ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

Katedra informačních a komunikačních technologií v lékařství

Modul hlasového ovládání domácnosti pro lidi se sníženou soběstačností

Home voice control module for people with low self-sufficiency

Bakalářská práce

Studijní program:	Biomedicínská a klinická technika
Studijní obor:	Informační a komunikační technologie v lékařství

Autor bakalářské práce:	Serhij Solomon
Vedoucí bakalářské práce:	Ing. Vít Janovský

Kladno 2021

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	**Solomon**	Jméno:	**Serhij**	Osobní číslo:	**465975**
Fakulta:	**Fakulta biomedicínského inženýrství**				
Garantující katedra:	**Katedra informačních a komunikačních technologií v lékařství**				
Studijní program:	**Biomedicínská a klinická technika**				
Studijní obor:	**Informační a komunikační technologie v lékařství**				

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Modul hlasového ovládání domácnosti pro lidi se sníženou soběstačností

Název bakalářské práce anglicky:

Home voice control module for people with low self-sufficiency

Pokyny pro vypracování:

Sestavte modul hlasového ovládání. Ten musí respektovat potřeby lidí se sníženou soběstačností (senioři a lidé na vozíku) a komptabilitu s rozhraním řídícího systému laboratoře (Tecomat Foxtrot). Implementujte příkazy pro ovládání domácnosti např.: rozsvit světlo, otevři okno, pusť rádio, vypni vše apod. Pokuste se vytvořit i složitější strukturu rozhovoru (výběr světla, intenzity, info o stavu). Výsledné řešení optimalizujte testováním s cílovou skupinou.

Seznam doporučené literatury:

[1] Codelabs, Výukový portál společnosti Google, -, https://codelabs.developers.google.com
[2] NEDVĚD, Jakub, Evaluace hlasových dialogových systémů, 2013
[3] Kenichiro Noda , Google Home: smart speaker as environmental control unit, Disability and Rehabilitation: Assistive Technology, ed. -, DOI: 10.1080/17483107.2017.1369589, 2018, ISBN -

Jméno a příjmení vedoucí(ho) bakalářské práce:

Ing. Vít Janovský

Jméno a příjmení konzultanta(ky) bakalářské práce:

Ing. Pavel Pelán

Datum zadání bakalářské práce: **20.01.2020**
Platnost zadání bakalářské práce: **19.09.2021**

doc. Ing. Karel Hána, Ph.D.
podpis vedoucí(ho) katedry

prof. MUDr. Ivan Dylevský, DrSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

30. 4. 2021
Datum převzetí zadání

Podpis studenta(ky)

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci s názvem „Modul hlasového ovládání domácnosti pro lidi se sníženou soběstačností“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k diplomové práci.

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 10.5.2021 …...……...……...

Serhij Solomon

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce, panu Ing. Vítu Janovskému, za vedení práce, podporu a poskytnuté konzultace. Děkuji také panu Ing. Pavlu Pelánovi za konzultace a poskytnuté rady ohledně programování PLC.

ABSTRAKT

Modul hlasového ovládání domácnosti pro lidi se sníženou soběstačností

Hlavním cílem této práce bylo sestavit modul hlasového ovládání pro osoby se sníženou soběstačností. V současné době není k dispozici žádný volně dostupný hlasový asistent s podporou českého jazyka, proto bylo zvoleno vlastní řešení pomocí mobilní aplikace a webové služby. Mobilní aplikace čeká na spouštěcí slovo, které jakmile zaznamená, tak aktivuje online převod řeči na text. Výsledný text následně posílá v HTTP požadavku webové službě, která provádí analýzu textu a zjišťuje záměr uživatele. Poté, co webová služba zjistí záměr uživatele, provede požadovanou akci – pomocí Modbus TCP protokolu pošle řídící jednotce bytu odpovídající instrukce.

Provedené testování výsledného modulu prokázalo, že i lidé odkázaní na vozík, nebo neschopni plně pohybovat pažemi, by byli schopni ovládat zařízení pouhým hlasem, například otevřít dveře. Avšak někdy je potřeba spouštěcí slovo zopakovat více krát, nebo se naopak spustí naslouchání příkazu i přes to, že spouštěcí slovo nebylo vysloveno.

Klíčová slova

Hlasový asistent, rozpoznávání řeči, PLC, Modbus TCP, webová aplikace

ABSTRACT

Home voice control module for people with low self-sufficiency

The main goal of this thesis was to develop home voice control module for people with low self-sufficiency. There are no available virtual assistants, which would support Czech language, so the custom solution was performed using smartphone application and web service. Smartphone application waits for wakeup word and when detects one, starts the online speech recognition service. When the speech recognition is done, the resulting text is sent to web service via HTTP request. Web service then analyzes the text containing command and attempts to understand user's intent. If the intent is known, web service sends the request with instructions to the home control unit via Modbus TCP protocol.

Performed tests showed up that module can help people, who are unable to fully move arms, to control devices using their voice, for example to open a door. But there are still relatively many false alarms on spotting wake-up word.

Keywords

Virtual assistant, speech recognition, PLC, Modbus TCP, web application

Obsah

Seznam symbolů a zkratek ... **6**

1 Úvod ... **7**

1.1 Hlasový asistent ... 7

1.2 Motivace ... 7

1.3 Laboratoř personalizované medicíny ... 7

2 Přehled současného stavu ... **8**

2.1 Přehled hlasových asistentů ... 8

2.1.1 Princip funkce hlasového asistenta ... 8

2.1.2 Přínos hlasových asistentů pro osoby se zdravotním postižením ... 8

2.2 Google Assistant ... 9

2.2.1 Vlastní příkazy pro Google Assistant ... 10

2.3 Amazon Alexa ... 11

2.4 Siri ... 12

2.5 Cortana ... 12

2.6 Bixby ... 13

2.7 Mycroft AI ... 13

2.8 Shrnutí rešerše hlasových asistentů ... 14

2.9 Rozpoznávání řeči ... 14

2.10 Skryté Markovovy modely ... 15

2.11 Neuronové sítě ... 15

2.12 CMUSphinx ... 16

2.12.1 Podporované jazyky ... 16

2.12.2 Vytvoření modelu pro další jazyk ... 16

2.12.3 PocketSphinx ... 17

2.13 Google API pro rozpoznání řeči ... 17

3 Cíle práce ... **18**

4 Návrh softwaru ... **19**

4.1 Potřeby lidí se sníženou soběstačností ... 19

4.2 Analýza požadavků výsledného software ... 19

4.2.1 Půdorys laboratoře ... 20

4.2.2 Webová aplikace ovládání ... 20
4.2.3 Ovladatelná zařízení v laboratoři ... 23
4.3 Návrh řešení hlasového ovládání ... 25
4.4 Řídící jednotka laboratoře ... 25
4.4.1 Programovatelný automat ... 25
4.4.2 Tecomat Foxtrot ... 26
4.4.3 Základní modul CP-1001 ... 26
4.4.4 Programování jednotky ... 27
4.5 Úprava PLC programu ... 28
4.5.1 Modbus ... 29
4.6 Analýza požadavků webové aplikace ... 30
4.7 Technická specifikace webové aplikace ... 31
4.8 Raspberry Pi ... 32
4.9 Návrh databáze ... 33
4.9.1 Tabulka Keywords ... 33
4.9.2 Tabulka Values ... 34
4.9.3 Tabulka Commands ... 34
4.9.4 Tabulka Skills ... 35
4.9.5 Tabulka ModbusCommands ... 35
4.9.6 Tabulka CommandSequence ... 36
4.9.7 Tabulka Response ... 36
4.9.8 Pohled ValuesView ... 36
4.9.9 Pohled CommandsView ... 37
4.9.10 Pohled CommandSequenceView ... 37
4.10 Analýza požadavků mobilní aplikace ... 38
4.11 Technická specifikace mobilní aplikace ... 38
5 Implementace ... 39
5.1 Přiřazení zařízení k Modbus objektům ... 39
5.2 Implementace webové aplikace ... 40
5.2.1 Komponenty typu core ... 41
5.2.2 Komponenty typu controller ... 44
5.2.3 Komponenty typu view ... 44

5.2.4 Komponenty typu model ... 44
5.2.5 Komponenty typu helper ... 47
5.2.6 Komponenty typu skill ... 49
5.2.7 Licence ... 49
5.3 Implementace mobilní aplikace ... 50
6 Uživatelská dokumentace ... 51
6.1 Hlasové příkazy asistenta ... 51
6.2 Administrační rozhraní webové aplikace ... 52
6.2.1 Správa hlasových příkazů ... 53
6.2.2 Správa Modbus příkazů ... 55
6.2.3 Správa klíčových slov ... 56
6.2.4 Správa přiřazených hodnot ke klíčovým slovům ... 57
7 Testování ... 58
7.1.1 Průběh testování ... 58
7.1.2 Vyhodnocení testování modulu hlasového ovládání ... 60
8 Diskuse ... 61
9 Závěr ... 63
Seznam použité literatury ... 64
Příloha A: Obsah přiloženého CD ... 68

Seznam symbolů a zkratek

Seznam zkratek

Zkratka	Význam
ASR	Automatické rozpoznávání řeči (Automatic Speech Recognition)
CPU	Centrální výpočetní jednotka (Central Processor Unit)
DBMS	Systém řízení báze dat (Database management system)
LED	Elektroluminiscenční dioda (Light-Emitting Diode)
MVC	Model-View-Controller
PLC	Programovatelný logický automat (Programmable Logic Controller)
PWM	Pulzně šířková modulace (Pulse Width Modulation)
RGB	Barevný model (Red, Green, Blue)
RTC	Obvod reálného času (Real Time Clock)
STT	Převod řeči na text (Speech To Text)
TTS	Syntéza textu do řeči (Text To Speech)
URL	Jedinečný lokátor zdroje (Uniform Resource Locator)
UCEEB	Univerzitní centrum energeticky efektivních budov

1 Úvod

V posledních letech jde vývoj nových technologií rychlým tempem dopředu, a díky tomu také vývoj asistivních technologií.

Lidé, kteří mají určité postižení nebo jsou po zranění, mívají problém vykonávat jinak běžné činnosti. Může se jednat například o zhoršení vnímání nebo úplnou ztrátu některého z lidských smyslů, omezení v pohybu apod. Asistivní technologie mají za úkol tyto funkce nahradit a usnadnit tak vykonávání daných činností.

1.1 Hlasový asistent

Hlasový asistent je jedním ze skupiny asistivních technologií. Jedná se o digitální zařízení, které prostřednictvím přirozené řeči interaguje s uživatelem. Může jít například o otázku, na kterou se tento asistent pokusí najít odpověď na internetu, nebo příkaz, který se pokusí vykonat, pokud mu bude rozumět. Díky tomu je možné například nakoupit v e-shopu při žehlení (nebo jiné fyzické činnosti), nastavit cíl navigace na určité místo, někomu zavolat, nebo vybrat stanici rádia při řízení, aniž by bylo nutné se na malou chvilku přestat věnovat řízení. V neposlední řadě mohou hlasoví asistenti sloužit jako společnost. Například Siri od Apple umí na požádání říct vtip.

1.2 Motivace

Kormě komfortu je ale tento druh asistence nenahraditelnou pomocí pro lidi se zrakovým nebo pohybovým postižením, kdy ovládání pomocí tlačítek a jiných ovládacích prvků není buď vůbec možné, nebo je velmi obtížné.

Tato práce se zabývá sestavením modulu hlasového ovládání domácnosti, který bude respektovat potřeby lidí se sníženou soběstačností (senioři a lidé na vozíku) a bude kompatibilní s řídícím systémem v laboratoři personalizované telemedicíny.

1.3 Laboratoř personalizované medicíny

Tato laboratoř se nachází v Univerzitním centru energeticky efektivních budov (UCEEB) u Buštěhradu.

Jedná se o laboratoř typu „Living lab“ – bezbariérová a moderně vybavená garsoniéra, která spojuje vědce, lékaře a samotné uživatelé asistivních technologií, čímž umožňuje testování a hodnocení zdravotnických prostředků v domácím prostředí.

Jako řídící jednotka bytu je zde nainstalován PLC počítač Tecomat Foxtrot řady CP-1001.

2 Přehled současného stavu

2.1 Přehled hlasových asistentů

Podle studie z května 2019 od Adobe Analytics [1] se v současné době nejvíce používá těchto 5 hlasových asistentů:

- Alexa (Amazon)
- Google Assistant (Google)
- Siri (Apple)
- Cortana (Microsoft)
- Bixby (Samsung)

Některé lze pořídit také jako samostatné zařízení, které je přizpůsobené speciálně pro účely hlasového asistenta. Obsahují reproduktor, mikrofon a řídící obvod se softwarem, kde software je právě ten nejdůležitější rozdílový faktor, podle kterého se jednotliví asistenti od sebe nejvíce liší. [2]

2.1.1 Princip funkce hlasového asistenta

V zásadě jsou hlasový asistenti koncipovány tak, že čekají na spouštěcí slovo, například „Alexa“, „Ok Google“, „Hey Siri“ apod., a poté začnou nahrávat náš hlas a přenášet na server zaznamenanou zvukovou stopu. Samotné zpracování zvuku, převod mluveného slova na text a analýza textu pro zjištění zamýšleného záměru uživatele se provádí také na serveru. Po rozboru vysloveného požadavku uživatele se následně provede požadovaná akce, případně se asistent zeptá na doplňující informace, pokud by příkazu nerozuměl. [3]

2.1.2 Přínos hlasových asistentů pro osoby se zdravotním postižením

Díky tomu, že pro interakci s hlasovým asistentem stačí náš hlas, tak se lidem se zdravotním postižením s příchodem této technologie otevírá spoustu nových možností.

To, co je pro jiné lidi jen pomůcka, která možná vypadá působivě, se pro osoby upoutané na lůžko nebo na invalidním vozíku jeví jako nenahraditelná pomoc, a umožňuje jim vykonávat velké množství činností samostatně. Například nastavit teplotu v místnosti, zapnout klimatizaci, rozsvítit, zhasnout, zapnout televizi, otevřít okno.

2.2 Google Assistant

Pravděpodobně nejznámější hlasový asistent v české republice je Google Assistant. Napomáhá tomu také fakt, že podle statistiky [4] je OS Android je nejrozšířenější platformou pro mobilní zařízení v česku, a s tímto OS je zde používáno 77 % mobilních zařízení.

Jde o nástupce asistenta se jménem Google Now, který původně sloužil pro hlasové vyhledávání relevantních informace na zadaný dotaz. To hlavně díky tomu, že Google uchovával data o uživatelích – předchozí vyhledávání, navštívená místa apod. Pomocí těchto dat mohl následně sestavit profil uživatele a odhadnout věk, práci, koníčky, oblíbený sport, seriál atd. [5]

Aktivační frází pro spuštění Google asistenta je „OK Google“ nebo „Hey Google“.

Díky velmi rozšířenému ekosystému Google zařízení, lze tohoto asistenta najít ve smartphonech, chytrých reproduktorech, hodinkách či televizích. [6]

Schopnosti asistenta

Tento asistent umí například:

- Ovládat zařízení v chytrém bytě
- Vytvářet a připomínat naplánované události v kalendáři
- Najít informace na internetu
- Ovládat přehrávání hudby
- Přehrát video na Chromecast a dalších podporovaných zařízeních
- Nastavit časovač a budík
- Odeslat zprávu
- Spustit nainstalovanou aplikaci na mobilu
- Přečíst aktuální notifikace
- Překládat do jiného jazyka v reálném čase
- Spustit navigaci na vybrané místo

Ovládání spotřebičů chytré domácnosti

Jak lze vidět z prvního bodu výčtu schopností tohoto asistenta, poradí si dokonce s ovládáním domácích spotřebičů v chytrém bytě. Je však potřeba, aby byl daný spotřebič tímto asistentem podporován. Podle dokumentace je už nyní podporováno přes 50 000 zařízení od více než 10 000 různých výrobců. [7]

Přidání nového zařízení

Přidání a nastavení nového zařízení nebo spotřebiče chytré domácnosti mezi známá zařízení asistenta je možné pomocí aplikace Google Home. V této aplikaci lze také nalézt výčet podporovaných zařízení, nastavit jednotlivým zařízením názvy a určit, ve které místnosti se dané zařízení nachází. [8]

Podporované jazyky

V následující tabulce (Tab. 2.1) je přehled aktuálně podporovaných jazyků tímto asistentem. Jak je z tabulky patrné, v současné době bohužel není podporovaná čeština.

Tab. 2.1: podporované jazyky Google asistenta [9]

Země nebo oblast	**Podporovaný jazyk**
Austrálie	angličtina
Rakousko	němčina
Kanada	angličtina, francouzština
Dánsko	dánština
Francie	francouzština
Německo	němčina
Indie	angličtina, hindština
Irsko	angličtina
Itálie	italština
Japonsko	japonština
Jižní Korea	korejština
Mexiko	španělština
Holandsko	holandština
Norsko	norština
Singapore	angličtina
Španělsko	španělština
Švédsko	švédština
Anglie	angličtina
Spojené státy	angličtina

2.2.1 Vlastní příkazy pro Google Assistant

V některých případech je potřeba vytvořit vlastní příkaz, kterému bude asistent rozumět a provede požadovanou akci. Například pokud chceme ovládat zařízení, které není asistentem podporované. Pro takové případy může posloužit služba IFTTT. Jméno znamená „if this, then that“ (volně přeloženo: „pokud tohle, potom tamto“). Pomocí této služby lze nastavit, že pokud například řekneme: „Je čas vstávat“, tak se zapne chytrý kávovar. [10]

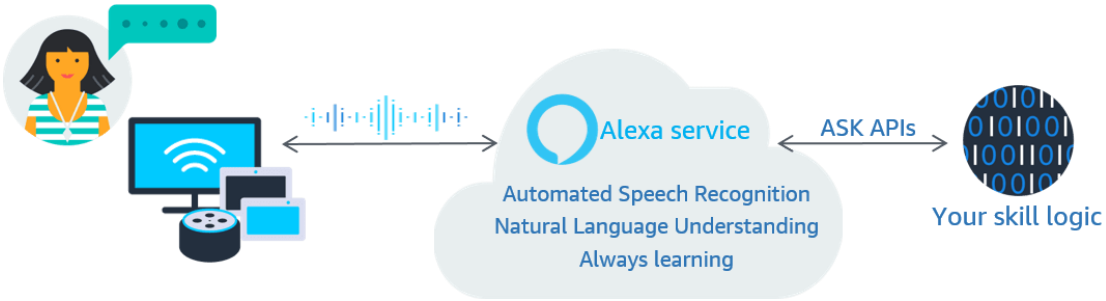
Pro vytvoření vlastního příkazu je potřeba se v IFTTT přihlásit, mezi službami najít „Google Assistant“, a kliknout na tlačítko „Connect“. Následně je možné vytvořit vlastní applet, který na požadovanou frázi vykoná přiřazenou akci pomocí jiné služby podporované IFTTT. Kromě jednoduché fráze je zde ještě možnost reagovat na frázi obsahující parametr, buď číselný nebo slovní, který lze předat cílové službě.

2.3 Amazon Alexa

Alexa je cloudový hlasový asistent založený na umělé inteligenci, vyvíjený společností Amazon. Podobně jako Google Assistent, má Alexa velkou škálu schopností, počínaje u plánování připomínek v určitý čas, přes vyhledání informací na internetu, ovládání zařízení chytré domácnosti, online nakupování, hraní slovních her, až po jednoduchou konverzaci s uživatelem. Pokud chceme, aby Alexa uměla ovládat i zařízení, které v současné době není podporované, tak je potřeba vytvořit tzv. „Skill“.

Skill se v tomto kontextu rozumí jako definice akce, která se má provést po určitém hlasovém příkazu. Alexa se postará o analýzu hlasového vstupu a zjistí záměr uživatele, poté pošle zpracovaný požadavek námi vytvořené aplikační logice. Ta by měla být umístěna na Alexa, AWS nebo jiné cloudové službě ve formě back-end aplikace. [11] Naše aplikace nakonec zpracuje POST požadavek obsahující JSON zprávu s parametry pro nastavení zařízení, nebo dotazem na zjištění aktuálního stavu, a naše API se již postará o vykonání požadované akce, případně odpovědi.

Na obrázku (Obr. 2.1) lze vidět diagram propojení vlastní API s Alexou.



Obr. 2.1: diagram propojení vlastní API s asistentem Amazon Alexa [12]

Alexa Skills Kit (zkráceně ASK) je framework, který obsahuje rozhraní pro různá odvětví schopností Alexy. V tabulce (Tab. 1.2) je přehled vybraných Smart Home Skills rozhraní, které je možne ve vlastní API implementovat.

Tab. 2.2: přehled Smart Home Skills rozhraní [13]

Název API	Funkce
BrightnessController	nastavení jasu
ColorController	nastavení barev
PercentageController	nastavení vlastnosti, kterou lze vyjádřit procenty
PowerController	zapnutí nebo vypnutí zařízení
SceneController	ovládání skupiny zařízení přiřazených do stejné skupiny
ThermostatController	ovládání chytrého termostatu, nastavení teploty
TimeHoldController	pozastavení činnosti zařízení

Každý z těchto rozhraní podporuje různé jazyky, avšak každý podporuje minimálně angličtinu. Zatímco TimeHoldController podporuje jen americkou angličtinu, tak BrightnessController podporuje celkem 12 jazyků, mezi nimiž je němčina, francouzština, italština, japonština, španělština, hindština, portugalština a několik dialektů angličtiny. [14]

Podobně jako Google Assistent, ani Alexa v současné době nepodporuje češtinu.

2.4 Siri

Siri je hlasová asistentka vytvořená společností Apple. Aktivační fráze je „Hey Siri“ a následně můžeme říct, co bychom chtěli najít, naplánovat, komu zavolat nebo napsat zprávu apod. Ostatně, jak tomu bylo u obou předchozích asistentů. Najdeme ji ve většině produktech od Apple: iPhone, iPad, AirPods, Apple Watch, Beats, HomePod, Mac (z roku 2018 a novější). [15]

Siri sice umí ovládat zařízení chytré domácnosti, ale jedná se jen o zařízení s certifikací od společnosti Apple: „Works with Apple HomeKit“ nebo „Works with Apple AirPlay“.

Seznam podporovaných zařízení lze nalézt na oficiálním webu společnosti Apple. [16]

Existují neoficiální postupy, jak vytvořit vlastní příkazy pro ovládání nepodporovaných zařízení, například pomocí SiriControl, IFTTT nebo pomocí vytvořené zkratky, která otevře Google Asistenta nainstalovaného na iOS, ale jsou komplikované.

Podle oficiálních webových stránek Apple je Siri dostupná ve 37 zemích. [17] Ale Česká republika mezi nimi není.

2.5 Cortana

Cortana je osobní hlasový asistent, vytvořený společností Microsoft, který byl zamýšlený pro pomoc s efektivitou práce. Je dostupný na počítačích se systémem Windows 10. Mezi schopnosti Cortany patří správa kalendáře a plánování událostí, připojení k probíhajícímu meetingu přes aplikaci Microsoft Teams, vytváření seznamu, spuštění aplikace a vyhledání informací pomocí vyhledávače Bing. [18]

Mezi podporované jazyky Cortany patří angličtina, francouzština, portugalština, němčina, italština, japonština, španělština a čínština (zjednodušená). [19] Jak je vidět, ani zde v současné době nenalezneme podporu pro češtinu.

Microsoft však nedávno na svém portále oznámil, že se rozhodl ukončit podporu Cortany pro většinu dovedností a na mobilních platformách. Nově bude tento asistent dostupný pouze v programovém balíčku Office 365. [20]

2.6 Bixby

Bixby je další z hlasových asistentů, tentokrát vyvíjený společností Samsung. Vedle klasických schopností hlasového asistenta – jako je vyhledávání informací na internetu, plánování událostí do kalendáře, vykonávání rutin, zavolání nebo napsání zprávy určité osobě, otevření aplikace v telefonu, a dalších – také umožňuje ovládat spotřebiče chytré domácnosti. Zařízení musí být propojené s aplikací SmartThings a označené logem s textem „Works with SmartThings“. [21]

Na oficiálních webových stránkách společnosti Samsung lze najít seznam podporovaných spotřebičů a zařízení touto aplikací. [21] Jsou zde televize, audio systémy, klimatizace, pračky, sušičky, robotické vysavače, chladničky, WiFi huby, senzory pohybu, fotoaparáty, multifunkční senzory, zvonky, žárovky, zámky, zásuvky, tlumiče světla, termostaty a další.

Bixby je dostupný ve třech dialektech angličtiny (americká, britská a indická), francouzštině, němčině, italštině, korejštině, mandarínské čínštině, španělštině a portugalštině [22], tedy podobně jako všichni předchozí zmiňovaní hlasoví asistenti, zatím nepodporuje češtinu.

Bixby Vision

Mimo jiné Samsung vyvíjí vizuálního asistenta Bixby Vision, nabízejícího funkce usnadnění, které pomáhají osobám se zrakovým postižením. Bixby Vision lze jednoduše nastavit, má snadné rozhraní a dokáže hlasem popisovat, co vidí. Tím pomáhá uživatelům vnímat okolní svět. [23]

2.7 Mycroft AI

Mycroft je první populární hlasový asistent s otevřeným kódem (open-source), což také vývojáři považují za hlavní výhodu tohoto asistenta. Věří, že umělá inteligence by měla být transparentní, a otevřená pro každého. [24] Mezi jeho další vlastnosti patří respektování soukromí a nenáročnost software, který lze provozovat například na Raspberry Pi 3.

Zdrojové kódy tohoto projektu jsou k dispozici pod svobodnou softwarovou licencí Apache 2.0 [25], která umožňuje svobodné užívání softwaru k jakémukoliv účelu, upravování a následnou redistribuci upravené verze softwaru, ale vyžaduje uvedení autorství a zřeknutí se odpovědnosti. [26]

Podporované jazyky asistenta Mycroft AI k polovině 2019 byly: němčina, francouzština, maďarština, italština, portugalština a švédština. V současné době se pracuje na podpoře pro dánský, holandský a španělský jazyk. [27]

2.8 Shrnutí rešerše hlasových asistentů

Pro tuto práci jsou relevantní parametry, zda asistent podporuje češtinu, nabízí možnost přidání vlastního příkazu (Skillu) a zda může fungovat offline. Přehled těchto parametrů nalezneme v tabulce (Tab. 2.3). Křížek „X“ znamená, že daná vlastnost není podporovaná.

Tab. 2.3: přehled relevantních parametrů hlasových asistentů pro tuto práci

Hlasový asistent	čeština	vlastní příkazy	offline
Google Assistant	X	Ano	X
Alexa	X	Ano	X
Siri	X	X	X
Cortana	X	Ano	Ano
Bixby	X	X	X
Mycroft AI	X	Ano	Ano

Z výsledků rešerše hlasových asistentů je patrné, že v současné době žádný z volně dostupných hlasových asistentů nepodporuje češtinu, která je pro tuto práci zásadní, jelikož do cílové skupiny patří lidé z České republiky, kteří nemusí nutně umět anglicky.

Bude tedy potřeba navrhnout vlastního hlasového asistenta, který bude rozumět českým příkazům.

2.9 Rozpoznávání řeči

Rozpoznávání lidské řeči, také známé jako ASR (automatic speech recognition), nebo STT (speech-to-text), je schopnost stroje převádět mluvené slovo do textové podoby.

Historie této technologie sahá až do roku 1952, kdy byl vyvinut rozpoznávací systém Audrey. Uměl rozpoznat vyslovenou cifru a jako zpětnou vazbu rozsvítil odpovídající žárovku u rozpoznané cifry. Toto rozpoznávání bylo řešené ještě za pomocí složitých elektrických obvodů a filtrů. [28]

O deset let později přišla společnost IBM s počítačem pojmenovaných „Shoebox“. Ten již uměl rozpoznat 16 slov, včetně čísel od 0 do 9 a aritmetických operátorů, jako plus, mínus, rovná se. Po vyslovení jednotlivých slov do mikrofonu byl výsledek rozpoznaného slova předán aritmetickému stroji, který po dokončení matematické operaci vytisknul výsledek. [29]

Obecně lze rozpoznávání rozdělit do tří oblastí podle složitosti:

- Rozpoznávání jednotlivých slov
- Rozpoznávání slovních spojení
- Rozpoznávání plynulé řeči

Dále lze rozdělit rozpoznávání podle závislosti na řečníkovi:

- Rozpoznávání závislé na řečníkovi
- Rozpoznávání nezávislé na řečníkovi

V případě rozpoznávání závislém na řečníkovi je nutné, aby byl rozpoznávací systém natrénován na hlas daného řečníka. Taková aplikace je však schopná rozpoznávat jen řeč jednoho konkrétního řečníka.

Dnes je větší zájem o systémy rozpoznávání řeči nezávislé na řečníkovi. Trénují se pomocí velkého množství dat (nahrávek) od mnoha různých řečníků. [29] Takto natrénovaný rozpoznávač je poté schopný rozumět i hlasu, se kterým se dříve nesetkal.

Pro rozpoznávání řeči se nejčastěji využívá skrytých Markovových modelů (anglicky Hidden Markov model), nebo neuronových sítí.

2.10 Skryté Markovovy modely

Skryté Markovové modely jsou statistické stavové automaty, které modelují prvky řeči pomocí stavů a přechodů mezi nimi. Stavy modelů jsou charakterizovány pravděpodobnostním rozložením parametrů určité části signálu, která je daným stavem modelována. Částmi signálu rozumíme například fonémy, slabiky nebo slova. [31]

2.11 Neuronové sítě

Umělé neuronové sítě se inspirovaly strukturou lidské nervové soustavy. Základním prvkem přirozené i umělé neuronové sítě je neuron neboli perceptron. Neurony jsou navzájem propojeny a předávají si signály. Platí přitom, že každý neuron může mít více vstupů, ale jen jeden výstup (tento výstup ale může být poslán i více než jednomu dalšímu neuronu). Vstupem neuronu může být buď výstup z jiného neuronu nebo informace z vnějšku (v našem příkladu by se jednalo o naměřené parametry stroje). Každý vstup má určitou váhu. [32]

Neuronové sítě se používají mimo jiné také pro rozpoznávání a kompresi obrazů nebo zvuků.

2.12 CMUSphinx

CMUSphinx je open-source projekt systému na rozpoznávání řeči, který je vyvíjen na Univerzitě Carnegieho-Mellonových (Carnegie Mellon University, CMU).

V současné době je k dispozici čtvrtá verze softwaru (CMUSphinx4), který je napsán kompletně v programovacím jazyce Java.

Mimo jiné tento projekt obsahuje software pro trénování a vytváření akustických modelů (SphinxTrain). Díky tomu je zde možnost vytvořit nový akustický model pro další jazyk, nebo zlepšit rozpoznávání u konkrétního nářečí.

2.12.1 Podporované jazyky

Momentálně je pro CMUSphinx oficiálně ke stažení 15 akustických a jazykových modelů: [33]

- Americká angličtina
- Francouzština
- Hindština
- Holandština
- Indická angličtina
- Italština
- Katalánština
- Kazaština
- Mexická španělština
- Němčina
- Portugalština
- Ruština
- Řečtina
- Španělština

2.12.2 Vytvoření modelu pro další jazyk

CMUSphinx pracuje se třemi druhy modelu pro rozpoznávání řeči.

Akustický model se stará o analýzu zvuku a přiřazení fonémů.

Fonémový slovník přiřazuje skupinám fonémů jednotlivá slova.

Jazykový model pomáhá při rozhodování z několika možností, o které slovo se s největší pravděpodobností jedná, podle četnosti výskytu ve spojení s předchozími a následujícími slovy.

Nejobtížnější úlohou pro vytvoření těchto modelů pro další jazyk, je získání potřebného množství nahrávek s přepsaným textem.

V závislosti na účelu použití výsledného modelu je potřeba:

- 1 hodina nahrávek řečníka pro hlasové ovládání závislé na konkrétním řečníkovi
- 5 hodin nahrávek od 200 různých řečníků pro hlasové ovládání nezávislé na řečníkovi
- 10 hodin nahrávek pro rozpoznávání hlasu závislé na řečníkovi
- 50 hodin nahrávek od 200 různých řečníků pro rozpoznávání hlasu nezávislé na řečníkovi

Kromě vytvoření nového modelu je zde ještě možnost adaptovat existující model. Této možnosti lze využít, pokud potřebujeme zlepšit přesnost rozpoznávání, nebo nemáme dostatek dat k natrénování nového modelu. Avšak takto odvozený model neposkytuje dostatečnou úspěšnost rozpoznávání, pokud se jedná o odlišný jazyk, než ze kterého byl odvozen. [34]

2.12.3 PocketSphinx

PocketSphinx je nenáročná knihovna pro rozpoznávání řeči napsaná v jazyce C. Je speciálně odladěná pro použití na platformách mobilních telefonů a zařízeních s menším výpočetním výkonem.

2.13 Google API pro rozpoznání řeči

Pro mobilní aplikace na operačním systému Android je zde možnost využít `SpeechRecognizer` API, které je k dispozici zdarma a podporuje češtinu.

Kromě zmíněné API zabudované do Android aplikace, Google nabízí ještě cloudové řešení s názvem Google Cloud Speech API, které už není zdarma, ale cena je relativně nízká. Ke dni psaní této práce byla cena 0,006 dolarů za každých započatých 15 sekund nahrávky, tedy 6 dolarů za 250 hodin nahrávky. [35] Podporuje 80 jazyků, mezi kterými je také čeština. Výhodou je, že tuto cloudovou službu je možné využít i na jiné platformě než Android.

3 Cíle práce

Hlavním cílem této práce je vytvoření modulu hlasového ovládání spotřebičů a zařízení chytré domácnosti v laboratoři personalizované telemedicíny v centru UCEEB.

Hlasové ovládání by mělo být v českém jazyce a kompatibilní s řídící jednotkou bytu.

Jelikož žádný z volně dostupných hlasových asistentů, zmíněných v předchozí kapitole, nepodporuje český jazyk, bude potřeba vytvořit vlastní řešení.

Jednotlivé cíle jsou následující:

- Najít a vyzkoušet službu rozpoznávání řeči podporující češtinu
- Vytvořit modul, který převede řeč na text
- Vytvořit seznam ovladatelných spotřebičů a zařízení
- Na základě tohoto seznamu vytvořit seznam příkazů
- Vytvořit relační databázi klíčových slov a odezev na jednotlivé kombinace těchto klíčových slov
- Vytvořit modul, který analyzuje text příkazu a zjistí záměr uživatele
- Přiřadit řídící proměnné ve stávajícím programu řídící jednotky bytu k Modbus objektům, aby bylo možné zadávat instrukce

4 Návrh softwaru

4.1 Potřeby lidí se sníženou soběstačností

Navrhovaný modul hlasového ovládání by měl sloužit lidem se sníženou soběstačností, do této skupiny bychom mohli zařadit seniory, osoby upoutané na lůžko, osoby na invalidním vozíku a osoby se zrakovým postižením.

Soběstačnost je důležitým kritériem kvality života. Za soběstačného je možné považovat člověka, který samostatně a bez pomoci nebo dohledu druhé osoby zvládá všechny denní činnosti a zastává potřebné či očekávané sociální role v prostředí, v němž žije. [35]

Další definice soběstačnosti uvádí schopnost člověka uspokojovat své základní životní potřeby, především vykonávat všední denní činnosti (toaleta, oblékání, stravování, nakupování, úkony spojené se základní péčí o domácnost). Hodnocení, udržení a rovněž i obnova soběstačnosti jsou cílem ucelené podpory osob se speciálními potřebami. [37]

Posuzovat stupeň závislosti lze podle zvládání životních potřeb:

- Mobilita
- Orientace
- Komunikace
- Stravování
- Oblékání a obouvání
- Tělesná hygiena
- Výkon fyziologické potřeby
- Péče o zdraví
- Osobní aktivity a péče o domácnost

4.2 Analýza požadavků výsledného software

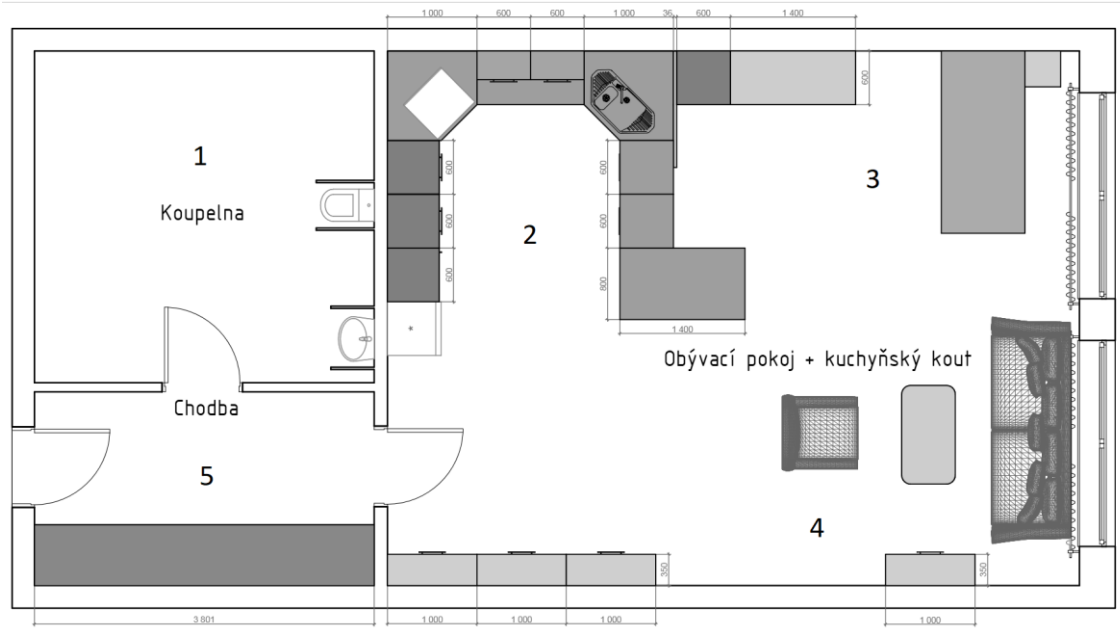
V této části se podáváme na aktuální možnosti ovládání spotřebičů a zařízení v laboratoři personalizované telemedicíny, a z toho vyvodíme požadavky na výsledný modul hlasového ovládání.

Momentálně je ovládání spotřebičů a zařízení realizované pomocí webové aplikace běžící jako program na řídící jednotce bytu (Tecomat Foxtrot). Tato webová aplikace je vytvořena nástrojem WebMaker ve vývojovém prostředí Mosaic, a propojuje ovládací prvky na webové stránce s proměnnými v prostředí samotného programu PLC, který už fyzicky spíná relé nebo pouští elektrické signály (PWM[1], komunikační) jednotlivým spotřebičům a zařízením připojeným k této řídící jednotce.

[1] Pulzně šířková modulace, používá se mj. pro řízení výkonu

4.2.1 Půdorys laboratoře

Jak již bylo zmíněno v úvodu, laboratoř má podobu garsoniéry, tedy jedné místnosti s oddělenou koupelnou a chodbou. Na obrázku (Obr. 4.1) je zobrazený půdorys laboratoře. Číslem jsou zde vyznačená jednotlivá místa kvůli referenci.



Obr. 4.1: půdorys laboratoře personalizované telemedicíny v UCEEB [38]

4.2.2 Webová aplikace ovládání

V této části si více přiblížíme webovou aplikaci pro ovládání zařízení připojených k řídící jednotce laboratoře. Tato aplikace a program pro PLC byly vytvořeny společností Elpramo s.r.o. v roce 2016.

Je dostupná pod lokální IP adresou řídící jednotky, konkrétně „192.168.0.10“, a portem 80 (HTTP protokol).

Po připojení je potřeba zadat přístupové údaje – uživatelské jméno a heslo, poté se zobrazí úvodní stránka, kterou můžeme vidět na obrázku (Obr. 4.2).

Menu obsahuje následující položky:

- Obývací pokoj
- Koupelna / Chodba
- Technologie
- Spotřebiče
- Probuzení
- Spotřeba elektřiny

Obr. 4.2: úvodní stránka webové aplikace pro ovládání laboratoře

Na této úvodní stránce lze ovládat dveře do obývacího pokoje nebo do koupelny, zapnout scénu „Disco!“, která rozbliká světla a rozhýbe rolety, ovládat televizi a rádio, nebo použít rychlé příkazy jako „Odchod“, „Jdu spát“ a „Probouzení“.

Obývací pokoj

V sekci „Obývací pokoj“ jsou další 4 podsekce:

- Ovládáni
- Osvětlení
- Denní režimy
- Nastavení scén

Ovládání

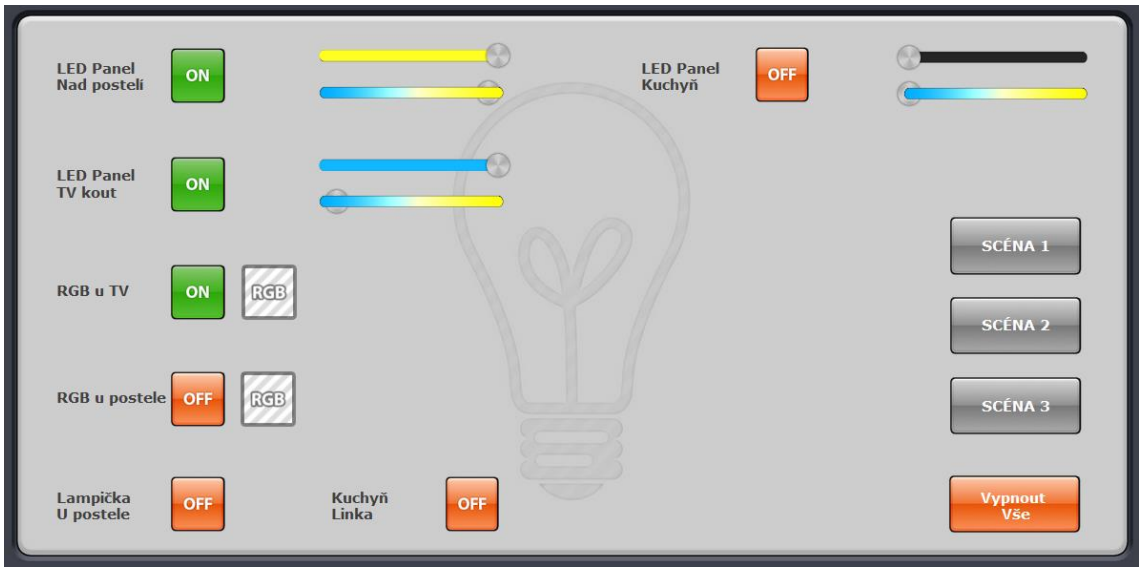
V sekci ovládání nalezneme ovládací prvky pro nastavení teploty a klimatizace v obývacím pokoji, také je zde zobrazena aktuální teplota. Na pravé straně se nachází ovládací panel rolet a okna. Na tomto panelu jsou tlačítka pro vytažení nebo stažení rolet u jednotlivých oken, možnost otevřít / zavřít okno u televize a tlačítko pro mikro-ventilaci.

Osvětlení

Jak název napovídá, v sekci „Osvětlení“ se nachází ovládání světel. Je zde možné zapnout nebo vypnout určité světlo, nastavit jas a barvu. V případě LED panelů na stropě lze nastavit chromatickou barvu (barevnou teplotu), u barevných LED lze nastavit konkrétní barvu pomocí barevného modelu RGB.

Vzhled ovládacích panelů teploty v místnosti a osvětlení můžeme vidět na následujících obrázcích (Obr. 4.3 a Obr. 4.4).

Obr. 4.3: panel ovládání teploty



Obr 4.4: panel ovládání osvětlení

Denní režimy

Program rozlišuje dva denní režimy pro nastavení teploty: „Low“ a „High“. Lze například nastavit, aby v režimu „Low“ byla teplota 18 °C, a v režimu „High“ 22 °C. V sekci „Denní režimy“ můžeme nastavit rozvrh, ve který den a v jakém časovém rozmezí se má použít jeden nebo druhý režim. Například od 7 do 22 hodin můžeme nastavit teplejší režim, a od 22 do 7 hodiny ranní režim chladnější. Při změně nastavené teploty v sekci ovládání, se toto nastavení uloží pro právě aktivní režim.

Nastavení scén

Tato sekce slouží pro nastavení tří scén, které je potom možné rychle vyvolat tlačítky v sekci „Osvětlení“. Ovládací panel zde vypadá také vzhledově stejně jako v sekci pro ovládání osvětlení, akorát je zde ještě navíc volba, o kterou scénu se jedná.

Koupelna / Chodba

V této sekci se nachází jen 2 podsekce, a to „Ovládání“ a „Denní režimy“.

V podsekci „Ovládání“ na levé straně nalezneme opět ovládací panel pro nastavení požadované teploty, ale tentokrát v koupelně. Na pravé straně je pak panel pro ovládání tří světel, u kterých lze nastavit intenzitu / jas. Jedná se o světlo na chodbě, hlavní světlo v koupelně a světlo u zrcadla, které je také v koupelně.

Denní režimy zde fungují stejně jako v případě obývacího pokoje.

Jelikož v těchto dvou místnostech nejsou žádná barevná světla, nastavovat scény by bylo zbytečné.

Technologie

Zde je přehled spotřeby studené i teplé vody, jak v koupelně, tak v kuchyni. Dále zde nalezneme aktuální stav ventilů a aktuální průtok. Případně si můžeme zobrazit graf pro jednotlivé možnosti: „studená voda v koupelně“, „teplá voda v koupelně“, „studená voda v kuchyni“ a „teplá voda v kuchyni“.

Spotřebiče

Sekce s přehledem nainstalovaných spotřebičů a jejich stavem (zapnuto / vypnuto).

4.2.3 Ovladatelná zařízení v laboratoři

Na základě analýzy webové aplikace ovládání laboratoře nyní můžeme sestavit seznam ovladatelných zařízení, která by měl umět ovládat výsledný modul hlasového ovládání. V tabulce na další straně (Tab. 4.2) nalezneme přehled zařízení, jejich umístění a podporované akce.

Tab. 4.1: seznam zařízení a podporovaných akcí

Umístění	Zařízení	Podporované akce
Koupelna (1)	Hlavní světlo	Zapnutí / vypnutí
		Nastavení jasu
	Světlo u zrcadla	Zapnutí / vypnutí
		Nastavení jasu
	Termostat	Nastavení teploty
Kuchyň (2)	LED panel na stropě	Zapnutí / vypnutí
		Nastavení jasu
		Nastavení chromatické barvy
	Světlo na kuchyňské lince	Zapnutí / vypnutí
U postele (3)	LED panel nad postelí	Zapnutí / vypnutí
		Nastavení jasu
		Nastavení chromatické barvy
	Barevná LED na stěně	Zapnutí / vypnutí
		Nastavení jasu
		Nastavení barvy
	Lampa u postele	Zapnutí / vypnutí
	Rolety / žaluzie	Vytažení nahoru
		Stažení dolů
	Termostat / klimatizace	Nastavení teploty
		Režim zahřívání / chlazení
TV kout (4)	LED panel na stropě	Zapnutí / vypnutí
		Nastavení jasu
		Nastavení chromatické barvy
	Barevná LED na stěně	Zapnutí / vypnutí
		Nastavení jasu
		Nastavení barvy
	Rolety / žaluzie	Vytažení nahoru
		Stažení dolů
	Okno	Otevření / zavření
		Mikro-ventilace
	Televize	Zapnutí / vypnutí
		Zesílení / ztlumení hlasitosti
		Přepnutí programu
	Rádio	Zapnutí / vypnutí
		Zesílení / ztlumení hlasitosti
Chodba (5)	Bodové světlo na stropě	Zapnutí / vypnutí
		Nastavení jasu
	Dveře do koupelny	Otevření / zavření
	Dveře do obývacího pokoje	Otevření / zavření

4.3 Návrh řešení hlasového ovládání

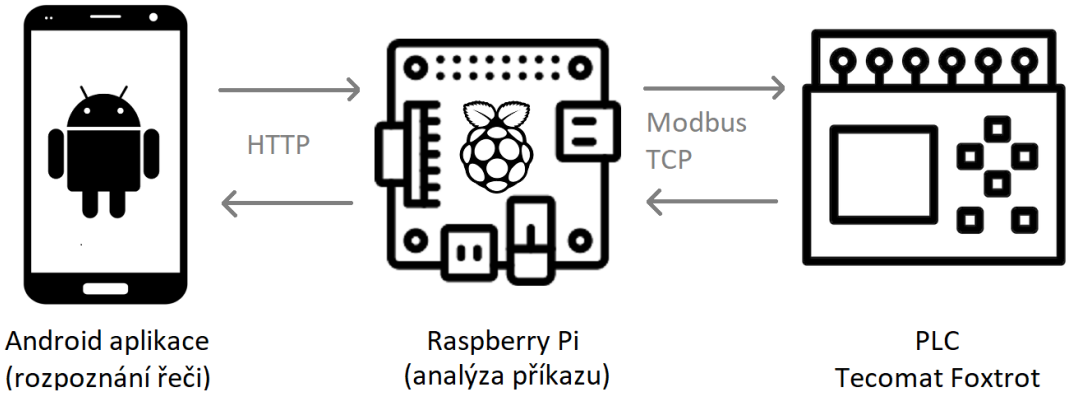
Hlasové ovládání laboratoře bude sestávat ze tří modulů.

Prvním je aplikace pro chytré telefony s operačním systémem Android, která bude mít za úkol zachycení hlasového vstupu uživatele, převod řeči na text, a následné odeslání výsledného textu druhému modulu.

Druhý modul bude ve formě webové služby, a bude mít na starost analýzu textu s příkazem. Touto analýzou se pokusí zjistit záměr uživatele a odeslat příslušnou instrukcí řídící jednotce domácnosti.

Třetí modul je samotná řídící jednotka domácnosti, která je již naprogramovaná, avšak zde bude potřeba provést určité úpravy, aby bylo možné přijímat instrukce.

Na obrázku (Obr. 4.5) je znázorněné blokové schéma propojení jednotlivých modulů.



Obr. 4.5: propojení modulů hlasového ovládání

4.4 Řídící jednotka laboratoře

Řídící jednotkou laboratoře je programovatelný automat Tecomat Foxtrot se základním modulem CP-1001 od firmy Teco a.s.

4.4.1 Programovatelný automat

Programovatelný automat (známý také pod pojmem PLC) je číslicový řídící elektronický systém určený pro řízení pracovních strojů a procesů v průmyslovém prostředí. PLC prostřednictvím číslicových nebo analogových vstupů a výstupů získává a předává informace z a do řízeného zařízení. Algoritmy řízení jsou uloženy v paměti uživatelského programu, který je cyklicky vykonáván. [39]

4.4.2 Tecomat Foxtrot

Programovatelné automaty TECOMAT FOXTROT představují malé kompaktní automaty s možností modulárního rozšíření. Spojují tak výhody kompaktních automatů co do velikosti a modulárních automatů co do rozšiřitelnosti a variability. Jsou určeny pro řízení technologií v nejrůznějších oblastech průmyslu i v jiných odvětvích. Jednotlivé moduly systému jsou uzavřeny v plastových ochranných pouzdrech, které se montují na U lištu ČSN EN 50022. [39]

Rozšiřovací periferní moduly se k centrální jednotce připojují pomocí sériové sběrnice. Díky tomu mohou být jednotlivé části systému TECOMAT FOXTROT rozmístěny decentralizovaně tak, že jednotlivé moduly jsou umístěny přímo u ovládaných technologií a šetří tak silovou kabeláž. [39]

4.4.3 Základní modul CP-1001

Základní modul CP-1001 je osazen centrální jednotkou (CPU) řady L, která je určena pro aplikace s požadavky na výkon. Obsahuje: [40]

- zálohovanou paměť CMOS RAM pro uživatelské programy, data, tabulky, uživatelské registry a DataBox
- paměť Flash pro zálohování uživatelského programu
- slot pro MMC/SD/SDHC paměťovou kartu
- obvod reálného času (RTC)
- rozhraní Ethernet 10/100 Mb/s
- dva sériové kanály
    - jeden s pevným rozhraním RS-232
    - druhý s pozicí pro volitelné sub-moduly
- dva komunikační kanály s rozhraním CIB pro připojení externích periferií
- systémové rozhraní TCL2 určené pro připojení rozšiřovacích modulů, které zvyšují počet I/O systému

Přehled systémových parametrů nalezneme níže v tabulce (Tab. 4.2).

Tab. 4.2: systémové parametry základního modulu CP-1001 [40]

Paměť uživatelského programu a tabulek	384 + 64 KB
DataBox (interní přídavná paměť dat)	512 KB
Paměť pro archivaci projektu	2 MB
Počet uživatelských registrů z toho remanentních registrů	192 KB 40 KB
Doba cyklu na 1000 logických instrukcí	0,2 ms

4.4.4 Programování jednotky

Programování centrální jednotky se provádí ve vývojovém nástroji Mosaic, který je dostupný na oficiálních webových stránkách společnosti Teco a.s. [41] Aby bylo možné do PLC nahrát program, je nutné, aby byl počítač připojený do stejné lokální síti jako centrální jednotka PLC.

Typy dat

V rámci společných prvků jsou definovány typy dat. Definování datových typů napomáhá prevenci chyb v samém počátku tvorby projektu. Je nutné definovat typy všech použitých parametrů. Běžné datové typy jsou BOOL, BYTE, WORD, INT, REAL, DATE, TIME, STRING atd. Z těchto základních datových typů je pak možné odvozovat vlastní uživatelské datové typy, tzv. odvozené datové typy. Tímto způsobem můžeme např. definovat jako samostatný datový typ analogový vstupní kanál a opakovaně ho používat pod definovaným jménem. [42]

Proměnné

Proměnné mohou být přiřazeny explicitně k hardwarovým adresám (např. vstupům, výstupům) pouze v konfiguracích, zdrojích nebo programech. Tímto způsobem je dosaženo vysokého stupně hardwarové nezávislosti a možnosti opakovaného využití softwaru na různých hardwarových platformách. Oblast působnosti proměnných je běžně omezena pouze na tu programovou organizační jednotku, ve které byly deklarovány. Pokud mají mít proměnné globální působnost v rámci celého projektu, pak musí být jako globální deklarovány (VAR_GLOBAL). Aby bylo možné správně nastavit počáteční stav procesu nebo stroje, může být parametrům přiřazena počáteční hodnota při startu nebo studeném restartu. [42]

Programové organizační jednotky

Funkce, funkční bloky a programy jsou v rámci normy IEC61131 nazývány společně programové organizační jednotky (Program Organization Units, POUs).

Funkce je pojmenovaný blok příkazů. Funkcím můžeme předávat argumenty. Funkce vrací návratovou hodnotu. [43]

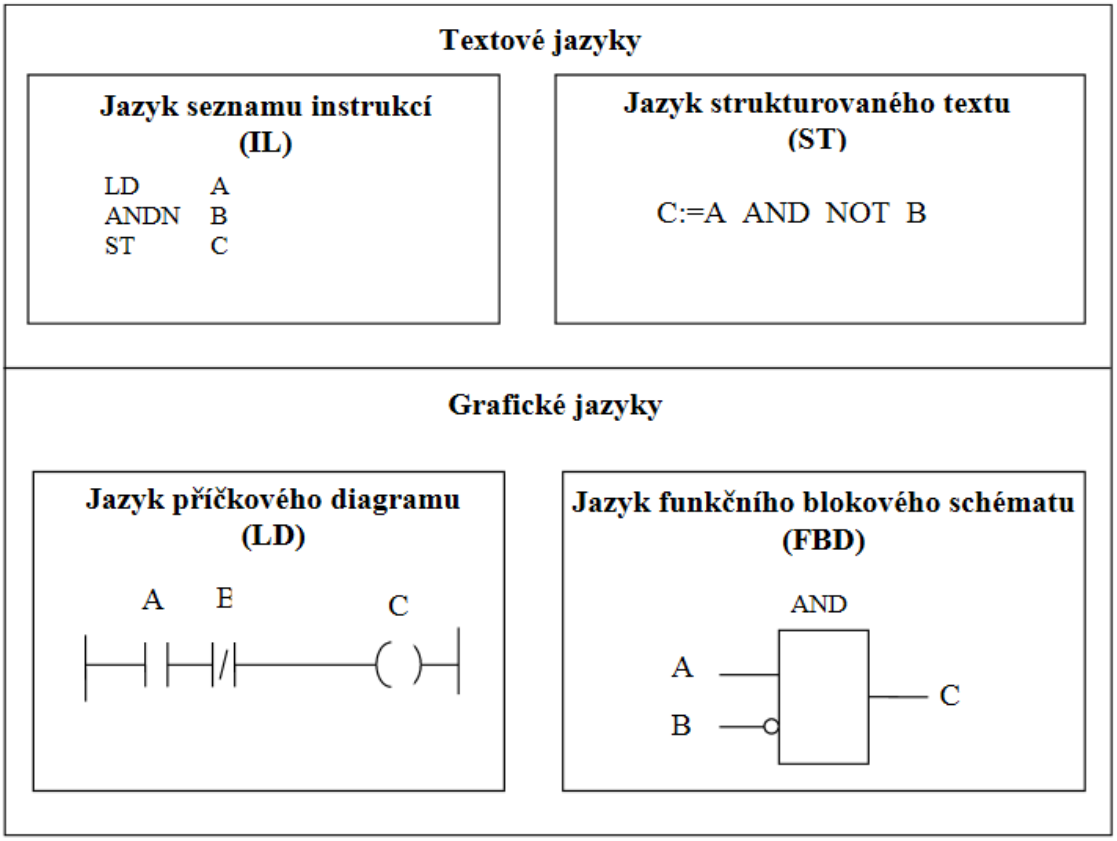
Na **funkční bloky** se můžeme dívat jako na integrované obvody, které reprezentují hardwarové řešení specializované řídicí funkce. Obsahují algoritmy i data, takže mohu zachovávat informaci o minulosti (tím se liší od funkcí). Mají jasně definované rozhraní a skryté vnitřní proměnné, podobně jako integrovaný obvod nebo černá skříňka. Umožňují tím jednoznačně oddělit různé úrovně programátorů nebo obslužného personálu. Klasickými příklady funkčního bloku jsou např. regulační smyčka pro teplotu nebo PID regulátor. [42]

Program je sítí funkcí a funkčních bloků. Program může být zapsán v libovolném z jazyků definovaných v normě. [42]

Programovací jazyky

V rámci standardu IEC61131-3 jsou definovány čtyři programovací jazyky. Jejich sémantika i syntaxe je přesně definována a neponechává žádný prostor pro nepřesné vyjadřování. Zvládnutím těchto jazyků se tak otevírá cesta k používání široké škály řídicích systémů, které jsou na tomto standardu založeny.

Programovací jazyky se dělí do dvou základních kategorií, znázorněných na obrázku (Obr. 4.6)



Obr. 4.6: Logicá funkce ANDN ve čtyřech záklaních jazycích [42]

4.5 Úprava PLC programu

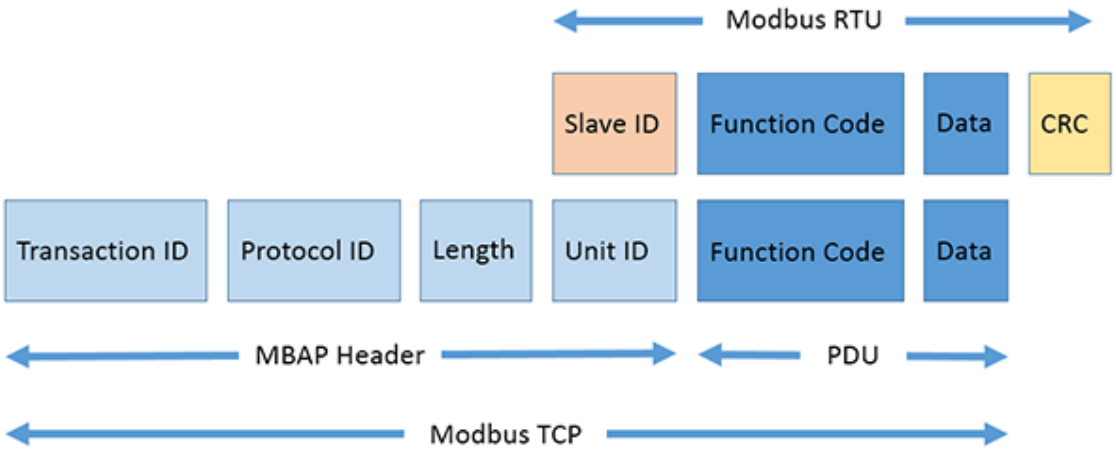
Aktuální verzi programu řídící jednotky je potřeba upravit tak, aby bylo možné nějakým způsobem předávat instrukce pro ovládání jednotlivých zařízení. K tomuto účelu lze využít Modbus TCP protokol.

4.5.1 Modbus

Modbus je komunikační protokol založený na architektuře master/slave nebo klient/server. Hlavním účelem protokolu je zajišťovat snadnou, spolehlivou a rychlou komunikaci mezi automatizačními zařízeními a zařízeními pro procesní instrumentaci. Jedna z hlavních výhod tohoto protokolu je, že se jedná o standardizovanou výměnu dat v souladu s normami, díky tomu je nezávislý na konkrétní technologii nebo výrobci. [44]

Rozlišujeme následující režimy přenosu dat:

- Modbus TCP: komunikace pomocí standardu Ethernet TCP/IP na základě modelu klient/server
- Modbus RTU: asynchronní sériový přenos prostřednictvím rozhraní RS-232 nebo RS-485
- Modbus ASCII: podobný jako protokol RTU, ale s jiným formátem dat; používá se relativně zřídka



Obr. 4.7: popis protokolu Modbus TCP [45]

Na obrázku (Obr. 4.7) vidíme, že se protokol Modbus TCP a Modbus RTU od sebe liší v tom, že Modbus TCP neobsahuje CRC součet – jelikož úspěšné doručení má na starost už použitý protokol TCP – a místo položky Slave ID je zde Unit ID.

Protokol Modbus TCP rozlišuje následující základní funkce, popsané níže v tabulce (Tab. 4.3).

Tab. 4.3: popis základních funkcí protokolu Modbus TCP [45]

Kód funkce	Co funkce dělá		Typ hodnoty	Typ přístupu
01 (0x01)	Čtení DO	Čtení stavu cívky	Diskrétní	Čtení
02 (0x02)	Čtení DI	Čtení stavu vstupu	Diskrétní	Čtení
03 (0x03)	Čtení AO	Čtení holding registrů	16bitová	Čtení
04 (0x04)	Čtení AI	Čtení vstupních registrů	16bitová	Čtení
05 (0x05)	Zápis jednoho DO	Nastavení stavu jedné cívky	Diskrétní	Zápis
06 (0x06)	Zápis jednoho AO	Nastavení jednoho registru	16bitová	Zápis
15 (0x0F)	Zápis více DO	Nastavení stavu více smyček	Diskrétní	Zápis
16 (0x10)	Zápis více AO	Nastavení více registrů	16bitová	Zápis

4.6 Analýza požadavků webové aplikace

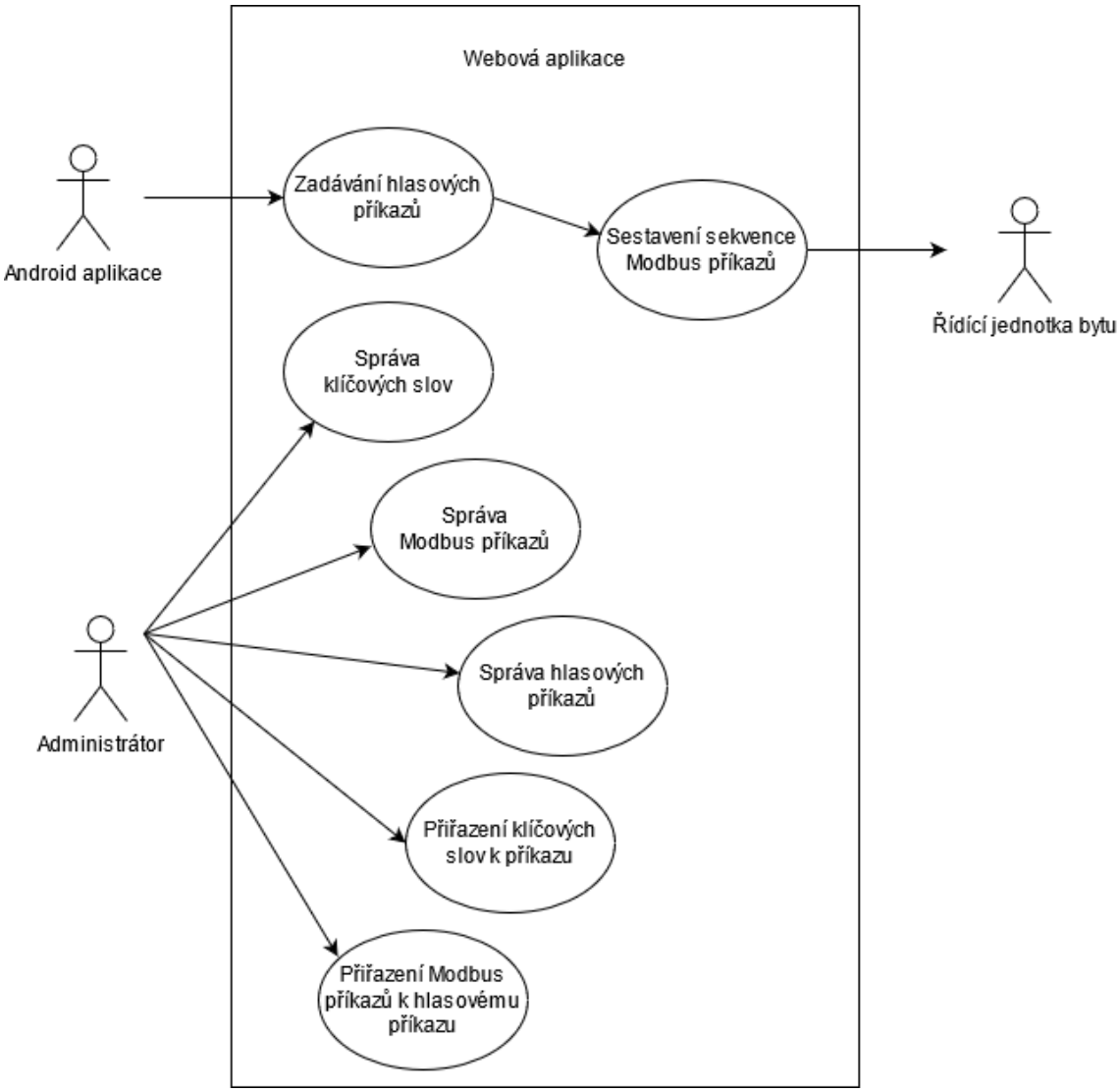
Webová aplikace bude zpracovávat text s příkazem a pomocí databáze – propojující klíčová slova s jednotlivými příkazy – bude zjišťovat záměr uživatele. Pokud nalezne odpovídající příkaz sestavený z přiřazených klíčových slov, předá řídící jednotce potřebné instrukce pro vykonání požadavku.

Mimo jiné bude také umožňovat spravovat databázi klíčových slov, příkazů a odezev a jejich přiřazení. Tato administrační sekce databáze bude přístupná po přihlášení administrátorským účtem a heslem.

Komunikace s mobilní (Android) aplikací bude probíhat ve formě HTTP POST požadavků, jejichž obsahem bude JSON objekt. Jako vstup bude aplikace očekávat objekt s parametrem `commandText`. Jako odpověď na takový požadavek webová aplikace vrátí také JSON objekt, který může být buď prázdný, nebo s parametrem `tts` nesoucím text, který by měla mobilní aplikace syntetizovat na řeč a přehrát uživateli.

Pro komunikaci s PLC vytvoří webová aplikace Modbus zprávu a odešle ji na nastavenou IP adresu a port prostřednictvím TCP socketu.

Diagram použití (use-case diagram) můžeme vidět na následující straně na obrázku (Obr. 4.8).



Obr. 4.8: diagram použití webové aplikace

4.7 Technická specifikace webové aplikace

Pro provoz webové aplikace bude potřeba zařízení s přístupem na síť a možností provozovat webovou službu – zpracovávat HTTP požadavky. Pro takový účel lze využít například stolní počítač, notebook, minipočítač Raspberry Pi nebo mikroprocesor ESP32. Jako zlatá střední cesta v poměru výkonu, spotřeby a schopností se zde nabízí právě Raspberry Pi.

Pro tuto volbu jsou k dispozici 3 základní varianty operačního systému:

- Raspberry Pi OS Lite
- Raspberry Pi OS with Desktop
- Raspberry Pi OS with desktop and recommended software

Pro účel použití zařízení jako serveru pro webovou aplikaci bude stačit nejodlehčenější verze operačního systému, tedy Raspberry Pi OS Lite. Tato verze neobsahuje grafické prostředí, a není tedy náročná na výkon a z toho plynoucí spotřebu energie.

Jako programové vybavení bude potřeba:

- Apache2
- PHP 7
- MySQL nebo MariaDB
- phpMyAdmin

Apache2 je HTTP server, který zpracovává příchozí webové požadavky a spouští příslušné PHP skripty nebo poskytuje obsah požadovaného souboru, pokud je nastaveno, že má být daný soubor veřejně přístupný.

PHP je skriptovací programovací jazyk, určený primárné k vytváření webových stránek a aplikací.

MariaDB je jedním z mnoha systémů řízení báze dat, takzvané DBMS (anglicky – database management system). DBMS je softwarové vybavení, které zajišťuje práci s daty, a tvoří aplikační rozhraní mezi aplikačními programy a uloženými daty. MariaDB uplatňuje relační model databáze.

phpMyAdmin je webový nástroj napsaný v jazyku PHP, který se používá pro návrh a správu databáze a databázových tabulek. Podporuje MySQL a MariaDB databáze.

4.8 Raspberry Pi

Raspberry Pi je malý, jednodeskový počítač o velikosti platební karty.

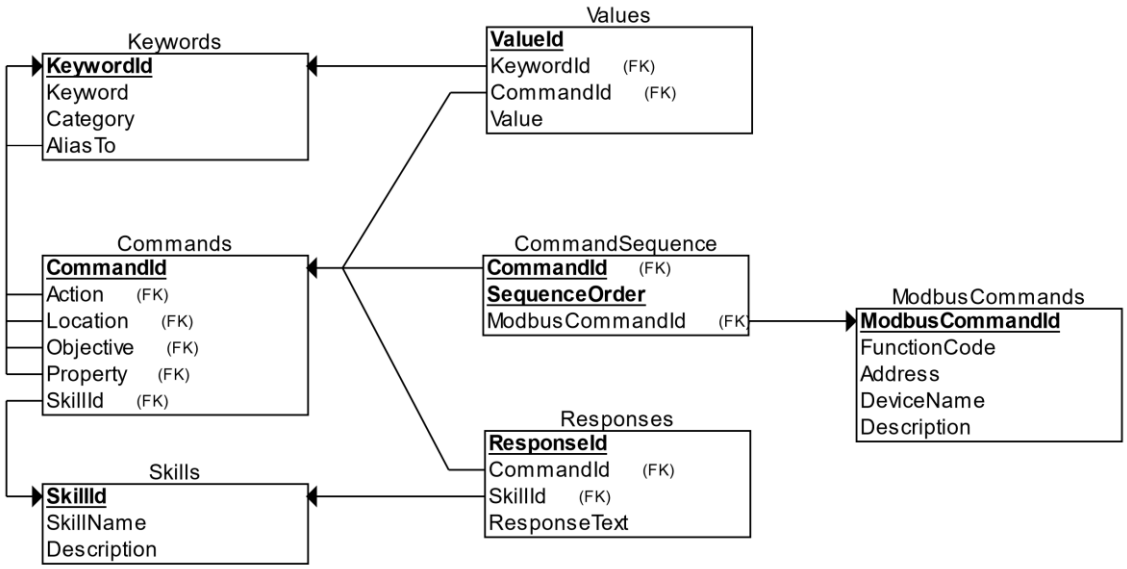
Obsahuje:

- čtyřjádrový procesor Broadcom BCM2837B0 (64-bit, 1,40 GHz)
- 1 GB operační paměti
- Gigabit Ethernet přes USB 2.0 (propustnost 300 Mbps)
- čtečku paměťových karet
- 4 USB 2.0 porty
- HDMI rozhraní
- Audio výstup
- WiFi ac
- Bluetooth 4.2 BLE
- 40 GPIO pinů
    - Některé piny slouží pro funkce jako PWM, SPI, I2C, UART

4.9 Návrh databáze

Databáze se skládá ze 7 tabulek propojených mezi sebou (Obr. 4.9).

Kromě zmíněných tabulek obsahuje také 3 pohledy spojující dvě nebo více tabulek.



Obr. 4.9: relační schéma databáze

4.9.1 Tabulka Keywords

Tabulka Keywords (Tab. 4.4) je nezávislá tabulka a slouží pro uchovávání jednotlivých klíčových slov, jejich kategorie a informace o tom, zda se jedná o alias k jinému slovu.

Tab. 4.4: struktura databázové tabulky Keywords

Název sloupce	Datový typ	Null	Další vlastnosti
KeywordId	INT	ne	Primární klíč, Auto Increment
Keyword	VARCHAR (20)	ne	
Category	ENUM	ne	ACTION, LOCATION, OBJECTIVE, PROPERTY, VALUE
AliasTo	INT	ano	Cizí klíč pro KeywordId ve stejné tabulce

Sloupec KeywordId je primární klíč této tabulky a jednoznačně určuje jednotlivé řádky. Ve sloupci Keyword budou uložená jednotlivá klíčová slova. Sloupec Category určuje kategorii daného slova – zda se jedná o určení akce, místa, cíle / zařízení, vlastnosti nebo hodnoty. Poslední sloupec AliasTo poskytuje referenci na původní klíčové slovo, pokud by se jednalo o jiný tvar slova se stejným významem.

4.9.2 Tabulka Values

Tabulka Values (Tab. 4.5) uchovává informace o číselném významu hodnoty některých klíčových slov. Například slovo „polovinu“ lze vyjádřit číslem 50 v kontextu procentové škály. Nebo barvu „červená“ jako číslo 16711680, které se v hexadecimálním tvaru zapíše jako „FF0000“, a v RGB modelu znamená červenou barvu.

Tato tabulka je závislá na tabulce klíčových slov (Keywords) a volitelně na tabulce příkazů (Commands), jelikož pro některé příkazy může mít dané klíčové slovo jiný význam hodnoty.

Tab. 4.5: struktura databázové tabulky Values

Název sloupce	Datový typ	Null	Další vlastnosti
ValueId	INT	ne	Primární klíč, Auto Increment
KeywordId	INT	ne	Cizí klíč na tabulku Keywords
CommandId	INT	ano	Cizí klíč na tabulku Commands
Value	INT	ne	

4.9.3 Tabulka Commands

Tabulka Commands (Tab. 4.6) propojuje jednotlivé kombinace klíčových slov se sekvencí instrukcí pro řídící jednotku bytu, a kontrolérem, který bude tento příkaz zpracovávat.

Tab. 4.6 struktura databázové tabulky Commands

Název sloupce	Datový typ	Null	Další vlastnosti
CommandId	INT	ne	Primární klíč, Auto Increment
Action	INT	ano	Cizí klíč na tabulku Keywords
Location	INT	ano	Cizí klíč na tabulku Keywords
Objective	INT	ano	Cizí klíč na tabulku Keywords
Property	INT	ano	Cizí klíč na tabulku Keywords
SkillId	INT	ne	Cizí klíč na tabulku Skills
CommandText	TEXT	ne	

Kromě primárního klíče obsahuje tato tabulka také unikátní klíč, který je tvořen sloupci Action, Location, Objective a Property. Jinými slovy zde nemohou být uložené dva různé příkazy pro dvě stejné kombinace klíčových slov. Zmíněné čtyři sloupce tvoří kombinaci klíčových slov, pod kterou bude přiřazený daný hlasový příkaz. Pomocí sloupce SkillId se k hlasovému příkazu přiřadí zpracovávající kontrolér, a ve sloupci CommandText bude samotný příkaz i s vedlejšími slovy a předložkami pro přehlednost.

4.9.4 Tabulka Skills

Po inspiraci od ASK API pro hlasového asistenta Alexa vznikl nápad na vytvoření systému kontrolérů, které budou zpracovávat jednotlivé hlasové příkazy. Právě v tabulce Skills (Tab. 4.7) se nachází seznam dostupných kontrolérů, které bude možné přiřadit.

Tab. 4.7: struktura databázové tabulky Skills

Název sloupce	Datový typ	Null	Další vlastnosti
SkillId	INT	ne	Primární klíč, Auto Increment
SkillName	VARCHAR (50)	ne	
Description	TEXT	ne	

Každý kontrolér, který bude v této tabulce uveden, musí být ve webové aplikaci také implementován. Jinak by při zpracovávání příkazu, ke kterému by byl přiřazený neimplementovaný kontrolér, vznikla chyba.

Sloupec SkillName určuje název třídy, pod kterou má být daný kontrolér implementován, a ve sloupci Description je popis, co kontrolér dělá.

4.9.5 Tabulka ModbusCommands

Tabulka ModbusCommands (Tab. 4.8) uchovává seznam přiřazení Modbus registrů k proměnným v programu řídící jednotky bytu. K této tabulce je vytvořen unikátní klíč nad sloupci DataType a Address.

Tab. 4.8: struktura databázové tabulky ModbusCommands

Název sloupce	Datový typ	Null	Další vlastnosti
ModbusCommandId	INT	ne	Primární klíč, Auto Increment
DataType	ENUM	ne	BOOL, UINT
Address	INT	ne	
DeviceName	VARCHAR (255)	ne	
Description	TEXT	ne	

Sloupec DataType určuje, zda se jedná o cívku (diskrétní hodnotu) nebo registr v rámci Modbus protokolu, a tedy jestli se má použít funkce s kódem 01 pro čtení a 05 pro zápis v případě cívky, nebo 03 pro čtení a 05 pro zápis v případě 16-bit registru. Ve sloupci Address je uložená adresa daného registru nebo cívky. Sloupce DeviceName a Description slouží pro popis, o které zařízení se jedná a který parametr je nastavován.

4.9.6 Tabulka CommandSequence

Jedná se o propojující tabulku hlasových příkazů se skutečně vykonávanými příkazy. Primární klíč je zde tvořen dvěma sloupci – CommandId a SequenceOrder. První sloupec je zároveň cizí klíč na tabulku hlasových příkazů. Druhý sloupec udává pořadí vykonávané instrukce. Strukturu této databázové tabulky nalezneme v tabulce (Tab. 4.9).

Tab. 4.9: struktura databázové tabulky CommandSequence

Název sloupce	Datový typ	Null	Další vlastnosti
CommandId	INT	ne	Primární klíč, Cizí klíč na tabulku Commands
SequenceOrder	INT	ne	Primární klíč
ModbusCommandId	INT	ne	Cizí klíč na tabulku ModbusCommands

4.9.7 Tabulka Response

Poslední databázovou tabulkou je tabulka Response (Tab. 4.10). V této tabulce ve sloupci ResponseText budou uložené textové odpovědi pro zpětnou vazbu uživateli, nebo dotazy na upřesňující informace k vyslovenému hlasovému příkazu.

Záznamy v této tabulce mohou být volitelně závislé na hlasovém příkazu nebo kontroléru.

Tab. 4.10: struktura databázové tabulky Response

Název sloupce	Datový typ	Null	Další vlastnosti
ResponseId	INT	ne	Primární klíč, Auto Increment
CommandId	INT	ano	Cizí klíč na tabulku Commands
SkillId	INT	ano	Cizí klíč na tabulku Skills
ResponseText	TEXT	ne	

4.9.8 Pohled ValuesView

Tento pohled spojuje tabulku hodnot (Values) s tabulkou klíčových slov (Keywords). Jako spojovací podmínka je zde použit sloupec KeywordId. Zahrnuté sloupce do pohledu jsou vyjmenované v tabulce (Tab. 4.11).

Tabulka 4.11: zahrnuté sloupce do pohledu ValuesView

Název sloupce	Zdrojová tabulka
ValueId	Values
KeywordId	Values / Keywords
CommandId	Values
Value	Values
Keyword	Keywords

4.9.9 Pohled CommandsView

Pohled CommandsView (Tab. 4.12) spojuje tabulku hlasových příkazů (Commands) s tabulkou kontrolérů (Skills), které zpracovávají jednotlivé hlasové příkazy. Jako spojovací podmínka je zde stejná hodnota ve sloupci SkillId.

Tab. 4.12: zahrnuté sloupce do pohledu CommandsView

Název sloupce	Zdrojová tabulka
SkillId	Skills / Commands
CommandId	Commands
Action	Commands
Location	Commands
Objective	Commands
Property	Commands
CommandText	Commands
SkillName	Skills
Description	Skills

4.9.10 Pohled CommandSequenceView

V pohledu CommandSequenceView (Tab. 4.13) je spojená tabulka jednotlivých instrukcí přiřazených k hlasovému příkazu (CommandSequenece) s tabulkou adres registrů v rámci Modbus protokolu.

Tab. 4.13: zahrnuté sloupce do pohledu CommandSequenceView

Název sloupce	Zdrojová tabulka
ModbusCommandId	ModbusCommands / CommandSequence
CommandId	CommandSequence
SequenceOrder	CommandSequence
DataType	ModbusCommands
Address	ModbusCommands
DeviceName	ModbusCommands
Description	ModbusCommands

4.10 Analýza požadavků mobilní aplikace

Aplikace bude mít za úkol čekat na spouštěcí slovo. Rozpoznávání, zda bylo vyslovené dané slovo, bude z důvodu šetření energie a zatížení sítě prováděno offline.

Po zaznamenání spouštěcího slova se aktivuje online rozpoznávací služba, která bývá zpravidla přesnější než offline.

Uživateli bude pípnutím oznámeno, že může vyslovit svůj požadavek. Když uživatel domluví, nebo po 5 vteřinách ticha, se online rozpoznávání ukončí a výsledný rozpoznaný text se pošle v JSON objektu, pomocí HTTP POST požadavku do webové aplikace ke zpracování.

Jako odpověď z webové aplikace může přijít text se zpětnou vazbou nebo dotazem na doplňující informace k požadavku. V takovém případě bude tento text syntetizován na řeč a přehrán uživateli. Pokud se bude jednat o otázku, tak se opět aktivuje online rozpoznávání řeči, aby uživatel mohl doplnit svůj požadavek. Poté co uživatel domluví, se znovu odešle požadavek pro webovou aplikaci s textem doplněné informace a objektem stavu minulého zpracování (už předtím zpracovaných klíčových slov), od kterého se webová aplikace bude mít možnost odrazit.

4.11 Technická specifikace mobilní aplikace

Mobilní aplikace bude využívat SpeechRecognizer API od Google, které je dostupné pouze pro operační systém Android.

Tato API podporuje češtinu jen v online režimu, z toho vyplívá, že další požadavek – aby byl telefon připojený k internetu.

5 Implementace

5.1 Přiřazení zařízení k Modbus objektům

Pro ovládání nainstalovaných zařízení v laboratoři byl vytvořený skript, který propojuje řídící proměnné PLC programu s Modbus objekty.

Modbus protokol rozlišuje 4 druhy objektů (Tab. 5.1), z nichž 2 lze použít jak ke čtení, tak i pro zápis. V tabulkách (Tab. 5.2) a (Tab 5.3) je přehled přiřazených funkcí zařízení.

Tab. 5.1: druhy objektů Modbus protokolu

Typ objektu	Přístup	Velikost paměti	Rozsah adres
Cívka	Čtení i zápis	1 bit	00001–09999
Diskrétní vstup	Pouze čtení	1 bit	10001–19999
Vstupní registr	Pouze čtení	16 bitů	30001–39999
Holding registr	Čtení i zápis	16 bitů	40001–49999

Tab. 5.2: přiřazení funkcí zařízení k Modbus objektu s typem diskrétní hodnoty (cívka)

Zařízení	Funkce	Adresa	Datový typ
Světlo nad postelí	Zapnutí / vypnutí	0x00	BOOL
Světlo u televize	Zapnutí / vypnutí	0x01	BOOL
Světlo v kuchyni	Zapnutí / vypnutí	0x02	BOOL
Dveře do obývacího pokoje	Otevření	0x03	BOOL
Dveře do koupelny	Otevření	0x04	BOOL
RGB LED u postele	Zapnutí / vypnutí	0x05	BOOL
RGB LED u televize	Zapnutí / vypnutí	0x06	BOOL
Lampa u postele	Zapnutí / vypnutí	0x07	BOOL
Světlo pro kuchyňskou linku	Zapnutí / vypnutí	0x08	BOOL
Světlo na chodbě	Zapnutí / vypnutí	0x09	BOOL
Světlo v koupelně	Zapnutí / vypnutí	0x10	BOOL
Světlo u zrcadla	Zapnutí / vypnutí	0x11	BOOL
Termostat v obývacím pokoji	Snížení teploty o 1 °C	0x12	BOOL
Termostat v obývacím pokoji	Zvýšení teploty o 1 °C	0x13	BOOL
Klimatizace	Režim zahřívání	0x14	BOOL
Klimatizace	Režim chlazení	0x15	BOOL
Termostat v koupelně	Snížení teploty o 1 °C	0x16	BOOL
Termostat v koupelně	Zvýšení teploty o 1 °C	0x17	BOOL
Žaluzie u postele	Stažení žaluzií dolů	0x18	BOOL
Žaluzie u postele	Vytažení žaluzií nahoru	0x19	BOOL
Žaluzie u televize	Stažení žaluzií dolů	0x20	BOOL
Žaluzie u televize	Vytažení žaluzií nahoru	0x21	BOOL

Tab. 5.3: přiřazení funkcí zařízení k Modbus objektu s typem 16-bit. registru (holding registr)

Zařízení	Funkce	Adresa	Datový typ
Světlo nad postelí	Nastavení jasu (0–100)	0x00	UINT
Světlo nad postelí	Nastavení barvy teploty (0–100)	0x01	UINT
Světlo u televize	Nastavení jasu (0–100)	0x02	UINT
Světlo u televize	Nastavení barvy teploty (0–100)	0x03	UINT
Světlo v kuchyni	Nastavení jasu (0–100)	0x04	UINT
Světlo v kuchyni	Nastavení barvy teploty (0–100)	0x05	UINT
RGB LED u televize	Červená složka barvy (0–255)	0x06	UINT
RGB LED u televize	Zelená složka barvy (0–255)	0x07	UINT
RGB LED u televize	Modrá složka barvy (0–255)	0x08	UINT
RGB LED u postele	Červená složka barvy (0–255)	0x09	UINT
RGB LED u postele	Zelená složka barvy (0–255)	0x10	UINT
RGB LED u postele	Modrá složka barvy (0–255)	0x11	UINT
Světlo na chodbě	Nastavení jasu (0–100)	0x12	UINT
Světlo v koupelně	Nastavení jasu (0–100)	0x13	UINT
Světlo u zrcadla	Nastavení jasu (0–100)	0x14	UINT
Termostat v obývacím pokoji	Aktuální teplota (desetinásobek)	0x15	UINT
Termostat v koupelně	Aktuální teplota (desetinásobek)	0x16	UINT
Termostat v obývacím pokoji	Nastavená teplota	0x17	UINT
Termostat v koupelně	Nastavená teplota	0x18	UINT

Zatímco pro jednoduché funkce jako je například zapnutí / vypnutí zařízení, simulaci stisku tlačítka a zjištění stavu stačí použití Modbus objektu s diskrétním typem hodnot, pro složitější funkce, které nastavují jas, barvu a teplotu, bylo potřeba použít objekt s 16-bit registry pro uložení nebo přečtení nastavené hodnoty.

5.2 Implementace webové aplikace

Webová aplikace byla napsána v jazyce PHP s využitím MariaDB databáze. Struktura aplikace je založená na architektuře MVC.

MVC (Model-View-Controller) je architektonický návrhový vzor, který odděluje datový model aplikace, uživatelské rozhraní a řídící logiku do tří nezávislých komponent tak, že modifikace kterékoliv z nich má jen minimální vliv na zbylé dvě.

Model je doménově specifická reprezentace informací. Patří sem například model databázové tabulky s metody pro získání a práci s těmito daty.

View (pohled) převádí data reprezentovaná modelem do vhodné podoby k prezentaci uživateli.

Controller (řadič) reaguje na události a zajišťuje změny v modelu.

Kromě základních tří komponent MVC architektury, obsahuje aplikace navíc komponenty typu **core**, **helper** a **skill**.

Komponenty typu **core** zajišťují základní funkce aplikace. Patří sem abstraktní třída řadičů a hlavní řadič, který má na starosti načtení požadovaného řadiče z požadavku, přesměrování nebo vypsání informace o chybě. Dále zde také nalezneme objekt pro připojení a práci s databází, objekt výjimky pro vyvolání přesměrování, objekt předávající jednotlivé instance vytvořených služeb mezi hlavním řadičem a cílovým řadičem a konfigurační objekt.

Komponenty typu **helper** slouží jako pomocné funkce pro vytváření jednotlivých částí HTML výstupu nebo pro komunikaci přes Modbus protokol.

Komponenty typu **skill** slouží pro přizpůsobené vykonávání jednotlivých příkazů. Některé příkazy například slouži jen pro zapnutí nebo vypnutí zařízení, ale jiné pro nastavení barvy, kdy je potřeba z 24-bit hodnoty, přiřazené konkrétní barvě, postupně vyčíst první tři bajty, a jejich hodnoty následně přiřazovat jednotlivým registrům Modbus objektu.

5.2.1 Komponenty typu core

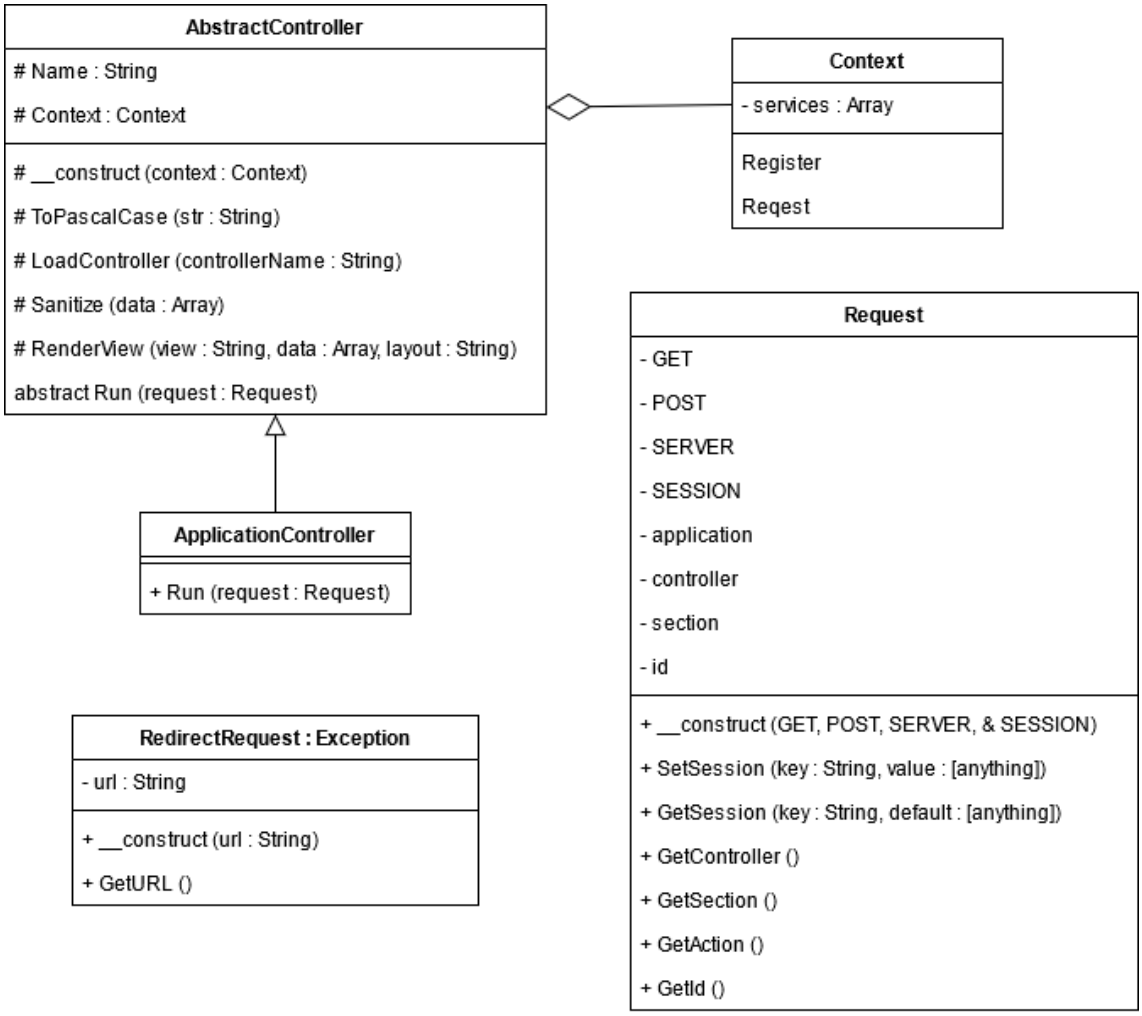
Sem patří abstraktní třída řadiče **AbstractController**, která má jednu abstraktní metodu se jménem Run. Tato metoda přijímá jako argument objekt Request, a je to hlavní metoda řadiče, která se volá pro spuštění zpracovávání algoritmu řadiče. Zmíněný objekt **Request** také patří mezi core komponenty. Jeho úlohou je vytvořit abstraktní vrstvu mezi řadičem a superglobálními proměnnými $_GET, $_POST, $_SERVER a $_SESSION. Při inicializaci analyzuje parametr v URL adrese, ze kterého zjistí hodnoty jednotlivých kategorií požadavku, jako je požadovaný kontrolér, sekce, akce a id.

Mimo jiné AbstractController obsahuje jednu instanci objektu **Context**, ve kterém jsou uložené reference na jednotlivé služby (instance závislostních objektů) aplikace. Služby se inicializují v hlavním aplikačním řadiči **ApplicationController**, a díky tomuto objektu je možné předat instance inicializovaných objektů požadovanému řadiči. V tomto případě má aplikace jen jednu službu, a tou je databáze.

Jako další komponenta je zde objekt **RedirectRequest**, který dědí ze zabudovaného objektu Exception. Tímto objektem je tedy možné vyvolat výjimku, která bude zachycená v hlavním aplikačním řadiči a přesměruje uživatele na požadovanou URL.

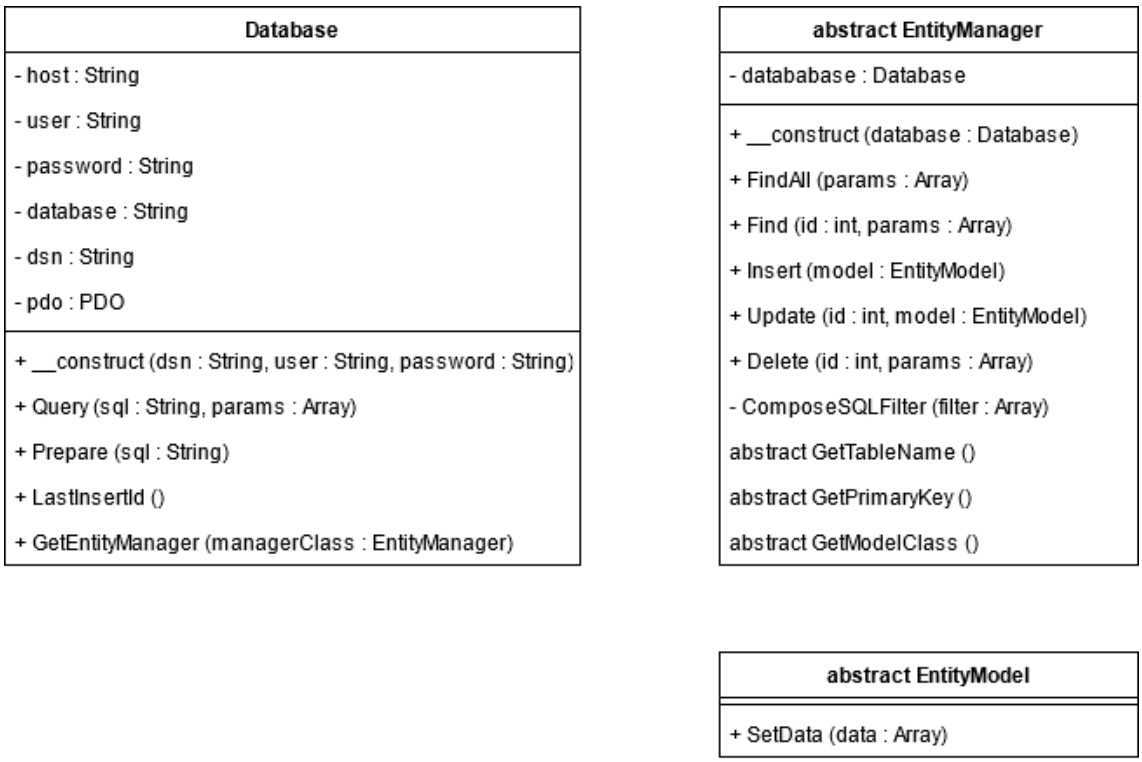
Tyto komponenty slouží pro zajištění základních funkcí samotné aplikace. Jelikož ale aplikace využívá také databázi, tak jsou do aplikace zahrnuty objekty pro zjednodušení práce s databází a zajišťují ORM – objektově relační mapování.

Diagram základních tříd je zobrazený na následujícím obrázku. (Obr. 5.1)



Obr. 5.1: Diagram tříd pro core komponenty

Mezi základní ORM objekty (Obr. 5.2) patří objekt Database a dva abstraktní objekty: EntityModel a EntityManager.



Obr. 5.2: Diagram tříd základních ORM objektů

Objekt **Database** uchovává referenci na aktivní připojení k databázi a disponuje metodami pro vytvoření dotazu (Query), vytvoření předpřipraveného dotazu (Prepare), zjištění id naposled vloženého záznamu (LastInsertId) a vytvoření řídícího objektu databázové tabulky (GetEntityManager).

Abstraktní objekt **EntityManager** zajišťuje, že každý odvozený objekt pro práci s databází bude implementovat tří metody:

- GetTableName,
- GetPrimaryKey
- GetModelClass

Díky informacím získaných z těchto tří metod je možné sestavit dotazy pro výběr dat, vložení nového záznamu, upravení stávajícího záznamu a odstranění záznamu.

Druhý abstraktní objekt **EntityModel** s metodou SetData zajišťuje, že se do instance objektu nedostanou atributy, které pro daný model nejsou definované. Jazyk PHP jinak umožňuje s objektem pracovat jako s asociativním polem, a za běhu dynamicky vytvářet nové atributy. To by mohlo způsobit problém při vytváření nového záznamu, kdy se očekává, že instance datového modelu bude obsahovat jen atributy, které jsou v tabulce.

5.2.2 Komponenty typu controller

Sem patří jednotlivé řadiče, které lze zavolat zadáním jejich názvu do URL.

Například relativní URL „/admin“ zavolá řadič AdminController.

Aplikace obsahuje 6 řadičů.

AdminController obsluhuje administrační sekci aplikace.

CommandController představuje hlavní část asistenta. Jedná se o řadič, který na základě analýzy textu s příkazem zjišťuje záměr uživatele a posílá požadované instrukce do PLC.

DefaultController se spustí, když uživatel nezadá žádný kontrolér. Jeho funkce je, že uživatele přesměruje do administrátorské sekce. Pokud ale uživatel není přihlášený, tak bude přesměrován na přihlášení.

Pokud při zpracovávání nastane nějaká chyba, neošetřená výjimka, nebo uživatel zadá jméno neexistujícího kontroléru, tak se zavolá **ErrorPageController**, který uživateli oznámí, že nastala chyba.

LoginController slouží pro přihlášení uživatele do administrátorské sekce.

LogoutController slouží pro odhlášení uživatele.

Každý z těchto 6 řadičů je potomkem objektu AbstractController a implementuje metodu Run.

5.2.3 Komponenty typu view

Na konci zpracování hlavní metody řadiče může být zavolaná metoda RenderView, která načte požadovaný pohled. Pohledem se myslí soubor s HTML výstupem, který může obsahovat malé kousky PHP kódu kvůli vypisování obsahu proměnných, cyklickému vytváření obsahu nebo podmíněnému větvení.

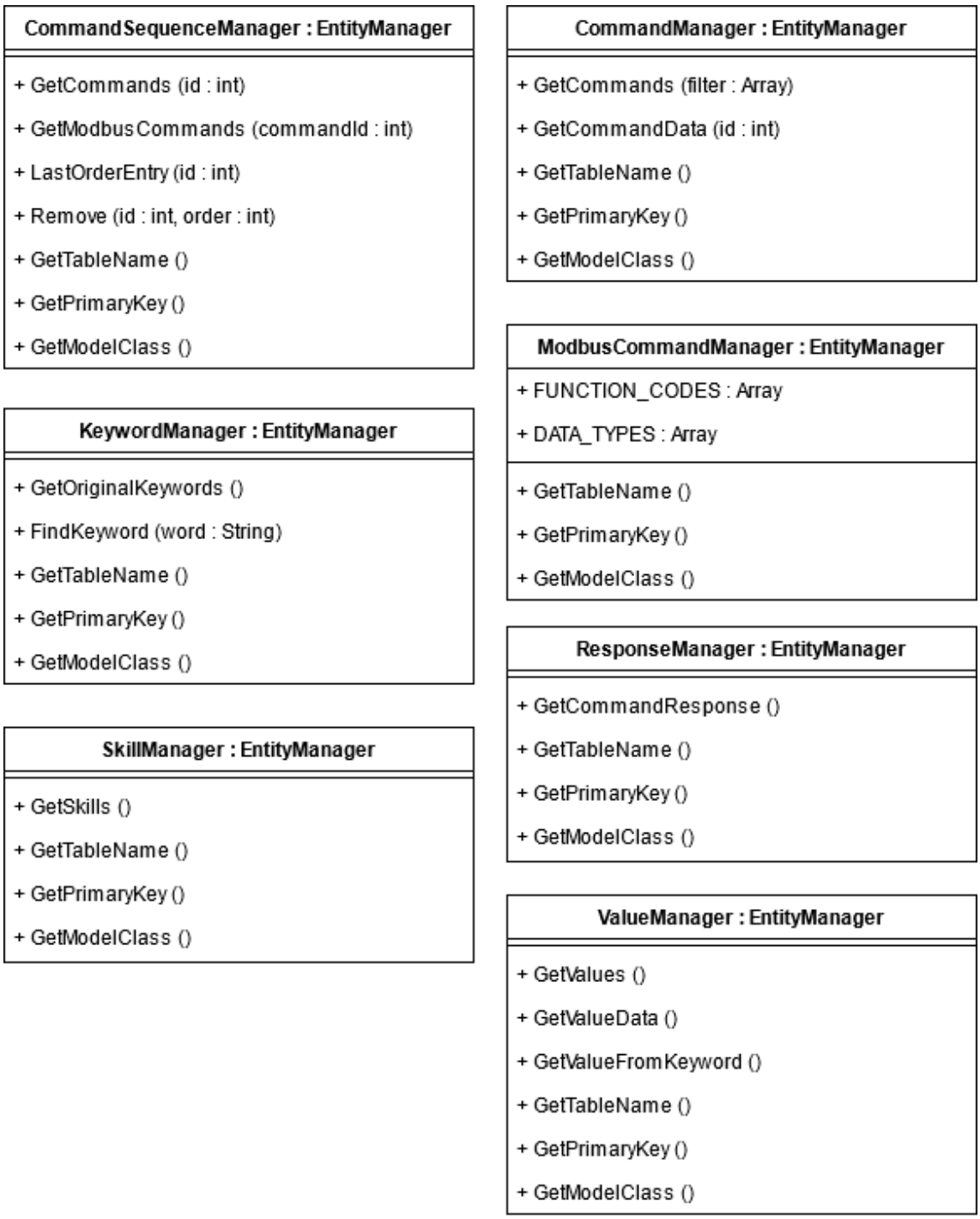
5.2.4 Komponenty typu model

Pro každou tabulku v databázi je vytvořen řídící objekt a datový model.

Řídící objekt dědí ze základního objektu EntityManager a slouží pro zajištění změn v databázi nebo vytažení dat. Některé řídící objekty implementují vlastní metody pro případy, kdy je potřeba vybírat data z pohledu, nebo daná tabulka nemá standardní primární klíč celočíselného typu a nad jediným sloupcem, jako například tabulka CommandSequence.

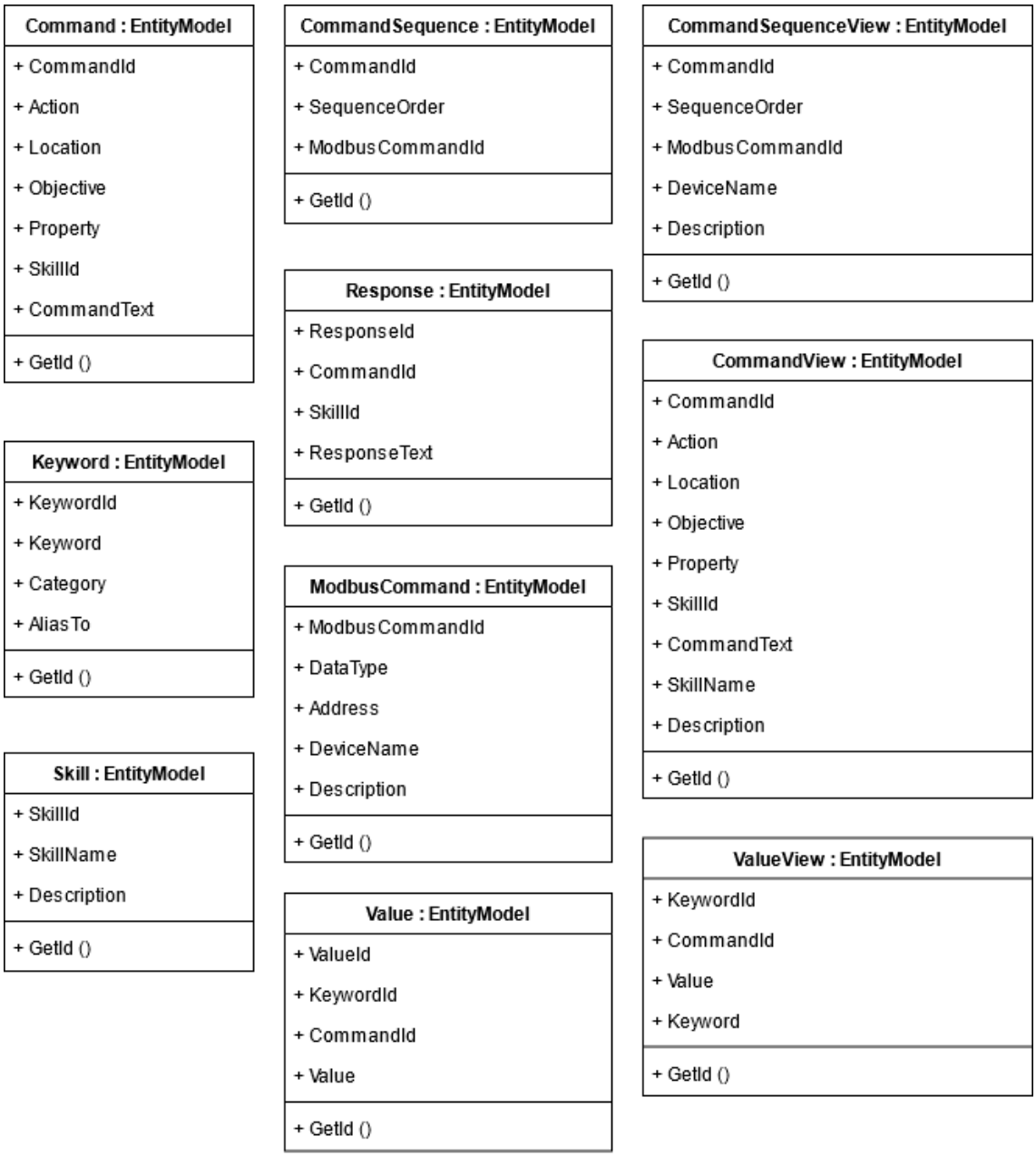
Každý datový model je potomkem základního ORM objektu EntityModel a reprezentuje datovou strukturu databázové tabulky nebo pohledu.

Diagram tříd řídících objektů je zobrazen na obrázku (Obr. 5.3)



Obr. 5.3: Diagram tříd řídících ORM objektů

Diagram tříd datových modelů je zobrazený na obrázku (Obr. 5.4).



Obr. 5.4: Diagram tříd datových modelů pro ORM

5.2.5 Komponenty typu helper

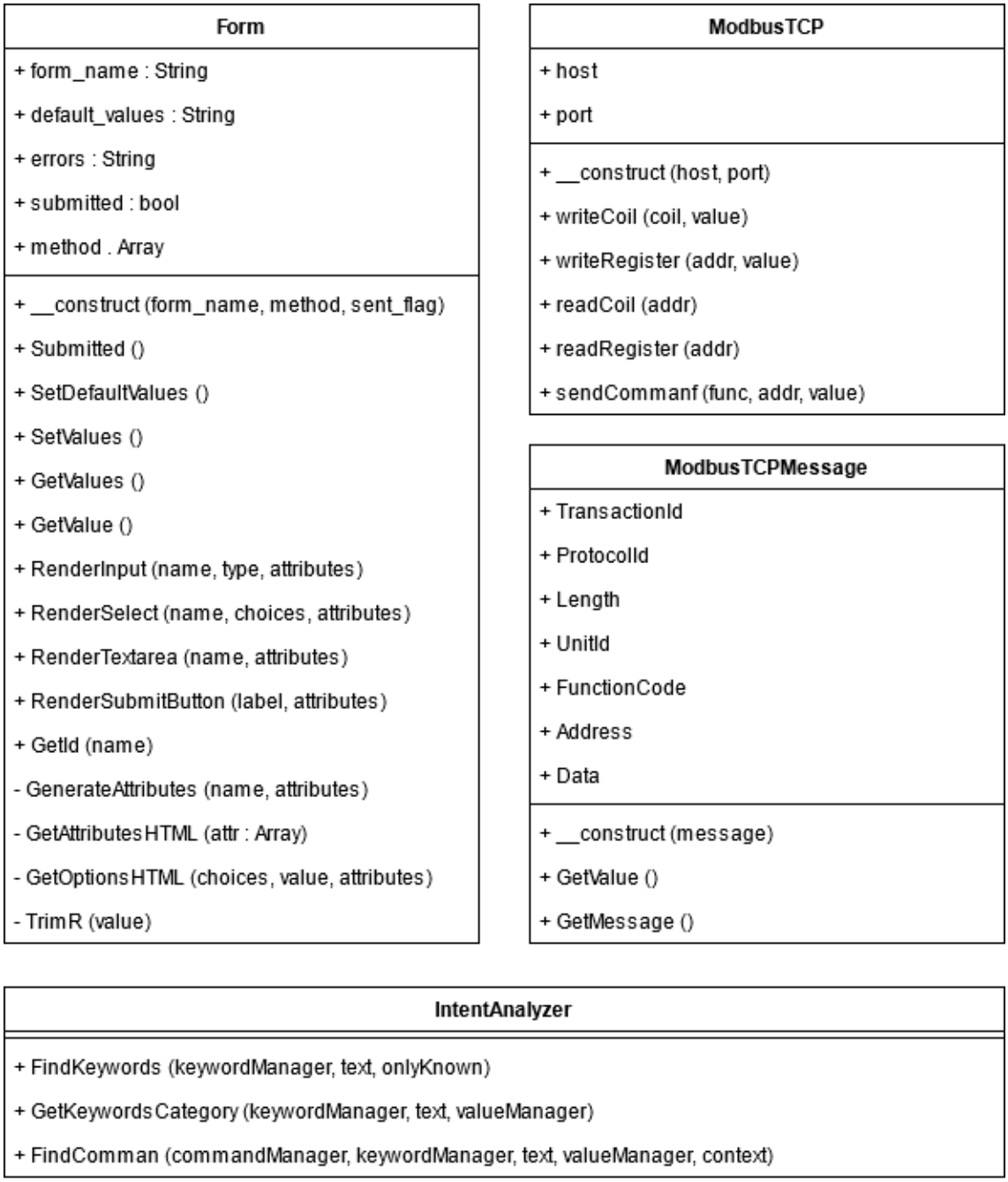
Sem patří pomocné třídy (Obr. 5.5), které pomáhají zajistit požadovanou funkcionalitu.

Třída **Form** pomáhá vytvářet HTML kód pro jednotlivé formulářové prvky.

Třída **ModbusTCP** zajišťuje odeslání ModbusTCP příkazu a přijetí odpovědi.

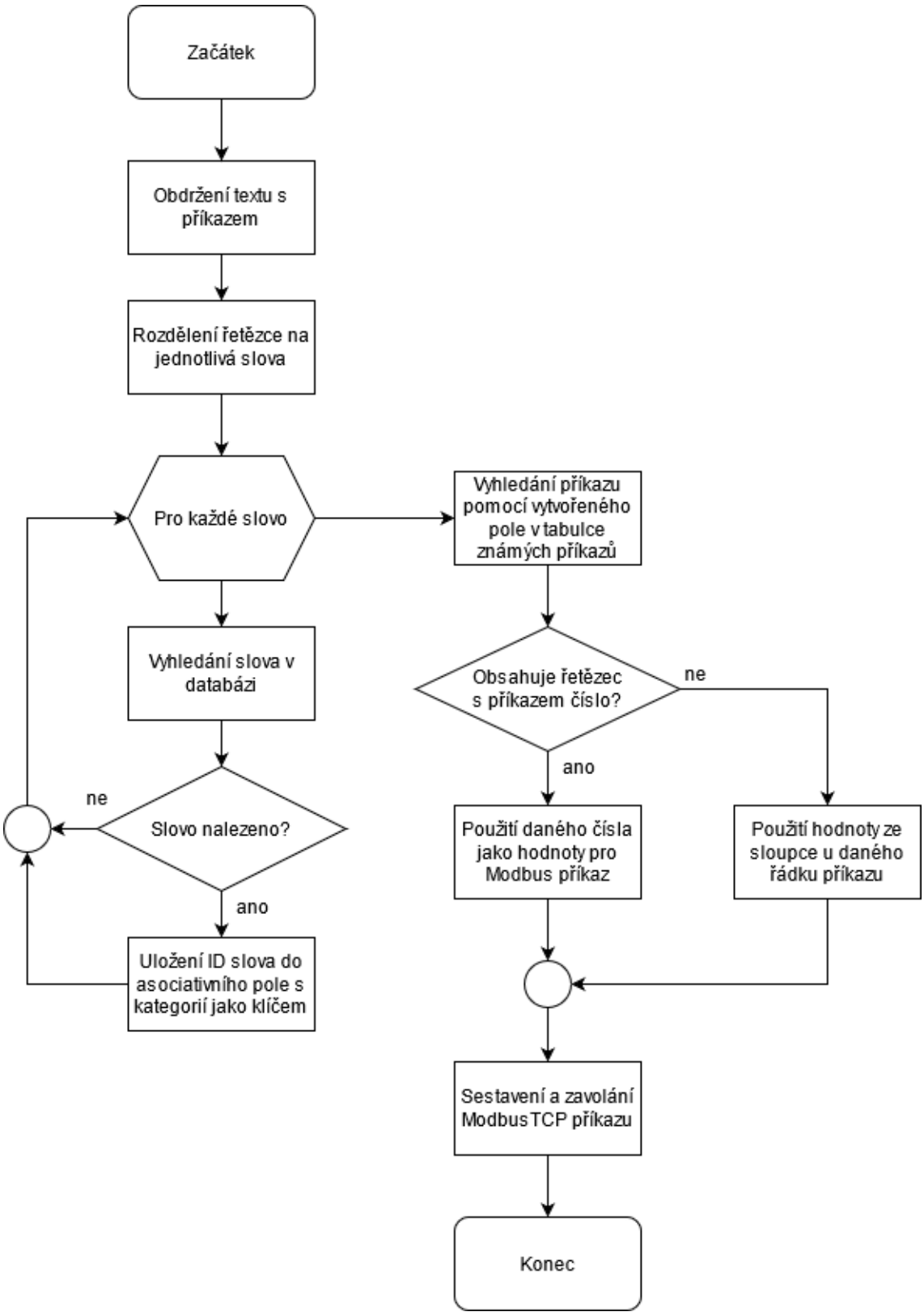
Třída **ModbusTCPMessage** převádí ModbusTCP zprávu do srozumitelné podoby.

Třída **IntentAnalyzer** obsahuje metody pro nalezení klíčových slov a jejich kategorií v textu, a nalezení instrukcí k hlasového příkazu.



Obr. 5.5: Diagram pomocných tříd

Vývojový diagram algoritmu hledání klíčových slov a přiřazeného hlasového příkazu je na obrázku (Obr. 5.6).



Obr. 5.6: Algoritmus hledání klíčových slov a přiřazeného příkazu

5.2.6 Komponenty typu skill

Poslední druh komponentů, které se ve zdrojových kódech webové aplikace vyskytují, nese název skill. Jedná se o řadiče jednotlivých hlasových příkazů.

Každý z těchto řadičů implementují abstraktní třídu SkillController, která obsahuje definovaný konstruktor a deklaruje jednu abstraktní metodu Run.

ButtonController má za úkol simulovat stisk tlačítka. To provede tak, že na přiřazenou adresu Modbus registru zapíše do objektu diskrétních hodnot jedničku. Hodnota se následně sama zase změní na 0. PLC program však mezitím stačí zaregistrovat změnu a provede požadovanou akci.

ColorController nastavuje barvu podle RGB modelu. Z obdržené číselné hodnoty nejdřív přečte první, potom druhý, a nakonec třetí bajt, a postupně zapíše tyto hodnoty do přiřazených Modbus registrů. Očekává, že k hlasovému příkazu budou přiřazené tři Modbus registry.

DimmableLightPowerController ovládá zapnutí světla s nastavitelným jasem. Po studeném restartu PLC bývá jas nastavený na 0 a obyčejné zapnutí světla by se neprojevilo. Proto tento kontrolér nejdříve zjistí, jaký je aktuálně nastavený jas, a pokud by byl pod 10 %, tak nastaví jas na 75 %.

LongPressButtonController simuluje dlouhý stisk tlačítka po dobu 2 sekund.

PercentageController nastavuje parametr, který lze vyjádřit v procentech, například jas.

PowerController se na rozdíl od kontroléru DimmableLightPowerController nestará o aktuální nastavenou hodnotu nějakého jiného parametru, ale jednoduše jen nastaví hodnotu na požadované adrese na 1 nebo 0.

ResponseController se používá pro sdělení uživateli zpětné vazby nebo pro dotaz na doplňující informace.

RoomTemperatureController nastavuje teplotu v místnosti simulací opakovaného stisknutí tlačítka, protože nebylo možné přímo zapisovat do proměnné nastavené teploty v PLC programu.

UnknownController se zavolá, pokud by nebyl nalezen žádný známý hlasový příkaz na vyslovenou kombinaci klíčových slov. Sdělí uživateli, že příkazu nebylo porozuměno.

5.2.7 Licence

Zdrojové kódy webové aplikace jsou dostupné pod licencí BSD.

5.3 Implementace mobilní aplikace

Mobilní aplikace je vytvořena pro platformu Android a je postavena na aplikaci PocketSphinx demo, dostupné na GitHubu. [46]

Využívá dva rozpoznávací systémy – PocketSphinx a SpeechRecognition API od Google.

PocketSphinx je v aplikaci použit pro offline rozpoznávání a detekci spouštěcího slova.

Inicializace každého z rozpoznávačů je prováděná ve zvláštní třídě, aby byl kód přehlednější.

Pro reakci na jednotlivé události je použitý princip posluchačů událostí.

Aplikace čeká na spouštěcí slovo, pokud ho zaznamená, zavolá metodu onWakeup na přiřazeném posluchači událostí. Implementace této metody je v hlavním vláknu aktivity.

Metoda onWakeup spustí SpeechRecognition API, které provede rozpoznávání řeči na serveru. Když bude mít rozpoznaný text příkazu, tak se zavolá metoda onResult, ve které se pomocí knihovny Volley vytvoří HTTP POST požadavek na webovou službu. V těle požadavku bude JSON objekt s parametrem „commandText“.

Po obdržení odpovědi z webové aplikace, nebo v případě nějaké chyby, se aplikace znovu vrátí do počátečního stavu, kde čeká na zaznamenání spouštěcího slova.

Zdrojové kódy k mobilní aplikaci jsou dostupné pod licencí BSD.

6 Uživatelská dokumentace

6.1 Hlasové příkazy asistenta

V této části je popsán návod pro používání hlasových příkazů.

Pro aktivaci online rozpoznávání hlasu je potřeba vyslovit spouštěcí slovo, které je „Zoro“. Pokud aplikace zaznamená, že bylo toto slovo vysloveno, podá pípnutím signál uživateli, že může vyslovit samotný příkaz.

Implementované příkazy jsou následující:

„Rozsviť / zhasni nad postelí.“

„Rozsviť / zhasni u televize.“

„Rozsviť / zhasni v kuchyni.“

„Rozsviť / zhasni na chodbě.“

„Rozsviť / zhasni v koupelně.“

„Rozsviť / zhasni u zrcadla.“

„Nastav jas světla u postele na {0–100} procent.“

„Nastav jas světla u televize na {0–100} procent.“

„Nastav jas světla v kuchyni na {0–100} procent.“

„Nastav jas světla na chodbě na {0–100} procent.“

„Nastav jas světla v koupelně na {0–100} procent.“

„Nastav jas světla u zrcadla na {0–100} procent.“

„Nastav barvu světla nad postelí na teplou/studenou/neutrální.“

„Nastav barvu světla u televize na teplou/studenou/neutrální.“

„Nastav barvu světla v kuchyni na teplou/studenou/neutrální.“

„Nastav barvu LED u postele na {název barvy}.“

„Nastav barvu LED u televize na {název barvy}.“

„Zhasni LED na zdi u postele.“

„Zhasni LED na zdi u televize.“

„Rozsviť / zhasni lampu u postele.“

„Zapni / vypni lampu u postele.“

„Otevři dveře do obývacího pokoje.“

„Otevři dveře do koupelny.“

„Vytáhni žaluzie u postele nahoru.“

„Stáhni žaluzie u postele dolů.“

„Vytáhni žaluzie u televize nahoru.“

„Stáhni žaluzie u televize dolů.“

„Nastav teplotu v obývacím pokoji na {10–30} stupňů.“

„Nastav teplotu v koupelně na {10–30} stupňů.“

„Zhasni vše.“

Následující příkazy obsahují přiřazenou zpětnou vazbu pro uživatele nebo dotaz na doplňující informace.

„Jaká je teplota obývacím pokoji?“ – „Teplota v místnosti je {číslo} stupňů.“

„Jaká je teplota v koupelně?“ – „Teplota v místnosti je {číslo} stupňů.“

„Rozsviť.“ – „Které světlo si přejete rozsvítit?“

„Zhasni“ – „Které světlo si přejete zhasnout?“

„Otevři dveře“ – „Které dveře si přejete otevřít – k obývacímu pokoji nebo do koupelny?“

6.2 Administrační rozhraní webové aplikace

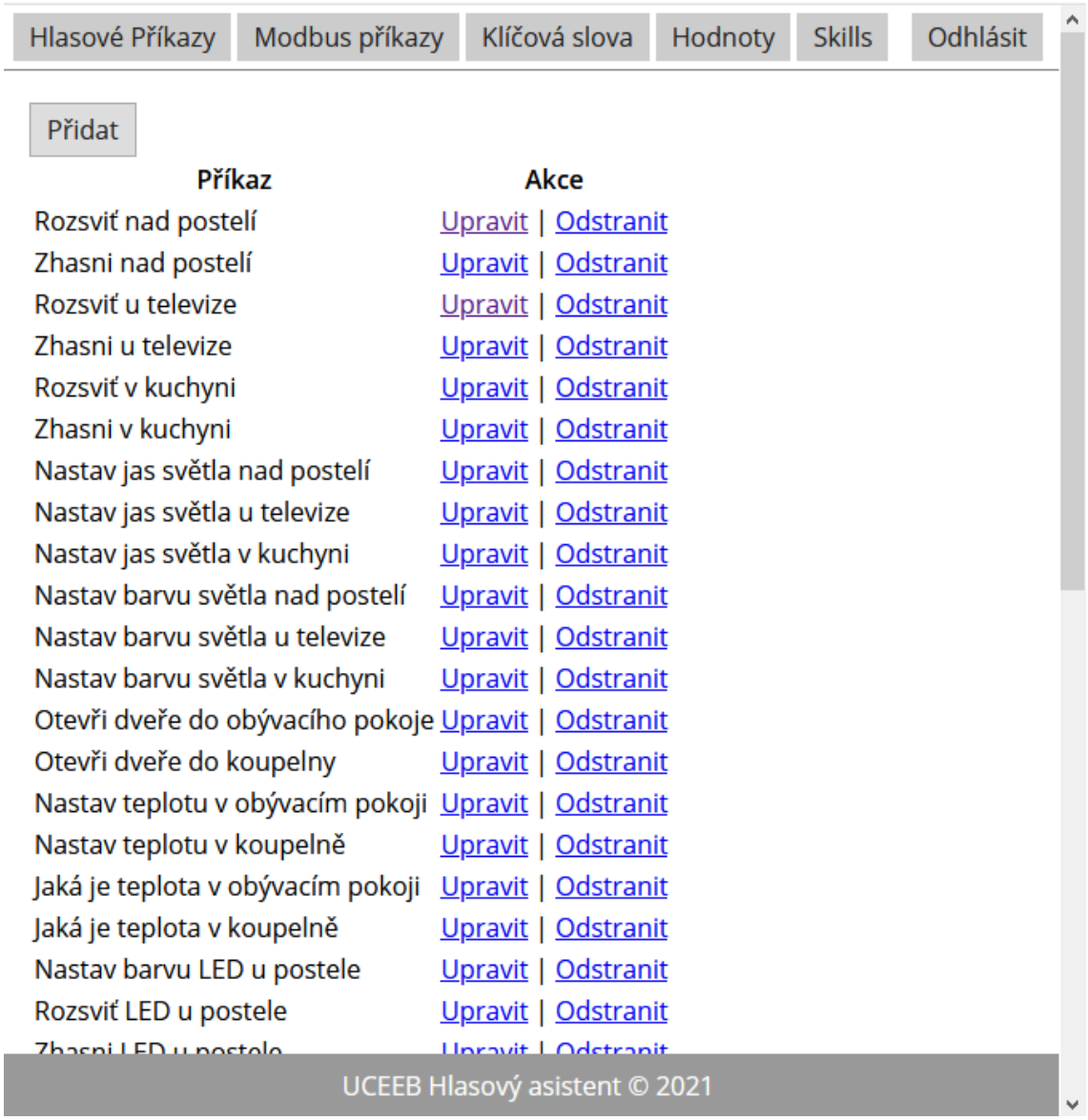
Pro pohodlnější správu databáze s hlasovými příkazy byla vytvořena administrační sekce. Do této sekce má přístup pouze administrátor, který má přihlašovací údaje.

Při prvním přístupu se zobrazí přihlašovací obrazovka. (Obr. 6.1)

Obr. 6.1: přihlašovací obrazovka

6.2.1 Správa hlasových příkazů

Po přihlášení bude administrátor přesměrován do sekce správy hlasových příkazů. Zde jsou vypsané všechny vytvořené hlasové příkazy a u každého je odkaz pro úpravu nebo odstranění hlasového příkazu. (Obr. 6.2)



Obr. 6.2: uživatelské rozhraní pro správu hlasových příkazů

Po kliknutí na tlačítko „Přidat“ nad tabulkou s příkazy, nebo na odkaz „Upravit“ u některého z hlasových příkazů se uživatel dostane k formuláři pro úpravu příkazu. (Obr. 6.3)

Stejný formulář se zobrazí i po kliknutí na tlačítko „Přidat“, akorát zde nebudou vyplněné formulářová pole.

Obr. 6.3: formulář pro editaci nebo vytvoření nového záznamu s hlasovým příkazem

Po vytvoření nového hlasového příkazu, nebo po kliknutí na tlačítko „Přidat“ pod formulářem, bude uživatel přesměrován do sekce Modbus příkazů (Obr. 6.4), akorát s tím rozdílem, že místo možností „Upravit“ a „Odstranit“ bude ve sloupci akce jen možnost „Zvolit“. Po kliknutí na tento odkaz u příslušného Modbus příkazu se přiřadí do sekvence instrukcí k upravovanému hlasovému příkazu.

6.2.2 Správa Modbus příkazů

V této sekci nalezneme přehled zařízení a funkcí přiřazených k jednotlivým registrům Modbus objektu. (Obr. 6.4) U každého příkazu je možnost upravit nebo odstranit záznam.

Přidat

Zařízení	Popis	Datový typ	Adresa	Akce
Světlo nad postelí	*Zapnutí / vypnutí světla nad postelí*	BOOL	0x00	Upravit | Odstranit
Světlo u televize	*Zapnutí / vypnutí světla u televize*	BOOL	0x01	Upravit | Odstranit
Světlo v kuchyni	*Zapnutí / vypnutí světla v kuchyni*	BOOL	0x02	Upravit | Odstranit
Dveře obývák	*Otevření dveří do obývacího pokoje (otevřou se na cca 10s)*	BOOL	0x03	Upravit | Odstranit
Dveře koupelna	*Otevření dveří do koupelny (otevřou se na cca 10s)*	BOOL	0x04	Upravit | Odstranit
RGB LED u postele	*Zapnutí / vypnutí barevné LED na zdi u postele*	BOOL	0x05	Upravit | Odstranit
RGB LED u televize	*Zapnutí / vypnutí barevné LED na zdi u televize*	BOOL	0x06	Upravit | Odstranit
Lampa u postele	*Zapnutí / vypnutí lampy u postele*	BOOL	0x07	Upravit | Odstranit
Stětlo na kuchyňské lince	*Zapnutí / vypnutí světla pro kuchyňskou linku*	BOOL	0x08	Upravit | Odstranit
Světlo na chodbě	*Zapnutí / vypnutí světla na chodbě*	BOOL	0x09	Upravit | Odstranit
Světlo v koupelně	*Zapnutí / vypnutí světla v koupelně*	BOOL	0x010	Upravit | Odstranit
Světlo u zrcadla	*Zapnutí / vypnutí světla u zrcadla (v koupelně)*	BOOL	0x011	Upravit | Odstranit
Termostat v obýváku	*Snížení nastavené teploty v obývacím pokoji o 1 °C*	BOOL	0x012	Upravit | Odstranit
Termostat v obýváku	*Zvýšení nastavené teploty v obývacím pokoji o 1 °C*	BOOL	0x013	Upravit | Odstranit
Klimatizace v obýváku	*Režim ohřívání*	BOOL	0x014	Upravit | Odstranit
Klimatizace v obýváku	*Režim chlazení*	BOOL	0x015	Upravit | Odstranit
Termostat v koupelně	*Snížení nastavené teploty v koupelně o 1 °C*	BOOL	0x016	Upravit | Odstranit
Termostat v koupelně	*Zvýšení nastavené teploty v koupelně o 1 °C*	BOOL	0x017	Upravit | Odstranit

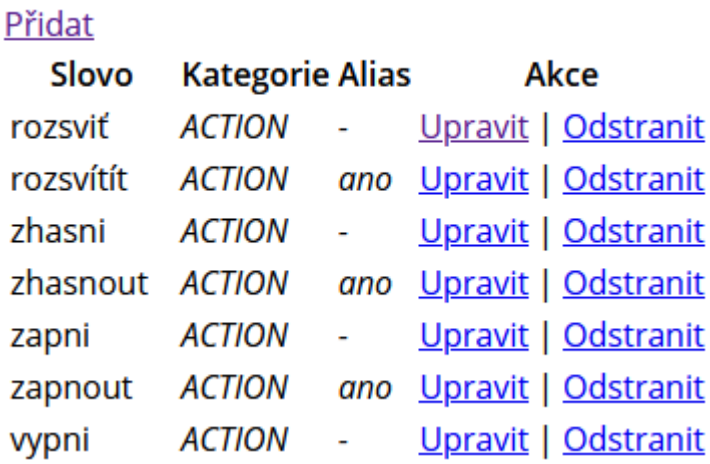
Obr. 6.4: přehled funkcí přiřazených k Modbus příkazům

Kliknutím na tlačítko „Přidat“ nad tabulkou nebo na odkaz „Upravit“ se přejde k formuláři pro editaci vybraného příkazu nebo vytvoření nového. (Obr. 6.5)

Obr. 6.5: formulář pro editaci nebo vytvoření nového zázamu Modbus příkazu

6.2.3 Správa klíčových slov

Zde je přehled vytvořených klíčových slov. (Obr. 6.6)



Obr. 6.6: ukázka přehledu klíčových slov uložených v databázi

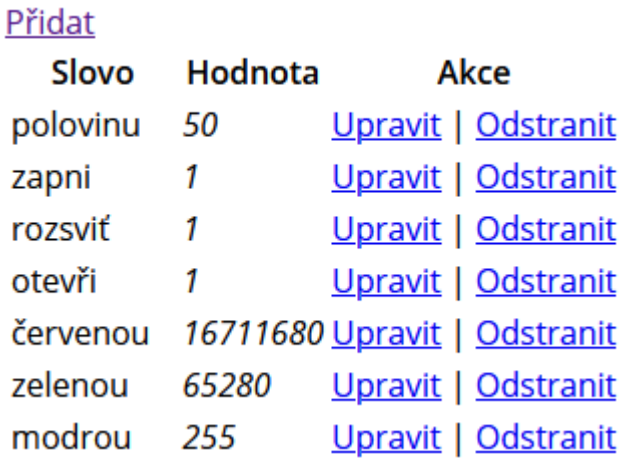
Podobně jako u předchozích sekcí, je tu také nad tabulkou tlačítko (odkaz) „Přidat“ pro vytvoření nového klíčového slova. U každého záznamu je také možnost upravit nebo odstranit vybraný záznam. Na obrázku (Obr. 6.7) je formulář pro úpravu nebo vytvoření nového klíčového slova.

Obr. 6.7: formulář pro editaci nebo vytvoření nového klíčového slova

6.2.4 Správa přiřazených hodnot ke klíčovým slovům

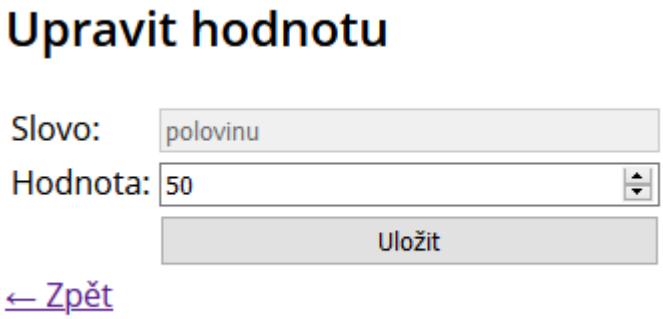
Některá klíčová slova mohou sloužit významově jen jako hodnota, nebo současně i jako jiná kategorie. Například slovo „rozsviť“ je vytvořené jako akce, ale je k němu přiřazena hodnota 1. To proto, aby aplikace mohla u hlasových příkazů „Rozsviť nad postelí“ a „Zhasni nad postelí“ – které mají přiřazený stejný Modbus registr – rozeznat, zda se má nastavit hodnota na 1 nebo na 0.

V sekci „Hodnoty“ nalezneme přehled slov, ke kterým je přiřazená nějaká číselná hodnota. (Obr. 6.8)



Obr. 6.8: přehled slov s přiřazenou číselnou hodnotou

Po kliknutí na tlačítko „Přidat“ nebo „Upravit“ se dostaneme na formulář pro úpravu nebo vytvoření nového slova s číselnou hodnotou. (Obr. 6.9)



Obr. 6.9: formulář pro upravení přiřazené hodnoty ke slovu

7 Testování

Modul hlasového ovládání skládající se z mobilní aplikace a webové aplikace byl nasazen a otestován v laboratoři personalizované medicíny v centru UCEEB.

Hlasové ovládání bylo vyzkoušené 5 osobami. Z důvodu Covid-19 epidemie nemohlo být provedené testování s lidmi z cílové skupiny, jelikož se jedná o ohroženou skupinu.

7.1.1 Průběh testování

Předmětem testování bylo ověření reakce na spouštěcí slovo „Zoro“ a následně úspěšnost rozpoznání záměru uživatele. Uživatelé byli seznámeni s příkazy, které mohou použít. Nejprve se vyzkoušeli jednodušší příkazy pro rozsvěcení a zhasínání jednotlivých světel, a následně složitější příkazy pro nastavení parametrů jako je jas, barva a teplota.

První uživatel

Prvním uživatelem, který testoval aplikaci, byl muž, 24 let.

V tabulce (Tab. 7.1) je přehled úspěšně a neúspěšně vykonaných příkazů.

Tab. 7.1: přehled úspěšnosti vykonání hlasových příkazů u prvního uživatele

Počet úspěšných rozpoznání spouštěcího slova	12
Počet neúspěšných rozpoznání spouštěcího slova	6
Počet falešných detekcí spouštěcího slova	0
Počet úspěšně vykonaných příkazů	10
Počet neúspěšně vykonaných příkazů	6

Druhý uživatel

Druhým uživatelem, který testoval aplikaci, byla žena, 46 let.

V tabulce (Tab. 7.2) je přehled úspěšně a neúspěšně vykonaných příkazů.

Tab. 7.2: přehled úspěšnosti vykonání hlasových příkazů u druhého uživatele

Počet úspěšných rozpoznání spouštěcího slova	15
Počet neúspěšných rozpoznání spouštěcího slova	8
Počet falešných detekcí spouštěcího slova	9
Počet úspěšně vykonaných příkazů	12
Počet neúspěšně vykonaných příkazů	3

Třetí uživatel

Třetím uživatelem, který testoval aplikaci, byl muž, 43 let.

V tabulce (Tab. 7.3) je přehled úspěšně a neúspěšně vykonaných příkazů.

Tab. 7.3: přehled úspěšnosti vykonání hlasových příkazů u třetího uživatele

Počet úspěšných rozpoznání spouštěcího slova	18
Počet neúspěšných rozpoznání spouštěcího slova	15
Počet falešných detekcí spouštěcího slova	3
Počet úspěšně vykonaných příkazů	14
Počet neúspěšně vykonaných příkazů	4

Během tohoto testování bylo nutné jednou restartovat mobilní aplikaci, protože se zdálo, že přestala reagovat na spouštěcí slovo.

Čtvrtý uživatel

Čtvrtým uživatelem, který testoval modul aplikaci byl muž, 36 let.

V tabulce (Tab. 7.4) je přehled úspěšně a neúspěšně vykonaných příkazů.

Tab. 7.4: přehled úspěšnosti vykonání hlasových příkazů u čtvrtého uživatele

Počet úspěšných rozpoznání spouštěcího slova	16
Počet neúspěšných rozpoznání spouštěcího slova	7
Počet falešných detekcí spouštěcího slova	10
Počet úspěšně vykonaných příkazů	14
Počet neúspěšně vykonaných příkazů	2

Pátý uživatel

Pátým uživatelem, který testoval aplikaci byla žena, 49 let.

V tabulce (Tab. 7.5) je přehled úspěšně a neúspěšně vykonaných příkazů.

Tab. 7.5: přehled úspěšnosti vykonání hlasových příkazů u pátého uživatele

Počet úspěšných rozpoznání spouštěcího slova	16
Počet neúspěšných rozpoznání spouštěcího slova	23
Počet falešných detekcí spouštěcího slova	9
Počet úspěšně vykonaných příkazů	12
Počet neúspěšně vykonaných příkazů	4

7.1.2 Vyhodnocení testování modulu hlasového ovládání

Tab. 7.5: součet všech parametrů úspěšnosti testovaného hlasového ovládání

Počet úspěšných rozpoznání spouštěcího slova	77
Počet neúspěšných rozpoznání spouštěcího slova	59
Počet falešných detekcí spouštěcího slova	31
Počet úspěšně vykonaných příkazů	62
Počet neúspěšně vykonaných příkazů	19

Úspěšnost rozpoznání spouštěcího slova „Zoro“ při použití knihovny PocketSphinx a akustického modelu pro americkou angličtinu je 56,61 %.

Úspěšnost rozpoznání řeči s příkazem pomocí SpeechRecognition API od Google s podporou češtiny je 76,54 %.

8 Diskuse

Z výsledků testování v předchozí kapitole je patrné, že největší problém u sestaveného modulu hlasového ovládání je rozpoznání spouštěcího slova s úspěšností 56,61 %. Při testování se poměrně často falešně detekovalo spouštěcí slovo, když bylo vyslovené jiné slovo, nebo se naopak občas stávalo, že se muselo spouštěcí slovo říct opakovaně, aby ho mobilní aplikace rozpoznala a spustila online rozpoznávání.

Při nastavení přísnějšího prahu detekce spouštěcího slova by se sice snížil počet případu falešné detekce, ale zároveň by se snížila úspěšnost správné detekce, která už nyní není příliš vysoká.

Tento stav je pravděpodobně způsobený použitím akustického modelu natrénovaného pro americkou angličtinu. Vytvoření akustického modelu pro češtinu by mělo úspěšnost rozpoznávání spouštěcího slova zlepšit.

Úspěšnost rozpoznávání vyslovených příkazů pomocí služby od Google byla 76,54 %. Některé případy neúspěšného rozpoznání byly způsobené pokusem vyslovit příkaz v přeházeném slovosledu, například místo „Rozsviť nad postelí“ bylo řečeno „Postelí nad rozsviť“. To bylo provedeno kvůli vyzkoušení, zda si webová aplikace, zjišťující záměr uživatele, poradí i s takovým příkazem. Po napsání příkazu přímo do webové služby v adresním řádku se příkaz vykonal a webová služba s tím tedy nemá problém. Z toho vyplívá, že problém bude ve službě pro rozpoznávání řeči, která používá mimo jiné i jazykový model, podle kterého je takové slovní spojení nepravděpodobné, a našla se nejspíš jiná kombinace slov, která by lépe odpovídala jazykovému modelu k rozpoznaným fonémům.

Všichni testovací uživatelé se po několika pokusech oslovení osvojili a aktivace asistenta byla úspěšnější. U lidí se sníženou fyzickou soběstačností (pohyb, zranění, ochrnutí) je běžné zachovat dobré schopnosti mluvy, proto si myslím, že po pár dnech bude pro uživatele již snadné aktivovat asistenta.

U skupiny lidí, která má problém s mluvou, je možné asistenta přizpůsobit adaptací akustického modelu. Schopnost interakce této skupiny lidí s hlasovým asistentem by bylo dobré podrobit dalšímu výzkumu. Minimálně pro detekci volání o pomoc je hlasový asistent dobrou pomůckou.

Navržený modul hlasového ovládání v současné době komunikuje jen s PLC, a je tedy schopný ovládat jen ta zařízení, která jsou připojená k této řídící jednotce bytu. Nicméně díky tomu, že PHP umožňuje spouštět systémové příkazy, je možné rozšířit schopnosti asistenta na systémové aplikace, například ovládání online rádia, nebo ovládat další zařízení s možností komunikace přes síť.

Kromě použití modulu čistě jen pro hlasové ovládání zařízení se nabízí také možnost vytvořit a natrénovat model na rozpoznávání zvuků rozbíjejícího se skla, padání předmětů, volání o pomoc apod. Následně pak bude možné na tyto podněty reagovat a přivolat pomoc.

9 Závěr

V této bakalářské práci byla provedena rešerše dostupných hlasových asistentů, avšak žádný z nich v současné době nepodporuje češtinu. Proto bylo pro splnění zadání nutné vytvořit vlastní aplikaci pro hlasové příkazy.

Toho bylo dosaženo vytvořením aplikace pro mobilní telefony s operačním systémem Android, a webové aplikace. Bylo také nutné upravit stávající program řídící jednotky chytré domácnosti, aby bylo možné zadávat instrukce z vytvořené webové aplikace.

Pro mobilní aplikaci byly použity dva systémy rozpoznávání hlasu – CMU Pocketsphinx a SpeechRecognizer API od Google.

Pocketsphinx byl použit pro offline detekci spouštěcího slova, zatímco SpeechRecognizer pro rozpoznání samotného hlasového příkazu. Využití offline řešení bylo zvoleno z důvodu šetření energie i datového toku.

Převedený hlasový příkaz na text je následně odeslán do webové aplikaci ke zpracování a zjištění záměru uživatele.

Webová aplikace rozděluje text na jednotlivá slova a porovnává je se záznamy v databázi. Pokud najde v databázi záznam odpovídající vyslovené kombinaci klíčových slov, tak pošle instrukce řídící jednotce bytu, a ta již vykoná požadovanou akci.

Výsledné řešení je vhodné pro lidi se sníženou soběstačností. Prokázalo se, že hlasové ovládání je vhodná alternativa pro běžné ovládací prvky v domácnosti (vypínače, tlačítka, kliky), ale i moderní (tablet, počítač), a umožňuje lidem s fyzickým handicapem plně interagovat se svou domácností, například si otevřít okno, dveře, rozsvítit apod.

Při testování bylo samotné rozpoznávání hlasu pomocí API od Google spolehlivé na 76,54 %, avšak detekce spouštěcího slova příliš spolehlivá nebyla (56,61 %).

Zdrojové kódy vytvořené webové i mobilní aplikace jsou dostupné pod licencí BSD.

Seznam použité literatury

[1] Adobe. State of Voice Technology for Brands. *Slideshare* [online]. 15.05.2019 [cit. 2021-4-13]. Dostupné z: https://www.slideshare.net/adobe/state-of-voice-technology-for-brands-145863065

[2] DOAN, Hoang. *Modul hlasového ovládání a jeho integrace do vybrané služby*. Kladno, 2019. Projekt III. FBMI, ČVUT.

[3] How Can Voice Assistants Understand Us? *Wonderpolis* [online]. [cit. 2021-4-13]. Dostupné z: https://wonderopolis.org/wonder/how-can-voice-assistants-understand-us

[4] Mobile & Tablet Operating System Market Share Czech Republic: Mar 2020 - Apr 2021. *StatCounter* [online]. [cit. 2021-4-20]. Dostupné z: https://gs.statcounter.com/os-market-share/mobile-tablet/czech-republic/#monthly-202003-202104-bar

[5] TILLMAN, Maggie. What is Google Assistant and what can it do? *Pocket-lint* [online]. 2021 [cit. 2021-4-20]. Dostupné z: https://www.pocket-lint.com/apps/news/google/137722-what-is-google-assistant-how-does-it-work-and-which-devices-offer-it

[6] MATÝSEK, Jiří V. Google Assistant v češtině: jak to s ním je a kdy se jej dočkáme v plné podobě? *Alza* [online]. 22. září 2020 [cit. 2021-4-20]. Dostupné z: https://www.alza.cz/google-assistant-v-cestine

[7] Services and smart devices that work with Google Assistant. *Google Nest Help* [online]. [cit. 2021-4-20]. Dostupné z: https://support.google.com/googlenest/answer/7639952

[8] Set up smart home devices using your speaker or display. *Google Nest Help* [online]. [cit. 2021-4-20]. Dostupné z: https://support.google.com/googlenest/answer/9159862

[9] Change the language of Google Assistant: Languages you can use. *Google Nest Help* [online]. [cit. 2021-4-20]. Dostupné z: https://support.google.com/googlenest/answer/7550584

[10] Create commands to control online services & devices. *Google Nest Help* [online]. [cit. 2021-4-2]. Dostupné z: https://support.google.com/googlenest/answer/7194656

[11] Get Started with the Alexa Skills Kit: How an Alexa Skill Works. *Amazon Alexa* [online]. [cit. 2021-5-21]. Dostupné z: https://developer.amazon.com/en-US/alexa/alexa-skills-kit/start

[12] What is the Alexa Skills Kit? *Amazon Alexa* [online]. [cit. 2021-4-21]. Dostupné z: https://developer.amazon.com/en-US/docs/alexa/ask-overviews/what-is-the-alexa-skills-kit.html

[13] Smart Home Skill APIs. *Amazon Alexa* [online]. [cit. 2021-4-21]. Dostupné z: https://developer.amazon.com/en-US/docs/alexa/device-apis/smart-home-general-apis.html

[14] List of Alexa Interfaces and Supported Languages. *Amazon Alexa* [online]. [cit. 2021-4-21]. Dostupné z: https://developer.amazon.com/en-US/docs/alexa/device-apis/list-of-interfaces.html

[15] Zařízení, která podporují „Hey Siri“. *Apple* [online]. [cit. 2021-4-23]. Dostupné z: https://support.apple.com/cs-cz/HT209014

[16] Home accessories. The list keeps getting smarter. *Apple* [online]. [cit. 2021-4-23]. Dostupné z: https://www.apple.com/ios/home/accessories/

[17] IOS and iPadOS Feature Availability: Siri. *Apple* [online]. [cit. 2021-4-23]. Dostupné z: https://www.apple.com/ios/feature-availability/#siri

[18] What is Cortana? *Microsoft: Support* [online]. [cit. 2021-4-23]. Dostupné z: https://support.microsoft.com/en-us/topic/what-is-cortana-953e648d-5668-e017-1341-7f26f7d0f825

[19] Cortana's regions and languages. *Microsoft: Support* [online]. [cit. 2021-4-23]. Dostupné z: https://support.microsoft.com/en-us/topic/cortana-s-regions-and-languages-ad09a301-ce3a-aee4-6364-0f0f0c2ca888

[20] Upcoming changes to Cortana. *Microsoft: Support* [online]. [cit. 2021-4-26]. Dostupné z: https://support.microsoft.com/en-us/topic/upcoming-changes-to-cortana-2d04871e-f576-7080-58b4-7c37131c3baf

[21] Propojený život se SmartThings: Propojte zařízení a získejte ještě více se SmartThings. *Samsung* [online]. [cit. 2021-4-26]. Dostupné z: https://www.samsung.com/cz/apps/smartthings/

[22] Dostupné jazykové verze Bixby. *Samsung* [online]. [cit. 2021-4-26]. Dostupné z: https://www.samsung.com/cz/apps/bixby/

[23] Bixby Vision pro každého. *Samsung* [online]. [cit. 2021-4-26]. Dostupné z: https://www.samsung.com/cz/apps/bixby/vision/

[24] Meet the Mycroft Team: Where we’re going. *Mycroft AI* [online]. [cit. 2021-4-26]. Dostupné z: https://mycroft.ai/team/

[25] GESLING, Kris. Why use Mycroft AI?: Open source. *GitBook: Mycroft AI* [online]. [cit. 2021-4-26]. Dostupné z: https://mycroft-ai.gitbook.io/docs/about-mycroft-ai/why-use-mycroft

[26] Licence Apache. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2021-4-25]. Dostupné z: https://cs.wikipedia.org/wiki/Licence_Apache

[27] Languages: Mycroft Core - Lingua Franca library. *GitBook: Mycroft AI* [online]. [cit. 2021-4-25]. Dostupné z: https://mycroft-ai.gitbook.io/docs/using-mycroft-ai/customizations/languages

[28] WARREN, Jennifer. Audrey: The First Speech Recognition System. *Asta Speaks* [online]. Asta Publications, 13. 10. 2014 [cit. 2021-4-25]. Dostupné z: https://astaspeaks.wordpress.com/2014/10/13/audrey-the-first-speech-recognition-system/

[29] IBM Shoebox. *IBM: IBM Archives* [online]. [cit. 2021-4-25]. Dostupné z: https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html

[30] CHALUPNÍČEK, Kamil. *Rozpoznávání diktované řeči pro medicínské aplikace*. Brno, 2004. Diplomová práce. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ.

[31] Tatarinov, J. – Pollák, P.: Řečové detektory využívající ergodické Markovovské modely. In Technical Computing Prague 2008 [online]. Praha: Humusoft, 2008, díl1, s. 1–6. ISBN 978-80-7080-692-0, Dostupné z: http://dsp.vscht.cz/konference_matlab/mAtLAB08/prispevky/109_tatarinov.pdf

[32] DURČÁK, Pavel. Neuronové sítě a princip jejich fungování. *NaPočítači.cz* [online]. Verlag Dashöfer, 8.9.2017 [cit. 2021-4-29]. Dostupné z: https://www.napocitaci.cz/33/neuronove-site-a-princip-jejich-fungovani-uniqueidgOkE4NvrWuNY54vrLeM670eFNQh552VdDDulZX7UDBY/

[33] CMU Sphinx Downloads: Models. *CMUSphinx* [online]. [cit. 2021-4-29]. Dostupné z: https://cmusphinx.github.io/wiki/download/

[34] Training an acoustic model for CMUSphinx. *CMUSphinx* [online]. [cit. 2021-4-29]. Dostupné z: https://cmusphinx.github.io/wiki/tutorialam/

[35] Pricing. *Google Cloud: Cloud Speech-To-Text* [online]. [cit. 2021-5-12]. Dostupné z: https://cloud.google.com/speech-to-text/pricing

[36] CHLUMECKÁ, J. *Hodnocení soběstačnosti v ergoterapii*. Diagnóza, 3/2005. ISSN 1 801-134

[37] OPATŘILOVÁ, Dagmar a Dana ZÁMEČNÍKOVÁ. Podpora soběstačnosti a nezávislosti. *Podpora rozvoje hybnosti osob s tělesným postižením* [online]. [cit. 2021-5-3]. Dostupné z: https://is.muni.cz/do/rect/el/estud/pedf/ps14/hybnost/web/pages/04-02-sobestacnost.html

[38] SEDLÁK, Radek. ARTSPECT, A.S. *Půdorys* [pdf]. 28.3.2012. číslo dokumentace: 12-037, revize R00.

[39] *PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROTCP-1000, CP-1001, CP-1020* [online]. In: březen 2017, s. 5-14 [cit. 2021-5-3]. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00430_01_foxtrot_cp_1000

[40] TECOMAT FOXTROT PROGRAMOVATELNÉ AUTOMATY: ZÁKLADNÍ DOKUMENTACE K MODULU CP-1001. *Teco: Advanced Automation* [online]. říjen 2013 [cit. 2021-5-3]. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv11001_00_foxtrot_cp-1001_cz_en

[41] Mosaic - pro vývoj PLC programu dle standardu IEC 61131-3. *Teco: Advanced Automation* [online]. [cit. 2021-5-3]. Dostupné z: https://www.tecomat.cz/ke-stazeni/software/mosaic/

[42] Programování PLC podle normy IEC61131-3 v prostředí Mosaic. *Teco: Advanced Automation* [online]. listopad 2007 [cit. 2021-5-3]. Dostupné z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00321_01_mosaic_progiec_cz

[43] Funkce. *Umíme programovat* [online]. [cit. 2021-5-3]. Dostupné z: https://www.umimeprogramovat.cz/programovani-funkce

[44] Rychlá komunikace mezi automatizačními zařízeními a zařízeními pro procesní instrumentaci: MODBUS. *Wago* [online]. 1. září 2020 [cit. 2021-5-3]. Dostupné z: https://www.wago.com/cz/modbus

[45] Podrobný popis protokolu Modbus TCP s příklady příkazů. *IPC2U* [online]. [cit. 2021-5-3]. Dostupné z: https://ipc2u.cz/articles/simple-decisions/podrobny-popis-protokolu-modbus-tcp-s-priklady-prikazu/

[46] Pocketsphinx-android-demo. *GitHub* [online]. [cit. 2021-5-8]. Dostupné z: https://github.com/cmusphinx/pocketsphinx-android-demo

Příloha A: Obsah přiloženého CD

Abstrakt_CZ.pdf	Abstrakt v českém jazyce
Abstrakt_EN.pdf	Abstrakt v anglickém jazyce
Bakalarska_prace.pdf	Kompletní bakalářská práce
Klicova_slova.pdf	Klíčová slova v českém i anglickém jazyce
Mobilni_aplikace.zip	Zdrojové kódy mobilní aplikace
PLC_program.zip	Upravené zdrojové soubory PLC programu
Webova_aplikace.zip	Zdrojové kódy k webové aplikaci
Zadani.pdf	Naskenované zadání bakalářské práce