**Vyšší odborná škola**

**a Střední průmyslová škola elektrotechnická**

**Plzeň, Koterovská 85**

**Dlouhodobá maturitní práce s obhajobou**

Téma:**Vytvoření Chatbota s orientací FAQ ohledně SPŠE Plzeň**

**Autor práce: Ladislav BORČ**

**Třída: 4.F**

**Vedoucí práce: Ing. Marek LUKŠÍK**

**Dne: 30. 3. 2023**

**Hodnocení:**

**Vyšší odborná škola**

**a Střední průmyslová škola elektrotechnická**

**Plzeň, Koterovská 85**

**Zadání dlouhodobé maturitní práce**

**Žák: Ladislav BORČ, Vladimír DACEJ, Daniel MAI**

**Třída: 4.F**

**Studijní obor: 18-20-M/01 Informační technologie**

**Zaměření: Vývoj aplikací, Správa počítačových sítí**

**Školní rok: 2022 – 2023**

*Téma práce:*  ***Vytvoření Chatbota s orientací FAQ ohledně SPŠE Plzeň***

***Pokyny k obsahu a rozsahu práce:***

* *Provést rešerši v oblasti Chatbota a popsat jaké druhy Chatbota jsou dnes k dispozici.*
* *Provést rešerši existujících platforem, které lze využít při implementaci Chatbota.*
* *Implementovat FAQ Chatbota pro potřeby SPŠE, což znamená:*
  + *Vytvořit mapy oblastí zájmů koncových uživatelů a sestavit nejčastější dotazy. Chatbot je natrénován pouze na prostředí SPŠE a na identifikované dotazy.*
  + *Implementovat databázi, ze které bude Chatbot čerpat data.*
  + *Implementovat metodu Webscraping pro získávání dat na otázky přímo  
    ze stránek SPŠE.*
  + *Implementovat metodu strojového učení (machine learning - využití dat a algoritmů k napodobování způsobu, jakým se lidé učí a postupně zlepšuje přesnost Chatbota.)*
    - *Využít metodu zpracování přirozeného jazyka (NLP – Natural Language Processing).*
  + *Implementovat REST API, které propojí frontend Chatbota s backendem. REST API musí obsahovat i autorizační mechanismus pro zabezpečený přístup   
    ke službě Chatbota.*
  + *Navrhnout a implementovat frontend Chatbota, tedy dialogové okno, které bude sloužit pro komunikaci mezi Chatbotem a koncovým uživatelem:*
    - *Chatbot bude získávat informace z databáze. Při sestavování databázového dotazu Chatbot použije uložené údaje. Odpověď z databáze zpracuje a připraví pro tvorbu textové odpovědi pro uživatele.*
      * *Pokud se bude jednat o otázku, jejíž odpověď se v čase mění, bude aplikována metoda webscrapingu a cache mechanismu.*
    - *Chatbot umožní uživateli přeformulovat dotaz po neúspěšném pokusu rozeznat* *smysl dotazu.*
* *Provést testování Chatbota s ohledem na rychlost odezvy (otázka – odpověď) a porozumění Chatbota otázkám (zda je schopen správně odpovídat na položené otázky).*

**Určení částí tématu zpracovávaných jednotlivými žáky:**

* ***Ladislav BORČ***
* *Provést rešerši v oblasti Chatbota a popsat, jaké druhy Chatbota jsou dnes  
   k dispozici.*
* *Vytvořit mapy oblastí zájmů koncových uživatelů a sestavit nejčastější dotazy. Chatbot je natrénován pouze na prostředí SPŠE a na identifikované dotazy.*
* *Implementovat databázi, ze které bude Chatbot čerpat data.*
* *Chatbot bude získávat informace z databáze. Při sestavování databázového   
  dotazu Chatbot použije uložené údaje. Odpověď z databáze zpracuje a připraví pro tvorbu textové odpovědi pro uživatele.*
* ***Vladimír DACEJ***
* *Provést rešerši existujících platforem, které lze využít při implementaci Chatbota.*
* *Provést rešerši python knihoven.*
* *Implementovat metodu strojového učení (machine learning – využití dat  
   a algoritmů k napodobování způsobu, jakým se lidé učí a postupně zlepšuje  
   přesnost Chatbota.)*
* *Využít metodu zpracování přirozeného jazyka (NLP – Natural Language Processing)*
* ***Daniel MAI***
* *Navrhnout a implementovat frontend Chatbota, tedy dialogové okno, které bude sloužit pro komunikace mezi Chatbotem a koncovým uživatelem.*
* *Implementovat metodu Webscraping pro získávání dat na otázky přímo ze   
  stránek SPŠE Plzeň.*
* *Implementovat REST API, které propojí frontend Chatbota s backendem. REST API musí obsahovat i autorizační mechanismus pro zabezpečený přístup ke službě Chatbota.*

***Požadavek na počet vyhotovení maturitní práce:*** *2 výtisky*

*Termín odevzdání:* ***30. března 2023***

*Čas obhajoby:* ***15 minut***

Vedoucí práce: **Marek LUKŠÍK**

Projednáno v **katedře VTT** a schváleno ředitelkou školy.

V Plzni dne: 21. září 2022 Mgr. Jan Syřínek, v.r.

*ředitel školy*

# Anotace

Cílem tohoto projektu je vytvořit umělou inteligenci, která je schopna odpovídat na otázky týkající se SPŠE Plzeň. Uživatelé mohou komunikovat prostřednictvím chatovacího okna   
umístěného na webové stránce. Výsledek práce je složen ze čtyř částí:

1. Na začátek jsem potřeboval obsáhlý zdroj otázek týkajících se SPŠE Plzeň. Tyto otázky jsem získal pomocí internetových dotazníků. Dotazy jsem rozdělil na základě jejich   
   záměru.
2. Druhým krokem bylo umožnění konverzace mezi uživatelem s chatbotem. Za tímto účelem jsem vytvořil webové dialogové okno.
3. Poslední krok se zabýval trénováním klasifikačního modelu. Získaná data jsem rozdělil na trénovací sadu a testovací sadu pomocí aplikace a následně použil k trénovaní   
   modelu. Ten byl otestován pro získání dat o přesnosti klasifikace a využit v rozhraní API.

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil(a) literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

„Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.“

V Plzni dne: …..................... Podpis: …...............................

Obsah

[Anotace 8](#_Toc132289681)

[1 Tabulka pojmů 6](#_Toc132289682)

[2 Metodika vývoje 8](#_Toc132289683)

[3 Požadavky 9](#_Toc132289684)

[3.1 Rešerše chatbotů 9](#_Toc132289685)

[3.1.1 Typy chatbotů 9](#_Toc132289686)

[3.1.2 Chatbot framework 10](#_Toc132289687)

[3.1.3 Vyhodnocení rešerše chatbotů 12](#_Toc132289688)

[3.2 Datová extrakce 12](#_Toc132289689)

[3.2.1 Web scraping 13](#_Toc132289690)

[3.3 Datová klasifikace 15](#_Toc132289691)

[3.4 Java aplikace 17](#_Toc132289692)

[3.5 Rasa framework 19](#_Toc132289693)

[3.5.1 Instalace Rasa Open Source 19](#_Toc132289694)

[3.5.2 Zdrojové kódy a popis jednotlivých souborů 21](#_Toc132289695)

[3.5.3 Trénování a testování NLU modelu 24](#_Toc132289696)

[3.6 Front-end 25](#_Toc132289697)

[3.6.1 HTML 25](#_Toc132289698)

[3.6.2 Java Script 26](#_Toc132289699)

[3.6.3 CSS 27](#_Toc132289700)

[4 Architektura řešení 28](#_Toc132289701)

[4.1 Softwarová architektura 28](#_Toc132289702)

[4.1.1 Controller.java 29](#_Toc132289703)

[4.1.2 Ostatní třídy 29](#_Toc132289704)

[5 Závěr 30](#_Toc132289705)

[6 Literatura 31](#_Toc132289706)

[7 Seznam obrázků a tabulek 33](#_Toc132289707)

[7.1 Seznam obrázků 33](#_Toc132289708)

[7.2 Seznam tabulek 33](#_Toc132289709)

# Tabulka pojmů

|  |  |
| --- | --- |
| AI | Umělá inteligence označuje simulaci lidské inteligence na strojích. Ty jsou naprogramovány tak, aby přemýšleli a jednali jako lidé. |
| API | Je rozhraní pro přístup k určitým funkcím nebo službám. |
| Bot | Je počítačový program, který automaticky provádí určité úkoly nebo interaguje s uživateli. |
| Hands-Free | Technologie, které umožňují uživateli komunikovat bez použití rukou. |
| Chatbot | Je počítačový program, který simuluje rozhovor s člověkem. |
| Intent | Je cíl nebo účel, se kterým uživatel vstupuje do rozhovoru s chatbotem. |
| Kanbanboard | Je to vizuální nástroj, který slouží k sledování průběhu práce na projektu. Je rozdělený na sloupce, které reprezentují fáze nebo stavy úkolů. |
| NLP | Zpracování přirozeného jazyka, které umožní počítačům porozumět lidskému jazyku. |
| NLU | Porozumění přirozeného jazyka, které umožní počítačům rozpoznávat význam slov a celých vět. |
| NLU model | Je to výstupem trénování. Je to uložený strojový model, který umožňuje rozumět lidské řeči. |
| Python | Programovací jazyk, který nabízí rozsáhlou knihovnu funkcí. |
| REST API | Rozhraní pro komunikaci mezi počítačovými systémy. Skládá se z http požadavků jako jsou GET a POST. |
| Selektor | Určuje, na co se budou vztahovat vlastnosti CSS |
| Speech-To-Text | Nebo také automatické přepisování řeči je technologie, která umožňuje převádět řeč na text. |
| Spider | Knihovna Scrapy ho využívá k automatizaci prohlížení stránek a získávání dat. |
| Sprint | Časový úsek, během kterého tým pracuje na určitých úkolech. Cílem je dokončit určité množství práce. |
| Story | Termín pro sekvence vstupů a odpovědí, které chatbot poskytuje uživateli během rozhovoru. |
| Tag | Element, který definuje strukturu a vzhled. |
| XPATH | HTML cesta k elementu nebo obsahu. |
| Yaml | Serializační formát pro ukládání dat. |

Tabulka - Tabulka pojmů

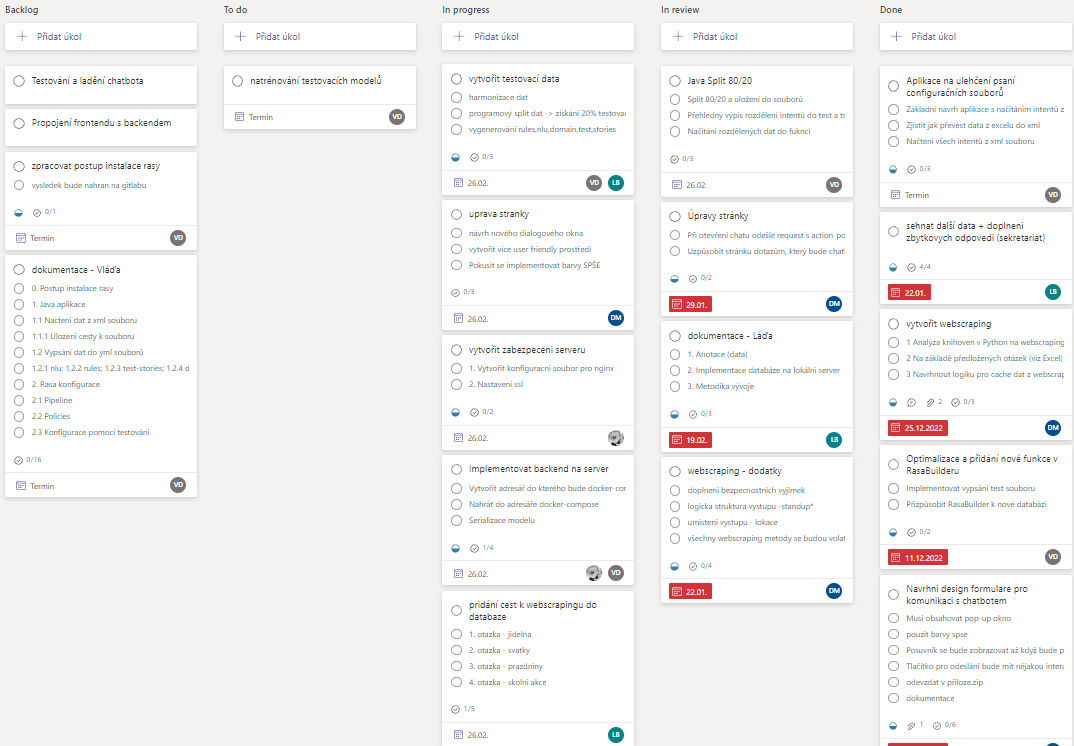
# Metodika vývoje

Vývoj probíhá metodou Scrumban, což je metodika kombinující metody scrumu a kanbanu. Práce je zadávána v menších částech a je sledována pomocí vizuální tabule. Práce je zadávána v podobě dvoutýdenních sprintů, což zajišťuje možnost měnit priority práce, která je potřeba udělat co nejdříve. [Scrumban]

Tým je složen ze čtyř členů. Před začátkem tvorby maturitní práce byla každému přiřazena práce na základě jeho preferencí. Během každého sprintu probíhají průběžné schůzky, během kterých musí každý člen týmu splnit tři kroky. Prvním krokem je uvést ostatní do problematiky, na které pracuje. Druhým krokem je uvedení dosavadního postupu, člen týmu ukazuje svoji práci a vysvětluje ji. Posledním krokem je uvedení problémů, které se při práci vyskytují. Tento postup se opakuje v průběhu každé práce alespoň jednou nebo dvakrát týdně. Pro   
přehlednost úkolů jsem využil vizuální tabuli, takzvaný kanbanboard. Ta disponuje pěti   
sekcemi:

* Backlog
* To Do
* In Progress
* In Review
* Done

Vytvořený úkol s rozepsanými pokyny je přidán do sekce Backlog. Pokud je úkol vyhodnocen jako prioritní, je přiřazen do sprintu daného člena a zařazen do sekce To Do. Jakmile začne člen týmu na práci pracovat, přesune si úkol do sekce In Progress. Hned poté co úkol   
vypracuje, je úkol přesunut ke zrevidování do sekce In Review. Zde dochází ke kontrole   
výstupu práce. Pokud něco není v pořádku, je navrácena danému členu k opravě. V případě, že je výstup adekvátní, je práce přesunuta do sekce Done.



Obrázek - Ukázka použité nástěnky (Zdroj: Autor)

# Požadavky

## Rešerše chatbotů

Cílem bylo vytvořit plně automatizovaný program na odpovídání dotazů na SPŠE Plzeň za   
pomoci dat získaných z internetových dotazníkůa znalostí programovacích jazyků Python, Java, JavaScript a MongoDB.

Chatbot (nebo také chatovací robot) označuje počítačový program, který je vytvořený   
za účelem automatizované komunikace. Odpovídá lidem na otázky na základě předem   
nadefinovaného scénáře, případně s přihlédnutím ke kontextu situace. Je nadefinován pro konkrétní účel. Rešerši druhů chatbotů jsme dělali proto, abychom byli schopni následně identifikovat typ bota, který by byl co možná nejvíce vhodný pro naší práci. [types of chatbots]

### Typy chatbotů

1. Chatbot na základě tlačítek je nejzákladnější typ chatbota aktuálně na trhu. Uživatel   
   si musí pouze vybrat z předem definovaných tlačítek k provedení určité akce. Není zde možné zadávat textový vstup, takže je možné, že se k požadovanému výsledku uživatel nedostane.
2. Chatbot na základě klíčových slov rozpoznává v textu konkrétní slova, aby mohl   
   dosáhnout požadovaného výsledku. S pomocí umělé inteligence a vlastním listem   
   klíčových slov, chatbot určí vhodnou reakci pro uživatele pomocí algoritmu. Tento chatbot začne selhávat, pokud se budou klíčová slova opakovat v několika souvisejících otázkách.
3. Chatbot na základě pravidel, označován také jako robot rozhodovacího stromu.   
   Je jazykově založen. Musí mít předem naprogramované otázky a odpovědi (jelikož není poháněn umělou inteligencí). Je vhodný pro menší čísla a jednoduché dotazy, jako je   
   rezervace stolu v restauraci nebo při požadavku o otevírací době. Nepotřebují rozsáhlé školení, které zrychluje a zkomplikuje proces implementace. Čím lépe bude   
   předdefinovaná struktura a odpovědi, tím snadněji bude schopen odpovídat a lépe se   
   chovat. Bohužel však nemůže zachytit překlepy, což znamená, že v některých případech nepochopí uživatele. Interakce s ním se zdá spíše robotická než konverzační. Nemůže se učit sám, takže kterékoliv zlepšení je třeba dělat ručně.
4. AI Chatbot je chatbot využívající zpracování přirozeného jazyka (NLP). Takže na rozdíl od chatbota založeného na pravidlech nebude k odpovědi používat klíčová slova, ale pokusí se pochopit záměr hosta. Čím více komunikuje s uživateli, tím lépe bude chápat záměr  
    a tím lépe bude schopen odpovídat na požadavky uživatelů. Rozumí překlepům   
   a gramatickým chybám, takže je stále schopen odpovědět na danou otázku. Neustále se zlepšuje bez další lidské pomoci. Rozhovor s ním je nejpřirozenější a nejvíce připomíná lidskou konverzaci.
5. Hybrid je spojení chatbota na základě pravidel s AI. Je poháněn umělou inteligencí   
   a zároveň používá strukturu založenou na pravidlech. To znamená, že uživateli položí   
   dodatečné otázky, ale pomocí umělé inteligence pochopí záměr uživatele. Implementace není tak dlouhá a komplikovaná jako u AI, protože používá předem definované struktury a odpovědi. Protože je však poháněn umělou inteligencí, tak se neustále zlepšuje,   
   aby pochopil záměr uživatele. Rozhovory s hybridním modelem tak jsou stále relativně konverzační a přirozené.
6. Hlasem aktivovaný chatbot je program, který umožňuje uživateli komunikovat   
   prostřednictvím hlasu. Je schopen přijmout příkaz ústní nebo písemnou formou.   
   Naprogramován je tak, aby odpověděl hlasem. To lze dále rozdělit do dvou typů. První typ je bot, který používá jenom hlas. Zvládá vše od jednoduchých úkolů, jako je nastavení   
   budíků až po složitější úkoly, jako je například proměna domu v chytrou domácnost. Druhý typ je hybridní model s podporou hlasu. Vstupním formulářem je řeč i text.

### Chatbot framework

Chatbot framework je software, který umožňuje vývojářům snadněji vytvářet a spravovat chatboty. Framework poskytuje vývojářům sadu nástrojů a knihoven, které jim pomáhají   
implementovat různé funkce chatbotů, jako je přijímání příkazů od uživatelů, generování   
odpovědí a integrace s dalšími službami. Chatbot framework také umožňuje vývojářům   
spravovat a monitorovat chování chatbotů, což jim pomáhá zlepšovat jejich funkčnost.

1. Microsoft Bot Framework je open-source nástroj, který obsahuje řadu šablon, které mohou pomoci při tvorbě bota. Tyto šablony sahají od základní služby sběru dat až po pokročilejší. LUIS NLU slouží k podpoře inteligentních konverzačních zážitků.   
   K efektivnímu nasazení vyžaduje zkušenosti s programováním. Platforma podporuje automatický překlad pro několik jazyků včetně češtiny. Obsahuje také REST API pro integraci s ostatními platformy. Dokáže pracovat jenom s intenty a entity.   
   Nepodporuje zpracování přirozeného jazyka.
2. Wit.ai je Facebookem vydaný open-source nástroj, který nabízí podporu více jak 130 jazyků. Pokročilé funkce hlasového rozhraní dělají z Wit.ai užitečný nástroj při   
   vytváření botů, kteří jsou specifičtí pro malá zařízení a domácí automatizaci, kde je hands-free prostředí považováno za žádoucí. Platforma umožňuje vývojářům nahrávat jejich vlastní data ve snaze zvýšení kvality NLP. Je používán především pro funkce   
   podpory zákazníků. Naopak se kvůli jeho open-source povaze nehodí k případům   
   použití, pokud se jedná o citlivé informace.
3. Rasa je open-source nástroj založený na Pythonu. Obsahuje zpracování přirozeného jazyka (NLP). Rasa nasazuje modely strojového učení nejen k rozpoznání entit   
   a intentů, ale take k určení toku konverzace. Díky tomu je vhodná k vytváření vysoce sofistikovaných konverzací, které se neustále vyvíjejí bez nutnosti ručního zásahu. Rasa NLU lze použít k vytváření textových I hlasových botů a integruje se většinou   
   z kanálů pro zasílání zpráv prostřednictvím REST API. Potřebuje pro svoji funkčnost hotový NLP model. Rasa podporuje již několik těchto nástrojů jako je například: spaCy, MitieNLP a NLTK. Také nabízí přidání svého vlastního NLP modelu.
4. DialogFlow, původně spuštěný v roce 2014 jako API.ai, je vývojový framework   
   od společnosti Google, který je nakonfigurován jako služba Google. To znamená, že se vývojáři mohou k platformě připojit prostřednictvím cloudových funkcí Google. Využívá algoritmus Speech-to-Text společnosti Google k přeměně hlasových příkazů na čitelná data. Postrádá ovšem některé důležité funkce pro NLP.
5. IBM Watson Assistant je přední vývojový framework pro chatboty. Je postaven na   
   neuronové síti, která využívá framework k porozumění a určení konverzačních   
   podnětů. Používá tři hlavní komponenty a těmi jsou entity, intenty a dialogy. Využíván je zejména většími firmy, které si ho mohou dovolit. Je využíván zejména ve finančním a zdravotním sektoru.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **MS bot framework** | **Wit.ai** | **Rasa** | **DialogFlow** | **IBM Watson** |
| **Multiplatformní** | + | + | + | + | + |
| **Strojové učení** | + | + | + | + | + |
| **Hosting** | Cloud | Cloud | Cloud, Lokálně | Cloud | Cloud |
| **Programovací jazyk** | C#, Java, JS, Python | Python, Ruby | Python | Typescript | C++, Java |
| **Vlastní akce** | + | + | + | + | + |
| **Podpora českého jazyka** | + | + | + | + | 0 |
| **Zpracování českého jazyka (NLP)** | 0 | + | + | 0 | 0 |

Tabulka - Přehled výsledků porovnání frameworků pro tvorbu Chatbota

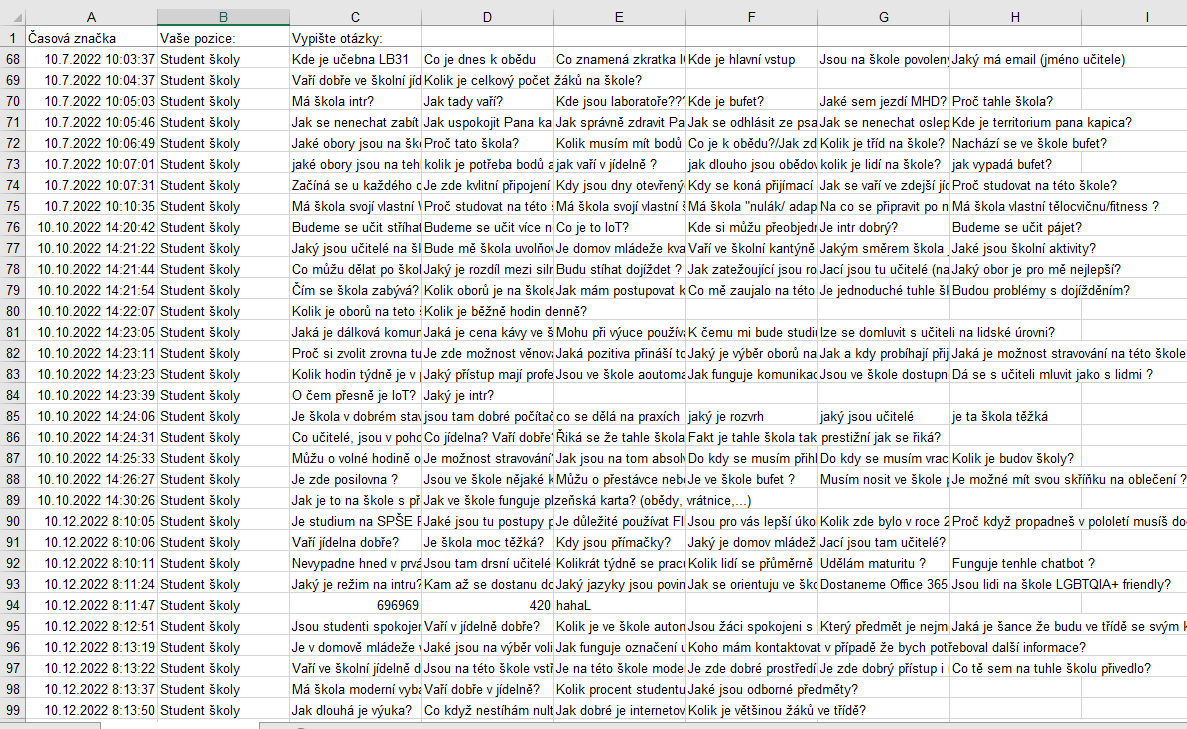
### Vyhodnocení rešerše chatbotů

Na základě předchozí rešerše jsem si vybral Rasu. Jeden z hlavních požadavků, který jsem měl je možnost lokálně spustit server a pracovat bez nutnosti cloudového hostingu. Dále oproti ostatním nabízí načtení jakéhokoliv vlastního NLP modelu včetně spaCy. Na rozdíl   
od ostatních předem zmiňovaných frameworků nenutí uživatele k používání předem   
definovaných funkcí a umožnuje tvorbu vlastních.

## Datová extrakce

Abych zajistil, že náš chatbot bude poskytovat odpovědi na otázky, musel jsem takové otázky získat. Pro ukládání těchto otázek jsem využil Microsoft Excel a začal s tvorbou databáze.

Data jsem získal přes dotazníky, které byly zaslány širokému okruhu lidí. Každé z otázek, které jsem získal, jsem vytvořil její alternativy v českém jazyce.



Obrázek - Ukázka získaných dat (Zdroj: Autor)

### Web scraping

Web scraping se využívá k automatizaci procesu získávání informací z internetu, například pro analýzu trhu nebo pro vytvoření vlastní databáze. Web scraping může být proveden pomocí programovacích jazyků, jako jsou Python, Java nebo R. Pro nás bude web scraping zařizovat aktualizaci odpovědí na otázky, které to vyžadují. Těmi mohou být například termíny   
přijímacích zkoušek nebo jídelníček školní jídelny.

#### Rešerše knihoven

Následoval výběr knihovny, která mi proces umožní. V Pythonu se "knihovna" označuje jako sada modulů nebo souborů kódu, které poskytují předem definované funkce, třídy   
a proměnné. Pro naše mé jsem vybíral ze dvou nejpoužívanějších knihoven v tomto odvětví “Beautiful soup” a “Scrapy”.

1. Beautiful Soup je velmi jednoduchá na použití a nabízí mnoho metod pro vyhledávání a filtrování dat. Beautiful Soup se používá k parsování HTML a XML dokumentů   
   a následnému vyhledávání specifických elementů nebo atributů. Knihovna poskytuje metody pro vyhledávání podle tagu, atributu nebo CSS třídy. Také nabízí možnost   
   vyhledávání elementů pomocí regulárních výrazů.
2. Je to knihovna pro komplexní web scraping. Scrapy používá "spider" k automatizaci prohlížení stránek a získávání dat. Scrapy také poskytuje možnosti pro řízení proxy   
   serverů a ochranu proti blokování, takže může pracovat s webovými stránkami, které mají ochranu proti automatizovanému scrapingu.
3. Uzavření rešerše knihoven. Zvolenou knihovnou se nakonec stala Scrapy.   
   Po otestování obou knihoven na demo scrapovacím kódu jsem dospěl k závěru,   
   že Scrapy je vyhovující knihovna. Hlavní důvod byl právě spider, který zvládne data automaticky posbírat.

#### Lokalizace dat

Základem bylo najít na internetu data která je třeba posbírat. Pro mě to znamenalo najít různé webové stránky s následujícím obsahem: prázdniny SŠ, den otevřených dveří na SPŠE, termíny přijímacích zkoušek a jídelníček školní jídelny. Hlavním cílem bylo najít web s kvalitní   
strukturou, který dostává konzistentní aktualizace.

|  |  |  |  |
| --- | --- | --- | --- |
| **Scrapovaná data** | **Provozovatel webu** | **URL** | **Popis** |
| Prázdniny středních škol | Ministerstvo školství, mládeže a tělovýchovy. | https://www.msmt.cz/vzdelavani/organizace-skolniho-roku-2022-2023-v-zs-ss-zus-a | Web s kvalitní strukturou a konzistentní aktualizací, která se děje nikoli na webu ale v URL. (př. “2022-2023” se mění na “2023-2024”) |
| Termíny přijímacích zkoušek | Cermat | https://prijimacky.cermat.cz/aktuality/aktualita/336-terminy-jpz-2022 | Web s kvalitní strukturou a konzistentní aktualizací pouze v termínech. |
| Den otevřených dveří | SPŠE Plzeň | https://www.spseplzen.cz/dny-otevrenych-dveri/ | Web s použitelnou strukturou, ale nejistotou v konzistenci aktualizací. Jediná možná volba pro scrapování těchto dat. |
| Jídelníček školní jídelny | Strava SPŠE Plzeň | https://strava.spseplzen.cz/ | Velmi složitá a nekonzistentní struktura webu. Konzistentní, každodenní, aktualizace. Jediná možná volba pro scrapování těchto dat. |

Tabulka - Výběr scrapovaných stránek

#### Web scraping script

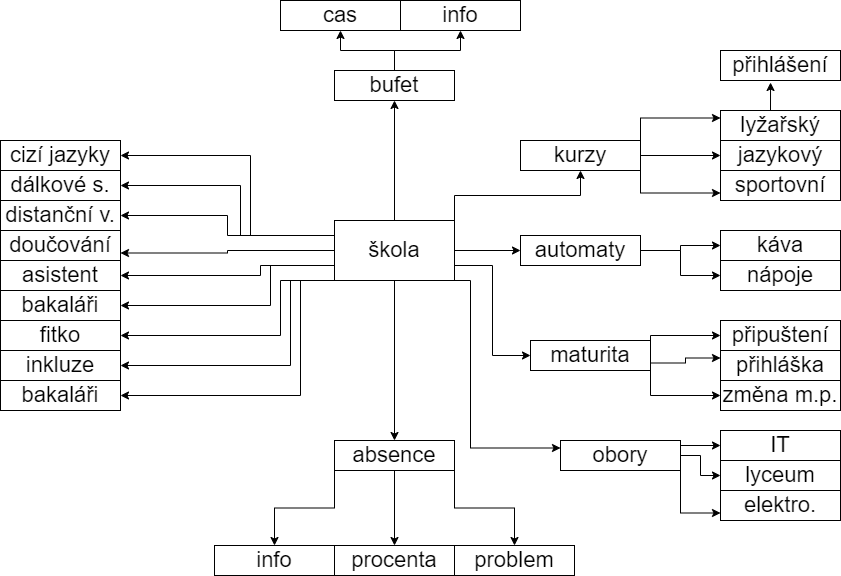
Další krok je napsat samotný script.

1. Za prvé jsem nainstaloval Scrapy do virtuálního prostředí a založil jsem nový projekt   
   a spidera.
2. Za druhé jsem definoval URL scrapované stránky a pomocí jednoduché funkce   
   a XPATH[[1]](#footnote-2) jsem určil která data se ze stránky budou stahovat. Otestoval jsem script pomocí terminálu.
3. Za třetí jsem script obohatil o správné formátování výstupových dat a o výjimky.   
   Například v případě, že se změní struktura stránky. Podobný postup jsem zopakoval pro každého spidera.
4. Za čtvrté jsem vytvořil script který spustí všechny spidery najednou, abych napomohl automatizaci scrapování dat.

## Datová klasifikace

Předtím než jsem mohl začít s tvorbou databáze, musel jsem definovat oblasti zájmů. Oblasti zájmů jsou otázky, na které je chatbot schopen odpovídat. Jsou vytvořeny ručně a každá   
oblast zájmů reprezentuje jednotlivý intent. Otázky získané od respondentů jsem kvůli   
rozsáhlosti rozdělil do čtyř skupin: škola, internát, jídelna a ostatní. Po rozdělení do těchto bloků jsem byl schopen jednotlivé otázky analyzovat a zjistit tak, jakou odpověď očekávají. Vždy když byla získaná otázka relevantní, vytvořil jsem jí jedinečný intent, podle kterého   
chatbot pozná, jak má dále reagovat. Na závěr jsem kvůli velkému množství dat několikrát zkontroloval určené intenty, jestli neexistují dva nebo více různých se stejným významem. V případě že se intentů se stejným významem vyskytovalo více, spojil jsem dané intenty   
do jednoho.

Zde je část vizualizace jednotlivých intentů ze sekce škola:



Obrázek - Ukázka vizualizace rozdělení intentů (Zdroj: Autor)

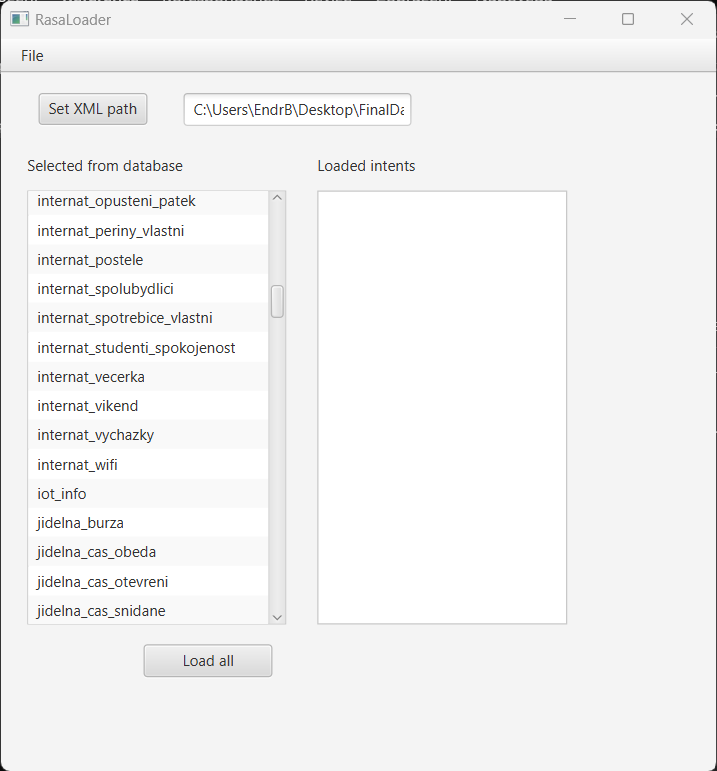
Získaná data jsem přidal do databáze a tu přizpůsobil tak, aby vyhovovala potřebám mojí aplikace. V následujících bodech jsou popsány jednotlivé sloupce.

1. První sloupec se nazývá text. Zde jsou všechny otázky zaznamenány a rozřazeny do jednotlivých skupin podle jejich významu.
2. Druhý sloupec se nazývá intent. Je to jedinečný význam otázky. Díky funkcím rasa frameworku jsem mohl ke každému intentu přiřadit unikátní akce.[[2]](#footnote-3) Akce se vykonají, pokud chatbot vyhodnotí, že byl dotázán na konkrétní intent. Všechny intenty jsou přidány ručně na základě vyhodnocení významu otázek, které jsem získal. Každý   
   využitý intent obsahuje skupinu deseti až čtyřiceti jinak zformulovaných otázek,   
   na které ale uživatel očekává stejnou odpověď.
3. Třetí sloupec se nazývá is\_action. Tento sloupec slouží k definování, zdali je   
   odpověď na otázku jen text. V opačném případě může odpověď obsahovat odkaz na webovou stránku, obrázek nebo aktivaci web scrapingu.
4. Čtvrtý sloupec se nazývá url. Uchovává v sobě adresu webové stránky. Odkazy na webové stránky posílá chatbot u intentů, na které není možné odpovědět dostatečně detailně.
5. Pátý sloupec se nazývá url\_text. Nachází se v něm text, který odkazuje na určitou webovou stránku.
6. Šestý sloupec se nazývá img\_src. Zde je uložena cesta k obrázku, který je uložen ve složce mojí webové stránky.
7. Sedmý sloupec se nazývá response\_text. Tento sloupec uchovává textové   
   odpovědi na všechny intenty. Během tvorby odpovědí jsem využíval školních stránek, školního řádu, domovního řádu domova mládeže a internetu. Ke tvorbě odpovědí na otázky, které nejsou veřejně přístupné, jsem využil služeb sekretariátu SPŠE Plzeň.   
   Během určování odpovědi jsem také definoval, je-li odpověď statická či nikoliv. Pokud se odpověď měnila v závislosti na čase, bylo nutné použít místo odpovědi web   
   scrapingový script.

## Java aplikace

Rasa framework pracuje s datovými formáty yml. Vytvořil jsem aplikaci, která je schopna   
převést data do požadovaného formátu. K určení přesnosti chatbota jsem data rozdělil na dvě sady, trénovací a testovací v poměru 80 ku 20.

Pro vývoj desktopové aplikace jsem zvolil JavaFX framework[[3]](#footnote-4). Nejprve jsem navrhnul vzhled a funkce v SceneBuilder[[4]](#footnote-5). Ten vytvoří FXML soubor, ve kterém propojím controller.[[5]](#footnote-6) Pomocí tlačítka „Set XML path“ uživatel vybere xml soubor s daty tlačítkem „Load all“ spustí celý   
proces zpracování dat.



Obrázek - Ukázka vzhledu aplikace (Zdroj: Autor)

V controlleru první metodou je getUserFile ta načte vybraný soubor.

Druhá metoda se nazývá LoadFromXML ta má za úkol načíst data z XML souboru. K načtení využívá metodu getPath k získání cesty k souboru z uživatelského rozhraní.

Další důležitou metodou je LoadFromSortedXML, která podle logického vstupu vrací   
testovací, nebo trénovací data.

Metoda DataSplit rozdělí data na trénovací a testovací a následně je uloží zvlášť do dvou soborů. Také vytvoří soubor s vizuálním rozdělením dat.

Obsah obrázku text

Popis byl vytvořen automaticky

Obrázek - Ukázka vizualizace rozdělení dat (Zdroj: Autor)

Dalším krokem bylo vytvořit třídy pro vytvoření potřebných souborů k trénování. K tomu jsem využil třídy Rules.java, nlu.java, test\_stories.java a Domain.java, která načítají data pomoci metody v controlleru LoadFromSortedXML.

## Rasa framework

Rasa Open Source poskytuje stavební bloky pro vytváření virtuálních asistentů nebo chatbotů. Nabízí robustní nástroje pro trénování přirozeného jazyka, rozpoznávání entit a záměrů,   
a správu dialogu. Po natrénování vznikne model chatbota, který můžeme otestovat.

### Instalace Rasa Open Source

K nainstalování všech potřebných balíčků jsem využil program Anaconda. Ten slouží   
ke zjednodušení správy a instalaci potřebných balíčků. V Anaconda Prompt následujícími   
příkazy vytvoříme prostředí s balíčky pro Rasa framework:

1. conda create -name „název\_prostředí“ python==3.8.0
2. conda activate „název\_prostředí“
3. conda install ujson
4. conda install tensorflow
5. pip install rasa
6. rasa init

Prvním příkazem vytvořím prostředí, kde budou naše balíčky nainstalovány. Dalším krokem aktivujeme prostředí a následně nainstaluji knihovny ujson a tensorflow. Poté příkazem pip install rasa nainstaluji veškeré potřebné knihovny pro Rasa framework. Posledním příkazem vytvoříme nový projekt s předem připravenými soubory pro snadnější používání.

|  |  |
| --- | --- |
| **Soubor** | **Popis** |
| actions/actions.py | Obsahuje vlastní metody a příkazy,  které může chatbot spustit. |
| config.yml | Obsahuje nastavení konfigurace a definuje, jak bude chatbot fungovat. |
| credentials.yml | Nastavení připojení k webu nebo aplikacím. |
| data/nlu.yml | Obsahuje strukturované příklady otázek s informacemi, podle kterých se trénuje NLU model. |
| data/rules.yml | Typ trénovacích dat pro naučení  jednoznačných odpovědí. |
| data/storeis.yml | Typ trénovacích dat pro naučení toku  dialogu. |
| domain.yml | Obsahuje odpovědi a specifikace intentů  a entit. |
| endpoints.yml | Nastavení koncových bodů[[6]](#footnote-7). |
| models | Zde se ukládají natrénované modely. |
| results | Zde se ukládají provedené testy. |
| tests/test\_stories.yml | Zde jsou zapsány možné toky konverzací, kterými je chatbot testován. |

Tabulka – Popis souborů rasa projektu

### Zdrojové kódy a popis jednotlivých souborů

#### Config.yml

V prvním kroku je důležité správně nastavit pipeline a policies. Pipeline slouží k určení   
posloupnosti komponent, které se používají pro zpracování vstupních dat konverzace.

|  |  |
| --- | --- |
| **Komponenta** | **Popis** |
| WhitespaceTokenizer | Rozdělí text na slova podle mezer. |
| LexicalSyntacticFeaturizer | Předpracování textu. Napomáhá k lepšímu porozumění textu. |
| CountVectorsFeaturizer | Transformuje informace do vektorové  reprezentace. |
| DIETClassifier | Slouží ke klasifikaci intentů a entit. |
| ResponseSelector | Vytváří vektorové reprezentace odpovědí  a porovnává je s vektorovou reprezentací vstupního dotazu. |

Tabulka – Pipeline komponenty

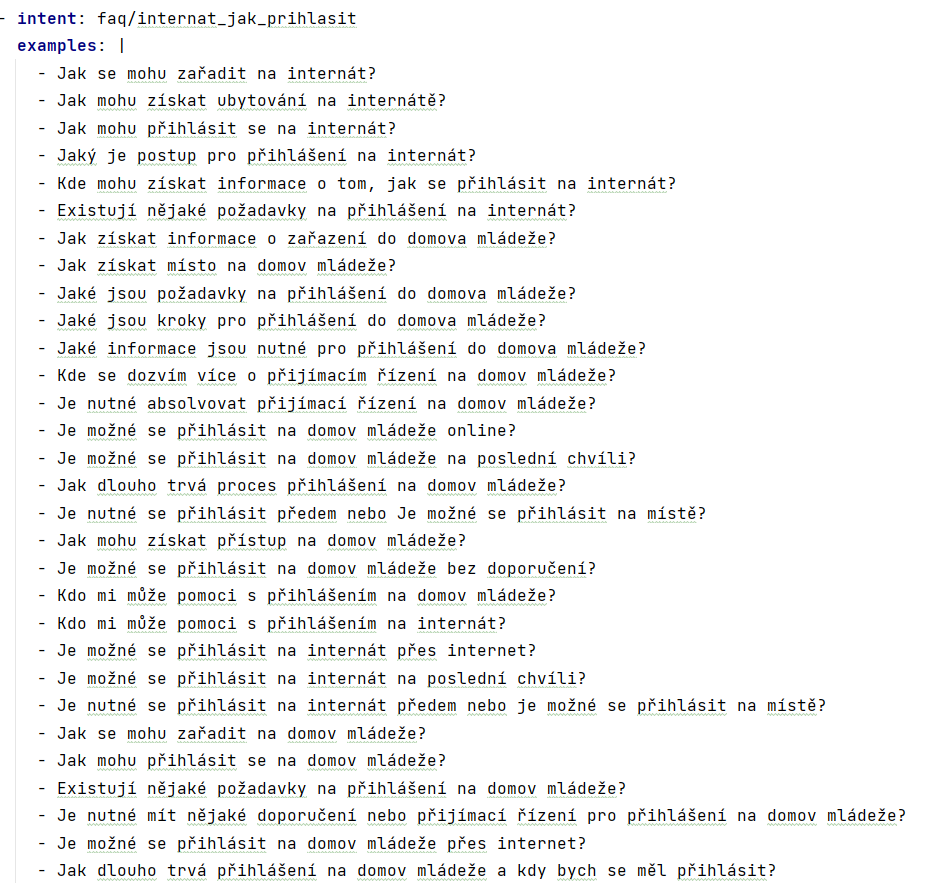
Dále musíme správně nastavit policies. Policies určují, jakým způsobem bude model určovat následující akci. Také se používají k vyhodnocování dialogového stavu a zjišťování, zda byl   
splněn určitý cíl.

|  |  |
| --- | --- |
| **Komponenta** | **Popis** |
| MemoizationPolicy | Ukládá historii konverzace a využívá jí  k vyhodnocení další akce. |
| RulePolicy | Umožňuje vytvářet pravidla odpovědí na základě uživatelského dotazu. |
| UnexpecTEDIntentPolicy | Používá se pro detekci neočekávaných nebo neznámých záměrů uživatele. |
| TEDPolicy | Používá strojové učení a neural network  k výběru nejlepší akce na základě aktuálního stavu konverzace a historie. |

Tabulka – Policies komponenty

#### nlu.yml

Je potřeba vypsat intenty s co nejvíce příklady otázek, které se můžou vyskytnout. Čím více příkladů dostane tím lépe bude rozumět jednotlivým intentům, proto jsem určil jako   
minimalní počet na jeden intent 10. Všechny intenty jsou podmnožinou intentu faq. To mi umožní vybírat všechny intenty najednou a v dalším kroku toho využít.



Obrázek – Ukázka intentu internat\_jaky\_je (Zdroj: Autor)

#### stories.yml

Zde připravím scénaře, které popisují krok za krokem průběhu konverzace. Vytvořil jsem smyčku pomocí checkpoint. Ta napomáhá na základě uživatelského vstupu určit jaký   
scénář bude následovat.

Obsah obrázku text

Popis byl vytvořen automaticky

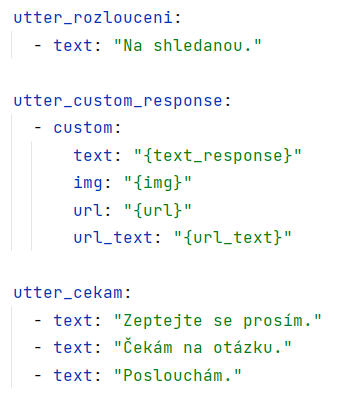
Obrázek – Ukázka stories (Zdroj: Autor)

#### rules.yml

Pravidla slouží k vyvolání určité akce na základě intentu bez přerušení scénáře.   
Například při dotazu, u kterého si není dostatečně jistý, se dotáže o opakování dotazu.

#### domain.yml

Zde se nachází veškeré informace o entitách a intentech. Dále obsahuje definice akcí, které může vykonat v odpověď na uživatelský dotaz. Ty můžou obsahovat textovou odpověď. Zde jsem vytvořil předlohu(template) odpovědi pro naší akci. Do té se vyplní informace v actions.py.



Obrázek – Ukázka akcí a předlohy odpovědi (Zdroj: Autor)

#### actions.py

Slouží pro vlastní akce, které se mají vykonat. Můžeme se zde pomocí jazyka python připojit k databázi. Zde si vytvořím akci action\_utter\_faq, kterou bude používat k vyhledání   
odpovědi v databázi. První problém byl získat určený intent. Třída Tracker ukládá   
informace o posledním dotazu. Pomocí instance latest\_message a metody get jsem   
získal určený intent. Metodou split rozdělíme určený intent od množiny faq.

full\_intent = (  
 tracker.latest\_message.get("response\_selector", {})  
 .get("faq", {})  
 .get("response", {})  
 .get("intent\_response\_key")  
)  
if full\_intent:  
 topic = full\_intent.split("/")[1]  
else:  
 topic = None

Dále se pomocí třídy vytvořené třídy DBconnect vytvoří objekt, který naváže spojení   
s databází. Třída obsahuje metodu vyhledat\_intent, která vyhledá v databázi intent a vrátí   
potřebné informace, jako jsou například text otázky, intent a další. Poté pošle zprávu pomocí vytvořeného template v json formátu zpět k uživateli.

### Trénování a testování NLU modelu

Příkazem rasa train nlu se natrénuje NLU model pomocí pipeline konfigurace.   
Následně příkazem rasa shell nlu spustíme testovací relaci v příkazovém řádku.   
Zde můžeme napsat zprávu a odpověď se nám vypíše a „CTRL+C“ jí vypneme.   
Pomocí příkazu rasa test otestujeme nejnovější natrénovaný model a výsledky se uloží do složky results.

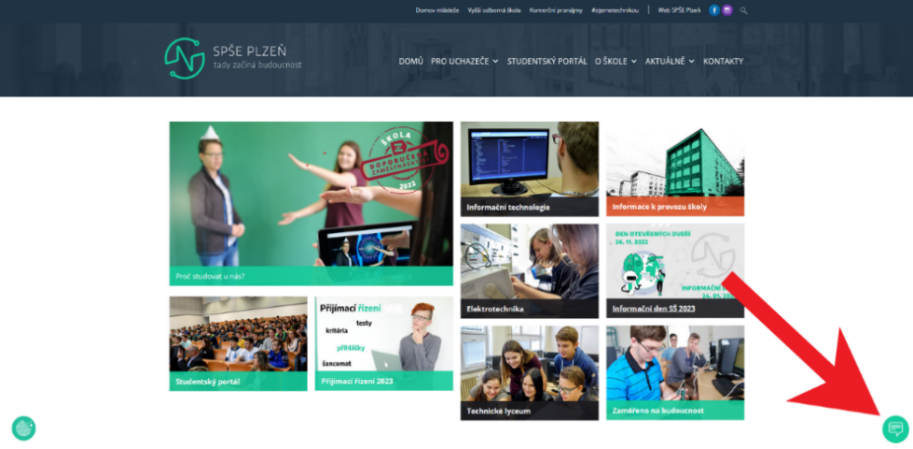
## Front-end

Front-end je design na webové stránce, který je propojen s programovou částí chatbota.   
Design stránky, který uživatel uvidí bude vyskakovací dialogové okno. Okno bude umožňovat konverzaci mezi uživatelem a Chatbotem.

### HTML

Neboli Hypertext Markup Language je značkovací jazyk. Právě HTML je jedním z hlavním   
programovacích jazyků pro vytváření webových stránek. Slouží k formátování textů a objektů, jako jsou třeba obrázky, které se ve webových prohlížečích zobrazují. Pro zobrazení HTML slouží webové prohlížeče.

Jako první bylo třeba napsat základní HTML soubor (index). Stránka je z počátku prázdná   
a obsahuje pouze tlačítko a příslušnou logiku pro otevření okna chatu.



Obrázek 9 – Chat widget[[7]](#footnote-8) (Zdroj: Autor)

V okně chatu se nachází vstupní pole, kam může uživatel psát své dotazy a tlačítko k odeslání.

|  |  |
| --- | --- |
|  |  |
| Obrázek 10 – Chat zavřený (Zdroj: Autor) | Obrázek 11 – Chat otevřený (Zdroj: Autor) |

### Java Script

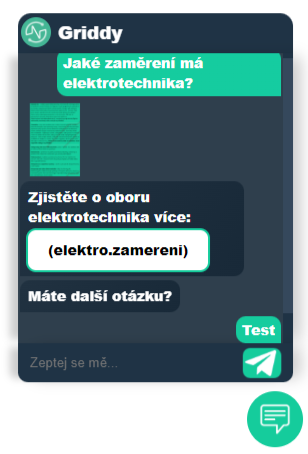
JavaScript je skriptovací jazyk, který se používá pro vývoj webových aplikací. Je to jeden   
z nejběžnějších jazyků používaných na webu a je implementován ve většině webových   
prohlížečů. Díky své jednoduchosti a flexibilitě je JavaScript využíván k vytváření interaktivních a dynamických webových stránek, které se mohou měnit v závislosti na akcích uživatele nebo na dalších podmínkách.

Tento soubor bude provádět veškerou komunikaci s koncovým bodem chatbota. Nejdříve bylo třeba definovat kdy bude JS aktivní. V případě, že je chatovací okno zavřené, tak je vstupní pole a tlačítko “odeslat” vypnuté a zapíná se při otevření chatbota. Přidal jsem   
posluchač události, který čeká na stisk klávesy enter nebo tlačítka “odeslat”. Následně odešle obsah vstupního pole na koncový bod chatbota a zároveň ho vypíše do chatového okna.   
Server se obratem vrátí s odpovědí v podobě textu, obrázku, odkazu anebo kombinací těchto možností. Odpověď je následně vypsána do chatovacího okna.

### CSS

Neboli Cascading Style Sheets (kaskádové styly), přiděluje tagům webových stránek jejich vlastnosti. Určuje se pomocí něj například styl písma, barvy, rozložení a další. Pomocí CSS   
se dá stránka přizpůsobit pro různé platformy jako třeba PC nebo mobilní zařízení, které mají různé rozměry zobrazení. Stejně jako HTML, tak i CSS je vyvinutý stejnou organizací   
a na základě toho je podporován všemi prohlížeči na všech platformách.

Jako poslední z trojice souborů je CSS. Tento soubor je určen k definování vzhledu chatbota. Vybral jsem základní barvy, které reprezentují SPŠE Plzeň a definoval je pomocí proměnných. Následně to byl jeden element za druhým, definování vzhledu hlavičky, těla, patičky a stínů. Jako další jsem definoval vzhled zpráv od uživatele a chatbota, což zahrnovalo odkazy   
a obrázky. také vlastní scrollovací lištu.



Obrázek 12 – Chatovací okno (Zdroj: Autor)

# Architektura řešení

## Softwarová architektura

Softwarová architektura je struktura komponent programu/systému, jejich vzájemné vazby, principy a předpisy určující jejich návrh a vývoj v průběhu času. Funguje jako plán sytému   
a vývojového projektu a stanoví úkoly, které je nutné provést konstrukčními týmy.

Obsah obrázku diagram

Popis byl vytvořen automaticky

Obrázek – Ukázka architektury aplikace (Zdroj: Autor)

|  |  |
| --- | --- |
| **Třída** | **Popis** |
| Controller.java | Celá logika aplikace. Načítá soubor s daty a rozdělí je na trénovací a testovací data a uloží je. |
| Domain.java | Vytvoří připravený soubor domain.yml  pro Rasa framework. |
| Main.java | Spouští aplikaci. |
| nlu.java | Vytvoří připravený soubor nlu.yml  pro Rasa framework. |
| rules.java | Vytvoří připravený soubor rules.yml  pro Rasa framework. |
| test\_stories.java | Vytvoří připravený soubor test\_stories.yml  pro Rasa framework. |

Tabulka – Popis tříd java aplikace

### Controller.java

|  |  |
| --- | --- |
| **Metoda** | **Popis** |
| Initialize(URL, resourceBundle) | Při spuštění aplikace vyvolá metody  pro načtení dat. |
| WriteData(Nodelist,ListView<String>) | Vypíše načtené intenty do tabulky. |
| LoadFromXML() | Načte data z xml souboru. |
| getPath() | Načte cestu k xml souboru z json souboru. |
| LoadFromSortedXML(boolean) | Vrací NodeList na základě potřeby s daty na trénování, nebo na testování. |
| SetPath() | Zkontroluje, jestli soubor není prázdný  a vypíše ho. |
| savePath(File) | Uloží cestu k souboru s daty do souboru json. |
| getUserFile() | Vyskakovací okno, ve kterém se vybere xml soubor. |
| LoadAll() | Akce, kterou vykoná tlačítko load all.  Vyvolá instance tříd pro výpis xml souboru. |
| DataSplit() | Rozdělí data v poměru 80:20 do dvou soborů a vizualizuje rozdělení do souboru. |
| writeValues(XMLStreamWriter,String[]) | Vypíše hodnoty do souboru. Vstupem jsou hodnoty k zapsání. |

Tabulka - Popis metod v Controller.java

### Ostatní třídy

Ostatní třídy jsou totožné, jen vypisují yml soubory s jinými daty.

|  |  |
| --- | --- |
| **Třída/metoda** | **Výstup** |
| nlu/write\_nlu() | nlu.yml |
| rules/write\_rules() | rules.yml |
| domain/write\_domain() | domain.yml |
| test\_stories/write\_test\_stories() | test\_stories.yml |

Tabulka - Popis tříd pro vypsání yml souborů.

# Závěr

Závěr dokumentace ukazuje, že získání velkého množství dat, ať už za pomocí webscrapingu, dotazníkových šetření nebo tvorby vlastních otázek je klíčový krok pro úspěšné strojové učení, které chatbot využívá. Vytvoření webové aplikace, která umožňuje odesílat a přijímat zprávy, ukazuje, jak lze využít strojové učení k interakci s uživateli. Správně nakonfigurovaný projekt pro trénování strojového učení je ovšem také nezbytný pro dosažení spolehlivých výsledků. Tyto klíčové body ukazují, že úspěšné použití strojového učení závisí na kvalitním a dobře   
připraveném datovém zdroji, spolehlivých algoritmech a správně nakonfigurovaném   
projektu. Celkově lze říct, že tyto kroky jsou nezbytné pro úspěšné nasazení chatbota,   
který využívá principů strojového učení. Naše práce na těchto krocích nám umožnila vytvořit úspěšný NLU model, který lze dále modifikovat bez nutnosti hlubšího porozumění.

Chtěl bych také poděkovat vedoucímu práce za jeho ochotu a pomoc, kterou nám během celého projektu poskytl.

# Literatura

Rasa. *Rasa docs* [online]. 28. 10. 2022 [cit. 2022-11-01]. Dostupné z: <https://rasa.com/docs/rasa/>

Artificial Intelligence: What It Is and How It Is Used [online]. 2022 [cit. 2023-03-29]. Dostupné z: https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp

Crumban — The unwanted Child [online]. 2021 [cit. 2023-03-29]. Dostupné z: https://medium.com/agileinsider/scrumban-the-unwanted-child-272b18482c0b

Types of chatbots - Which is best for your business?.  [online]. 2021 [cit. 2022-11-01]. Dostupné z: <https://www.engati.com/blog/types-of-chatbots-and-their-applications>

Chatbot Tech Trends in 2019. [online]. 2019 [cit. 2022-11-01]. Dostupné z: <https://techaffinity.com/blog/best-chatbot-development-frameworks/>

25 Chatbot Platforms: A Comparative Table. [online]. 2020 [cit. 2022-11-01]. Dostupné z: <https://chatbotsjournal.com/25-chatbot-platforms-a-comparative-table-aeefc932eaff>

Top Chatbot Development Frameworks You Should Know. [online]. 2020 [cit. 2022-11-01]. Dostupné z: <https://thirdeyedata.io/top-chatbot-development-frameworks-you-should-know-exploring-open-source-solutions-part-2/>

10 Types of Chatbots and How to Choose the Best One for Your Business. [online]. 2022 [cit. 2022-11-01]. Dostupné z: [https://helpcrunch.com/blog/types-of-chatbots/#Keyword\_recognition-based\_chatbots](https://helpcrunch.com/blog/types-of-chatbots/%23Keyword_recognition-based_chatbots)

9+ Best Open Source Chatbot Frameworks Compared. [online]. 2022 [cit. 2022-11-01]. Dostupné z: <https://www.tidio.com/blog/chatbot-framework/>

The begginer's guide to scrumban [online]. [cit. 2023-01-27]. Dostupné z: <https://monday.com/blog/rnd/the-beginners-guide-to-scrumban/>

What is natural language processing (NLP)? [online]. 2022 [cit. 2023-03-29]. Dostupné z: <https://www.ibm.com/topics/natural-language-processing>

Vodafone [online]. 2023 [cit. 2023-03-29]. Dostupné z: https://www.vodafone.cz/uzitecne-odkazy/

What is natural language understanding? [online]. 2022 [cit. 2023-03-29]. Dostupné z: <https://www.qualtrics.com/experience-management/natural-language-understanding/>

Scrapy dokumentace [online]. 2023 [cit. 2023-03-31]. Dostupné z: <https://docs.scrapy.org/en/latest/>

Beautiful Soup dokumentace [online]. [cit. 2023-03-31]. Dostupné z: <https://beautiful-soup-4.readthedocs.io/en/latest/>

W3schools [online]. [cit. 2023-03-27]. Dostupné z: <https://www.w3schools.com>

Installing Rasa Open Source 3.x: Windows 10 (64 bit) [online]. 2021 [cit. 2023-04-12]. Dostupné z: <https://www.youtube.com/watch?v=RVoFqxmG8p0&t=209s>

TutorialsPoint [online]. [cit. 2023-03-27]. Dostupné z: <https://www.tutorialspoint.com/index.htm>

# Seznam obrázků a tabulek

## Seznam obrázků

[Obrázek 1 - Ukázka použité nástěnky (Zdroj: Autor) 8](#_Toc132289846)

[Obrázek 2 - Ukázka získaných dat (Zdroj: Autor) 12](#_Toc132289847)

[Obrázek 3 - Ukázka vizualizace rozdělení intentů (Zdroj: Autor) 15](#_Toc132289848)

[Obrázek 4 - Ukázka vzhledu aplikace (Zdroj: Autor) 17](#_Toc132289849)

[Obrázek 5 - Ukázka vizualizace rozdělení dat (Zdroj: Autor) 18](#_Toc132289850)

[Obrázek 6 – Ukázka intentu maturita\_zmena\_mat\_prace (Zdroj: Autor) 22](#_Toc132289851)

[Obrázek 7 – Ukázka stories (Zdroj: Autor) 23](#_Toc132289852)

[Obrázek 8 – Ukázka akcí a předlohy odpovědi (Zdroj: Autor) 24](#_Toc132289853)

[Obrázek 9 – Chat widget (Zdroj: Autor) 25](#_Toc132289854)

[Obrázek 10 – Chat zavřený (Zdroj: Autor) 26](#_Toc132289855)

[Obrázek 11 – Chat otevřený (Zdroj: Autor) 26](#_Toc132289856)

[Obrázek 12 – Chatovací okno (Zdroj: Autor) 27](#_Toc132289857)

[Obrázek 13 – Ukázka architektury aplikace (Zdroj: Autor) 28](#_Toc132289858)

## Seznam tabulek

Tabulka 1 - Tabulka pojmů 7

Tabulka 2 - Přehled výsledků porovnání frameworků pro tvorbu Chatbota 11

Tabulka 3 - Výběr scrapovaných stránek 14

Tabulka 4 – Popis souborů rasa projektu 20

Tabulka 5 – Pipeline komponenty 21

Tabulka 6 – Policies komponenty 21

Tabulka 7 – Popis tříd java aplikace 28

Tabulka 8 - Popis metod v Controller.java 29

Tabulka 9 - Popis tříd pro vypsání yml souborů. 29

1. XPATH je HTML cesta k objektu [↑](#footnote-ref-2)
2. Unikátní akcí je odpověď na otázku. Může to být aktivace web scrapingového scriptu, odkaz na webovou stránku nebo obrázek. [↑](#footnote-ref-3)
3. Open source nástroj pro vytváření počítačových nebo mobilních aplikací. [↑](#footnote-ref-4)
4. Vizuální nástroj pro vytváření uživatelského rozhraní. [↑](#footnote-ref-5)
5. Třída, která slouží k oddělení logiky uživatelského rozhraní od zbytku kódu aplikace. [↑](#footnote-ref-6)
6. Jsou to specifické adresy, které představují cíl pro komunikaci mezi klientem a serverem v rámci architektury REST. [↑](#footnote-ref-7)
7. Widget - též gadget, jsou miniaplikace, kterými lze rozšířit a obohatit webové stránky. [↑](#footnote-ref-8)