedná se o komponent se kterým může pracovat user (uživatel). Tento komponent je spuštěn pouze v případě kdy se uživatel přihlasí pomocí správného uživatelské hesla která odpovídá heslu v datábazy které je označeno jako uživatelské. Tato komponenta komunikuje s main, databází a deviceloaderem. Toto okno se skládá ze tří framů Obrázek 5.9.

Obrázek 5.9 Zdroj: vlastní

Uživatelské GUI pro nezávislé ovládání 5 svalů

První frame jménem ,,Left Frame‘‘ obsahuje tlačítka pro jednotlivé akce. …

Drým druhým widgetu je CTkSwitch – jedná se o komponetu která je bud on nebo of a pomocí tohoto si uživatel může zvolit zdali chce light mode nebo dark mode. Když uživatel poprvé zapne toto okno tak se mu automaticky zapne to ce je předvoleno podle systému ale tuto možnosti si může změnit podle svých preferencí.

Druhým framemem je ,,Right Frame‘‘ . Tato část obsahuje radiobuttony pro výběr jednotek. Tyto jednotky si může uživatel vybrat dle své prefence v jakých jednotkách mu přijde nejpříjemnější ovládání pneumatické soustavy. Komponenta byla zvolena radiobutton neboť je zde nutnost zabezpečení že uživatel si vybere pouze jeden sval pro ovládání proto výběr chechboxu by musela být zabezpečena funkcí která by ošetřovala že pokaždé co by uživatel vybral novou jednotku tak by program musel automaticky odoznačit předchozí vybraný checkbox. Tento radiobutton má vytvořenou i vlastní validaci která zabezpečuje že uživatel který nevybere jednotky pro zadávání bude na tuto skutečnost upozorněn a nebude vpuštěn dál.

Druhým prvkem v tomto framu je tlačítko main. Pomocí tohoto tlačítka se může uživatel dostat zpět na první okno kde zadával helso pro přihlašení. Toto tlačítko je vhodné především pro administrátory kteří zárověn pracují jako uživatelé a zapoměnily změnit převodní vzorce, tak toto tlačítko omezí nutnost vypnutí a znovu spouštění aplikace.

Hlavním prvek okna je ,,main frame‘‘. V tomto prvku bude uživatel schopen zadávat hodnoty o které chce zvýšit nebo zmenšit tlak. Každý entry pole je vybaveno příslušným jménem svalu jako palce holder. Zároven pro bezpečnost že uživatel nebude schopen zadat do entery pole nic jiného než číslo, neboť program není schopen příjímat text jelikož neví na jakou číselnou hodnotu ho má převest, tak bylo nutné zabezpečit že uživatel bude schopen zadat pouze čísla. To se provedlo validací. Stejný druh validace se provadí i v ostatních částech programu pomocí regulárních výrazů. Dalším bezpečnostím prvek který tento frame obsahuje je CTkProgressBar na obrázku 5.9 se nachází nad entery polem. A jedná se o komponentu která zobrazuje v jakém stavu natlakování je pneumatický sval. Toto jako informační prvek pro uživatele neboť je schopen určit jak je schopen ještě natlakovat soustavu než bude na konci nebo naopak o kolik může tlak snížit. Tento prvek není ovšem nastaven jenom jako vyzuální kontrola ale je také nastaven aby pokud uživatel bude chtít limit překročit jak ze spodní tak horní hranice bude na tuto skutečnost upozorněn že je již na povolené hranici svalu a zároveň mu tato akce nebude povolena. V horní hranici je tato zábrana především z důvodu aby nedošlo k nevratnému poškození pneumatického svalu a tudíž i ohrození pacienta který bude v plízkosti těchto svalů. A spodní hranice je zde nastavena především z důvodu pokud by se stalo že ve se uživatel snažil neustále ubírat tlak i když už by byl ventil dávno zavřen hrozilo by že be se tento ventil mohl poškodit.

Tento frame obsahuje jediné tlačítko které spouští celkový krok celé soustavy najednou. Soustava je nastavena tak že pokud by se měla posunout o 0 kroků tak se tento krok vůbec neuskuteční a přeskočí se.

V UserWindow bylo nejdůležitější a zároveň nejsložitější částí navrhnout výpočet pro správné přepočty mezi jednotlivými jednotkami. Důležité bylo zaměřit se na omezení redundance. Ta zde hrozila hned ve dvou případech. V této soustavě se nachází 5 svalů a k jejímu správnému fungování jsou potřeba 4 převodní vzorce.

Nejednoduším jednotkou pro práci jsou kroky ty se nemusí nijak upravovat u nich stačí pouze jejich přesnou hodnotu přenést do funkce go\_forward nebo go\_backward.

V tomto projektu se vyřešila možnost zadávání záporné hodnoty tak že se nastavila podmínka která rozděluje tyto hodnoty na kladné a záporné a následně pokud se jednalo o zápornou hodnotu která má vstoupit do funkce go\_backward() tak byla tato hodnota vložena jako záporná tudíž se její zápornost znegovala a hodnota tak vstupovala do funkce jako kaldná.

Redundace v této části kódu se vyřišila elegantním zapisem pomocí využítí jednoho slovnáku a jednoho pole Kód 5.5. Zde je program schopen na jednom řádku docílit stejného výsledku ke kterému by bylo jinak zapotřebý 5x nápsat tuto funkci s příslušným svalem zde stačí použíjede pouze jeden for cyklus který provede akci pro každý sval.

Kód 5.4: Omezení redundance

svaly[i].go\_forward(10, results[i])

Druhým již složitějším výpočtem je pokud uživatel chce provádět změnu tlaku za pomoci milivotů. V tomto případě je nutné využít převdní rovnici z databáze. Jelikož ale v databázi není pouze jedna převodní rovnice tak si program vůbec musí zjistit z jiné tabulky vůbec jaký převodní vzorec má z požadované tabulky načítat to se provádí pomocí tohoto Kód 5.5. V tomtu kódu je opět omezena redundace pomocí toho že se načítá pro každý index který může nabívat hodnot od 1-5 to se spojí se stringem sval a pak je ještě nutné to spojit s příslušnou jednouttkou se kterou pracujeme ta se načítá z radiobuttonu. Následě po tomto prvním kroku se může až provést výběr celého požadovaného řádku s příslušným svalem.

Kód 5.4: Výběr příslušné rovnice z databáze

cur.execute(f"SELECT sval{index}\_{jednotka} FROM sval{index}")

# Spojení navráceno SQL příkazu do stringu

cislo\_vzorce = cur.fetchone()

cur.execute(f"SELECT \* FROM sval{index}\_{jednotka} WHERE id IN ({int(cislo\_vzorce[0])})")

Následně co je příslušný převodní vzorec načten provádí se výpočet konečné hodnoty na kterou má program dojet. Pomocí načtení aktuální hodnoty v mV + posunu který si uživatel volí sám. Poté je nutnost převést dané startovní a konečné hodnoty v mV na jednotky které je krokový motor schopný přijmout proto je nutné spočítat startovní kroky kolik kroků bylo provedeno od začátku procesu. DeviceLoader obsahuje funcki která je v sobě schopná uchovávat hodnoty kolik kroků bylo provedeno ovšem ale tento výpočet se vymaže pokaždé když motor provede krok zpět proto je tato funkce nepoužitelná další možností by bylo si tuto informaci uchovávat v programu kde by to šlo ale z důvodu přesnoti nastavování hodnot se tato hodnota počítá pokaždé znovu. Následně se vypočítá kolik kroků motor musí provést aby se do této pozice dostal z technické nuly. Nakonec se tyto dvě hodnoty od sebe odečtou a program si tento výsled uloží do pole do kterého až uloží veškeré hodnoty svalů tak začne provádět pohyb krokových motorů.

Nejtěžším ukolem bylo vyřešení dvou jednotek které nemají žádné referenční hodnoty ze soustavy. Milivolty je soustava schopna odečíst z integrovaných voltmetrů. Ovšem pro tlak a ani pro vzdálenost tato možnost neexistuje. Jedna z možností by bylo přidání senzorů které by tyto referenční hodnoty dodaly to by ovšem nebylo příliš ekonomické a musela by se přepracovávat celá sostava. Proto se zvolila metoda kdy se jako refenční hodnoty využíjí hotnovy mV které se převodu na příslušné jednotky pomocí převodní funkce. Tudíž oba dva druhy převodů fungují na stejném principu že pomocí refenční hodnoty se vypočítá počáteční a konečný krok a tyto dvě hodnoty se pak od sebe odečtou a je výsledek. Pouze u mm a mbar je nutný ještě jeden mezi krok kdy se převede aktuální hodnota z mV na požadovanou jednotku a tím se vytvoří referenční hodnota.

Následně poté co se provedou veškeré upravy tlaků se přepočítají hodnoty progres barů na nové hodnoty aby se zajistilo že při dalším opakování meření nedojde k překročení těchto hranic.

# Uživatelská dokumentace

Manuál pro práci s aplikaci. Mj. se v této kapitole prezentuje vzhled a ergonomičnost aplikace.

# Výsledky (Testování)

# Diskuse

# Závěr

# Seznam použité literatury

|  |  |
| --- | --- |
| [1] | BANSAL, Rishabh. *Python GUI – tkinter*. Online. GeeksforGeeks. 2023. Dostupné z: <https://www.geeksforgeeks.org/python-gui-tkinter/>. [cit. 2024-03-15]. |
| [2] | *Co je to Open source? Zdroj: https://it-slovnik.cz/pojem/open-source/?utm\_source=cp&utm\_medium=link&utm\_campaign=cp*. Online. IT SLOVNÍK. Dostupné z: <https://it-slovnik.cz/pojem/open-source>. [cit. 2024-03-15]. |
| [3] | ALI, Francis. *Začínáme s Kivy pro vývoj grafického uživatelského rozhraní*. Online. Python GUIs. 2023. Dostupné z: <https://www.pythonguis.com/tutorials/getting-started-kivy/>. [cit. 2024-03-15]. |
| [4] | *Co je to Framework?* Online. A star search. 2021. Dostupné z: <https://a-starsearch.cz/blog/co-je-to-framework>. [cit. 2024-03-15]. |
| [5] | VIKTORIN, Petr a HRONČOK, Miro. *GUI v Pythonu: PyQt5*. Online. Nauč se Python!. 2017. Dostupné z: <https://naucse.python.cz/lessons/intro/pyqt/>. [cit. 2024-03-15]. |
| [6] | GEON. *Co je to Tkinter?* Online. PY.CZ. 2005. Dostupné z: <https://www.py.cz/Tkinter>. [cit. 2024-03-15]. |
| [7] | SCHIMANSKY, Tom. *CustomTkinter*. Online. Git Hub. 2021. Dostupné z: <https://github.com/TomSchimansky/CustomTkinter/tree/master>. [cit. 2024-03-15]. |
| [8] | SCHIMANSKY, Tom. *Documentation Introduction*. Online. CustomTkinter. 2021. Dostupné z: <https://customtkinter.tomschimansky.com/documentation/>. [cit. 2024-03-15]. |
| [9] | *Co je to databáze?* Online. Oracle. Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>. [cit. 2024-03-16]. |
| [10] | MIHALCEA, Vlad. *How does MVCC (Multi-Version Concurrency Control) work*. Online. Vlad Mihalcea. 2022. Dostupné z: <https://vladmihalcea.com/how-does-mvcc-multi-version-concurrency-control-work/>. [cit. 2024-03-16]. |
| [11] | *Rozhraní DB-API 2.0 pro SQLite databáze¶*. Online. Python. Dostupné z: <https://docs.python.org/3/library/sqlite3.html>. [cit. 2024-03-16]. |
| [12] | VASUDEV4. *Modul hashlib v Pythonu*. Online. GeeksforGeeks. 2023. Dostupné z: <https://www.geeksforgeeks.org/hashlib-module-in-python/>. [cit. 2024-03-16]. |
| [13] | *Časový přístup a převody*. Online. Python. Dostupné z: <https://docs.python.org/3/library/time.html>. [cit. 2024-03-16]. |
| [14] | *Openpyxl 3.1.2*. Online. Pypi. Dostupné z: <https://pypi.org/project/openpyxl/>. [cit. 2024-03-16]. |
| [15] | IHRITIK. *Modul Pathlib v Pythonu*. Online. GeeksforGeeks. Dostupné z: <https://www.geeksforgeeks.org/pathlib-module-in-python/>. [cit. 2024-03-16]. |
| [16] | SEOPRAKTICKY.CZ. *Co je to regulární výraz a k čemu se používá*. Online. Collabim. 2023. Dostupné z: <https://www.collabim.cz/akademie/knihovna/co-je-to-regularni-vyraz-a-k-cemu-se-pouziva/#:~:text=Už%20víme%2C%20že%20se%20regulární%20výrazy%20skládají%20ze,3%20závorky%20se%20používají%20pro%20seskupování%20množin%20znaků.>. [cit. 2024-03-16]. |
| [17] | *Operace s regulárními výrazy*. Online. Python. Dostupné z: <https://docs.python.org/3/library/re.html>. [cit. 2024-03-16]. |

# Příloha A: Požadavky na formátování práce

# Příloha B: Základní typografické zásady

# Příloha C: Další doporučení pro přehlednost textu

# Příloha D: Obsah přiloženého CD