**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](#_Toc162335915)

[1 Úvod 6](#_Toc162335916)

[2 Přehled současného stavu 7](#_Toc162335917)

[2.1 Motivace pro vznik projektu 7](#_Toc162335918)

[2.1.1 Základní principy intubačních přístrojů 7](#_Toc162335919)

[2.1.2 Principy magnetické rezonance 7](#_Toc162335920)

[2.2 Současná pneumatická soustava a její fungování 7](#_Toc162335921)

[2.2.1 Komponenty pneumatické soustavy 7](#_Toc162335922)

[2.2.2 Operační systém pro řízení soustavy 7](#_Toc162335923)

[2.3 Pneumatické svaly a jejich typy 7](#_Toc162335924)

[2.3.1 Popis různých typů pneumatických svalů 7](#_Toc162335925)

[2.3.2 Principy práce pneumatických svalů 7](#_Toc162335926)

[2.4 Softwarové a robotické aspekty 7](#_Toc162335927)

[2.4.1 Vývoj softwaru pro ovládání pneumatických svalů 7](#_Toc162335928)

[2.4.2 Integrace pneumatických svalů do robotických struktur 7](#_Toc162335929)

[2.5 Python 7](#_Toc162335930)

[2.5.1 Výhody a nevýhody oproti ostatním programovacím jazykům 7](#_Toc162335931)

[2.5.2 Proč jsem si ho vůbec vybral a proč se předešlá část soustavy taky programovala v pythonu 7](#_Toc162335932)

[2.6 SQL 8](#_Toc162335933)

[2.7 UML 8](#_Toc162335934)

[3 Cíle práce 9](#_Toc162335935)

[4 Návrh aplikace 11](#_Toc162335936)

[4.1 Funkční specifikace 11](#_Toc162335937)

[4.1.1 Knihovna pro GUI 11](#_Toc162335938)

[4.1.2 Databázová knihovna 15](#_Toc162335939)

[5 Implementace 21](#_Toc162335940)

[5.1 Pneumatická soustava 21](#_Toc162335941)

[5.2 DeviceLoader 21](#_Toc162335942)

[5.2.1 getBoardX() 22](#_Toc162335943)

[5.3 Main 23](#_Toc162335944)

[5.4 AdminWindow 26](#_Toc162335945)

[5.5 UserWindow 31](#_Toc162335946)

[6 Uživatelská dokumentace 36](#_Toc162335947)

[7 Výsledky (Testování) 37](#_Toc162335948)

[8 Diskuse 38](#_Toc162335949)

[9 Závěr 39](#_Toc162335950)

[Seznam použité literatury 40](#_Toc162335951)

[Příloha A: Požadavky na formátování práce 42](#_Toc162335952)

[Příloha B: Základní typografické zásady 42](#_Toc162335953)

[Příloha C: Další doporučení pro přehlednost textu 42](#_Toc162335954)

[Příloha D: Obsah přiloženého CD 42](#_Toc162335955)

# 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 rozčleněn do jednotlivých komponent (modulů) s cílem zvýšit přehlednost a usnadnit správu výsledného softwarového produktu. Tento přístup byl zvolen zejména s ohledem na rozsah a komplexnost projektu, stejně jako na potřebu spolupráce více jednotlivců, kteří se na projektu podílejí. Je třeba zdůraznit, že tým pracoval na této soustavě již před zahájením této bakalářské práce, a bude i nadále po jejím ukončení, přičemž se plánuje neustálé udržování a vylepšování softwarového produktu. Proto je klíčové, aby byla struktura projektu správně navržena s ohledem na snadnou budoucí údržbu a rozšiřitelnost. Projekt byl rozdělen do šesti hlavních komponent, jak je znázorněno na Obrázku 5.1.



Obrázek 5.1 Zdroj: Vlastní

Komponent diagram soustavy této bakalářské práce vyhotoven v jazyce UML.

## Pneumatická soustava

Pneumatická soustava není předmětem této bakalářské práce. Jedná se o autonomní systém, který umožňuje nezávislé ovládání pěti krokových motorů, jež řídí ventily a tím mění tlak na jednotlivých výstupech. Tyto výstupy jsou navrženy tak, aby umožňovaly připojení pneumatických svalů, což jsou zařízení schopná konvertovat tlakovou energii vzduchu na mechanický pohyb. Tento systém poskytuje robustní řízení a flexibilitu při manipulaci s pneumatickými svaly a dalšími příslušenstvími v rámci příslušné aplikace.

## DeviceLoader

Tato komponenta je navržena tak, aby mohla efektivně komunikovat se zbytkem softwarového systému a řídit jednotlivé krokové motory prostřednictvím interakce se systémem. Tato třída slouží jako rozhraní pro komunikaci se soustavou, přičemž její hlavní funkcí je ovládání krokových motorů. Zbytek softwarového systému interaguje s touto komponentou jako prostředníkem pro komunikaci se soustavou. Přestože se tato komponenta skládá z řady funkcí, pro potřeby projektu je využívána především jedna základní funkce, která umožňuje řízení krokových motorů.

### getBoardX()

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

* on() – Tato funkce, která je zásadní pro aktivaci a přípravu desky pro další operace, musí být deklarována na začátku každého programu, ve kterém je tato soustava používána. Její účelem je inicializovat desku a nastavit ji do aktivního stavu, aby bylo možné provádět další operace v rámci programu. Tato inicializační funkce zajišťuje správnou funkci celé soustavy a připravuje ji k provádění požadovaných úkolů.
* readA0() – Tato funkce je určena k odečtení hodnoty ve mV z integrovaného voltmetru v rámci soustavy. S každou deskou je spojen svůj vlastní voltmetr, což vyžaduje specifikaci desky, pro kterou se měření provádí. Každé měření vyžaduje tedy specifikaci cílové desky, aby bylo zajištěno správné odečtení hodnoty z odpovídajícího voltmetru. Tímto způsobem se zajišťuje přesnost a spolehlivost měření v rámci celé soustavy.
* go\_forward(speed, steps) – Tato funkce představuje základní pracovní mechanismus, který umožňuje softwaru nastavit požadovaný posun krokového motoru. Její parametry zahrnují dvě proměnné: speed (rychlost) a steps (kroky).

Proměnná speed určuje rychlost, s jakou se provede jednotlivá akce. Například, pokud je hodnota speed nastavena na 20, znamená to, že se krokový motor bude točit rychlostí 20 kroků za sekundu.

Druhá proměnná, steps, určuje, o kolik kroků se má krokový motor posunout. Pokud tedy bude funkce volána s parametry go\_forward(20, 100), znamená to, že se krokový motor bude pohybovat vpřed po dobu 5 sekund, přičemž každý krok bude odpovídat 100 krokům..

* go\_backward(speed, steps) – Tato funkce je analogií předchozí funkce, avšak s tím rozdílem, že umožňuje provádět jednotlivé kroky pozpátku. Stejně jako předchozí funkce, i tato má schopnost přijímat záporné hodnoty, což umožňuje flexibilitu v použití. V některých případech může být jedna z těchto funkcí redundantní, protože obě mají podobnou funkčnost. Nicméně, v závislosti na konkrétním použití může být jedna z funkcí preferována před druhou.

## Main

Tato první komponenta, vyvinutá v rámci této bakalářské práce, představuje základní kámen celého grafického uživatelského rozhraní. Právě v této komponentě probíhá inicializace a vytváření grafického uživatelského rozhraní (GUI). Komponenta navíc komunikuje s dalšími třemi komponentami: AdminWindow, UserWindow a databází.

A screenshot of a computer

Description automatically generatedV této komponentě se nachází základní třída nazvaná App. Tato třída slouží k inicializaci hlavní okenní aplikace, jak je znázorněno na Obrázku 5.2. Jejím úkolem je zahájit proces vytváření uživatelského rozhraní a umožnit interakci s ním prostřednictvím různých funkcionalit poskytovaných dalšími komponentami.

Obrázek 5.2 Zdroj: Vlastní

Základní okno grafického uživatelské rozhraní

Prvním oknem, které se zobrazí při spuštění aplikace, je zároveň autentizačním oknem. Toto okno je vytvořeno pomocí grafické knihovny CustomTkinter. Jeho pozice, stejně jako pozice všech ostatních oken, je vypočítávána dynamicky. Kód 5.1 obsahuje parametry width (šířka) a height (výška), které slouží k 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

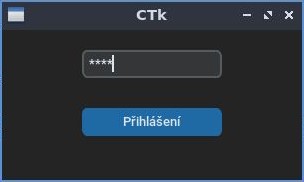
screen\_y = (screen\_height - height) // 2

# Nastavení jednotlivých parametrů

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

Na Obrázku 5.2 v pravém horním rohu je zobrazena trojice základních tlačítek, která slouží k manipulaci s oknem, jako je minimalizace na lištu, přizpůsobení obrazovce a zavření okna. Tyto funkce jsou automaticky vytvořeny prostřednictvím grafické knihovny CustomTkinter, a proto se v kódu nenachází žádný specifický kód spojený s těmito tlačítky. Tato funkčnost je standardní součástí grafického uživatelského rozhraní a umožňuje uživatelům snadnou interakci s oknem aplikace.

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

 Prvním widgetem je CTkEntry pole. Jedná se o políčko, do kterého může uživatel zapisovat znaky. Následně je možné pomocí funkce get() získat tyto znaky jako řetězec a pracovat s nimi v rámci projektu. Tato komponenta má dva parametry.

Obrázek 5.3 Zdroj: Vlastní

GUI v tmavém režimu

Prvním z nich je placeholder\_text, což je text, který je uvnitř pole a slouží jako nápověda pro uživatele, co má do daného pole zadat. Tento placeholder se smaže poté, co uživatel začne vyplňovat hodnoty. Druhým parametrem je parametr show. Tento parametr je především vhodný v případě, kdy je potřeba skrýt, co uživatel do pole zadává, jako například při zadávání hesla.

Druhým widgetem je CTkButton, což je tlačítko, které má dva parametry. Prvním parametrem je text, který umožňuje napsat do tlačítka libovolný text. Druhým parametrem je command (příkaz), který slouží k přiřazení specifického příkazu k tlačítku.

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

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

Například v tomto případě je tlačítko použito k provedení série akcí, které autentizují uživatele. Tato funkce je současně přiřazena speciální bindovací funkci, která umožňuje určit, jakou klávesovou zkratku použijeme pro vyvolání tohoto úkonu. V tomto kódu byla zvolena standardní klávesa Enter.

Autentizační mechanismus byl navržen jednoduše: pokud pracovník zadá uživatelské heslo, dostane se na uživatelské okno; pokud zadá heslo pro administrátora, dostane se do administrátorské části. Tato hesla jsou ověřována v databázi a pokud databáze najde shodu, uživateli je umožněn přístup na základě informace, kterou mu databáze vrátí.

Pro zajištění bezpečnosti uložených hesel je heslo v databázi hashováno. To znamená, že když uživatel potvrdí své heslo, nejprve se převede na hash a až poté je posláno na ověření do databáze. Pokud uživatel zadá špatné přístupové heslo, obdrží odpovídající hlášku, jak je znázorněno na Obrázku 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 okno typu CTkToplevel, buď pro administrátorskou nebo uživatelskou část. Toto nové okno se otevře a současné okno, ve kterém probíhalo zadávání hesla, bude automaticky zavřeno.

CTkToplevel je speciální typ okna, který uchovává informace o svém nadřazeném okně. Díky tomu je schopno se k nadřazenému oknu vrátit, aniž by muselo být vytvořeno nové okno. To umožňuje efektivněji spravovat okna v aplikaci a poskytuje uživateli plynulý zážitek z používání.

## AdminWindow

Komponenta AdminWindow je přístupná pouze po zadání správného administrátorského hesla. Administrátor by měl být osobou, které lze svěřit náročnější úkoly bez takové míry kontroly jako u normálních uživatelů. Tato premisa byla zohledněna i v tomto projektu, což umožňuje administrátorovi provádět změny v systému.

A screenshot of a computer

Description automatically generatedPo úspěšném přihlášení pomocí správného hesla uvidí administrátor první obrazovku, kde bude moci vybrat mezi dvěma úkony, jak je znázorněno na Obrázku 5.5. Tato obrazovka mu poskytuje možnost provést specifické úkony v rámci svých administrátorských práv..

Obrázek 5.5 Zdroj: Vlastní

GUI pro administrátora pro výběr jeho činnosti

V nejspodnější části okna se nachází tlačítko s názvem "Hlavní Stránka". Toto tlačítko umožňuje návrat zpět na autentizační obrazovku. Je vhodné především pro administrátory, kteří chtějí například otestovat nový převodní vzorec po jeho nastavení. Tlačítko jim poskytuje možnost vrátit se na autentizační obrazovku, aniž by museli celou aplikaci vypínat a znovu spouštět. Po stisknutí tlačítka se aktivuje funkce, která zavře aktuální okno a znovu otevře autentizační obrazovku.

V prostřední části okna se nachází 5 tlačítek. Tyto tlačítka umožňují administrátorovi přistoupit k databázovému oknu, jak je znázorněno na Obrázku 5.6. Byla zvolena metoda vytvoření tlačítka pro každý sval z důvodu, že jméno tohoto tlačítka je předáno do dalšího okna, jak je uvedeno v Kódu 5.3, a následně s těmito hodnotami lze 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 je následně využívána k vybrání požadovaných tabulek z databáze, jak je znázorněno v Kódu 5.4. V jazyce Python je k dispozici funkce f-string, která umožňuje vkládat proměnné přímo do řetězců pomocí složených závorek {} před nimi. Tím se kód stává přehlednější, než je tomu u klasického zápisu, kdy se musí jednotlivé řetězce a proměnné spojovat pomocí logického operátoru plus +. V tomto případě byla tato metoda použita pro všechny čtyři převodní vzorce, což usnadňuje čitelnost a údržbu kódu.

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 pěti hlavních částí. První část obsahuje tlačítko pro návrat na hlavní administrátorskou obrazovku, kde si uživatel může vybrat jiný sval pro práci s jeho databází nebo provést jinou akci.

Dále jsou zde čtyři téměř identické části, lišící se pouze tím, z jaké tabulky v databázi načítají data. Význam jednotlivých převodních vzorců je dále vysvětlen v kapitole 5.5 UserWindow. Tyto části slouží k práci s databází a umožňují administrátorovi provádět různé operace v závislosti na zvoleném svalu.

Vedle názvu tabulky se nachází číslo, které indikuje, který převodní vzorec je zrovna aktivní. Například na obrázku 5.6, vedle vzorce pro sval1 pro převodní vzorec mezi kroky a mV je číslo 1. To značí, že pokud se uživatel rozhodne využít výpočet, který bude počítat s tímto vzorcem, bude použit právě tento převodní vzorec z databáze.

Každá část okna obsahuje tři prvky. Prvním prvkem jsou vstupní pole, ve kterých se nachází sklon, posun a popis funkce. Tento prvek byl záměrně zvolen jako vstupní pole, aby administrátor nemusel složitě spouštět editování jednotlivých vzorců. Stačí pouze změnit libovolnou část převodního vzorce na požadovanou hodnotu a následně využít druhý prvek, kterým je tlačítko "upravit".

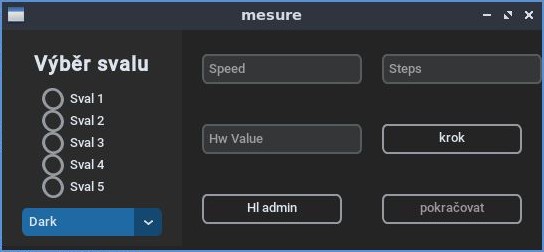
Toto tlačítko slouží k uložení provedených změn do databáze. Zároveň disponuje funkcí, pomocí které se volí aktuální převodní vzorec. V prostředí, kde soustava obsahuje mnoho převodních vzorců pro různě těžké hlavy pacientů, je rychlá změna mezi těmito vzorci nezbytná. Stačí jednoduše zmáčknout tlačítko "upravit" a tím se nastaví nový aktuálně používaný převodní vzorec. Tímto způsobem je administrátor schopen rychle a efektivně pracovat s různými převodními vzorci v systému.

A screenshot of a computer

Description automatically generatedPosledním prvkem je tlačítko "nový". Pomocí tohoto tlačítka je administrátor schopen vytvářet nové převodní vzorce. Po stisknutí tlačítka se objeví nová vstupní pole spolu s tlačítkem "uložit", jak je znázorněno na Obrázku 5.7. Zde musí administrátor vyplnit vstupní pole příslušnými hodnotami. Po dokončení vyplnění stiskne tlačítko "uložit" a tím se nový převodní vzorec uloží do databáze. Tímto způsobem je umožněno administrátorovi snadno a rychle přidávat nové převodní vzorce do systému.

Obrázek 5.7 Zdroj: Vlastní

GUI databáze, když administrátor vytváří nový převodní vzorec.



Obrázek 5.8 Zdroj: Vlastní

Grafické uživatelské rozhraní pro administrátora pro vytváření měření.

Druhé pracovní okno administrátorské části slouží k vytváření měření pro následnou práci na vytvoření nových převodních vzorců (viz Obrázek 5.8). Toto okno je rozděleno do dvou rámců, což umožnilo vytvoření přehledného uživatelského prostředí s jasně rozdělenými prvky.

V levé části okna se nachází CtkRadioButton, což je speciální typ tlačítka, který kombinuje vlastnosti tlačítka a výběrového pole. Uživatel má možnost vybrat jednu možnost ze seznamu, ale na rozdíl od běžného výběrového pole může vybrat pouze jednu možnost. Tato komponenta je velmi vhodná pro případy, kdy je třeba uživateli umožnit vybrat pouze jednu možnost, jako například v tomto případě, kde je nutné zajistit, aby uživatel mohl ovládat pouze jeden sval.

Ve spodní části tohoto rámu se nachází komponenta CtkOptionMenu. Tato komponenta představuje rozbalovací seznam, který zobrazuje pouze aktivní prvek. Když je tato komponenta rozkliknuta, zobrazí všechny možnosti nastavení, jako jsou "Dark, Light, System". Tato nastavení určují barevné schéma aplikace. Díky CustomTkinteru je možné nastavit tmavý nebo světlý režim manuálně nebo automaticky podle systému. V tomto projektu byla vybrána možnost automatického barevného režimu podle systému, což umožňuje aplikaci automaticky přizpůsobit nastavení systému, na kterém je spuštěna.

V druhém fragmentu se nachází hlavní pracovní část tohoto okna. Zde jsou umístěna tři vstupní pole, do kterých uživatel musí zadat číselnou hodnotu.

Prvním vstupním polem je pole speed, kam administrátor musí zadat požadovanou rychlost, kolik kroků chce, aby soustava vykonala.

Druhým vstupním polem je pole steps, kam administrátor musí zadat, kolik kroků má soustava vykonat.

Posledním vstupním polem je pole s place holder textem "HW Value", kam administrátor musí ručně zadávat hodnoty z externího tlakoměru v jednotkách mbar.

Tato vstupní pole také disponují validačním mechanismem, který zamezuje zadání jiného vstupu než čísla. Tím je zajištěna integrita dat a správnost vstupů od uživatele.

Tlačítko "Krok" je schopno potvrdit celý proces a provést měření provedením požadovaného postupu na krokovém motoru.

S tímto tlačítkem "Krok" je spojeno také tlačítko "Pokračovat". Toto tlačítko slouží jako bezpečnostní prvek pro ověření, zda administrátor připojil všechny potřebné části a je připraven provádět měření na novém svalu. Po stisknutí tlačítka "Pokračovat" je provedena kontrola správnosti nastavení a připojení součástí a teprve poté je umožněno provést krokové měření. Tímto způsobem je zajištěna bezpečnost a spolehlivost celého procesu měření.

Proces měření dat je navržen tak, aby administrátor měl k dispozici jeden externí tlakoměr, pomocí kterého může měřit tlak v jednotlivých svalech soustavy. Proto soustava potřebuje určité zajištění, aby se zabránilo situacím, kdy by administrátor začal provádět měření na svalu, aniž by například již měl přepojený tlakoměr na nový sval.

Toto tlačítko slouží jako pojistka: když se administrátor rozhodne měřit jiný sval, soustava v programu ověří, zda je současný sval jiný než minulý sval, a tím spustí sérii akcí. Soustava aktivuje tlačítko pro potvrzení, že vše je připraveno k měření. Dokud toto tlačítko nebude stisknuto, soustava nedovolí administrátorovi pokračovat v práci na soustavě, čímž zajišťuje bezpečnost a správnost prováděných operací.

Další akcí, která se spustí, je automatické navrácení původního měřeného svalu na technickou nulu. Program si pamatuje, kolik kroků od začátku udělal, a poté, když uživatel zvolí jiný sval, program začne automaticky sjíždět na technickou nulu. Až poté, co soustava sjede na technickou nulu, bude administrátorovi umožněno stisknout tlačítko "Pokračovat" a provést následné měření.

Tímto způsobem je zajištěno, že soustava se před každým novým měřením vrátí do výchozí polohy, což je důležité pro správnost a konzistenci prováděných měření. Administrátor je také chráněn před nechtěnými chybami, protože soustava automaticky provádí proces sjíždění na technickou nulu, než mu umožní pokračovat v další práci.

Hlavní funkcí tohoto měření je zaznamenávání jednotlivých hodnot měření do souboru .xlsx. Tuto funkcionalitu zajišťuje knihovna openpyxl. Program do tohoto souboru ukládá následující informace:

* Počet kroků motoru od začátku měření
* Aktuální hodnota interního voltmetru v mV
* Hodnota fyzického tlakoměru v mbar
* Rychlost, jakou se v danou chvíli krokový motor pohyboval

Všechny tyto hodnoty jsou ukládány do souboru, dokud administrátor nezmění sval. Poté soustava provádí všechny kroky spojené s tímto úkonem a uloží tento soubor na plochu počítače, aby byl snadno dostupný pro další práci. Tím je zajištěno, že veškerá data jsou systematicky zaznamenána a uložena pro budoucí použití.

Tento program je navržen tak, aby fungoval pro 5 jednotlivých svalů, přičemž každý z nich funguje úplně stejným způsobem, s výjimkou jejich názvu. Existuje však riziko vážné redundance kódu, pokud by se každý sval musel řešit individuálně, což by mohlo vést k duplikaci kódu a zvýšení náročnosti na údržbu.

Aby se tomu předešlo, vytvořil se slovník, ve kterém jsou definovány jednotlivé desky, které komunikují s jednotlivými svaly. Poté se ze slovníku vybírá ta správná deska podle čísla svalu. Tímto způsobem se minimalizuje duplikace kódu a zajišťuje se, že veškeré operace se svaly jsou prováděny efektivně a jednotně.

Kód 5.5 obsahuje tuto logiku, která umožňuje programu dynamicky pracovat s více svaly a minimalizuje riziko chyb spojených s duplikací kódu.

Kód 5.5: 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 5.5 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

UserWindow představuje poslední část této bakalářské práce a je určena pro uživatele (user). Tato komponenta je spuštěna pouze v případě, že se uživatel přihlásí pomocí správného uživatelského hesla, které odpovídá heslu uloženému v databázi a je označeno jako uživatelské. UserWindow komunikuje s hlavním modulem (main), databází a modulem pro načítání zařízení (deviceloader).

A screenshot of a computer

Description automatically generatedToto okno se skládá ze tří rámců (framů), které jsou zobrazeny na Obrázku 5.9. Každý z těchto rámců slouží k organizaci a přehlednému zobrazení různých funkcí a informací, které jsou uživateli k dispozici. UserWindow je navrženo tak, aby uživatel mohl snadno a efektivně pracovat s aplikací a využívat všechny její funkcionality v souladu s jeho oprávněními a potřebami.

Obrázek 5.9 Zdroj: vlastní

Uživatelské GUI pro nezávislé ovládání 5 svalů

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

Druhým typem widgetu je CTkSwitch, což je komponenta umožňující uživateli zapnout nebo vypnout určitou funkci. V tomto případě slouží k přepínání mezi light mode a dark mode. Když uživatel poprvé spustí toto okno, automaticky se mu zapne režim, který je předvolený podle systému. Avšak uživatel má možnost tuto volbu změnit podle svých preferencí.

CTkSwitch poskytuje uživateli snadný způsob, jak přizpůsobit zobrazení aplikace svým potřebám a vkusu. Tímto způsobem může uživatel přizpůsobit prostředí aplikace tak, aby mu co nejlépe vyhovovalo a usnadňovalo práci či interakci s aplikací.

Druhým rámcem je "Right Frame". Tato část obsahuje radiobuttony pro výběr jednotek, které uživatel může vybrat podle svých preferencí ohledně ovládání pneumatické soustavy. Radiobuttony byly zvoleny, protože je nutné zajistit, aby uživatel vybral pouze jednotnou jednotku pro ovládání. Výběr pomocí chechboxů by vyžadoval další funkce, které by zajistily, že každý nový výběr uživatele automaticky zruší předchozí výběr. Radiobuttony řeší tuto potřebu automaticky tím, že uživatel může vybrat pouze jednu možnost ze seznamu.

Tento radiobutton má také vlastní validaci, která zajišťuje, že uživatel, který nevybere žádnou jednotku, bude upozorněn na tuto skutečnost a nebude mu umožněno pokračovat dále. Tím je zajištěno, že uživatel bude informován o nutnosti vybrat jednotky pro zadávání a že nebude možné pokračovat v procesu, dokud tato podmínka není splněna.

Druhým prvkem v tomto rámu je tlačítko "Main". Pomocí tohoto tlačítka se uživatel může vrátit zpět na první okno, kde zadával heslo pro přihlášení. Toto tlačítko je vhodné především pro administrátory, kteří současně pracují jako uživatelé a mohli by zapomenout změnit převodní vzorce. Díky tomuto tlačítku se eliminuje potřeba vypnout a znovu spustit aplikaci, což uživatelům ušetří čas a zlepší jejich uživatelskou zkušenost.

Hlavním prvkem okna je "main frame". V tomto prvku bude uživatel schopen zadávat hodnoty, které ovlivní tlak pneumatické soustavy. Každé entry pole je označeno názvem příslušného svalu jako placeholder. Pro zajištění bezpečnosti a správného zadávání čísel byla provedena validace vstupu. Program je schopen přijímat pouze číselné hodnoty, protože neví, na jakou hodnotu by měl převést text. Stejný typ validace se používá i v dalších částech programu pomocí regulárních výrazů.

Dalším bezpečnostním prvkem, který tento rámec obsahuje, je CTkProgressBar, který se nachází nad entry polem. Tato komponenta zobrazuje aktuální stav tlaku pneumatického svalu. Slouží jako informační prvek pro uživatele, který jim umožňuje určit, jaký je momentální stav natlakování soustavy a kolik je ještě možné přidat nebo ubrat tlaku. Tento prvek není pouze vizuální kontrolou, ale je také nastaven tak, aby upozornil uživatele, pokud se pokusí překročit povolenou hranici tlaku svalu. To je důležité zejména pro ochranu pacienta, který je v blízkosti těchto svalů.

Horní hranice je nastavena tak, aby se zabránilo možnému nevratnému poškození pneumatického svalu a tím i ohrožení pacienta. Spodní hranice je stanovena tak, aby se zabránilo poškození ventilu v případě, že by uživatel neustále snižoval tlak, i když by byl ventil již zavřen. Tím se minimalizuje riziko poškození ventilu a zajišťuje bezpečnost a spolehlivost systému.

Tento rámec obsahuje jediné tlačítko, které spouští celkový krok celé soustavy najednou. Soustava je konfigurována tak, že pokud by měla být posunuta o 0 kroků, tento krok se 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, která hrozila zejména ve dvou případech. V této soustavě se nachází 5 svalů, pro její správné fungování jsou potřeba 4 převodní vzorce.

Nejjednodušší jednotkou pro práci jsou kroky, protože se nemusí nijak upravovat. Stačí pouze přenést jejich přesnou hodnotu do funkce go\_forward nebo go\_backward.

V tomto projektu byla možnost zadávání záporných hodnot řešena tak, že byla nastavena podmínka, která rozděluje tyto hodnoty na kladné a záporné. Pokud se jednalo o zápornou hodnotu, která měla vstoupit do funkce go\_backward(), byla tato hodnota vložena jako záporná, čímž se její zápornost zneplatnila a hodnota tak vstupovala do funkce jako kladná. Tímto způsobem byla zachována konzistence a správné fungování funkce pro pohyb krokových motorů.

Redundance v této části kódu byla elegantně vyřešena pomocí jednoho slovníku a jednoho pole, jak je patrné z Kódu 5.6. Tímto způsobem je program schopen na jednom řádku dosáhnout stejného výsledku, kterého by jinak dosáhl opakovaným psaním téže funkce pro každý sval. Stačí použít jeden for cyklus, který provede akci pro každý sval, což výrazně zjednodušuje a zkracuje kód..

Kód 5.6: Omezení redundance

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

Pro změnu tlaku za pomoci milivoltů je nutné použít převodní rovnici z databáze. Vzhledem k tomu, že v databázi není pouze jedna převodní rovnice, program musí zjistit, jaký převodní vzorec má načíst z dané tabulky. K tomu slouží Kód 5.6. Tento kód omezuje redundanci tím, že načítá informace pro každý index, který může nabývat hodnot od 1 do 5. Poté se tyto informace spojí se stringem "sval" a dále se spojí s příslušnou jednotkou, kterou získáme z radiobuttonu. Následně se může provést výběr celého požadovaného řádku s příslušným svalem. Tímto způsobem je zajištěno efektivní a dynamické načítání převodních vzorců bez zbytečné redundance v kódu.

Kód 5.7: Výběr příslušné rovnice z databáze

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

# Spojení navráceného SQL příkazu do stringu

cislo\_vzorce = cur.fetchone()

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

Po načtení příslušného převodního vzorce program provádí výpočet konečné hodnoty, na kterou má dojet. Toho dosahuje pomocí aktuální hodnoty v mV načtené z tlakoměru, ke které přičítá posun, který si uživatel sám volí.

Následně je nutné převést startovní a konečné hodnoty v mV na jednotky, které je krokový motor schopný přijmout. Pro toto je potřeba spočítat startovní kroky, tedy kolik kroků bylo provedeno od začátku procesu. Funkce v rámci DeviceLoaderu uchovávající informace o počtu provedených kroků se vymaže pokaždé, když motor provede krok zpět, proto je nepoužitelná. Alternativní možností by bylo uchovávat tuto informaci v programu, ale kvůli přesnosti nastavení hodnot je tato hodnota počítána pokaždé znovu.

Poté se vypočítá, kolik kroků motor musí provést, aby se dostal do této pozice z technické nuly. Nakonec se tyto dvě hodnoty odečtou a program si tento výsledek uloží do pole, do kterého se ukládají veškeré hodnoty svalů. Po naplnění tohoto pole program začne provádět pohyb krokových motorů, čímž dosáhne požadovaného tlakového stavu.

Vyřešení dvou jednotek, které nemají žádné referenční hodnoty ze soustavy, bylo zásadním úkolem. Pro milivoltovou jednotku je soustava schopna odečíst hodnoty z integrovaných voltmetrů, což je relativně snadné. Ovšem pro jednotky tlaku a vzdálenosti tato možnost neexistuje. Jednou z možností by bylo přidání senzorů, které by dodaly referenční hodnoty, ale to by nejen bylo ekonomicky náročné, ale také by vyžadovalo přepracování celé soustavy.

Proto byla zvolena metoda, kdy se jako referenční hodnoty využívají hodnoty v mV, které se následně převedou na příslušné jednotky pomocí převodní funkce. Obě tyto metody fungují na stejném principu: pomocí referenční hodnoty se vypočítají počáteční a konečný krok, které se následně odečtou a získá se výsledek. Pouze u jednotek mm a mbar je potřebný ještě jeden mezistupeň, kdy se aktuální hodnota z mV převede na požadovanou jednotku, čímž se vytvoří referenční hodnota.

Po provedení veškerých úprav tlaků se přepočítají hodnoty progress barů na nové hodnoty, aby se zajistilo, že při dalším opakování měření nedojde k jejich překročení. Tímto způsobem je zajištěno, že proces měření probíhá bezpečně a spolehlivě.

# Uživatelská dokumentace

Aplikace se spouští

# 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