Vývoj komunikačního bota pro Programming Support Centre

David Rábel

Bakalářská práce
2021

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**David Rábel**
Osobní číslo:	**A18072**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Softwarové inženýrství**
Forma studia:	**Prezenční**
Téma práce:	**Vývoj komunikačního bota pro Programming Support Centre**
Téma práce anglicky:	**The Development of Chatbot for a Programming Support Centre**

Zásady pro vypracování

1. Vypracujte literární rešerši na téma vývoje komunikačního „bota", včetně popisu technologií, které plánujete pro tento účel využít.
2. Navrhněte a vytvořte komunikačního bota tak, aby dokázal automaticky komunikovat s uživatelem prostřednictvím grafického uživatelského rozhraní.
3. Implementujte možnost rezervace termínu konzultací na vybraný předmět ke konkrétnímu tutorovi.
4. Navrhněte a vytvořte databázi obsahující informace o předmětech, tutorech, termínech, rezervacích a uživatelích, kteří si zarezervovali termín.
5. Propojte komunikačního bota s vytvořenou databází a vhodně využívejte data při komunikaci s uživatelem.
6. Popište implementaci bodů zadání a zdokumentujte zdrojové kódy komunikačního bota.

Forma zpracování bakalářské práce: **Tištěná/elektronická**

Seznam doporučené literatury:

1. MULDOWNEY, Oisin. *Chatbots: An Introduction And Easy Guide To Making Your Own*. Dublin, Ireland: Curses & Magic, 2017. ISBN 978-1-9998348-0-7.
2. SAFKO, Lon. *The Artificial Intelligence Chatbot: Unexpected Positive Consequences*. United States of America: Safko, 2019. ISBN 978-1070979656.
3. KOTHARI, Amit, Rania ZYANE a Joshua HOOVER. *Chatbots for eCommerce: Learn how to build a virtual shopping assistant*. Santa Rosa: Bleeding Edge Press, 2017. ISBN 97819399902481.
4. Messenger Platform. *Facebook for Developers* [online]. Facebook, © 2020 [cit. 2020-11-17]. Dostupné z: https://developers.facebook.com/docs/messenger-platform
5. CHUA, David. *GitHub: Pymessener. GitHub* [online]. GitHub, © 2020 [cit. 2020-11-17]. Dostupné z: https://github.com/davidchua/pymessenger
6. Data to Fish. *Data to Fish: How to Connect Python to SQL Server using pyodbc* [online]. Data to Fish, © 2020 [cit. 2020-11-17]. Dostupné z: https://datatofish.com/how-to-connect-python-to-sql-server-using-pyodbc/

Vedoucí bakalářské práce: **Ing. Tomáš Vogeltanz**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **15. ledna 2021**
Termín odevzdání bakalářské práce: **17. května 2021**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
DĚKAN
L.S.

prof. Mgr. Roman Jašek, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2021

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

v. r. David Rábel
podpis studenta

ABSTRAKT

Tato bakalářská práce se zabývá návrhem a vytvořením chatbota pro Programming Support Centre. V teoretické části je popsána historie a vývoj chatbotů. Také jsou v práci ukázány jednotlivá řešení chatbotů na klíč a následně jsou teoreticky popsány použité technologie vytvořeného chatbota. V praktické části je uveden návrh chatbota a jeho funkcí a také jeho kompletní implementace. Dále se zde nachází návrh a realizace databáze a popis vytváření vlastního chatbota na platformě Facebook Messenger. V poslední části je uvedena ukázka nasazení chatbota na server s Linuxovou distribucí.

Klíčová slova: Chatbot, Databáze, Facebook, Server, Linux, Programming Support Centre, Messenger

ABSTRACT

This bachelor thesis deals with the design and creation of a chatbot for the Programming Support Center. The theoretical part describes the history and development of chatbots. The individual solutions of turnkey chatbots are also shown and then the used technologies of the created chatbot are described. The practical part presents the design of the chatbot and its functions, as well as its complete implementation. Furthermore, the design and implementation of a database are pointed out and a description of creating your own chatbot on the Facebook Messenger platform is explained. The last part shows an example of deploying a chatbot on a server with a Linux distribution.

Keywords: Chatbot, Database, Facebook, Server, Linux, Programming Support Centre, Messenger

Rád bych poděkoval svému vedoucímu práce panu Ing. Tomáši Vogeltanzovi za jeho vedení, dobré rady, čas a velkou trpělivost, bez kterého by tato práce neměla šanci vzniknout.

Dále bych chtěl poděkovat Adamovi Mirremu za pomoc a trpělivost s nastavováním Linuxového serveru.

Díky také patří organizaci Programming Support Centre, která mi poskytla možnost vytvořit tuto bakalářskou práci. Rád bych poděkoval své rodině a kamarádům za velkou podporu a trpělivost nejen při tvorbě této práce, ale i po celou dobu studia.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD

Problematika chatbotů je obrovské technologické téma, které se neustále vyvíjí a přináší spousty nových problémů. S rostoucím výpočetním výkonem a příchodem nových technologií jsou chatboti stále chytřejší a poskytují interaktivní prostředí pro nové i stávající zákazníky. To je také jeden z důvodu, proč byl navržen projekt vývoje chatbota pro Programming Support Centre.

Existuje mnoho technologií a platforem, na kterých lze takového chatbota vytvořit, ovšem existuje jedna platforma, která ve vytváření chatbotů vyčnívá. Jedná se nejspíše o nejznámější sociální síť Facebook, která má momentálně více než 3 miliardy aktivních uživatelů. Díky této obrovské základně uživatelů je to i nejlepší a nejlevnější cesta, jak oslovit skupinu potenciálních uživatelů. Právě z tohoto důvodu v roce 2016 spustila společnost Facebook možnost vytvářet aplikace na jejich platformě, včetně aplikace chatbota. [1]

Tato práce má za cíl navrhnout a vytvořit komunikačního bota tak, aby dokázal komunikovat s uživatelem právě pomocí grafického rozhraní. Dále by měl být uživatel schopný rezervovat jednotlivé termíny a získat informace o organizaci skrze chatbota. Součástí práce je také kompletní návrh databáze včetně fyzického modelu, který by měl obsahovat všechny informace pro plynulý chod chatbota. Nakonec je provedeno nasazení na server právě proto, aby tento chatbot mohl být spuštěn veřejně.

Přínosem práce je ulehčení aktuální situace, která je způsobena probíhající pandemií a nemožností se účastnit prezenční výuky. Toto navržené řešení poskytuje alternativní možnost rezervace studentů na termíny doučování, a také umožňuje studentům získat informace o projektu Programming Support Centre. Velkým přínosem je také to, že v kódu aplikace je implementováno vytváření prvků uživatelského rozhraní dynamicky, a tudíž aplikace dokáže flexibilně reagovat na aktuální potřeby centra.

I. TEORETICKÁ ČÁST

1 CHATBOT

Chatbot je počítačový program, který je určený k automatizované konverzaci. Nejčastěji jsou chatboti vytvářeni tak, aby komunikovali s běžnými lidmi. Nalezneme i výjimky, kde chatboti komunikují mezi sebou. V komerční sféře se chatboti používají stále více, a to hlavně z důvodu, že jsou v dnešní době schopni nahradit i webové stránky a e-shopy. Namísto toho, aby musel uživatel prohledávat celou webovou stránku sám, chatbot uživateli nabídne interaktivní variantu komunikace, která mu ukáže, čím se firma zabývá, jaké produkty nabízí a dovolí uživateli produkty zakoupit s tím, aby uživateli byla poskytnuta co nejlepší zákaznická péče. Podobné využití chatbotů můžeme najít také u různých organizací (pojišťovny, banky, doučovací centra), kde se chatbot primárně využívá k poskytnutí důležitých informací pro klienty. Chatbot tedy může navést klienta na potřebné webové stránky, popř. zodpovědět důležité dotazy tykajících se například podmínek pojištění. [2]

Ve světě existují stovky různých druhů chatbotů, ale mezi nejvíce oblíbené patří chatboti pro zabavení uživatele. Nejznámější chatbot pro zabavení uživatele se jmenuje Mitsuku. Avšak nesmíme opomenout to, že tento chatbot vyhrál již pětkrát Turingův test, díky čemuž drží světový rekord. Mitsuku lze také nalézt na různých platformách včetně Facebook Messengeru. [2]

V dnešní době můžeme chatboty rozdělit na dvě velké sekce. Chatboty reagující na hlas (chytrý asistent), mezi nejznámější patří Google Assistant, Microsoft Cortana a Apple Siri. Jako druhou sekci chatbotu můžeme považovat chatboty založené na psaném textu. Velké množství těchto chatbotů můžeme najít na známých platformách Facebook Messenger, Skype, Viber, WhatsApp, Telegram, WeChat, Kik, kde lze také nalézt různé knihovny a dokumentace pro jejich vytvoření. [2]

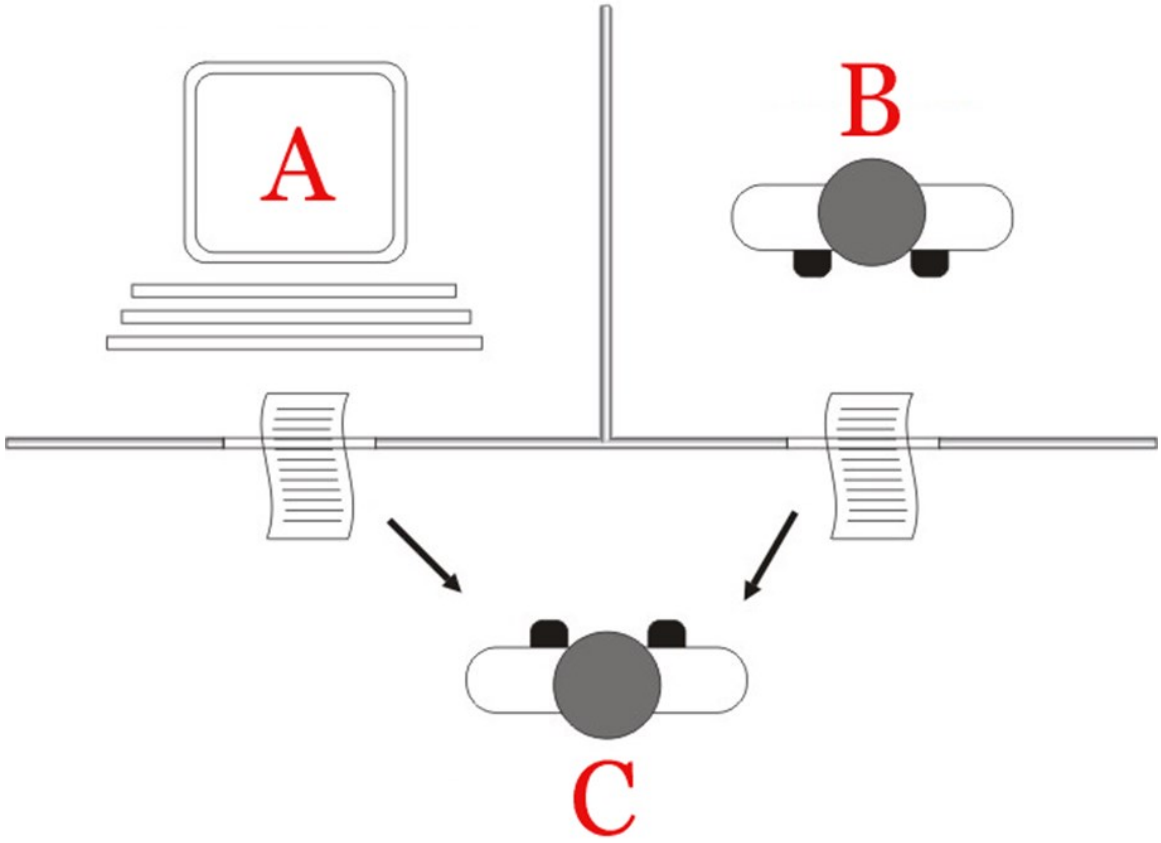
1.1 Historie

Tato podkapitola zahrnuje všechny podstatné milníky, bez kterých by dnes chatboti nebyly tak „inteligentní“, jak je známe dnes.

1.1.1 Turingův test

V roce 1950 si krypto analytik a zakladatel moderní počítačové vědy Alan Turing položil otázku „Může počítač komunikovat takovým způsobem, aby konverzace byla k nerozeznání

od běžného člověka?“ K odpovědi na tuto otázku slouží test, který dnes můžeme znát pod anglickým názvem „Turing test“. [2]



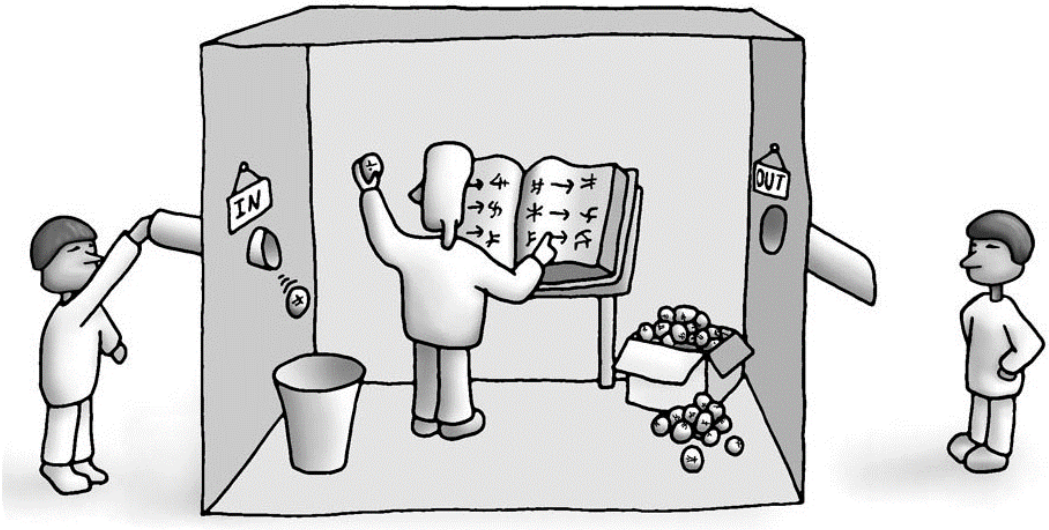
Obrázek 1. Turingův test [57]

Turingův test je imitace hry, kde probíhá slepá textová komunikace mezi člověkem a testovacím subjektem (člověk/robot nebo člověk/člověk). Na konci této komunikační hry se musí osoba rozhodnout, jestli komunikovala s živou osobou, anebo s chatbotem viz. Obrázek 1. Díky této komunikační hře je odpovězeno na otázku, jestli stroj umí myslet. [3]

V roce 1990 americký investor a aktivista Hugh Loebner vytvořil soutěž, která byla založena na Turingově testu v praxi. Hlavní cenou této soutěže bylo 100 000 amerických dolarů pro prvního programátora, anebo tým, který bude schopen sestrojit chatbota, který projde Turingovým testem před předem sestavenou porotou. I přes všechnu snahu tato cena nebyla zatím nikomu udělena. [3]

Dlouhou dobu byl Turingův test považován za základní měřítko schopnosti umělé inteligence i přes všechny aspekty tohoto testu, dnes však takto vnímán už není. To hlavně z důvodu tzv. „argumentu čínského pokoje“.[3]

Argument čínského pokoje je hypoteticky řečeno uzavřená místnost s velkým množstvím čínských textů s každou smysluplnou větou čínského jazyka. Do takového pokoje je umístěn člověk, který neumí čínsky ani slovo, ale má znalost na to, aby nalezl na základě předem daného kontextu odpověď na konkrétní otázku. Tazatel otázek z vnějšku místnosti by se mohl domnívat, že daný člověk v místnosti umí plně čínsky, ale opak je pravdou, protože pracuje pouze mechanicky s pro něj neznámými symboly viz. Obrázek 2. [4]



Obrázek 2. Argument Čínského Pokoje [58]

1.1.2 ELIZA

V roce 1964 byl stvořen jednoduchý počítačový program jménem ELIZA. Jedná se o nejstaršího a nejznámějšího chatbota. Byl vytvořen Německo-Americkým profesorem Massachusettského technologického institutu (MIT) Josephem Weizenbaumem, který se díky tomuto projektu stal inspirací pro mnoho dalších programátorů. [3][5][6]

Tento počítačový program je pojmenovaný podle postavy z divadelní hry „Pygmalion“. Program se snaží imitovat nevzdělanou anglickou dívku, která se učí, jakým způsobem má komunikovat jako žena, aby ohromila vysoko postavenou Londýnskou společnost. [1][3][3]

První scénář tohoto počítačového programu je nazván DOCTOR, který měl za úkol simulovat Rogeriánského psychoterapeuta, který funguje na základě vracení otázek zpátky pacientovi ve snaze přimět ho k hovoru. Díky tomuto jednoduchému scénáři jej bylo jednoduché prezentovat i za branami MIT. Část odborné lékařské veřejnosti se domnívala, že bude možné tímto počítačovým programem nahradit nedostatek psychiatrů. Obrázek 3 je příkladem komunikace chatbota Eliza. [3][5][6]

```
> Ahoj, Já jsem Eliza.
* Ahoj
> Jak se dnes máte.. O čem byste si chtěl promluvit?
* Jsem v depresi.
> Přišel jste, protože jste v depresi?
* Ano
> Aha.
* To kvůli tomu, že mám problém se svými rodiči.
> Můžete to rozvést?
* Ano, myslím že je to kvůli mému otci.
> Opravdu si to myslíte?
* Ano, on umí být někdy opravdu náladový.
> Rozumím.
...
```

Obrázek 3. ELIZA [59]

1.1.3 PARRY

V roce 1972 spatřil světlo světa další chatbot jménem PARRY. Byl vytvořen americkým psychiatrem Kennethem Colbym. Tento chatbot se choval jako naprostý opak ELIZY, choval se totiž jako pacient trpící schizofrenií. [1][3]

Cílem PARRYHO bylo vyvolat kontroverzi za účelem získat od uživatele propracovanější odpovědi. Díky tomuto chování byl i tento chatbot použit v lékařském průmyslu, a to konkrétně jako pomůcka pro mladé psychiatry, aby se naučili, jak komunikovat s pacientem, který trpí schizofrenií. [1][3][6]

Ve stejném roce se ale stalo to, co každý očekával. Doktor (ELIZA) začal konverzaci s Pacientem (PARRY). Obrázek 4 ilustruje příklad této konverzace. [3]

ELIZA: Would you like to discuss your nerves?

PARRY: I guess so.

ELIZA: What does that suggest to you?

PARRY: What are you getting at?

ELIZA: What answer would please you most?

PARRY: It's not a matter or like or dislike.

ELIZA: What else would you like to discuss?

PARRY: I am getting tired of answering the same questions over and over.

ELIZA: What resemblance do you see?

PARRY: Why do you need to know that?

ELIZA: What do you think?

PARRY: I don't understand your motives.

ELIZA: Why don't you understand my motives?

Obrázek 4. PARRY [60]

1.1.4 DR. SBAITSO

Velký zvrat nastal v roce 1991, kdy byl vytvořen chatbot, který využíval novou technologii „text to speech“, kterou poskytovala společnost Creative Labs za pomocí jejich nové zvukové karty „Sound Blaster sound card“. Chatbot je pojmenován právě po této kartě Dr. Sbaitso (Sound Blaster Artificial Intelligent Text to Speech Operator). Jak již z názvu vyplývá, jednalo se o prvního předchůdce dnes známých hlasových asistentů a poprvé byl také použit na osobních počítačích se systémem MS-DOS. [3][7]

Nejenže tento chatbot byl schopný rozpoznat základní věty od uživatele, ale byl schopný na ně i zpětně odpovědět a to verbálně. Velký problém byl v tom, že bot byl příliš málo komplexní a kvůli tomu pak uživateli vracel stále stejné odpovědi např. “Proč se tak cítíš??” Největším problém však bylo to, že pokud větu nerozeznal, odpověděl „To není můj problém“. Naopak největší výhodou bylo, že uměl číst text, který mu uživatel označil pomocí slova „say“. [7]

1.1.5 A.L.I.C.E

Roku 1995 byl vytvořen další významný chatbot s jménem A.L.I.C.E (Artificial Linguistic Internet Computer Entity) a to Richardem Wallacem. [9]

Tento chatbot využívá heuristické vzory pro smysluplnější konverzaci. Jednalo se o prvního chatbota, který využil XML schéma, též známé jako AIML, které určovalo pravidla toho, jak bude konverzace vypadat.[9]

V roce 1998 byl tento chatbot přepsán do Javy a dnes ho můžeme vyzkoušet na volně dostupných internetových stránkách.[9]

1.1.6 SMARTCHILD

V roce 2001 nastoupil na scénu chatbot s jménem SmartChild, který byl vytvořen společností ActiveBuddy. Dá se považovat za přímého předchůdce dnes všem známého chatbota Siri. [7][9]

Tento chatbot byl nasazen na AOL Instant Messenger a Windows Live Messenger proto, aby zabavil uživatele právě na těchto sociálních sítích. Chatbot byl napsaný v jazyku Perl a zvládal jednoduché počty, konvertování měny a odpovídání na otázky jako např. „Jaké je počasí?“. [7]

1.1.7 IBM WATSON

Roku 2010 byl do světa poslán chatbot od společnosti IBM, který je pojmenován po prvním výkonném řediteli Thomasem J. Watsonem. Tento chatbot je založený na zpracování přirozeného jazyka. [8]

Je znám především tím, že byl vyvinut za účelem odpovídat na otázky v Americké televizní show Jeopardy, což se také v roce 2011 stalo a byl postaven před dosavadního šampiona této televizní show, kde IBM Watson s přehledem zvítězil. [10]

Cílem tohoto chatbota bylo vytvořit novou technologickou generaci, která je schopna daleko efektivněji odpovídat na nestrukturovaná data lépe, než tomu bylo doposud. Což se také díky tomuto vítězství v Jeopardy podařilo. [10]

1.1.8 SIRI

Taktéž v roce 2010 byl spuštěn první inteligentní chatbot (založený na hlasové komunikaci) pro uživatele iOS, a to společností Apple. Jedná se o inteligentního osobního asistenta, který využívá přirozeného jazyka proto, aby reagoval na podněty uživatele. [2]

Apple si také nechal patentovat Siri, a to hlavně proto, aby umožnil ostatním developerům využít Siri v jejich aplikacích, za předem daných podmínek. [9]

V dnešní době je Siri schopná prakticky všeho, co byste od digitálního asistenta mohli očekávat. [9]

1.1.9 GOOGLENOW

V roce 2012 byl vypuštěn do světa další chytrý chatbot založený na hlasové komunikaci, tentokrát však od společnosti Google. [10]

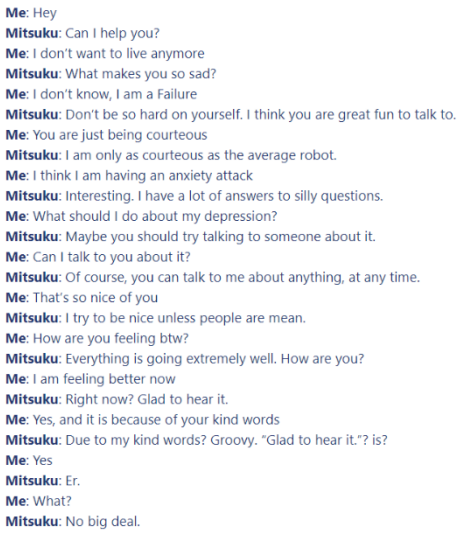
GoogleNow byl vytvořen proto, aby byl schopný odpovídat na otázky uživatele, obsluhovat jeho webové služby, a hlavně pomáhat s vyhledáváním. [10]

1.1.10 MITSUKU

Mitsuku, známá jako Kuki, je chatbot vytvořený na technologii značkovacího jazyka pro umělé inteligence (AIML) Stevem Worswickem. Je vytvořen tak, aby napodoboval chování osmnáctileté ženy. Obsahuje všechny dostupné AIML soubory od chatbota Alice a mnoho dalších vlastních souborů, které získává při navazování nových konverzací s uživateli. [10][11]

Inteligence Mitsuku je založená na schopnosti reagovat na konkrétní objekty (slova). Například pokud se někdo zeptá „Umíš sníst dům?“ Mitsuku bude hledat objekt dům, a to konkrétně hodnotu „z čeho je vyroben“. Tato hodnota je rovna „cihle“ tím pádem Mitsuku odpoví „ne“. Obrázek 5 je příkladem této reakce. [10]

Tento chatbot je také dostupný na mnoha známých platformách například Facebook Messenger, Twitch group, Kik, Telegram chat. Velkou část své existence byl také dostupný na Skypu, ale musel být z této platformy odstraněn. Od roku 2015 komunikuje s více než 250,000 lidmi každý den. [11]



Obrázek 5. Mitsuku [61]

1.1.11 CORTANA

Chatbot Cortana byl poprvé ukázán na Microsoft konferenci v roce 2014 a hned poté byl nasazen na obě platformy, a to jak Windows 10, tak Windows Phone. Tento chatbot je také založen na hlasové komunikaci, stejně jako je to u jeho konkurence (Alexa, Siri). [3]

Proto, aby byl uživatel schopný začít konverzaci s Cortanou, musí nejprve napsat svůj dotaz do speciální kolonky, popřípadě zvolit mikrofon a dotaz položit. Cortana primárně slouží k tomu, aby ulehčila uživateli práci se systémem Windows, a proto jsou také její funkce k tomu uzpůsobené. Například disponuje schopností ukládat poznámky do Windows kalendáře, prohledávat adresáře, upravovat uživatelské nastavení systému Windows a mnoho dalšího. [12]

1.1.12 ALEXA

Amazon Alexa Chatbot, známý pod jménem Alexa, byl představen v roce 2014. Jedná se o virtuálního chatbota založeného na hlasové komunikaci. Alexa je schopná zjistit aktuální dopravní situaci, počasí, jako ostatní chatboti, ale navíc je přímo integrovaná na Amazon a na všechny jeho chytré spotřebiče. [5]

V dnešní době se primárně využívá právě pro chytrou domácnost a můžeme se ní setkat prakticky všude, díky tomu že Amazon od doby, kdy byla Alexa uvedená na trh, již vypustil mnoho zařízení, na kterých je schopná pracovat, ať se již jedná o EchoDot anebo, dnes již známá zařízení Alexa HomePod a HomePod mini. [13]

V dnešní době Alexa podporuje již 8 světových jazyků. [14]

1.1.13 BOTS FOR MESSENGER

V roce 2016 nastal velký boom mezi komerčními chatboty. Jednalo se o spuštění chatbotů pro platformu Facebook Messenger. V prvních šesti měsících bylo vytvořeno více než 30 000 chatbotů na Facebook Messengeru a dnes jich již máme více než 300 000. [15]

Právě díky obrovské síle Facebook Messengeru, který dnes již používá více než 1,3 miliard aktivních uživatelů, jsou právě tito chatboti budoucností marketingu a budoucí podoby toho, jak bude vypadat reklama na platformě Facebook. [15][16]

Messenger chatboti jsou, na rozdíl od lidí, schopni reagovat neustále. Chatboti mohou také zastávat roli nákupního košíku na Facebooku. Jsou schopni vytvářet specifické reklamy a interakce, které normální uživatel poslat nemůže. [17]

Mezi nejlepší možné chatboty, které můžeme najít na platformě Messenger, musíme zařadit chatbota firmy Marriott Rewards, která s chatbotem přišla pro zlepšení pohodlí zákazníka. Dnes je uživatel schopen přes tohoto chatbota vybrat konkrétní destinaci, čas příletu a udělat si rezervaci do konkrétního hotelu, který mu chatbot nabídne, a to vše bez toho, aniž by musel opustit chat. Mezi ostatní známé chatboty můžeme také zařadit chatbota firmy UNICEF, popřípadě chatbota firmy Sephora. [15][16]

2 ROZDĚLENÍ CHATBOTŮ

Chatboty lze rozdělit na dvě hlavní kategorie: konverzační, kde se chatbot snaží vést konverzaci s uživatelem a transakční, kde se bot netváří jako člověk, ale snaží se plnit konkrétní funkce, které jsou po něm požadované. Může se jednat například o bota, který má za úkol řešit rezervační systém hotelu, nebo sloužit jako turistický průvodce. [18]

Chatboty lze rozdělit také na hlasové a textové. U hlasových typů (voice based) je primární nezbytností mít kvalitně zpracovanou technologii na rozpoznání. Textové (text based) jsou oproti tomu méně náročné, což je jeden z důvodů, proč na této technologii běží většina dnešních chatbotů. [18]

2.1 Konverzační

Konverzačního chatbota můžeme považovat za nejčastěji užívaný typ chatbota. Jak již název může napovídat, jedná o chatbota, který vede konverzaci. Jinak řečeno, uživatel napíše otázku a chatbot na ní odpoví. [19][20]

Konverzační chatboti jsou primárně založeni na technologii NLP (Zpracování přirozeného jazyka). Cílem je, aby uživatel nepoznal, že se jedná o chatbota a měl pocit, že komunikuje s člověkem (Turingův test). [19][20]

Problém s konverzačními chatboty začíná v případě, když chatbot pochopí špatně kontext věty a kvůli tomu vrátí odpověď, která nedává smysl. Obecně platí, že čím složitější jazyk, tím větší s tím technologie NLP bude mít obtíže. [19][20]

2.2 Transakční

Transakční chatbot, občas také znám jako chatbot orientovaný na úkoly, má pouze jeden účel – zautomatizování konkrétního úkolu či funkce. [20]

Tento druh chatbota je vždy navržen tak, že má uživatel vždy na výběr konkrétní sadu možností výběru, v závislosti na tom, co chce uživatel udělat. Poté si uživatel vybere konkrétní možnost a chatbot ho provede procesem, při kterém mu poskytne všechny dostupné možnosti do té doby, dokud uživatel nebude spokojený s odpovědí, popřípadě ho přepojí na živého operátora. [20]

Transakční chatboti využívají také přirozený jazyk, ale jen k tomu, aby správně pochopili dotaz uživatele. K tomuto druhu chatbotů je potřeba znát strukturovaná data konkrétního

podniku/problému, který chceme řešit, a to hlavně z důvodu, že transakční chatboti se používají primárně pro zpracování běžných otázek, nebo jednoduchých transakcí. [20]

V dnešní době je tento druh chatbotů nejvíce rozšířený, a to hlavně na platformě Facebook Messenger. [20]

2.3 Hlasově ovládání

Hlasově ovládání chatboti jsou ti, kteří jsou schopni komunikovat a interagovat pomocí hlasové komunikace. Jsou schopni přijímat vstupy, jak skrz mluvené slovo, tak přes slovo psané. Tento druh chatbotů můžeme rozdělit ještě dále na hlasově-textové chatboty a pouze hlasové chatboty. [22]

2.3.1 Hlasově-textový chatbot

Jedná se o klasické textové chatboty, kteří mají navíc schopnost rozpoznat jednoduché hlasové vstupy. U těchto chatbotů je možné zadat vstup oběma způsoby, a to jak hlasově, tak textově. [21][22]

2.3.2 Čistě hlasový chatbot

Tento druh chatbotů je schopný zvládnout jednoduché úkoly jako je nastavit budík, pustit muziku, a také komplexnější věci jako je ovládání celého domu. Hlavními zástupci této skupiny chatbotů jsou například Amazon Echo nebo Google Home. [21][22]

2.4 Textově ovládání

Tak jako jméno vypovídá, textově založení chatboti jsou ti, kteří interagují za pomocí psaného slova. Tento druh chatbota je schopný posílat obrázky, videa a popřípadě ovlivňovat UI elementy, jako jsou například karusely a tlačítka. [21]

Textově založení chatboti jsou extrémně efektivní, a to hlavně z důvodu, že uživateli poskytnou řešení problému takřka v reálném čase. [21]

V dnešní době se jedná o nejrozšířenější typ chatbotů pro interagovaní byznysů se zákazníkem. Velkou výhodou textově ovládaných chatbotů je také možnost, že je lze snadno spojit se sociálními sítěmi, jako je např Facebook Messenger. [21]

3 CHATBOTI NA KLÍČ

Chatbot na klíč je dnes jedna z nejpoužívanějších verzí chatbotů, a to primárně z důvodu, že je to nejlevnější a nejlehčí způsob pro velké firmy, jak aplikovat chatbota pro svůj byznys. Velkou výhodou tohoto řešení je cena, jednoduchá konfigurace (není potřeba znalost programování), a především rychlost dodání. Dá se říct, že je to skoro jako web na klíč například WordPress. Naopak velkou nevýhodou těchto řešení je dlouhodobá návratová investice (měsíční platby) a zvláště menší variabilita konkrétního řešení a závislost na konkrétním řešení na klíč. [23]

3.1 ManyChat

ManyChat je služba, která je určena především pro vytváření chatbotů na platformu Facebook Messenger. ManyChat funguje jako služba na klíč, tudíž nastavení zabere jen několik málo minut. [24]

Zásadní výhodou této služby je jednoduchá integrace na platformu Facebook Messenger a jednoduché uživatelské rozhraní pro základní úpravy chatbota. [24]

ManyChat nabízí tři základní stupně plánů Free, Pro a Business. [24]

3.1.1 Free

Jak již z názvu vyplývá, jedná se o plán, který je poskytován platformou ManyChat zcela zdarma, avšak má spousty funkčních omezení jako je například omezení některých důležitých funkcí messengeru, anebo uvádění vlastního brandingu ManyChat v chatu. [24]

Velkou výhodou tohoto plánu je také to, že i přes to, že plán je zcela zdarma, je nabízena emailová podpora. [24]

3.1.2 Pro

Dále je nabízena varianta plánu Pro, která je určena především pro uživatele, kteří již vědí, co od takového chatbota chtějí, aby zvládl. Na rozdíl od varianty zdarma jsou zde zahrnuty všechny důležité funkce, a to včetně možnosti zaplatit zboží přímo v messengeru přes platební bránu. [24]

Nevýhodou tohoto plánu je, že cena začíná na 10 $ měsíčně, což se nemusí při dlouhodobém používání vyplatit, zvláště pokud plánujeme oslovit více zákazníku, protože se do ceny projeví počet oslovených uživatelů. [24]

3.1.3 Business

Cenový plán Business slouží primárně pro firmy, kde je potřeba špičková komunikace s potenciálním klientem. Tohoto výsledku je docíleno především, protože jsou dostupné všechny nástroje a plná podpora, kterou ManyChat nabízí. [24]

Cenová kategorie plánu Business není známá, a to především z důvodu, že záleží na konkrétních požadavcích a rozsahu projektu. [24]

3.2 WingBot

Jedná se o českou společnost, která se zaměřuje především na vytváření profesionálních chatbotů na míru. To má také za následek daleko vyšší pořizovací cenu a mnohonásobně lepší kvalitu, než může nabídnout služba na klíč. [25]

Velkou výhodou tohoto řešení je možnost implementace přirozeného jazyka pro plynulejší konverzaci. [25]

Mezi nejznámější klienty této společnosti patří např AXA, Krušovice, anebo všem známá Škoda Auto. [25]

3.3 ChatFuel

Jedná se o světově nejznámější a nejvíce používanou technologii na klíč pro vytváření Facebook Messenger chatbotů. Na této platformě je uživatel schopný vytvořit chatbota bez znalosti programovacího jazyka pro platformy Messenger, Instagram a Facebook.

Mezi nejznámější firmy využívající ChatFuel řešení je například Netflix, T-Mobile, Lego, Visa atd. [26]

ChatFuel nabízí tři základní plány Free, Pro a Premium [26]

3.3.1 Free

V tomto základním plánu je interakce omezena pouze na 50 uživatelů, avšak jsou dostupné všechny základní funkce messenger bota.

Velkou výhodou tohoto plánu je, že je zcela zdarma. [27]

3.3.2 Pro

Jde primárně o plán, který využijí firmy, které chtějí dopřát svým zákazníkům nejkomplexnější služby. Cena tohoto plánu začíná na 15$ za měsíc, ale roste v případě, že s chatbotem interaguje vice uživatelů. [27]

Velkou výhodou tohoto plánu je možnost implementace nákupního košíku přímo do chatbota, anebo plná podpora od chatfuelu. Naopak velkou nevýhodou je zvyšující se cena na základě využití uživateli. [27]

3.3.3 Premium

Jako poslední plán, který tato platforma nabízí, je plán s názvem Premium. Jedná se o plán, při kterém jsou odborníci z této platformy nápomocni při vytváření chatbota. [27]

Velkou výhodou tohoto řešení je, že se do jisté míry jedná o řešení na zakázku včetně kompletního nastaveni reklam pro váš business na platformě Facebook. Jako jedinou nevýhodu lze uvést cenu, která předem není známá. [27]

3.4 ChatBot

Jedná se o společnost, která vytváří chatboty na klič, a to pro platformy LiveChat, Facebook Messenger WordPress atd. [28]

Výhodou tohoto řešení je možnost mít více chatbotů na jednou (dle plánu) a velmi přehledné prostředí pro vytváření chatbota, které je však více technicky náročné, a tak není určeno úplným začátečníkům. [28]

Naopak velkou nevýhodou je dražší cena, která v nejlevnějším plánu začíná na 50$ za měsíc. [28]

4 POUŽITÉ TECHNOLOGIE

V této kapitole jsou popsány technologie, které byly použity během vývoje chatbota. Jedná se o technologie, které spolu velmi dobře umí komunikovat a jsou nezbytné pro vývoj chatbota. Jsou zde uvedeny technologie, které byly použity jak na serveru při nasazení aplikace, tak na lokálním zařízení při samotném vývoji.

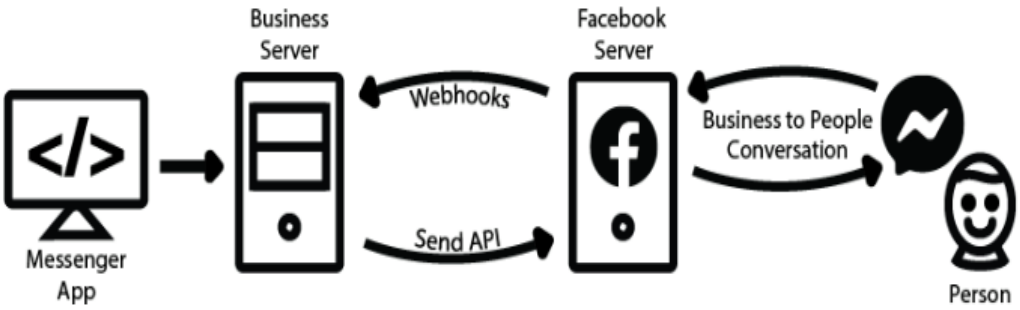
4.1 Facebook-Messenger

Jedná se o technologii od společnosti Facebook, která slouží k propojení vlastního serveru se serverem Facebooku. [29]

Vše musí začít u aplikace, kterou si uživatel vytvoří pro danou stránku na Facebook-Messengeru. (Tuto aplikaci lze vytvořit pouze pro Facebook stránky, nikoliv pro osobní profil.) Po úspěšném založení obdrží uživatel unikátní token, pomocí kterého bude komunikovat s Facebook serverem. Poté, když je vše nastaveno a uživatel pošle zprávu na jeho stránku za pomocí messenger klienta nebo webového rozhraní Facebooku, Facebook-server pošle webhook na požadované URL, již předem nastaveného serveru s aktuální zprávou klienta a podrobnosti o zprávě a klientovi (ID_Uživatele, Jméno, Text, Čas). Poté pomocí Send API může uživatel odpovědět zpátky na Facebook server, kde se právě pomocí unikátního tokenu ověří, jestli zprávu posílá z onoho serveru a poté je přeposlána konkrétnímu klientovi, anebo v případě, že by token neodpovídal, zahozena. Všechna zmíněná komunikace probíhá v řádu jednotek sekund pro maximální možné pohodlí klienta. Obrázek 6 znázorňuje celý workflow komunikace. [29]

Proto, aby tato aplikace mohla komunikovat se serverem, je zapotřebí i zabezpečená doména (https) a to z důvodu, že dle pravidel Facebooku není možné komunikovat přes nezabezpečenou doménu(http). [29]

Chatbot je funkční pouze pro majitele stránky, popřípadě ověřené developery a testery až do doby, než pošle svou aplikaci na schválení a zveřejnění Facebooku. Poté, pokud je vše provedeno v pořádku a dle předem daných zásad Facebooku, je aplikace vypuštěna veřejně mezi uživatele a jakýkoliv uživatel Facebooku může chatbota využívat. [29]



Obrázek 6. Messenger Flow [62]

PyMessenger

Jedná se o wrapper pro platformu Facebook Messenger napsaný v jazyku Python. Tento wrapper byl napsán uživatelem s přezdívkou davidchua a jeho dokumentace se nachází na githubu. Poskytuje mnoho důležitých funkcí pro platformu Facebook Messenger, jako jsou například funkce send_text_message a mnoho dalších. Je také důležité podotknout, že se nejedná o oficiální wrapper a se společností Facebook nemá nic společného. [30]

Wrapper

V počítačové vědě je wrapper jakákoliv entita, která zapouzdřuje jinou entitu (knihovnu, set funkcí). Primárně se wrapper používá pro dva účely – k převodu dat do kompatibilního formátu anebo ke skrytí složitosti původní entity, a to pomocí použití abstrakce. [31]

4.1.1 Facebook Messenger Funkce

V této kapitole jsou popsány funkce, které byly použity během vývoje chatbota. Jedná se o funkce, které vychází z oficiální dokumentace technologie Facebook Messenger. Tyto funkce jsou obsaženy v knihovně Pymessenger.

URL Button

Jedná se o typ tlačítka, který je, díky správnému složení payloadu, schopný otevírat jednotlivé webové stránky v Messenger webview. Tlačítko se definuje pomocí klíčového slova WebView v klíčové hodnotě Type. Poté je potřeba uvést dodatečný klíč URL, kde jako hodnota je uveden odkaz na naší konkrétní webovou stránku. [32]

PostBack Button

Po stisknutí tlačítka typu postback se pošle message_postback s předem definovaným payloadem na daný webhook, což nám dovolí reagovat na stisk tohoto tlačítka například tím, že pošleme uživateli specifickou zprávu. [32]

Send Generic Message

Jedná se u metodu v knihovně PyMessenger, která vytvoří jednoduchou strukturovanou zprávu, která obsahuje titulek, podtitulek, obrázek a tlačítka (maximálně tři). Metoda přijímá dva parametry. Prvním z nich je ID uživatele, kterému má daná zpráva dojit a druhý parametr je template payload, kde je definována podoba celkové zprávy jako jsou obrázky, titulky, tlačítka atd. [33]

Send Quick Message

Jedná se u metodu v knihovně PyMessenger, která umožnuje programátorovi vytvořit zprávu, která bude obsahovat až 13 tlačítek. Každé z těchto tlačítek obsahuje titulek, obrázek (nepovinný) a payload. Tato metoda je schopná také zjistit uživatelovu adresu, číslo anebo jeho emailovou adresu. [34]

V tu chvíli, když uživatel stlačí tlačítko, menu výběru zmizí a zpráva je odeslána na webhook, kde na něj může reagovat aplikace. [34]

Send Image Url

Metoda send_image_url se nachází v knihovně Pymessenger. Tato metoda umožnuje uživateli poslat obrázek/gif přímo do chatu, a to pomocí odkazu. Metoda vyžaduje dva povinné parametry, první z nich je recipient_id a druhým již zmiňované URL. [35]

Button Template

Jedná se o standardizovanou podobu slovníku, kde jsou jako klíče uvedeny type,title a payload. Do klíče „type“ můžeme uvést URL button, Postback Button, Call Button, Log In Button, Log Out Button a Game Play Button. V našem případě máme v projektu použité pouze dva typy tlačítek, a to Postback button a URL button. [36]

Další typ klíče, který je v našem případě vyžadován, je klíč title, do kterého zadáváme již zmíněné názvy jednotlivých tlačítek. [36]

Poslední klíč, který je vyžadován v případě tlačítka Postback, je payload, který je zde vnímán jako data, které si přeneseme do další interakce. Právě díky správné konfiguraci payloadu je

poté aplikace schopná interagovat například na to, když uživatel stiskne tlačítko „Dotaz“. [36]

4.2 Python

Python je vysokoúrovňový objektově orientovaný programovací jazyk, který navrhl Guido van Rossum. Nabízí dynamickou kontrolu datových typů, což jej činí v kombinaci s dynamickým vstupem a dynamickou vazbou velmi atraktivním pro rychlý a kvalitní vývoj všech různých aplikací. Lze jej také použit jako skriptovací jazyk. [37]

Syntaxe Pythonu je velmi snadná, a díky tomu je tento jazyk také tak oblíbený. Python je vyvíjen jako open source projekt, a tudíž je zcela zdarma a je k němu možné dohledat nepřeberné množství balíčku a knihoven. V dnešní době je Python dokonce součástí základní instalace GNU/Linux. [37]

Tento jazyk je rychlý a oblíbený, a to hlavně z důvodu, že při chybě anebo špatně zadaném vstupu nikdy nezpůsobí chybu segmentace, ale místo toho vždy vyvolá výjimku, kde uživateli poskytne bližší informace o tom, co se nepovedlo. [37]

Jedná se o multiparadigmatický jazyk a díky tomu nabízí, jak objektové, tak procedurální programování. Z důvodu, že psaní kódu se překládá na základě odsazení, je kód vždy přehledný a díky tomu velmi příjemný i pro začínající uživatele. [37]

4.3 MSSQL

Microsoft SQL Server je relační databázový systém, který byl vytvořen firmou Microsoft. Je napsán v jazyku C/C++ a je schopný běžet jak na systému Windows, tak na systému Linux. Jedná se o druh softwaru, který má za úkol ukládat a získávat data na základě požadavků uživatele (SQL dotazů). [38]

MSSQL server je dnes velice populární řešení databázových problémů, a to zejména z důvodu, že je jednoduchý na použití a naučení. Pro soukromé a komerční užití je poskytován zcela zdarma. Mezi klíčové vlastnosti databázového enginu patří to, že je schopný ukládat data pomocí instančních tabulek a má možnost importu a exportu dat pomocí různých formátu jako je například XML atd. [38]

Díky tomu, že je dnes tak populární, je MSSQL schopen běžet jak systému Linux (Red hat, Suse), tak také v Docker kontajneru. [38]

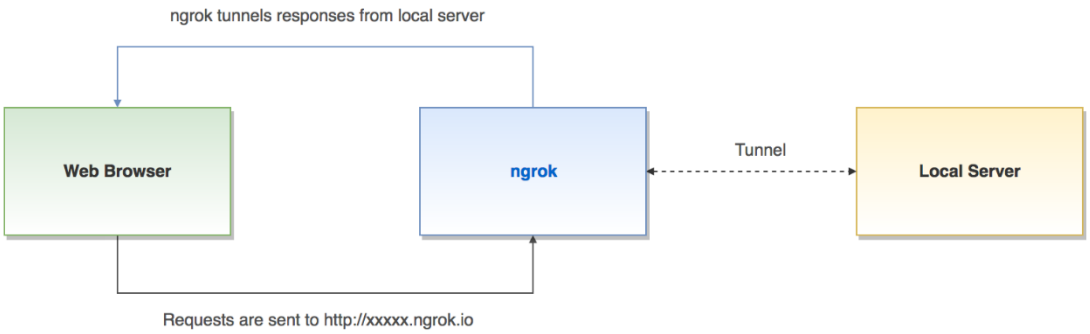
4.4 Ngrok

Jedná se o multiplatformní aplikaci, která umožnuje vývojářům odhalit lokální server na Internet s minimálním úsilím. Ngrok je schopný zobrazit lokální server na jedné z Ngrok subdomén, díky čemuž není potřeba veřejná IP anebo webová doména. [40][41]

Tento software je schopen obejít i NAT a Firewall, a to pomocí vytvoření TCP tunelu z náhodně vygenerované subdomény Ngrok.com. Po specifikování portu, na kterém bude server poslouchat Ngrok, vytvoří zabezpečené připojení na jejich server, tudíž se můžou vytvářet požadavky na lokální server, který ho přesměruje na Ngrok pomocí tunelu tak, jak ukazuje Obrázek 7. [40][41]

Ngrok je zcela zdarma a jeho servery pro veřejné použití jsou umístěny po celém světě, díky čemuž je tento software také tak oblíbený. [40][41]

Při vytvoření běžného tunelu se automaticky vytvoří jak http, tak https subdoména, protože dnes již mnoho webových aplikací vyžaduje certifikované domény. [40]



Obrázek 7. Ngrok [63]

4.5 Flask

Flask je webový mikro Framework, který je určený pro jazyk Python. Byl vyvinut Arminem Ronacherem. Jedná se o mikro Framework, a to z důvodu, že proto, aby fungoval, nepotřebuje žádné další knihovny nebo nástroje. Tento Framework reprezentuje kolekci knihoven, které usnadní vývojáři práci bez toho, aniž by se musel zabývat věcmi, jako je konfigurovaní protokolů atd. [42]

Flask nemá žádné vlastní ORM, tudíž dává uživatelům svobodu si zvolit vlastní framework pro práci s databází. Je založen na WSGI Werkzeug nástroji a na Jinja2 šabloně. [42]

4.5.1 Werkzeug

Jedná se o WSGI (Web Server Gateway Interface) nástroj, který slouží k implementaci požadavků a přijímání odpovědí od serveru. [42][43]

4.5.2 jinja2

Jinja2 je populární šablona pro jazyk Python, a to z důvodu, že bez větších problémů dovoluje kombinovat jazyk HTML a Python dohromady a tím vyvářet dynamické webové stránky. [42][43]

4.6 Gunicorn

Jedná se o Python WSGI HTTP server pro linuxové distribuce, který vychází z Ruby's Unicorn projektu. Pomocí WSGI.py souboru, který je vygenerován za pomocí frameworku, se server následně spojí s aplikací a díky tomu se je schopný komunikovat s webovou aplikací a serverem. [45]

4.7 Pyodbc

Pyodbc je open source Python modul, který vytváří přístup k OBDC. OBDC je rozhraní (tzv. API), které umožnuje komunikovat s databázovým systémem. Díky tomu je program schopný komunikovat s databází a jejími daty. [46]

Operace připojení

Jedná se u funkcionalitu, která má za úkol připojit se ke konkrétní databázi, aby na ní mohli být volány jednotlivé funkce jako je select, delete a insert. Operace připojení vyžaduje po uživateli definovat OBDC driver, který databáze používá, také jméno serveru, na kterém je uložená databáze, dále jméno konkrétní databáze a přihlašovací jméno a heslo. [47]

Operace spuštění dotazu

Vyvolává se pomocí příkazu cursor execute, kde cursor je připojená databáze dle funkce operace připojení a do parametru execute se vkládá konkrétní příkaz, o který má uživatel zájem (například Select). Tato funkce přijímá všechny dostupné příkazy, které jdou volat na databázi. Tyto záznamy jsou následně vráceny jako soubor výsledků, ve kterém je poté uživatel schopný iterovat. [48]

4.8 Nginx

Nginx je open source software, který slouží jako webový server. Může být také využit jako reverse proxy, load balancer atd. Nginx v dnešní době využívá mnoho velkých firem včetně Microsoftu, Googlu a Apple. [49]

V roce 2004 byl poprvé publikován Igorem Sysoevem, díky kterému byl vyřešen problém C10k, který spočíval v tom, jak obsloužit 10000 připojení v rozumném čase. [49]

4.8.1 Princip

Nginx je vytvořen tak, aby obsloužil co nejvíce klientů za co nejmenšího využití paměti. Místo toho, aby vytvářel nové procesy pro každý webový požadavek zvlášť, používá asynchronní přístup založený na jednotlivých událostech, díky čemuž jsou zpracovány pouze za využití jednoho vlákna. [49]

4.8.2 Reverse proxy

Chová se jako prostředník mezi klientem a serverem. Klientské žádosti, které následně obdrží, jsou přesměrovány na server, kde se zpracují a odpovědi jsou poté navráceny a tváří se, jako by přišly přímo z webového serveru. Nejčastějšími důvody pro využití reverse proxy je komprese dat anebo SSL zabezpečení. [50]

4.9 Let's Encrypt

Jedná se o certifikovanou autoritu (CA), která poskytuje webové certifikáty, které jsou potřebné na to, abychom si aktivovali HTTPS protokol na webové stránce. Pro využití této certifikační autority je potřeba používat ACME protokol, který musí běžet na serveru, kde budeme hostovat webovou stránku. [51]

Tato certifikační autorita je zcela zdarma a byla spuštěna skupinou Internet Security Research Group (ISRG). Všechny tyto certifikáty, které jsou vydány, jsou však platné pouze 90 dnů a pak je nutné je znovu obnovit. [51]

4.10 Linux Fedora Server

Jedná se o Linuxovou distribuci serveru, tato distribuce byla vyvinuta komunitou, která si říkala Fedora Project a byla podporována společnostmi Red Hat anebo IBM. Mezi hlavní výhodu této distribuce patří také to, že je plně open-source a zároveň není zatížena žádnou licencí, a to ani pro komerční použití. Za velkou výhodu této linuxové distribuce můžeme

také považovat to, že je adaptivní k novým technologiím, díky čemuž vychází aktualizace v šesti měsíčních cyklech. [52]

Stabilita a výhody Fedory zapříčinili to, že se tato distribuce stala oficiální distribucí Red Hat Enterprise Linuxu, díky čemuž se také stala jedna z nejpopulárnější a nejlepší možností pro dedikované servery. [52]

Linux Fedora Server nabízí velice rychlé načtení celého serveru, možnost instalace na všechny webové servery typu Apache, dále podporu 3 D grafického rozhraní, automatické aktualizace, a hlavně vysokou úroveň bezpečnosti v případě, že je postupováno podle oficiálních dokumentací. [52]

4.11 Docker

Docker je Open source projekt, který se využívá pro automatizované nasazení aplikací. Aplikace a její části jsou uloženy do balíčku (kontejneru), který lze následně spouštět jako celek. Mezi velkou výhodu Dockeru patří také to, že je multiplatformní a lze spouštět lokálně i v cloudu. [53][54]

Stejně tak, jako přepravní kontejnery umožnují, aby se zboží přepravilo na určité místo bez ohledu na obsah kontejneru, fungují stejně i Docker kontejnery, které mají za úkol spustit/nasadit jakýkoliv kontejner bez ohledu na kód a závislosti vně balíčku. Podstatnou výhodou je také to, že se díky kontejnerům aplikace dokážou izolovat na stejném operačním systému a to, protože běží nad hostitelem systému. Díky tomu mají mnohem menší nároky na daný server nebo virtuální počítač. [53][54]

Zároveň je každý kontejner schopný spustit celou webovou aplikaci, a to za pomocí Docker Compose souboru, kde je předem nakonfigurované, v jakém pořadí se budou dané kontejnery spouštět. [53]

4.11.1 Docker Compose

Jedná se o nástroj pro běh a zprávu Docker kontejnerů. Využívá YAML souborů proto, aby nakonfiguroval jednotlivé aplikace, díky čemuž je poté vytvořen jediný kontejner, který je spustitelný pouze jedním příkazem. [53][54]

II. PRAKTICKÁ ČÁST

5 NÁVRH CHATBOTA

V této kapitole je uveden návrh chatbota. Během návrhu probíhal sběr a analýza požadavků, došlo k definování funkčních a nefunkčních požadavků, a také k vytvoření diagramu případů užití s jednotlivými scénáři.

5.1 Souhrnný popis funkcí chatbota

Kapitola se zaobírá návrhem funkcí aplikace. Jsou zde podrobně popsány všechny funkcionality, které byly předem konzultovány s vedoucím Programming Support Centre. Tyto funkce byly navrženy tak, aby byly co nejjednodušší pro budoucí uživatele.

5.1.1 Dotaz

Jedná se o funkcionalitu, kde uživatel pomocí jednoduchého tlačítka bude moci položit dotaz, který následně bude přeposlán na email, který bude sloužit pro správu aplikace. Po stisknutí tlačítka by se uživateli mělo vyvolat jednoduché kontextové menu, kde v případě, že je nově příchozí uživatel (není v systému), bude požádán o to, aby zadal emailovou adresu, na kterou si přeje poslat odpověď na daný dotaz. V případě, že uživatel je v systému již zaveden, bude informován o tom, jaký email má v systému uložen a jestli si na něj přeje poslat i danou odpověď. V případě zvolení tlačítka „Ano“ bude vybrán email ze systému. V opačném případě bude uživatel vyzván k zadaní emailu, který se následně přepíše i v systému pro případ, že by uživatel znovu pokládal dotaz.

5.1.2 FAQ

Programming Support Centre, stejně tak jako ostatní organizace, se každý den setkává s tím, že jej objevují potenciálně noví uživatelé, kteří potřebují znát informace o projektu a organizaci, i proto by zde měla být implementována funkce pro často kladené otázky (FAQ).

Tato funkcionalita zobrazí uživateli odpovědi na základní otázky typu: „Kde nás najdete?“ „K čemu slouží Programming Support Centre?“, „Má Programming Support Centre webovou stránku?“ aj.

Konkrétní implementace by měla být schopná posílat jak textové, tak obrázkové odpovědi (mapy). FAQ by uživateli měly být nabídnuty jako sada tlačítek, kde by po kliknutí na příslušné tlačítko byla uživateli poslána odpověď do chatu, a to buď v textové nebo obrázkové podobě. Poté by mělo být uživateli opět navráceno úvodní menu START.

5.1.3 Doučování

Návrh funkce Doučování lze rozdělit na tři základní části. První část by měla uživateli nabídnout možnost výběru předmětů a konkrétních tutorů, kteří jsou aktuálně dostupní. Tyto informace/data budou dynamicky generovat GUI z databáze, a to přímo do chatu uživatele.

Druhá část této funkcionality by se měla týkat rezervace konkrétního termínu u předem vybraného tutora. Uživateli by měly být nabídnuty konkrétní volné dny, kdy se tutor v Programming Support Centre nachází a má volné místo. Všechny tyto termíny by se měly uživateli zobrazit na základě dat uložených v databázi. Funkce by měla být schopná vracet volné a aktuální termíny. V případě, že by si uživatel vybral konkrétní termín, by se měl navýšit počet obsazení, aby v případě, kdyby se termín naplnil, nebyl nabízen ostatním studentům.

Poslední část funkce Doučování by se měla týkat informování Programming Support Centre prostřednictvím emailu o tom, že se student zaregistroval na konkrétní termín. Email by měl obsahovat datum a čas termínu, předmět, o který se jedná, jméno a příjmení konkrétního tutora a email, který si při registraci termínu zvolil daný student.

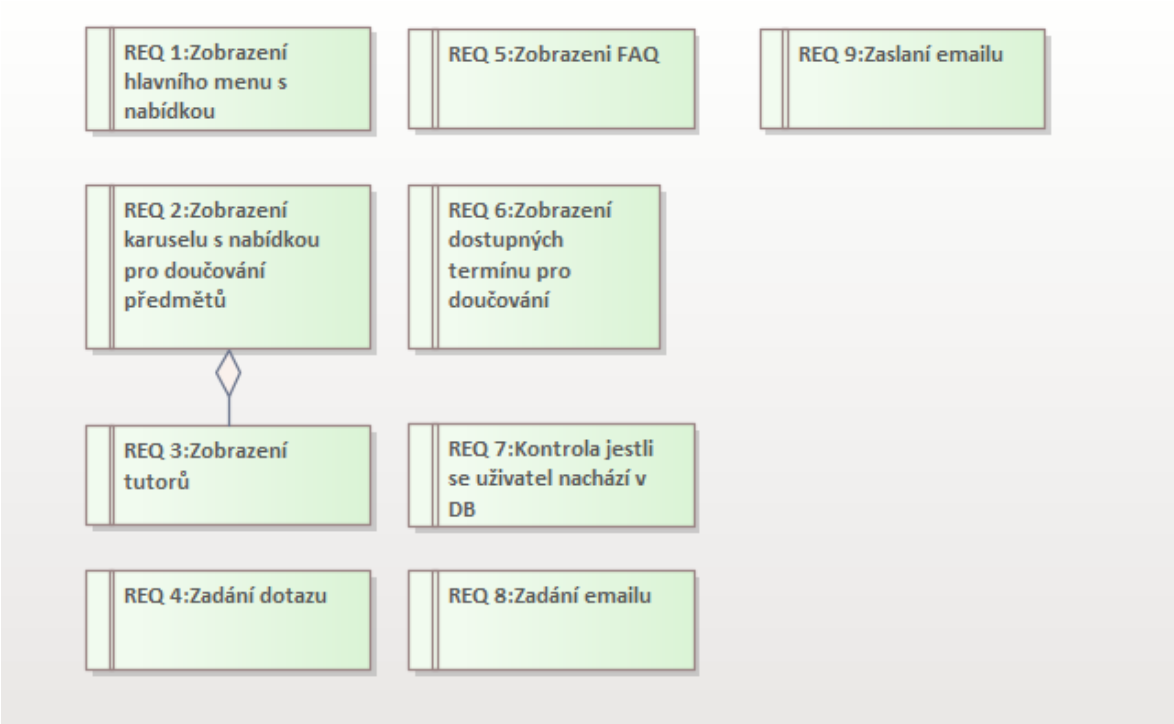
Aplikace tedy musí být schopna dynamicky generovat prvky GUI na základě dat získaných z databáze. Proto musí být implementace funkcí provedena obecně (je nutné se co nejvíce vyvarovat příliš specifických operací a konstant). Tento přístup má výhodu ve flexibilitě aplikace, jejíž obsah tak může být měněn bez nutnosti nasazení nové verze.

5.1.4 Start

Mělo by se jednat o základní funkcionalitu, která uživatele provede celou konverzací a bude pro něj sloužit i jako odrazový můstek k další interakci. Tato funkcionalita by měla uživateli nabídnout „most“ pro spojení ostatních funkcí jako je Dotaz, Doučování a FAQ. Start menu by mělo být nabídnuto uživateli při první interakci s chatbotem, nebo po ukončení každého požadavku. Zároveň by uživatel měl být poučen, jak toto menu vyvolat, ideálně klíčovým slovem START.

5.2 Funkční požadavky

V aplikaci existuje devět různých funkčních požadavků. Všechny tyto požadavky popisují kompletní funkcionalitu aplikace a jejich grafickou podobu znázorňuje Obrázek 8.



Obrázek 8. Funkční požadavky

REQ1 Zobrazení hlavního menu s nabídkou:

Jedná se o funkční požadavek na zobrazení hlavního menu s nabídkou. Uživateli se menu buď zobrazí automaticky při prvním přístupu, anebo musí být informován, jak jej zobrazit. Menu bude obsahovat minimálně 2 tlačítka: Dotaz a Doučování. Pro toto menu bude také uveden titulek, např. „Vyber si, jak začít“.

REQ2 Zobrazení karuselu s nabídkou pro doučování předmětů:

Jedná se o funkční požadavek na zobrazení karuselu s nabídkou pro doučování předmětů. Uživateli se menu zobrazí po kliknutí na tlačítko „Doučování“. Menu bude obsahovat tlačítka předmětů a tlačítka tutorů tak, jak je popsáno v navazujícím požadavku REQ3 Zobrazení tutorů. Uživatel Obdrží zprávu, ve které bude zobrazen karusel s jednotlivými předměty a tutory, kteří doučují daný předmět.

REQ3 Zobrazení tutorů:

Tento funkční požadavek slouží k rozšíření REQ2. Uživateli se bude u jednotlivých předmětů na doučování zobrazovat seznam dostupných tutorů.

REQ4 Zadaní dotazu:

Jedná se o funkční požadavek pro zadání dotazu. Uživateli se jednotlivé kroky zobrazí automaticky při kliknutí na tlačítko „Dotaz“ z hlavního menu. Uživatel bude muset splnit předem

danou sekvenci požadavků proto, aby se dokončilo podání dotazu, a dotaz byl přeposlán na PSC email.

REQ5 Zobrazení FAQ:

Jedná se o funkční požadavek pro zobrazení FAQ. Uživateli se zobrazí menu s FAQ, a to vždy při zaslání hlavního menu. Toto menu by pro uživatele mělo být také interaktivní a mělo by poskytovat základní informace o PSC.

REQ6 Zobrazení dostupných termínů pro doučování:

Jedná se o funkční požadavek pro zobrazení dostupných termínů pro doučování po vybrání konkrétního tutora v REQ2. Následně se uživateli zobrazí nejbližší tři dostupné termíny doučování, které bude tutor nabízet.

REQ7 Kontrola, jestli se uživatel nachází v DB:

Jedná se o funkční požadavek pro kontrolu, jestli se uživatel nachází v databázi. Systém by měl zkontrolovat, jestli se uživatel nachází v DB dle jeho ID, které je mu automaticky přiřazeno Facebookem. Na základě této kontroly by měl program být schopný vrátit, zda je uživatel aplikaci znám anebo je nově příchozí.

REQ8 Zadání Emailu:

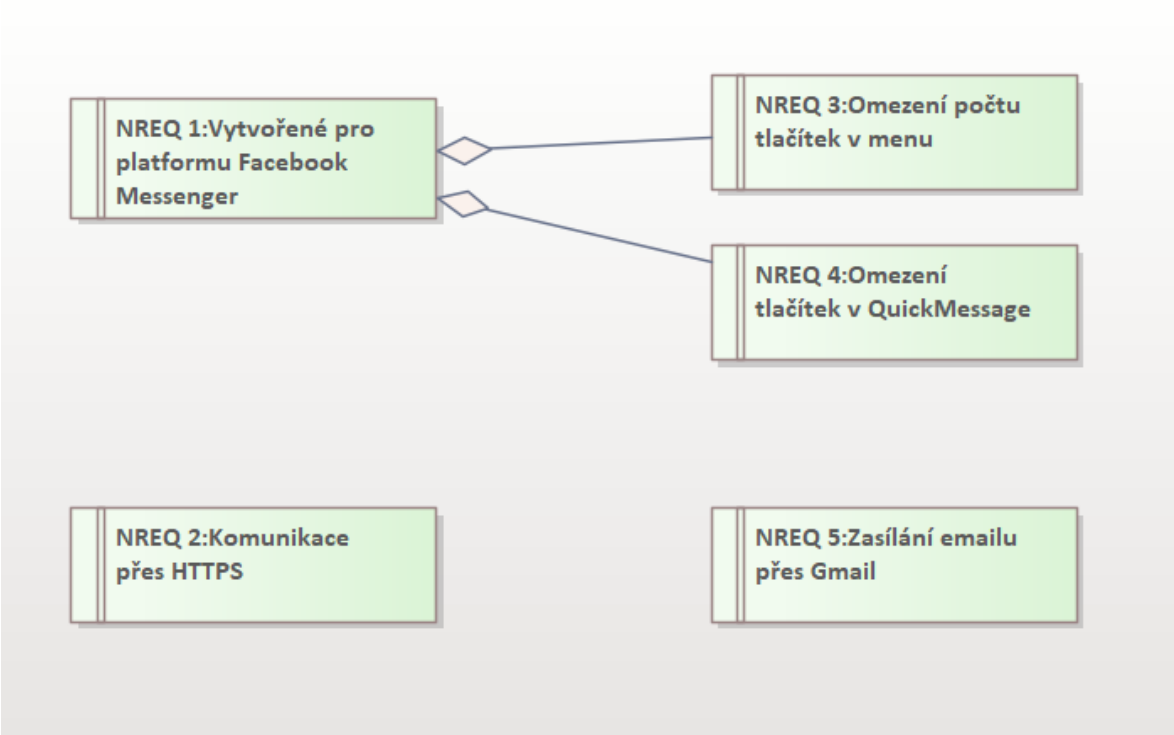
Jedná se o funkční požadavek zadání emailu. Systém by měl vyzvat uživatele, aby zadal svou emailovou adresu v případě, že je uživatel nově příchozí. Toto vyzvání by měl systém provést v případě REQ2 a REQ4.

REQ9 Zaslání Emailu:

Jedná se o funkční požadavek pro zaslání emailu. Systém by měl zpracovat email, který je zadán pomocí REQ8. Po zpracování, by měl být tento email přeposlán na PSC dle konkrétního požadavku.

5.3 Nefunkční požadavky

V aplikaci existuje pět nefunkčních požadavků. Všechny tyto nefunkční požadavky popisují omezení systému. Jejich grafickou podobu znázorňuje Obrázek 9.



Obrázek 9. Nefunkční požadavky

NREQ1 Vytvořené pro platformu Facebook Messenger:

Jedná se o omezení systému, protože je aplikace vytvořena na platformě Facebook Messenger. Kvůli tomuto omezení zde vznikají další nefunkční požadavky, které jsou popsány v NREQ3 a NREQ4

NREQ2 Komunikace přes HTTPS:

Vzhledem k tomu, že je systém vytvořen na platformu Facebook Messenger je nutno používat zabezpečenou doménu na přijímání webhooků.

NREQ3 Omezení počtu tlačítek v menu:

Z důvodu omezení na platformě Facebook Messenger je možno uvést pouze tři tlačítka do každého menu, které je uživateli posláno.

NREQ4 Omezení tlačítek v QuickMessage:

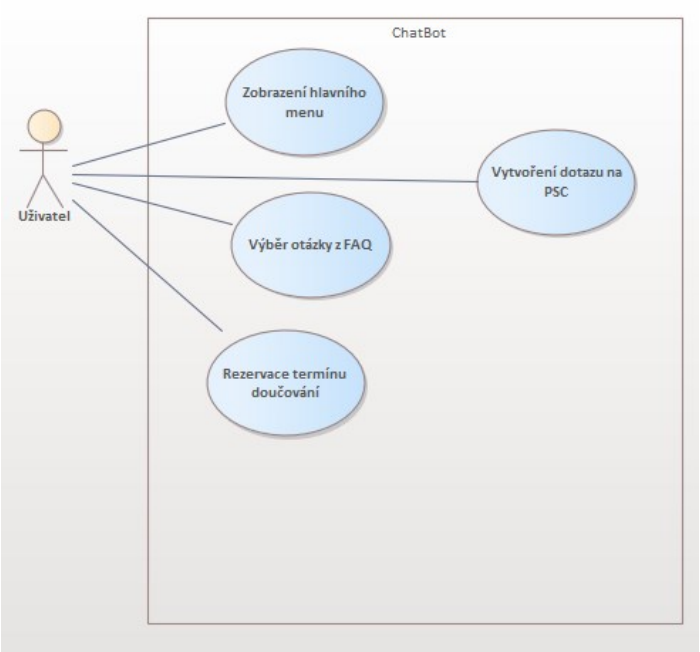
Z důvodu omezení na platformě Facebook Messenger je možno uvést třináct tlačítek do každé QuickMessage, která je uživateli poslána.

NREQ5 Zasílání emailu přes Gmail:

Omezení systému spočítá ve využitém SMTP klientu, který je součástí Pythonu. PSC email je veden na platformě Gmail i z tohoto důvodu je zde zvoleno posílání emailů přes tuto platformu, respektive přes port 587.

5.4 Případy užití

Obrázek 10 zobrazuje diagram případu užití. V tomto případě zde máme čtyři hlavní případy užití. Jedná se o situaci, kdy si uživatel zobrazí hlavním menu a interaguje s ním tak, jak ukazuje Tabulka 1.



Obrázek 10. Use Case

Tabulka 1: Use Case Zobrazení hlavního menu

<table>
<tr><td colspan="3">Název: Zobrazení hlavního menu</td></tr>
<tr><td colspan="3">ID: UC001</td></tr>
<tr><td colspan="3">Charakteristika:

Uživatel zobrazuje hlavní menu</td></tr>
<tr><td colspan="3">Primární aktér:

Uživatel</td></tr>
<tr><td colspan="3">Vedlejší aktéři:

Nejsou</td></tr>
<tr><td colspan="3">Vstupní podmínky:

Uživatel musí vlastnit Messenger účet</td></tr>
<tr><td colspan="3">Výstupní podmínky:

Není</td></tr>
<tr><td colspan="3">Hlavní scénář:</td></tr>
<tr><td>Krok</td><td>Aktér/Sys-tém</td><td>Popis</td></tr>
<tr><td>1</td><td>Uživatel</td><td>Uživatel otevře Facebook Messenger</td></tr>
<tr><td>2</td><td>Uživatel</td><td>Uživatel napíše slovo Start nebo zvolí Get Started</td></tr>
<tr><td>3</td><td>Systém</td><td>Je zobrazeno hlavní menu Start/Doučování/Konec</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3">Alternativní scénáře:

Žádné</td></tr>
</table>

Poté se zde nachází use case vytvoření dotazu na PSC i s její kompletní obsluhou tak, jak popisuje Tabulka 2, Tabulka 3 a Tabulka 4

Tabulka 2: Use Case Vytvoření dotazu na PSC

<table>
<tr><td colspan="3">Název: Vytvoření dotazu na PSC</td></tr>
<tr><td colspan="3">ID: UC002</td></tr>
<tr><td colspan="3">Charakteristika:
Uživatel posílá dotaz na PSC</td></tr>
<tr><td colspan="3">Primární aktér:
Uživatel</td></tr>
<tr><td colspan="3">Vedlejší aktéři:
Nejsou</td></tr>
<tr><td colspan="3">Vstupní podmínky:
Uživatel musí vlastnit Messenger účet, Uživatel musí vlastnit Email.</td></tr>
<tr><td colspan="3">Výstupní podmínky:
Není</td></tr>
<tr><td colspan="3">Hlavní scénář:</td></tr>
<tr><td>Krok</td><td>Aktér/Sys-tém</td><td>Popis</td></tr>
<tr><td>1</td><td>Uživatel</td><td>Uživatel klikne na tlačítko Dotaz</td></tr>
<tr><td>2</td><td>Systém</td><td>Systém vyzve uživatele, proto aby napsal dotaz</td></tr>
<tr><td>3</td><td>Uživatel</td><td>Uživatel napíše dotaz</td></tr>
<tr><td>4</td><td>Systém</td><td>Systém zkontroluje, jestli se uživatel nachází v sys-tému</td></tr>
<tr><td>5</td><td>Systém</td><td>Systém zpracuje dotaz a vyzve uživatele, aby zadal email</td></tr>
</table>

6	Uživatel	Napíše svůj vlastní email
7	Systém	Systém zpracuje email a odešle jej na PSC email
8	Systém	Systém zobrazí uživateli hlavní menu
Alternativní scénáře: UC002a, UC002b		

Tabulka 3: Alternativní scénář existující email

Název – Alternativní scénář: Existující uživatel		
ID: UC002a		
Charakteristika: Tento alternativní scénář může nastat v kroku 4. Zachycení alternativního toku případu užití v případě, že je již uživatel v systému a přeje si rezervaci na email, který je uložen v systému.		
Alternativní scénář:		
Krok	Aktér/Systém	Popis
1	Systém	Systém rozpozná uživatele
2	Systém	Systém se zeptá uživatele, jestli si přeje použít jeho uložený email
3	Uživatel	Uživatel zvolí, že chce použít existující email
4	Systém	Zpracuje email a odešle jej na PSC email
5	Systém	Pošle uživateli hlavní menu

Tabulka 4: Alternativní scénář existující email změna

Název – Alternativní scénář: Existující uživatel změna emailu		
ID: UC002b		
Charakteristika: Tento alternativní scénář může nastat v kroku 4. Zachycení alternativního toku případu užití v případě, že uživatel, který je již v systému si přeje rezervaci na jiný email.		
Alternativní scénář:		
Krok	Aktér/Systém	Popis
1	Systém	Systém rozpozná uživatele
2	Systém	Systém se zeptá uživatele, jestli si přeje použít jeho uložený email
3	Uživatel	Uživatel zvolí, že chce nechce použít existující mail
4	Systém	Poprosí uživatele o zadaní emailu
5	Uživatel	Zadá nový email
6	Systém	Uloží email ke konkrétnímu uživateli
7	Systém	Zpracuje email a odešle jej na PSC email

Jako předposlední use case se zde nachází Výběr otázky z FAQ, což zahrnuje kompletní výběr a zobrazení často kladených otázek tak, jak ukazuje Tabulka 5 a Tabulka 6.

Tabulka 5: Use Case Výběr otázky z FAQ

<table>
<tr><td colspan="3">Název: Výběr otázky z FAQ</td></tr>
<tr><td colspan="3">ID: UC003</td></tr>
<tr><td colspan="3">Charakteristika:

Uživatel vybírá otázku z FAQ</td></tr>
<tr><td colspan="3">Primární aktér:

Uživatel</td></tr>
<tr><td colspan="3">Vedlejší aktéři:

Nejsou</td></tr>
<tr><td colspan="3">Vstupní podmínky:

Uživatel musí vlastnit Messenger účet</td></tr>
<tr><td colspan="3">Výstupní podmínky:

Není</td></tr>
<tr><td colspan="3">Hlavní scénář: Výběr otázky z FAQ</td></tr>
<tr><td>Krok</td><td>Aktér/Sys-tém</td><td>Popis</td></tr>
<tr><td>1</td><td>Uživatel</td><td>Uživatel napíše slovo Start nebo zvolí Get Started</td></tr>
<tr><td>2</td><td>Systém</td><td>Systém zobrazí menu s FAQ otázkami</td></tr>
<tr><td>3</td><td>Uživatel</td><td>Uživatel klikne na jednu z FAQ otázek</td></tr>
<tr><td>4</td><td>Systém</td><td>Systém zobrazí textovou odpověď na FAQ otázku</td></tr>
<tr><td>5</td><td>Systém</td><td>Systém zobrazí uživateli hlavní menu a FAQ otázky</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3">Alternativní scénáře:

UC003a</td></tr>
</table>

Tabulka 6: Alternativní scénář FAQ „Kde se nacházíme“

Název – Alternativní scénář: FAQ „Kde se nacházíme?“		
ID: UC003a		
Charakteristika: Tento alternativní scénář může nastat v kroku 3. Zachycení alternativního toku případu užití, v případě že uživatel zvolí FAQ „Kde se nacházíme?“		
Alternativní scénář:		
Krok	Aktér/Systém	Popis
1	Uživatel	Zvolí FAQ „Kde se nacházíme?“
2	Systém	Systém pošle uživateli mapu a konkrétní cestu v obrázkové podobě
3	Systém	Systém zobrazí uživateli hlavní menu a FAQ otázky

Poslední use case v aplikaci je Rezervace termínu doučování, ve kterém je detailně popsáno, jak taková rezervace probíhá i s alternativními scénáři, které nám ukazuje Tabulka 7, Tabulka 8 a Tabulka 9

Tabulka 7: Use Case Rezervace termínu doučování

Název: Rezervace termínu doučování		
ID: UC004		
Charakteristika: Uživatel vybírá termín na doučování		
Primární aktér: Uživatel		
Vedlejší aktéři: Nejsou		
Vstupní podmínky: Uživatel musí vlastnit Messenger účet, Uživatel musí vlastnit Email		
Výstupní podmínky: Není		
Hlavní scénář:		
Krok	Aktér/Systém	Popis
1	Uživatel	Uživatel klikne na možnost doučování
2	Systém	Systém zobrazí karusel s jednotlivými předměty a tutory, které jsou k nabídce na doučování.
3	Uživatel	Vybere konkrétní předmět a tutora o kterého má zájem
4	Systém	Systém zobrazí uživateli tři termíny, které nejsou plné a jsou aktuální.
5	Uživatel	Vybere si konkrétní termín, který mu vyhovuje.
6	Systém	Systém vyžádá od uživatele email, na který si přeje rezervovat termín
7	Uživatel	Zadá email, který chce použít

8	Systém	Systém potvrdí registraci na konkrétní termín, a zvýší obsazenost tohoto termínu v databázi.
9	Systém	Systém zašle email na PSC o tom, že se uživatel zaregistroval na termín.
10	Systém	Systém zašle uživateli hlavní menu
Alternativní scénáře: UC004a, UC004b		

Tabulka 8: Alternativní scénář Tutor nemá volný termín

Název – Alternativní scénář: Tutor nemá volný termín		
ID: UC004a		
Charakteristika: Tento alternativní scénář může nastat v kroku 3. Zachycení alternativního toku případu užití, kdy tutor, kterého si uživatel vybral nemá volný termín		
Alternativní scénář:		
Krok	Aktér/Systém	Popis
1	Uživatel	Zvolí tutora, který již nemá volný termín.
2	Systém	Systém řekne uživateli, ať si zvolí nějaký jiný termín, protože tutor nemá žádný volný
3	Systém	Systém zobrazí uživateli karusel s jednotlivými předměty a tutory, které jsou k nabídce na doučování.

Tabulka 9: Alternativní scénář Uživatel si tento termín již zarezervoval

<table>
<tr><td colspan="3">Název – Alternativní scénář: Uživatel si tento termín již zarezervoval</td></tr>
<tr><td colspan="3">ID: UC004b</td></tr>
<tr><td colspan="3">Charakteristika:

Tento alternativní scénář může nastat v kroku 7. Zachycení alternativního toku případu užití, kdy si uživatel chce zarezervovat termín, na který je již přihlášený.</td></tr>
<tr><td colspan="3">Alternativní scénář:</td></tr>
<tr><td>Krok</td><td>Aktér/Systém</td><td>Popis</td></tr>
<tr><td>1</td><td>Uživatel</td><td>Vybere si konkrétní termín, který mu vyhovuje.</td></tr>
<tr><td>2</td><td>Systém</td><td>Systém vyžádá od uživatele email, na který si přeje rezervovat termín</td></tr>
<tr><td>3</td><td>Uživatel</td><td>Zadá email, který chce použít</td></tr>
<tr><td>4</td><td>Systém</td><td>Systém zobrazí uživateli zprávu, že tento termín má již rezervovaný</td></tr>
<tr><td>5</td><td>Systém</td><td>Systém zobrazí uživateli karusel s jednotlivými předměty a tutory, které jsou k nabídce na doučování.</td></tr>
<tr><td colspan="3"></td></tr>
</table>

6 POŽADAVKY NA DATABÁZI A JEJÍ NÁVRH

Pro ukládání dat byl vybrán MSSQL Server, který bude spravován pomocí Microsoft SQL Server Management Studio 18. Pro správu a využívání databázových příkazů v aplikaci byla zvolena Python knihovna Pyodbc.

6.1 Požadavky na databázi

Na základě požadavků na aplikaci byla navržena databáze, která obsahuje:

1. Tabulku předmětů, které se budou vyučovat v Programming Support Centre
2. Tabulku tutorů, kteří budou vyučovat v Programming Support Centre
3. Tabulku s rezervacemi studentů na termíny
4. Tabulku termínů konzultací
5. Tabulku často kladených otázek (FAQ)
6. Tabulku jednotlivých dat se začátkem a koncem konzultace a počtem obsazení

6.2 Návrh databáze

Cílem bylo navrhnout fyzický model databáze na základě předchozích požadavků, vytvořit tabulky, zpracovat jednotlivé požadavky na databázi a vypracovat funkční relační spojení mezi jednotlivými tabulkami. V návrhu je popsána struktura databáze, a také vazby mezi jednotlivými tabulkami.

6.2.1 Identifikace relací

V této části můžeme pozorovat všechny tabulky, se kterými bude aplikace pracovat viz Tabulka 10.

Tabulka 10. Seznam Tabulek

Název Tabulky	Popis
Buttons	Seznam jednotlivých předmětů, které se budou vyučovat v PSC
Učitelé	Seznam jednotlivých tutorů, kteří budou učit v PSC
Uživatelé	Seznam jednotlivých uživatelů, kteří budou navštěvovat PSC
Uživatelé_Kalendář	Propojovací tabulka
Kalendář	Seznam jednotlivých termínů s časem začátku a konce
Propojení	Propojovací tabulka Buttons,Kalendar,Ucitele
FAQ	Seznam často kladených otázek (FAQ)

Tabulka Buttons

Tato tabulka slouží primárně pro vytváření tlačítek pro aplikační karusel a k naplnění parametrů funkce send_button_message. V tabulce Buttons se nachází Name_of_button, který definuje názvy jednotlivých tlačítek/předmětů. Dále se zde nachází Image_url, kde se ukládá odkaz na fotku, který se zobrazí u předmětů/tlačítek. Řádek SubTitle slouží k tomu, aby definoval popis konkrétního tlačítka/předmětu. Poté zde lze nalézt také unikátní identifikační klíč Button_ID, který má nastavenou auto inkrementaci. Jako poslední část tabulky zde lze nalézt řádky PayLoad[1-3], do kterého je definován příslušný payload proto, aby aplikace přiřadila konkrétního tutora k vybranému předmětu. Celý tento popis uvádí Tabulka 11.

Tabulka 11. Tabulka Buttons

Název tabulky:	**Buttons**		
Integrita	**Název**	**Typ**	**Popis**
	Name_of_button	Nchar (50)	Název Předmětů/Tlačítek
	Image_url	Nchar (250)	Obrázek tlačítka/předmětu v aplikaci
Primární klíč	Button_ID	Int	Unikátní identifikační klíč
	SubTitle	Nchar (100)	Popisek předmětu/tlačítka
	PayLoad[1]	Nchar (100)	Aplikační payload na reakci
	PayLoad[2]	Nchar (100)	Aplikační payload na reakci
	PayLoad[3]	Nchar (100)	Aplikační payload na reakci

Tabulka Učitelé

Tabulka Učitelé slouží pro uchovávání konkrétních tutorů, kteří vyučují v Programming Support Centre. Nachází se zde sloupec s názvem ID_Učitele, který slouží pro unikátní identifikaci. Tento sloupec má nastavenou auto inkrementaci. Dále se zde nachází Jméno a Příjmení učitele s datovým typem Nchar nastaveným na maximální velikost 10 znaků. Poslední sloupec tabulky Učitelé je Titul. Ten slouží k nastavení titulu konkrétního tutora. Pokud se nechá sloupec nevyplněný je automaticky nastaven na NULL. Celý tento popis zobrazuje Tabulka 12.

Tabulka 12. Tabulka Učitelé

Název tabulky:	Učitelé		
Integrita	**Název**	**Typ**	**Popis**
Primární klíč	ID_Ucitele	Int	Unikátní identifikační klíč
	Jméno	Nchar (10)	Definuje jméno učitele
	Příjmení	Nchar (10)	Definuje příjmení učitele
	Titul	Nchar (10)	Definuje titul učitele

Tabulka Uživatelé

Tato tabulka je vytvořena primárně pro ukládání základních informací o uživatelích. Tyto informace jsou poté využity pro vytvoření uživatelského pohodlí. Tabulka Uživatelé obsahuje sloupec ID_Uživatele, který slouží pro unikátní identifikaci a má nastavenou auto inkrementaci. Poté se zde nachází sloupec FacebookID, který slouží pro identifikaci uživatele. Každý uživatel má toto ID předem přidělené při vytváření svého Facebook účtu. Poslední sloupec v této relaci slouží pro ukládání uživatelského emailu, a to hlavně z toho důvodu, aby mohla aplikace znovu používat email u již známých uživatelů a tím pádem nemusel uživatel psát svůj email znovu. Celý tento popis uvádí Tabulka 13.

Tabulka 13. Tabulka Uživatelé

Název tabulky:	Uživatelé		
Integrita	**Název**	**Typ**	**Popis**
Primární klíč	ID_Uzivatele	Int	Unikátní identifikační klíč
	FacebookID	Nchar (150)	Uživatelské ID, které poskytuje platforma FB messenger
	Email	Nchar(50)	Uživatelský email

Tabulka Uživatelé_Kalendář

Tabulka Uživatelé_Kalendář slouží pouze jako pomocná tabulka pro propojení mezi ostatními relacemi. Obsahuje pouze dva základní sloupce. První sloupec je ID_Kalendar (cizí klíč), který slouží proto, aby se tabulka byla schopná spojit s relací propojení. Druhý sloupec je ID_Uživatele (cizí klíč), který slouží proto, aby se tabulka byla schopná spojit s relací uživatele. Prakticky se dá říct, že vytváří most mezi relací Uživatele a zbytkem databáze. Konstrukci popisuje Tabulka 14.

Tabulka 14. Tabulka Uživatelé_Kalendář

Název Tabulky:	**Uživatelé_Kalendář**		
Integrita	**Název**	**Typ**	**Popis**
Cizí klíč	ID_Kalendar	Int	ID na propojeni k relaci Propojení
Cizí klíč	ID_Uživatele	Int	ID na propojeni k relaci Uživatelé

Tabulka Kalendář

Jednou z důležitých relací databáze je tabulka s názvem Kalendář. Slouží k tomu, aby ukládala všechny dostupné termíny k doučování a hlídala jejich obsazenost. První sloupec v tabulce je ID_Kalendar. Tento sloupec má nastavenou automatickou inkrementaci a je označen jako primární klíč. Poté se zde nachází sloupec Datum, který slouží proto, aby zaznamenával konkrétní datum, kdy má probíhat doučování. Tento sloupec má formát date a ten je uveden jako „YYYY-MM-DD“. Poté se zde nachází dva důležité sloupce a to Začátek_Konzultace a Konec_Konzultace. Oba tyto sloupce jsou uvedeny ve formátu Time a slouží proto, aby hlídaly, kdy má konzultace začít a kdy skončit. Další sloupec, který se zde nachází je Obsazenost, který kontroluje aktuální obsazenost jednotlivých termínů. Na tuto relaci Obsazenost navazuje další sloupec s názvem Max_Obsazenost, který uvádí, jaký je maximální počet studentů na termínu. Poslední sloupec má název PayLoad a slouží k tomu, aby aplikace byla schopná rozeznat jaký termín si uživatel vybral. Konstrukce tabulky Kalendář demonstruje Tabulka 15.

Tabulka 15. Tabulka Kalendář

Název Tabulky:	**Kalendář**		
Integrita	**Název**	**Typ**	**Popis**
Primární klíč	ID_Kalendar	Int	Unikátní identifikační klíč
	Datum	Date	Datum konzultace
	Začátek_Konzultace	Time (7)	Začátek konzultace (čas)
	Konec_Konzultace	Time (7)	Konec konzultace (čas)
	Obsazenost	Int	Aktuální obsazenost termínu
	Max_Obsazenost	Int	Maximální počet studentů
	PayLoad	Nchar (255)	Aplikační payload na reakci

Tabulka Propojení

Jedná se o klíčovou relaci, která slouží k tomu, aby propojila celou databázi dohromady a umožnila ostatním relacím komunikovat mezi sebou. První sloupec v této relaci je Button_ID. Je nastaven jako cizí klíč a slouží pouze k tomu, aby propojoval tabulku Propojení s tabulkou Buttons. Poté se zde nachází sloupec ID_Učitele a ID_Kalendar, které jsou rovněž nastavené jako cizí klíč a propojují se s tabulkou Učitelé a s tabulkou Propojení. Poslední sloupec s názvem ID je nastaven jako primární klíč a slouží k tomu. aby zajistil Integritu tabulky. Celý návrh této tabulky zobrazuje Tabulka 16.

Tabulka 16. Tabulka Propojení

Název tabulky:	**Propojení**		
Integrita	**Název**	**Typ**	**Popis**
Cizí Klíč	Button_ID	Int	ID na propojeni k relaci Buttons
Cizí Klíč	ID_Učitele	Int	ID na propojeni k relaci Učitelé
Cizí Klíč	ID_Kalendar	Int	ID na propojeni k relaci Propojení
Primární klíč	ID	Int	Unikátní identifikační klíč

Tabulka FAQ

Poslední tabulka v databázovém systému je tabulka s názvem FAQ. Tato tabulka nemá žádné propojení v databázi a slouží pouze k tomu, aby uchovávala data, která se následně používají pro funkci často kladené otázky v aplikaci.

Tabulka je rozdělená na čtyři jednotlivé sloupce. Prvním z nich je sloupec Title, kde se vkládají konkrétní otázky, respektive jejich znění. Dalším sloupcem je Payload, ten slouží k tomu, aby aplikace věděla, na kterou otázku má reagovat a k jaké otázce přiřadit konkrétní odpověď. Jako další sloupec je ImageURL, který určuje, jaký obrázek/miniatura bude přiražena ke konkrétní otázce (v tomto případě jsou nastavené červené kruhy). Jako poslední sloupec se v relaci nachází Reply, do kterého se vkládá odpověď na otázku. Návrh této tabulky zobrazuje Tabulka 17.

Tabulka 17. Tabulka FAQ

Název tabulky:	**FAQ**		
Integrita	**Název**	**Typ**	**Popis**
	Title	Nchar (21)	Text Otázky
	Payload	Nchar (20)	Aplikační payload na reakci
	ImageURL	Nchar (250)	Odkaz na obrázek k otázkám
	Reply	Nchar (500)	Odpověď na otázku

6.2.2 Vztahy mezi tabulkami

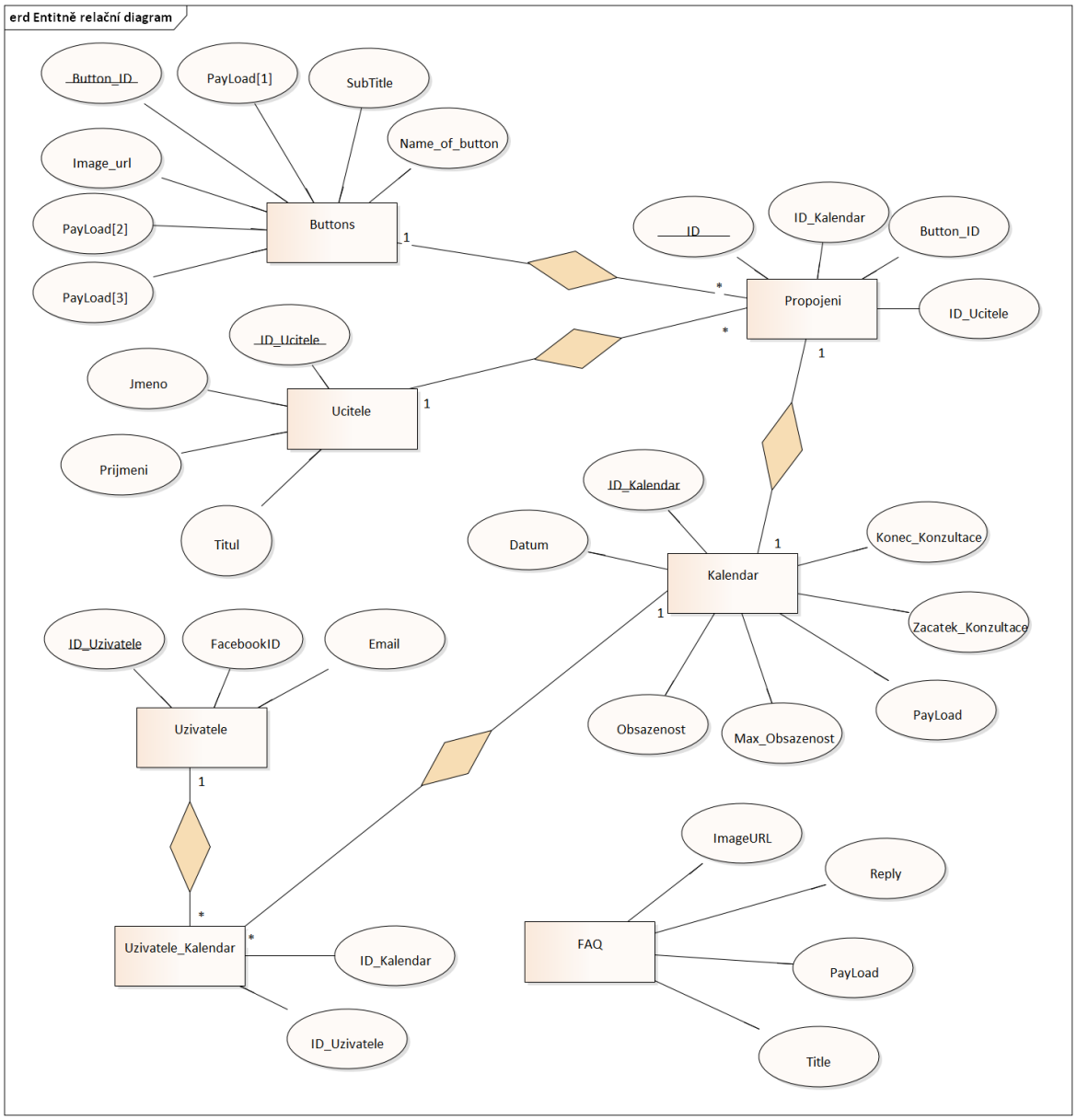
Tabulka 18 zobrazuje jednotlivé vztahy mezi tabulkami v databázi.

Tabulka 18. Vztahy mezi tabulkami

Tabulky	Vztah	Popis
Buttons-Propojení	1: N	Tabulka Buttons je napojena na propojovací tabulku Propojení
Ucitele-Propojení	1: N	Tabulka Ucitele je napojena na propojovací tabulku Propojení
Propojení-Kalendář	1: 1	Tabulka Propojení je napojena na tabulku Kalendář
Kalendář-Uživatele Kalendář	1: N	Tabulka Kalendář je napojena na tabulku Uživatele Kalendář
Uživatele Kalendář-Uživatele	N:1	Tabulka Uživatele Kalendář je napojena na tabulku Uživatele
FAQ	0	Tato tabulka nemá žádné vazby

6.2.3 Entitně relační diagram databáze

V ER diagramu můžeme vidět všechny možné vazby mezi jednotlivými tabulkami v databázi. Dále jsou zde také vidět cizí a primární klíče, které se nachází v databázi. Tyto vazby a tabulky ukazuje Obrázek 11.



Obrázek 11. ERD diagram

7 VYTVOŘENÍ CHATBOTA

7.1 Implementace funkcionalit

Cílem této kapitoly je ukázat implementaci funkcí, které byly popsány v kapitole **Chyba! Nenalezen zdroj odkazů.**. Bude zde popsána konkrétní podoba výsledných funkcí, včetně použitých metod, které nabízí Facebook Messenger.

7.1.1 Start

Finální podoba funkce Start je založena na metodě send_button_message z knihovny PyMessenger. Funkce vyžaduje tři povinné parametry, kde první parametr je recipient_id. Jedná se o unikátní identifikační číslo, které je přiřazeno každému uživateli aplikace Facebook Messenger.

Druhý povinný parametr této funkce je text neboli zpráva, kterou chceme uživateli zaslat v rámci zprávy s tlačítky. V našem případě se jedná o text „Vyber si, jak chceš začít“.

Poslední parametr funkce je nejsložitější, protože vyžaduje Button Template, jeho struktura je přesně definována platformou Facebook Messenger. V našem případě se jedná o tři tlačítka typu postback, s title (Doučování, Dotaz a Konec). Popis Button Template znázorňuje Obrázek 12.

```
Start_Vyber_button = [{'type': 'postback', 'title': 'Doučování', 'payload': '0x00000002'},
                      {'type': 'postback', 'title': 'Dotaz', 'payload': '0x00000003'},
                      {'type': 'postback', 'title': 'Konec', 'payload': '0x00000001'}]
```

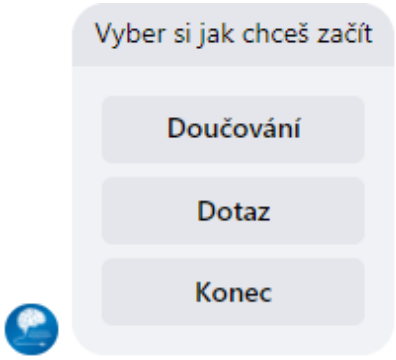
Obrázek 12. Button Template Start

Funkce se zavolá pouze v případech, kdy uživatel poprvé interaguje s chatbotem, anebo dokončí jakoukoliv jinou sekvenci, anebo napíše do chatu klíčové slovo START. Vše je přehledně vidět v kódu viz. Obrázek 13.

```
    elif formatted_message == "start":
        bot.send_button_message(sender_id, "Vyber si jak chceš začít", Buttons.Start_Vyber_button)
        bot.send_quick_message(sender_id, "Často kladené otázky: ", Buttons.FAQ)
```

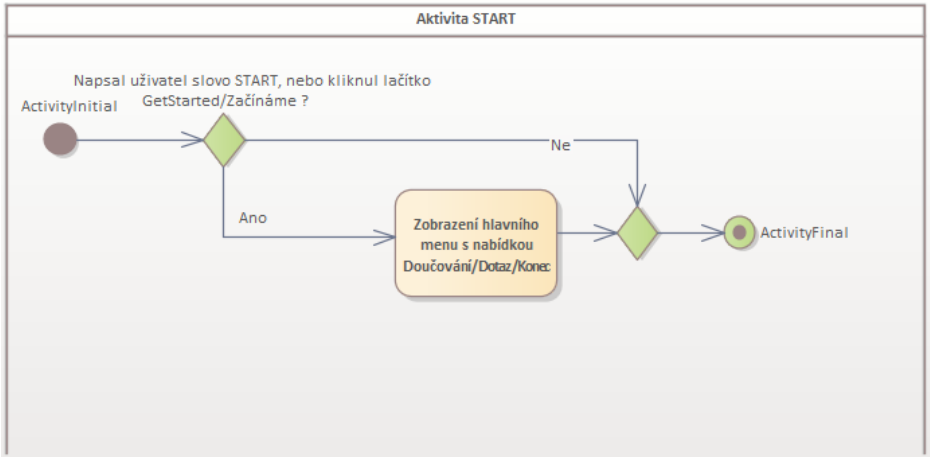
Obrázek 13. Start Funkcionalita

Finální grafickou podobu funkce zobrazuje Obrázek 14.



Obrázek 14. START

Aktivitní diagram k funkcionalitě start ilustruje Obrázek 15.



Obrázek 15. Aktivitní diagram START

7.1.2 Doučování

Funkcionalita Doučování je pro tento projekt klíčová, protože spouští dlouhou sekvenci, díky které se může student zaregistrovat na termín doučování. Tato funkcionalita se dá spustit pouze tím, že uživatel klikne na postback tlačítko „Doučování“, které je vyvoláno funkcionalitou Start.

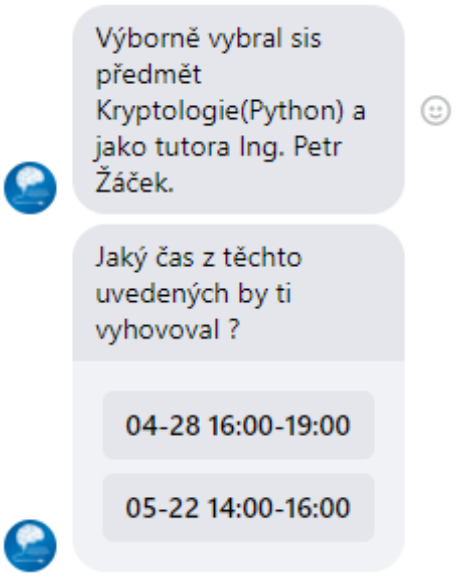
Při stisknutí tlačítka je poslán webhook s paylodem „0x00000002“, který je následně zpracován. Poté je zavolána první metoda send_text_message z knihovny PyMessenger. Díky tomuto kroku je docíleno toho, že uživateli přijde zpráva „Nyní si prosím vyber předmět, o který máš zájem :)“. Posléze je zavolána další metoda z knihovny PyMessenger a to metoda send_generic_message, která vytvoří uživateli karusel s předměty a dostupnými tutory v podobě postback tlačítek tak, jak uvádí Obrázek 16. Tato část funkcionality je přehledně vidět v kódu viz. Obrázek 17.

Obrázek 16. Tlačítko Doučování

```
if formatted_message == "0x00000002":
    bot.send_text_message(sender_id, "Nyní si prosím vyber předmět o který máš zájem :)")
    print(Buttons.buttony)
    bot.send_generic_message(sender_id, Buttons.buttony)
```

Obrázek 17. Doučování payload reakce

Poté v případě, že si uživatel vybere konkrétního tutora, je znovu poslán message_postback s konkrétním webhookem na náš server, který aplikace zpracuje a v případě, že má vybraný tutor nějaké aktuální volné termíny, jsou uživateli opět poslány pomocí send_text_mesage zprávy „Výborně, vybral sis předmět (dynamicky generový z databáze dle výběru) a jako tutora (dynamicky generový z databáze dle výběru).“ Poté je ve stejný okamžik zavolána ještě další funkce send_button_message, která pošle uživateli tři nejaktuálnější volné termíny z databáze v podobě tlačítek. Konkrétní grafickou podobu funkce znázorňuje Obrázek 18.



Obrázek 18. Termíny Doučování

V případě, že daný tutor nebude mít žádný volný termín, je uživateli zaslána zpráva „Omlouvám se, ale nenašel jsem žádný volný termín, zkus si vybrat jiného tutora! 🙂“ a znovu je poslána generická zpráva s karuselem pro výběr předmětů. Tato funkcionalitu přehledně znázorňuje Obrázek 19 a Obrázek 20.

```
for x in Buttons.PayBacks:
    for y in x:
        if formatted_message == y[2]:
            datumyList = []
            datumyButton = []
            DictOfUsers[sender_id][1].clear()
            MySQL.cursor.execute("SELECT TOP(3) Programing_Support_Centrum.dbo.Kalendar.Datum, "
                                 "Programing_Support_Centrum.dbo.Kalendar.Zacatek_Konzultace, "
                                 "Programing_Support_Centrum.dbo.Kalendar.Konec_Konzultace, "
                                 "Programing_Support_Centrum.dbo.Kalendar.Obsazenost, "
                                 "Programing_Support_Centrum.dbo.Kalendar.Max_Obsazenost, "
                                 "Programing_Support_Centrum.dbo.Propojeni.ID_Ucitele, "
                                 "Programing_Support_Centrum.dbo.Kalendar.PayLoad FROM    Programing_Support_Centrum.dbo.Kalendar "
                                 "INNER JOIN Programing_Support_Centrum.dbo.Propojeni ON dbo.Kalendar.ID_propojeni = "
                                 "Programing_Support_Centrum.dbo.Propojeni.ID_Propojeni INNER JOIN "
                                 "Programing_Support_Centrum.dbo.Ucitele ON "
                                 "Programing_Support_Centrum.dbo.Propojeni.ID_Ucitele = "
                                 "Programing_Support_Centrum.dbo.Ucitele.ID_Ucitele "
                                 "WHERE  (dbo.Propojeni.ID_Ucitele = {})".format(
                y[3]) + "AND DATEDIFF(day, GETDATE(), "
                        "Programing_Support_Centrum.dbo.Kalendar.Datum) > 0"
                        "AND Programing_Support_Centrum.dbo.Kalendar.Obsazenost < "
                        "Programing_Support_Centrum.dbo.Kalendar.Max_Obsazenost")
            for row in MySQL.cursor:
                datumyList.append(row)
            if len(datumyList) == 1:
                stringtobutton = (str(datumyList[0][0])[5:] + " " + str(datumyList[0][1])[:5] + "-" + str(
                    datumyList[0][2])[:5])
                datumyButton = [{'type': 'postback', 'title': stringtobutton,
                                 'payload': str(datumyList[0][6]).replace(" ", "")}]
                DictOfUsers[sender_id][1].append(y[0])
```

Obrázek 19. Doučování termíny I. část

```
        elif len(datumyList) == 0:
            bot.send_text_message(sender_id,
                                  "Omlouvám se ale nenašel jsem žádný volný termín zkus si vybrat jiného tutora! :)")
            bot.send_generic_message(sender_id, Buttons.buttony)
            return
        elif len(datumyList) == 2:
            stringtobutton = (str(datumyList[0][0])[5:] + " " + str(datumyList[0][1])[:5] + "-" + str(
                datumyList[0][2])[:5])
            stringtobutton2 = (str(datumyList[1][0])[5:] + " " + str(datumyList[1][1])[:5] + "-" + str(
                datumyList[1][2])[:5])
            datumyButton = [{'type': 'postback', 'title': stringtobutton,
                             'payload': str(datumyList[0][6]).replace(" ", "")},
                            {'type': 'postback', 'title': stringtobutton2,
                             'payload': str(datumyList[1][6]).replace(" ", "")}]
            DictOfUsers[sender_id][1].append(y[0])
        elif len(datumyList) == 3:
            stringtobutton = (str(datumyList[0][0])[5:] + " " + str(datumyList[0][1])[:5] + "-" + str(
                   datumyList[0][2])[:5])
            stringtobutton2 = (str(datumyList[1][0])[5:] + " " + str(datumyList[1][1])[:5] + "-" + str(
                   datumyList[1][2])[:5])
            stringtobutton3 = (str(datumyList[2][0])[5:] + " " + str(datumyList[2][1])[:5] + "-" + str(
                   datumyList[2][2])[:5])

            datumyButton = [{'type': 'postback', 'title': stringtobutton,
                             'payload': str(datumyList[0][6]).replace(" ", "")},
                            {'type': 'postback', 'title': stringtobutton2,
                             'payload': str(datumyList[1][6]).replace(" ", "")},
                            {'type': 'postback', 'title': stringtobutton3,
                             'payload': str(datumyList[2][6]).replace(" ", "")}]
            DictOfUsers[sender_id][1].append(y[0])
        bot.send_text_message(sender_id, "Výborně vybral sis předmět {} a jako tutora {}.".format(y[0], y[1]))
        bot.send_button_message(sender_id, "Jaký čas z těchto uvedených by ti vyhovoval ?", datumyButton)
```

Obrázek 20. Doučování termíny II. část

Pokud jsou tedy uživateli vráceny termíny a on si nějaký zvolí, tak je zkontrolováno, jestli databáze již uživatele zná a jestli má vedený i jeho email. V případě, že uživatel bude

v databázi uveden, je mu poslána button_message s textem „Díky za rezervaci termínu, chceš provést rezervaci na email: (doplněno dynamicky z databáze)“ a možnost Ano/Ne. V případě zvolení tlačítka „Ano“ je provedena rezervace na konkrétní email a uživateli je zaslána zpráva „Tvůj termín je zarezervován, budeme tě očekávat! :)“. V opačném případě (zvolení tlačítka Ne) je uživatel vyzván pomocí zprávy "Tvé přání je mým rozkazem :-). Napiš tedy email, na který si přeješ provést rezervaci“ proto, aby do chatu napsal email, na který si přeje provést konkrétní rezervaci. Následně je tento email vepsán do databáze k jeho ID a provede se rezervace stejně tak, jako při zvolení možnosti ANO. V obou těchto případech, je poslán i email na Programming Support Centre s bližší informací na kdy, kam a jaký student (rozeznáno pomoc emailu) se registroval na termín, a to vše pomocí SMTP klienta. Poté je uživateli navráceno úvodní Start Menu a FAQ.

V případě, že je uživatel nově příchozí, tudíž si ještě nikdy nerezervoval termín, je veden stejně, jako by zvolil tlačítko „NE“ při vybírání emailu s tím rozdílem, že je do databáze zavedeno i jeho ID.

Existuje zde také možnost, že uživatel je na daný termín již přihlášený, v tomto případě je uživateli vrácena textová zpráva „Tento termín máš již rezervovaný :-(musíš dát prostor také ostatním studentům“ a je vrácen na hlavní Start menu. Tato část funkcionality je přehledně znázorněna skrze Obrázek 21 a Obrázek 22.

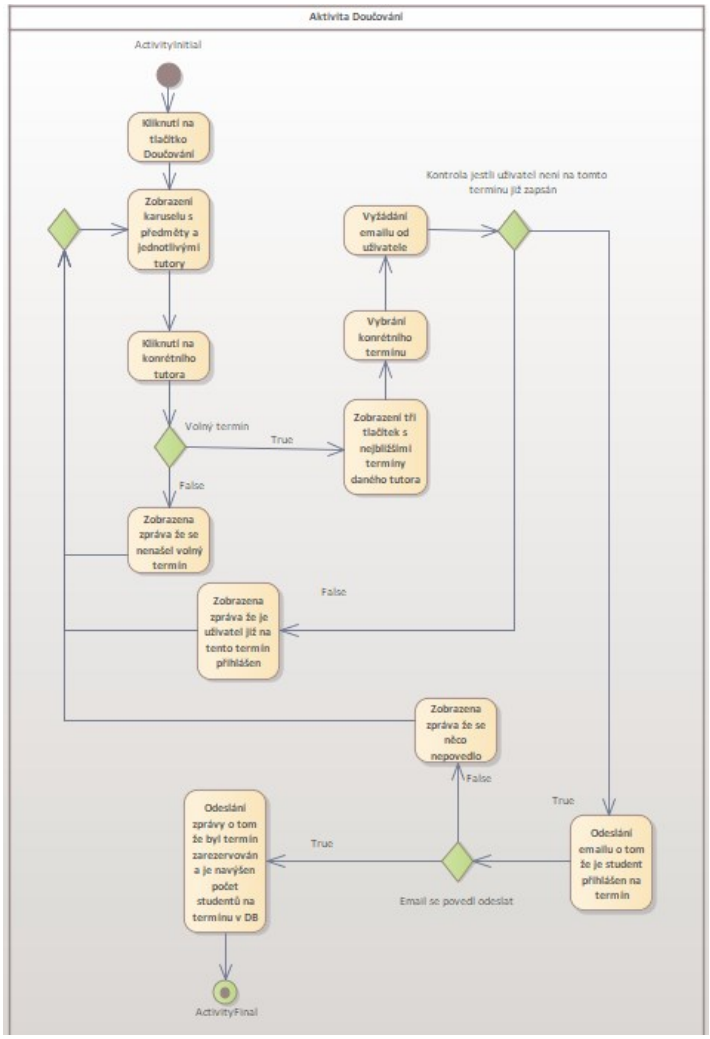
```
elif formatted_message == "mailano":
    body_text = "Uzivatel {} se prihlasil na predmet {} dne {} v {} az {} kde aktualni obsazenost je " \
                "{}/{}".format(str(DictOfUsers[sender_id][1][7]).strip(), DictOfUsers[sender_id][1][0],
                               str(DictOfUsers[sender_id][1][5]).replace(" ", ""),
                               str(DictOfUsers[sender_id][1][1]).replace(" ", "")[:5],
                               str(DictOfUsers[sender_id][1][2]).replace(" ", "")[:5],
                               str(int(str(DictOfUsers[sender_id][1][3]).replace(" ", "")) + 1),
                               str(DictOfUsers[sender_id][1][4]).replace(" ", ""))
    try:
        MySQL.cursor.execute("SELECT Programing_Support_Centrum.dbo.Uzivatele.ID_Uzivatele "
                             "FROM Programing_Support_Centrum.dbo.Uzivatele "
                             "WHERE Programing_Support_Centrum.dbo.Uzivatele.FacebookID = {}"
                             .format(sender_id))
        for IDS in MySQL.cursor:
            hello = [str(x).strip(' ') for x in IDS]
        UserID = int("".join(hello))
        MySQL.cursor.execute(
            "Select Programing_Support_Centrum.dbo.Uzivatel_Kalendar.ID_propojeni, "
            "Programing_Support_Centrum.dbo.Uzivatel_Kalendar.ID_Uzivatele from "
            "Programing_Support_Centrum.dbo.Uzivatel_Kalendar Where "
            "Programing_Support_Centrum.dbo.Uzivatel_Kalendar.ID_propojeni = {} AND "
            "Programing_Support_Centrum.dbo.Uzivatel_Kalendar.ID_Uzivatele = {}".format(str(
                DictOfUsers[sender_id][1][6]).strip(), UserID)
        )
```

Obrázek 21. Doučování rezervace termínu I. část

```
        if MySQL.cursor.fetchone() is not None:
            raise Exception
        MySQL.cursor.execute("UPDATE Programing_Support_Centrum.dbo.Kalendar SET "
                             "Programing_Support_Centrum.dbo.Kalendar.Obsazenost = Programing_Support_Centrum.dbo.Kalendar.Obsazenost + 1 "
                             "WHERE "
                             "Programing_Support_Centrum.dbo.Kalendar.ID_propojeni = {}"
                             .format(str(DictOfUsers[sender_id][1][6]).replace(" ", "")))
        MySQL.conn.commit()
        MySQL.cursor.execute(
            "INSERT INTO Programing_Support_Centrum.dbo.Uzivatel_Kalendar(ID_propojeni,ID_Uzivatele) VALUES "
            "({},{})".format(str(DictOfUsers[sender_id][1][6]).strip(), UserID))
        MySQL.conn.commit()
        Email_sender.SendMail(str(DictOfUsers[sender_id][1][7]).strip(), body_text)
        bot.send_text_message(sender_id, "Tvůj termín je zarezervován budeme tě očekávat ! :)")
        bot.send_button_message(sender_id, "Vyber si jak chceš začít",
                                Buttons.Start_Vyber_button)
        DictOfUsers[sender_id][1].clear()
    except:
        bot.send_text_message(sender_id,
                              "Tento termín máš již rezervovaný :-( musíš dát prostor také ostatním studentům")
        bot.send_button_message(sender_id, "Vyber si jak chceš začít",
                                Buttons.Start_Vyber_button)
        DictOfUsers[sender_id][1].clear()
elif formatted_message == "mailne":
    bot.send_text_message(sender_id,
                          "Tvé přání je mým rozkazem :-). Napiš tedy email na který si přeješ provést rezervaci")
```

Obrázek 22. Doučování rezervace termínu II. Část

Aktivitní diagram k funkcionalitě Doučování ilustruje Obrázek 23.



Obrázek 23. Aktivitní diagram Doučování

7.1.3 Dotaz

Finální funkcionalita Dotaz je závislá na funkcionalitě START. Při stisknutí tlačítka Dotaz je poslán webhook s paylodem 0x00000003. Tento payload je následně zpracován naší aplikací a uživateli je poslána zpráva „Dobrá, teď tě jen poprosím, aby si uvedl svůj dotaz “ a to pomocí funkce send_text_message viz. Obrázek 24.

```
    elif formatted_message == "0x00000003":  # Dotaz BUtton
        bot.send_text_message(sender_id, "Dobrá ted tě jen poprosím aby si uvedl svůj dotaz :-P")
        DictOfUsers[sender_id][0].append("Dotaz")
```

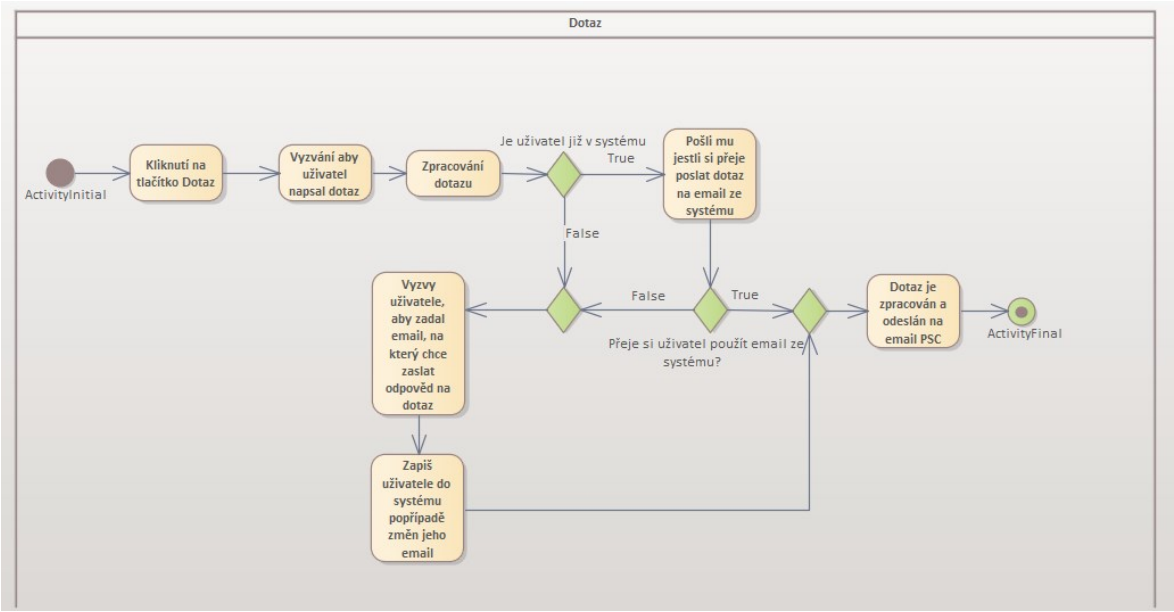
Obrázek 24. Dotaz payload reakce

Poté aplikace zpracuje uživatelský dotaz a pomocí funkce send_text_message je uživateli poslána zpráva „Tvůj dotaz nyní vypadá takto :(Uživatelův dotaz) poprosím tě tedy o to, aby si zadal email na " "který si přeješ odpověď"". Následně je zpracován i email uživatele a je mu poslána další textová zpráva „Díky za dotaz, odpovíme ti co nejrychleji to půjde na email: (Email, který zadal uživatel)“. Poté je pomocí SMTP klienta poslán email na Programming Support Centre s již zpracovaným dotazem. Tuto část funkcionality zobrazuje Obrázek 25.

```
                # Zpracování dotazu zacátek
                if len(DictOfUsers[sender_id][0]) == 1:
                    bot.send_text_message(sender_id, "Tvůj dotaz nyní vypadá takto :\n"
                                                     "[{}]\n poprosím tě tedy o to, aby si zadal email na "
                                                     "který si přeješ odpověď".format(text))
                    DictOfUsers[sender_id][0].append(text)
                elif len(DictOfUsers[sender_id][0]) == 2:
                    bot.send_text_message(sender_id,
                                          "Díky za dotaz odpovíme ti co nejrychleji to půjde na email :\n{}".format(
                                              text))
                    Email_sender.SendMail(text, DictOfUsers[sender_id][0][1])
                    DictOfUsers[sender_id][0].clear()
                    bot.send_button_message(sender_id, "Vyber si jak chceš začít", Buttons.Start_Vyber_button)
                # Zpracování dotazu konec
```

Obrázek 25. Finální část dotazu

Aktivitní diagram k funkcionalitě Dotaz ilustruje Obrázek 26.



Obrázek 26. Aktivitní diagram Dotaz

7.1.4 FAQ

Funkcionalita FAQ je velmi důležitá, a proto je dostupná skoro po celou dobu komunikace s chatbotem. Je vyvolána automaticky, a to vždy na začátku konverzace, anebo po jakékoliv jiné dokončené funkcionalitě.

Je založená na funkci send_quick_message z knihovny Pymessenger, která vyžaduje tři povinné parametry Sender_ID, textovou zprávu a payload. V payloadu jsou definované často kladené dotazy, které se načítají z databáze, aby je bylo možné měnit bez zásahu do kódu. Tuto část dynamického generování GUI z naší databáze ilustruje Obrázek 27.

```
def process_payload(payload, sender_id):                    #Process payload QuickMessages(FAQ)
    for x in MySQL.list_of_FAQ:
        if payload == str(x[1]).strip():
            if payload == "SecondFAQ":
                bot.send_text_message(sender_id, str(x[3]).strip())
                bot.send_image_url(sender_id,"https://fai.utb.cz/wp-content/uploads/2020/10/1NP_PSC_white.png")
                bot.send_button_message(sender_id, "Vyber si jak chceš začít", Buttons.Start_Vyber_button)
                bot.send_quick_message(sender_id, "Často kladené otázky: ", Buttons.FAQ)
            else:
                bot.send_text_message(sender_id, str(x[3]).strip())
                bot.send_button_message(sender_id, "Vyber si jak chceš začít", Buttons.Start_Vyber_button)
                bot.send_quick_message(sender_id, "Často kladené otázky: ", Buttons.FAQ)
```

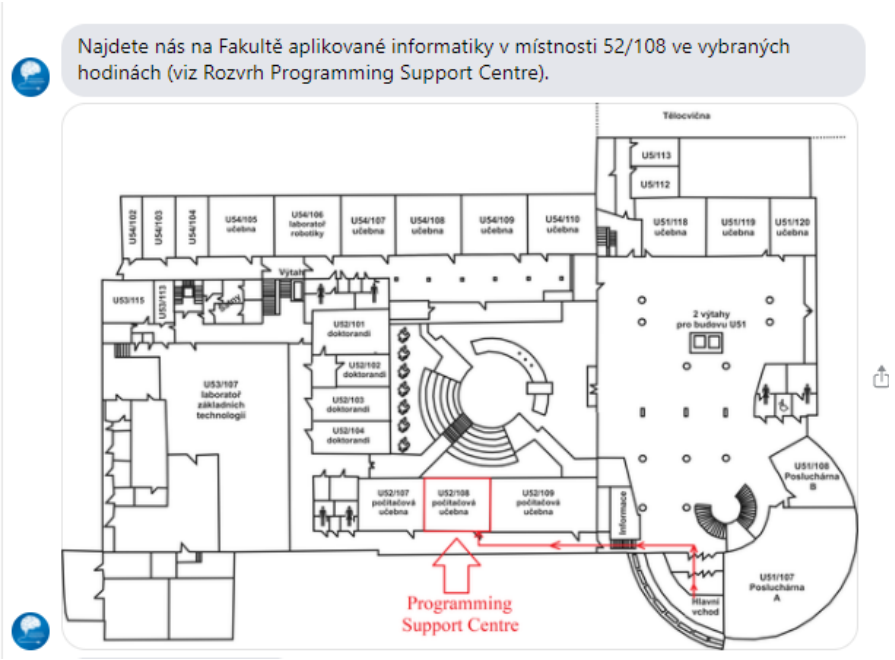
Obrázek 27. Dynamické generování GUI FAQ

Payload je ve formátu Json a vyžaduje čtyři povinné informace. V našem případě se jedná o content_type, title, payload, a img_url tak, jak zobrazuje Obrázek 28. Pokud uživatel klikne na jakékoliv tlačítko z FAQ nabídky, je mu doručena textová zpráva s příslušnou odpovědí, a to pomocí funkce send_text_message. Jediná výjimka nastává v případě kliknutí na FAQ

„Kde se nacházíme?“ V tom případě je místo funkce send_text_message použita funkce send_image_url, která uživateli pošle mapu fakulty, kde se Programming Support Centre nachází tak, jak je ukazuje Obrázek 29.

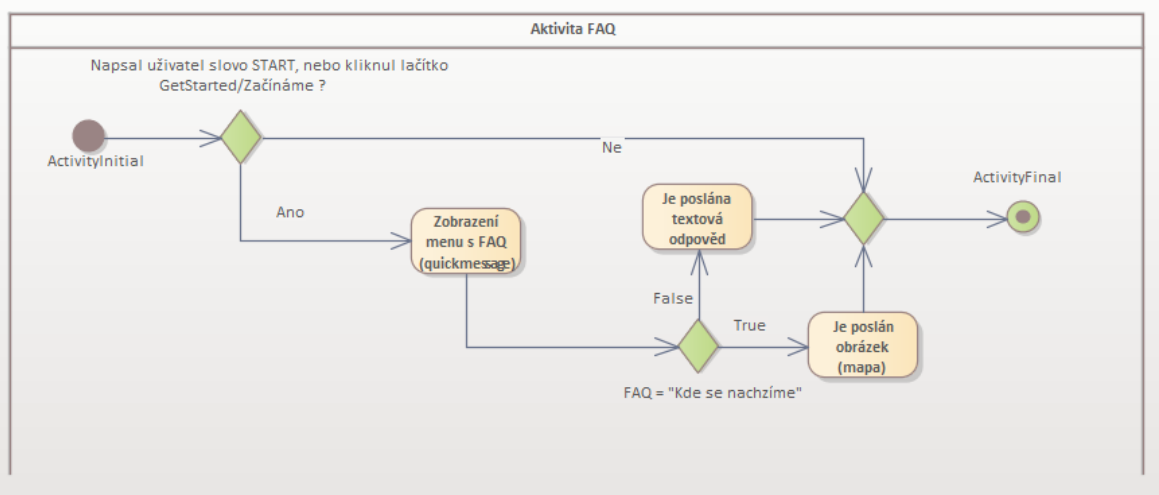
```
FAQ = []
for FAQ_Question in MySQL.list_of_FAQ:
    FAQ.append({
        "content_type": "text",
        "title": str(FAQ_Question[0]).strip(),
        "payload": str(FAQ_Question[1]).strip(),
        "image_url": str(FAQ_Question[2]).strip()
    })
```

Obrázek 28. Payload FAQ



Obrázek 29. Mapa

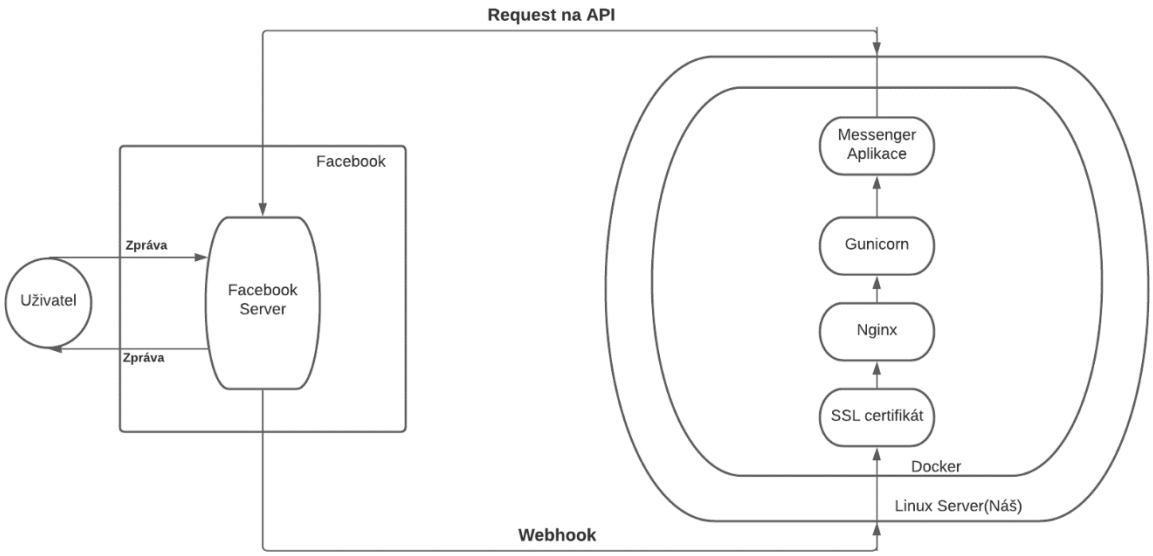
Aktivitní diagram k funkcionalitě FAQ ilustruje Obrázek 30.Obrázek 26. Aktivitní diagram Dotaz



Obrázek 30. Aktivitní diagram FAQ

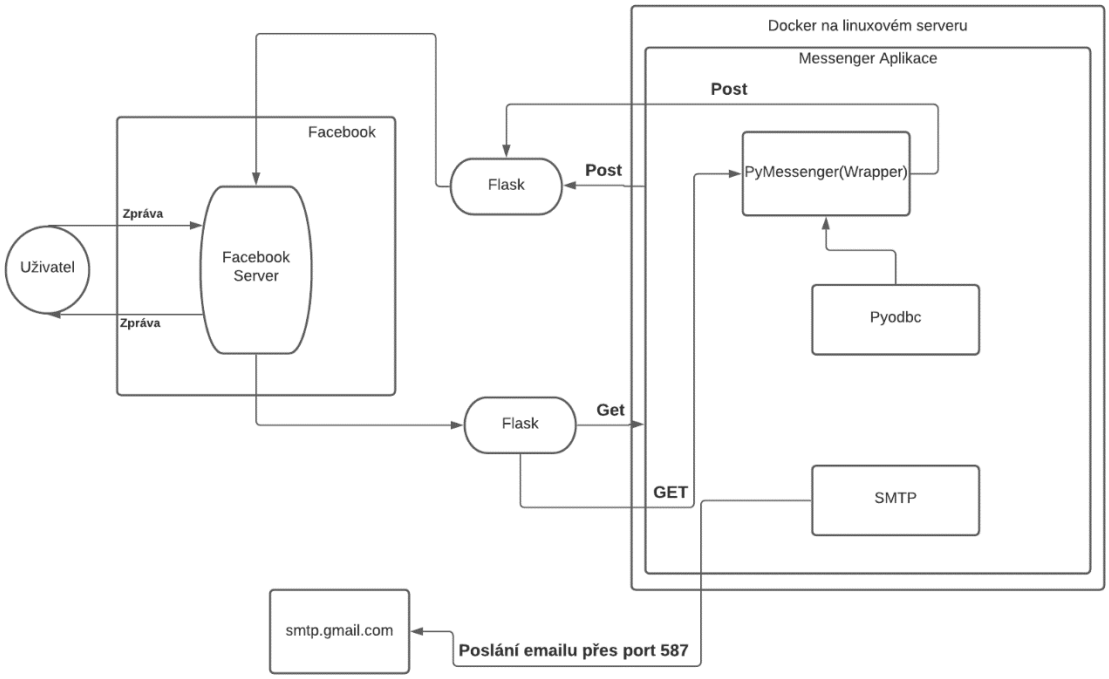
7.2 Propojení jednotlivých technologií

Tato kapitola pojednává o propojení všech technologií, které jsou použity. Obrázek 31 popisuje propojení jednotlivých technologií, které ukazují, jak mezi sebou komunikuje náš Linuxový server a server společnosti Facebook. Přehledně zde můžeme vidět že náš server respektive (aplikace) posílá požadavky na API, kde následně Facebook server odpoví pomocí Webhooku. Na obrázku lze také pozorovat, že celá naše aplikace běží v technologii Docker.



Obrázek 31. Propojeni technologií I

Obrázek 32 ilustruje podrobné propojení naší aplikace s Facebook serverem. Lze pozorovat, že celá aplikace komunikuje pomocí frameworku Flask, který přijímá a posílá GET a POST požadavky na Facebook server. Tyto požadavky využívá právě náš Wrapper PyMessenger, ze kterého se volají jednotlivé metody. Tyto metody jsou dále napojeny na další technologii, a to Pyodbc, odkud se vytahují data, která chceme uživateli zobrazit. Jako poslední technologie naší aplikace je SMTP klient, který posílá emaily přes port 587. Messenger Aplikace je spuštěna na Linuxovém serveru pomocí technologie Docker.



Obrázek 32. Propojení technologií II

7.3 Bezpečnost aplikace

Zabezpečení aplikace není aktuálně plně dořešeno. Nicméně v bezpečnosti aplikace jsou určité limity a hrozí zde bezpečnostní rizika. Z důvodu snížení těchto rizik běží celá aplikace v dockeru na našem linuxovém serveru. Dále jsou zde uzavřeny všechny porty, které aplikace nevyužívá pro svůj chod.

Za minoritní nevýhodu se dá považovat využití bezpečnostního certifikátu Let's Encrypt a to z důvodu, že některé instituce nerespektují ověření tohoto certifikátu, protože tento certifikát využívají také podvodné web stránky, které by jinak svůj certifikát takto jednoduše nezískaly. Při spuštění této certifikační autority bylo až 15 tisíc certifikátů zneužito pro phishingovým weby, které obsahovaly v názvu PayPal. V budoucnu je v plánu přejít na spolehlivější certifikační autoritu. [55]

Je také potřeba připomenout, že společnosti Facebook v nedávné době uniklo velké množství uživatelských dat. Vzhledem k tomu, že je aplikace na platformě Facebook nepřímo závislá, tak lze tuto hrozbu v jistém směru považovat i za bezpečnostní riziko pro naši aplikaci. Z tohoto důvodu se také v aplikaci ukládá pouze unikátní identifikační číslo uživatele,

které je přiděleno společností Facebook (toto je veřejně dohledatelné) a email, který si přeje uživatel sdílet. [56]

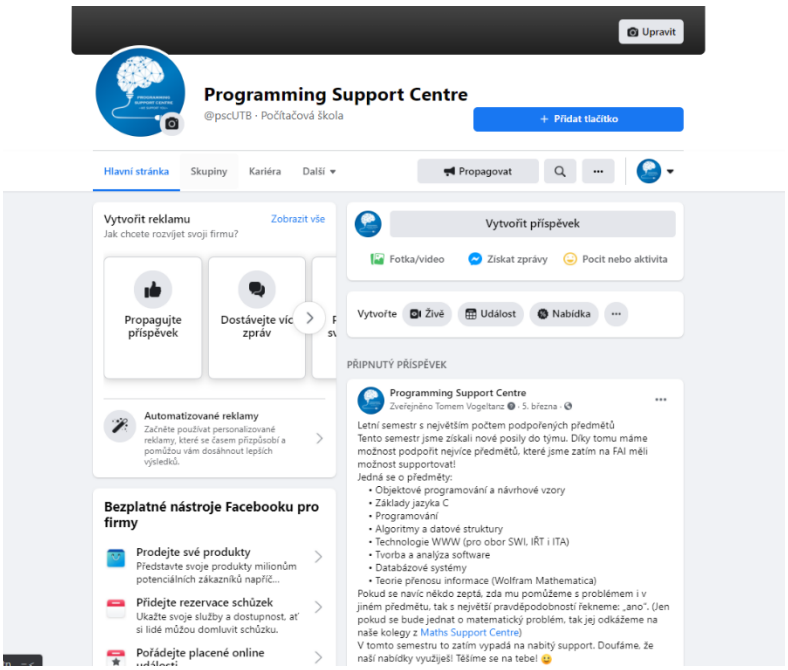
7.4 Vytvoření aplikace na platformě Facebook messenger

Tato podkapitola pojednává o tom, jakým způsobem se vytváří aplikace na stránce Facebook Messenger. Jaké informace je potřeba zadat a jaké jsou podmínky pro vytvoření vlastní aplikace pro messenger chatbota. Dále bude popsán vývojářský učet a ukázka toho, jak přidat testery této aplikace.

7.4.1 Stránka

Nejprve je potřeba vytvořit stránku na sociální síti Facebook, a to hlavně z toho důvodu, že vlastní aplikace lze tvořit pouze na stránkách, nikoliv na soukromých účtech. Na stránce je potřeba definovat její název, v našem případě se volil název organizace. Poté je potřeba vybrat fotku stránky, která bude reprezentovat náhled jak v miniatuře, tak později v aplikaci.

Všechny tyto atributy, které je potřeba zadat při vytváření stránky, se nesmí podcenit, a to z toho důvodu, že se jedná o první interakci, kterou uživatel bude mít. Obrázek 33 ukazuje náhled na finální stránku.



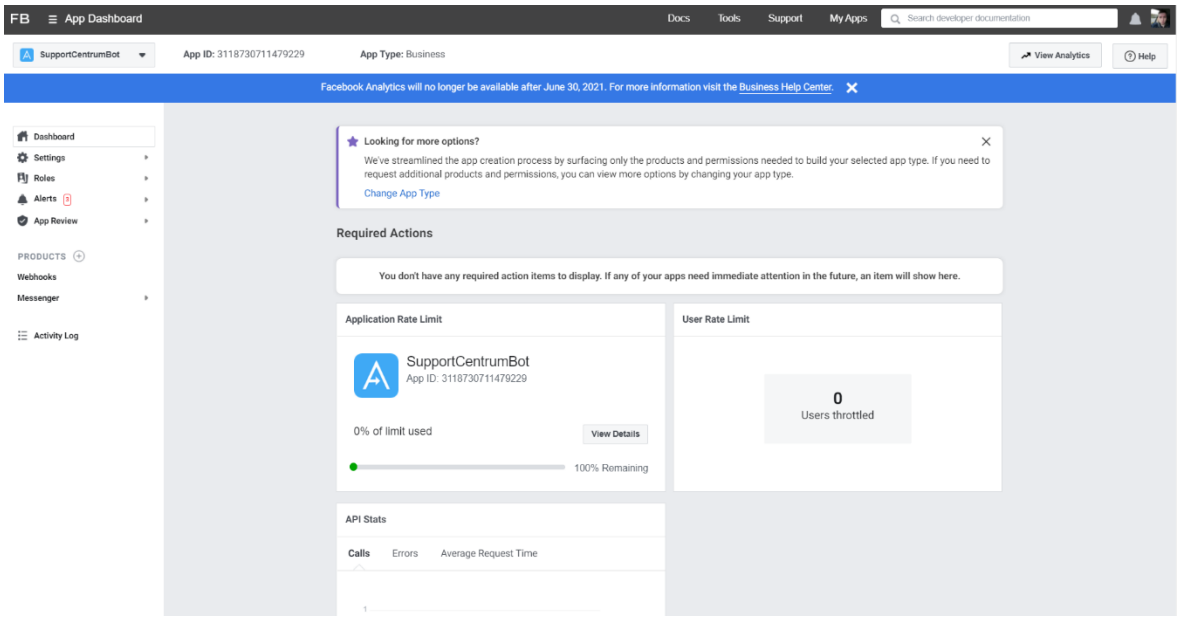
Obrázek 33. Programming Support Centre stránka

7.4.2 Založení Aplikace

Nyní si je potřeba založit aplikaci, toho lze docílit tím způsobem, že si otevřeme https://developers.facebook.com/apps/?show_reminder=true. Na této stránce je nyní potřeba zvolit „Create App“, díky čemuž vyskočí dialogové okno, ve kterém je nutné zaškrtnout Manage Business Integrations z toho důvodu, že chceme messenger bota.

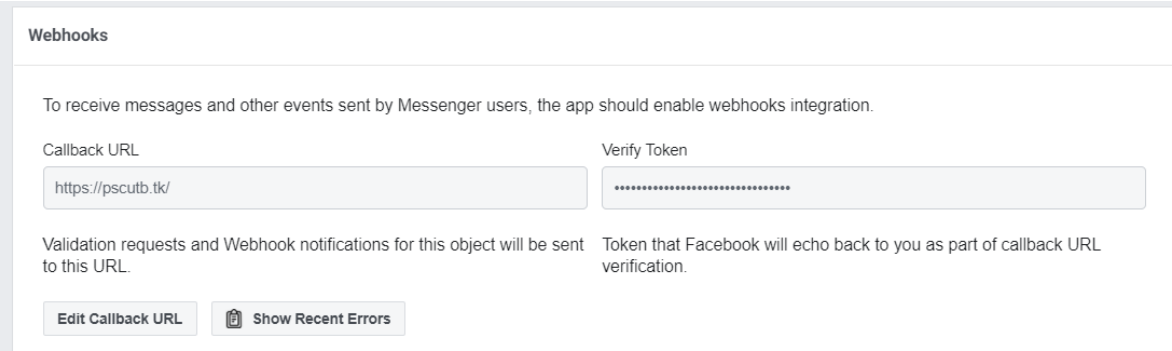
Poté co projdeme tímto procesem, je potřebné vyplnit název nové aplikace, kontaktní email (email, přes který budeme komunikovat s Facebook podporou), účel aplikace a stránku, na kterou bude aplikace vázána viz kapitola 7.4.1. Při vyplňování účelu aplikace je potřeba zvolit položku Clients, a to z toho důvodu, že je určena pro vytváření messenger chatbota.

Pokud je vše provedeno správně, měli bychom být přesměrováni na podobný náhled jako nám ukazuje Obrázek 34. V tomto náhledu je potřeba zvolit Products a zde zaškrtnout položky Messenger a Webhooks.



Obrázek 34. Náhled vytvořené aplikace

Poté je potřeba zvolit produkt messenger a zde zvolit kolonku settings, ve které je důležitá možnost pro chod aplikace, a to generate token, díky kterému poté Facebook bude vědět, že s ním komunikuje pouze naše aplikace. Dále zde najdeme velice důležitou část, a to Callback URL a Verify Token tak, jak uvádí Obrázek 35. Položka Callback URL slouží proto, aby Facebook věděl, na jaké URL odesílat webhooky s požadavky (GET, POST). Tato doména musí být zabezpečená. Poté se zde nachází token, který je ověřen vždy, když se aplikace připojí na server.

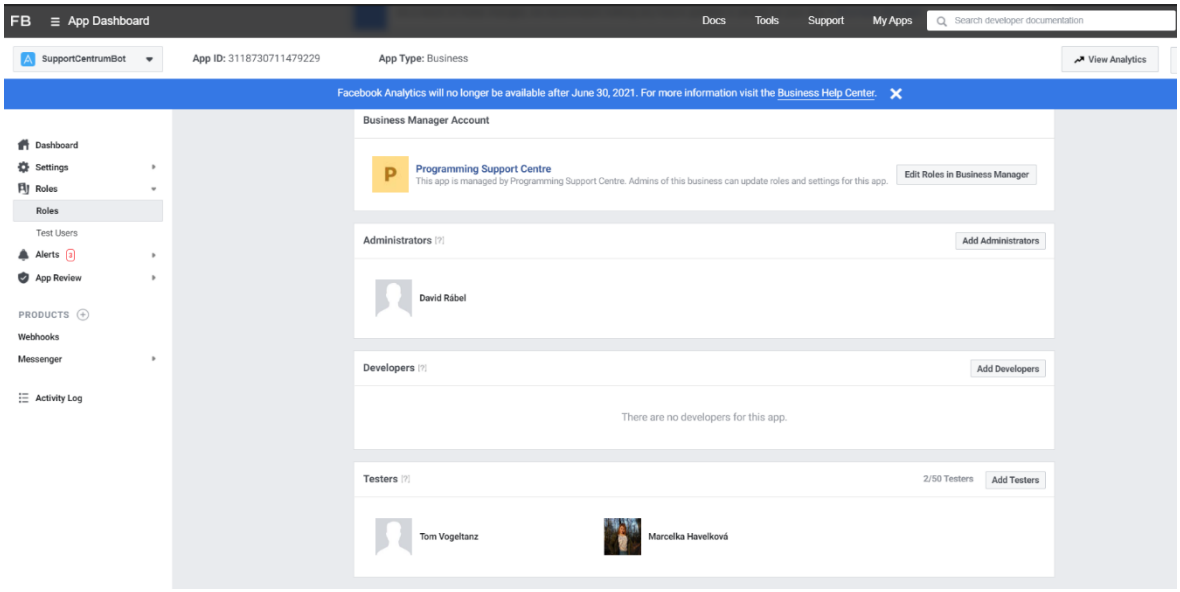


Obrázek 35. CallBack URL

7.4.3 Vývojářský účet

Do doby, než je aplikace schválená veřejně, messenger bot nereaguje na žádné zprávy, které přijdou od uživatelů. Z důvodu, aby bylo možné tohoto bota vyzkoušet a otestovat před zažádáním o schválení, je nutno v aplikaci vytvořit vývojářský učet, popřípadě přidat do aplikace developery anebo testery. V případě, že bude uživatel označený jako Developer anebo Tester, bude na něj aplikace reagovat i bez předchozího schválení firmou Facebook.

Tento krok provede tím způsobem, že si otevřeme správu aplikace, kde zvolíme položku Roles a zde klikneme na tlačítko Add a vyplníme email, který je propojený s existujícím Facebook účtem, ze kterého chceme udělat Developera/Testera. Poté, co je tento krok proveden, je uživatel upozorněn, že byl zvolen jako Developer/Tester dané aplikace. V případě, že s danou rolí uživatel souhlasí, musí si ověřit účet, a to pomocí telefonního čísla. Následně tuto roli může přijmout, díky čemuž na něj bot bude reagovat. Přehled developerů a testerů naší aplikace přehledně zobrazuje Obrázek 36.



Obrázek 36. Tester/Developer

7.5 Schvalování aplikace na platformě FB messenger

V případě, že chceme našeho chatbota spustit veřejně, je nejprve nutno projít schvalovacím procesem, který poskytuje společnost Facebook. Tento proces najdeme v menu pod názvem App Review → Request. V tomto procesu je nejprve nutné si stanovit, o jakou část přístupu máme zájem, na výběr je jich tu velmi mnoho. V našem případě jde jen o jednu, a to pages_messaging. Tato funkcionalita není automaticky přístupná, tudíž je nutné si o ní požádat, to provedeme pomocí tlačítka request.

Pro přístup k page_messaging je potřeba nejprve vyplnit dlouhý dotazník, kde musíme vyplnit otázky typu: Jak budeme používat tuto funkci? Vysvětlit krok po kroku, jak funguje naše aplikace a jak jí má testovaná osoba obsluhovat. Nakonec bylo potřeba natočit jednoduché video, kde byl znázorněn průchod funkčností naší aplikace.

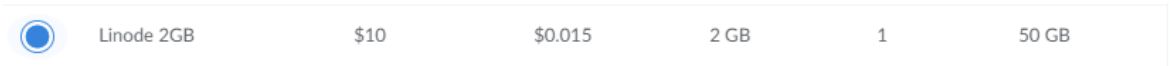
Pokud vše provedeme správně, měla by být aplikace otestována a schválena do pěti pracovních dnů. Je potřeba podotknout, že aplikace je testována zaměstnanci Facebooku, tudíž je testována většinou v brzkých ranních hodinách a je potřeba myslet na to, že musíme mít aplikaci spuštěnou po celou dobu této žádosti. Pokud je aplikace schválena, je možné jí pustit veřejně, a to pomocí posuvníku ve správě naší aplikace společnosti Facebook. Schvalování aplikace v našem případě trvalo jeden pracovní den.

8 NASAZENÍ APLIKACE NA SERVER

Nyní je potřeba aplikaci nasadit na server právě z toho důvodu, že chatbot funguje na principu zaslání požadavku a následném zpracování. Abychom mohli takové požadavky zpracovat je potřeba funkční server, který bude veřejně vystaven do Internetu. Server nám bude sloužit jako most mezi uživatelem a Facebook API.

8.1 Server

Jako server pro našeho chatbota byl vybrán server s linuxovou distribucí, a to konkrétně Linux Fedora Server. Služba pro hosting našeho serveru byla vybrána služba Linode, a to hlavně z důvodu nižší ceny a příjemného uživatelského prostředí. Konfigurace serveru byla po dohodě s vedoucím práce a náročnosti zvolena tak, jak ilustruje Obrázek 37. Větší paměť RAM byla vybrána, protože celá aplikace je na serveru spuštěna pomocí technologie Docker, kde je pro běh MSSQL potřeba minimálně 2 GB RAM.



Obrázek 37. Linode konfigurace

8.2 Konfigurace souborů

Celá aplikace běží na našem Linode serveru v technologii Docker. Zároveň je však propojena s serverem pomocí technologie Nginx a Gunicorn. Proto je nutné si nejprve tyto technologie nastavit pomocí konfiguračních souborů.

Nejprve je potřeba si na serveru vytvořit soubor s názvem Procfile, do kterého se musí vložit název aplikace - v našem případě „web: gunicorn app:FacebookBot“. Díky tomuto souboru bude technologie Gunicorn vědět, že má spouštět a hledat všechny serverové požadavky právě v našem souboru FacebookBot.py, kde je kompletní konfigurace webového frameworku Flask.

Dále je potřeba konfigurovat složku s technologií Nginx, ve které je potřebné mít tři soubory: DockerFile, Nginx.conf a project.conf. Výslednou podobu Nginx.conf souboru je možno stáhnout z oficiální dokumentace k technologii nginx a zůstává v originální podobě. Soubor project.conf je nutné nakonfigurovat dle naší aplikace a portu, na kterém aplikace bude poslouchat. Obrázek 38 zobrazuje finální podobu tohoto souboru.

19 lines (15 sloc) | 435 Bytes Raw Blame

```
server {

    listen 8000;
    server_name docker_flask_gunicorn_nginx;

    location / {
        proxy_pass http://Python_Chat_Bot_Flask_ngrok:8000;

        # Do not change this
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /static {
        rewrite ^/static(.*) /$1 break;
        root /static;
    }
}
```

Obrázek 38. Nginx-project.conf

Poslední soubor, který je potřeba nastavit pro správnou funkčnost aplikace na našem serveru, se týká DockerFile. Tento soubor slouží pouze proto, že obalí naší technologii Nginx do kontejneru, a tudíž potom půjde spouštět jako celek bez toho, aniž by vytvářel soubory na našem serveru.

Další soubor, který je potřeba pro funkčnost aplikace na našem serveru, je requirements.txt. Tento soubor říká, jaké balíčky využívá naše aplikace. Díky tomuto souboru je náš server schopný tyto balíčky doinstalovat. Tento soubor můžeme získat jednoduše, a to z lokálního počítače pomocí příkazu $pip freeze> requirements.txt.

Poté je potřeba nastavit další soubor a to install-packages.sh, ve kterém je definováno, jaké balíčky musíme doinstalovat na naší distribuci, aby nám zde fungovaly všechny naše závislosti. Zde je potřeba vložit příkaz „dnf install – y unixODBC unixODBC-devel gcc-c++ python3-devel crypto-policies-scripts crypto-policies“ z důvodu, že žádná linuxová distribuce nemá nainstalované unixOBDC, která je potřebná pro běh našeho balíčku Pyodbc.

Nyní si je potřeba nastavit další docker file, tentokrát však pro celou aplikaci a instalaci jednotlivých balíčků pomocí souboru requirements.txt a install-packages.sh. Finální podobu tohoto souboru ukazuje Obrázek 39.

```
FROM fedora:33

WORKDIR /tmp
ADD https://packages.microsoft.com/config/rhel/8/prod.repo /etc/yum.repos.d/mssql-release.repo
RUN ACCEPT_EULA=Y dnf install -y msodbcsql17 mssql-tools \
    && echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bash_profile \
    && echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc \
    && source ~/.bashrc \
RUN dnf install -y unixODBC-devel
COPY install-packages.sh .
RUN ./install-packages.sh
RUN update-crypto-policies --set LEGACY
WORKDIR /Python_Chat_Bot_Flask_ngrok
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
RUN echo "LANG=en_US.UTF-8" >> /etc/locale.conf \
    && echo "LC_ALL=en_US.UTF-8" >> /etc/environment
RUN echo -e "MinProtocol = TLSv1.0\nCipherString = DEFAULT@SECLEVEL=1" >> /etc/pki/tls/open
```

Obrázek 39. Docker file

8.3 Spuštění aplikace na serveru

Poté, co máme nastavené všechny potřebné soubory pro běh na serveru, je nutné tuto aplikaci spustit. Celá aplikace běží v Dockeru a vzhledem k tomu, že již máme vytvořené dva docker file jak pro celou aplikaci a instalaci balíčků, tak pro Nginx, je potřeba tyto soubory spojit a říct Dockeru, že je chceme spouštět jako celek. To provedeme tak, že si vytvoříme v našem repositáři soubor docker-compose.yml, do kterého vložíme příslušné názvy jednotlivých kontejnerů a jejich lokální parametry. V našem případě se jedná o kontejnery s názvy „Python_Chat_Bot_Flask_ngrok“ a „nginx“.

Poslední soubor, který budeme muset vytvořit, je script s názvem run docker.sh, který nám poslouží pouze pro spuštění našeho docker_compose souboru. Obrázek 40 ukazuje finální podobu našeho skriptu.

5 lines (4 sloc) | 116 Bytes Raw Blame

```
echo killing old docker processes
docker-compose rm -fs

echo building docker containers
docker-compose up --build
```

Obrázek 40. Run_docker script

Pokud je vše vytvořeno správně, měla by se při vložení příkazu ./run_docker.sh do našeho Linuxového server terminálu spustit naše celá aplikace a na server se automaticky doinstalovat všechny potřebné balíčky a komponenty tak, jak jsme si nastavili v kapitole 8.2. Při ukončení běhu Dockeru se všechny závislosti smažou, tudíž nebudou zabírat místo na našem serveru.

ZÁVĚR

V teoretické části bakalářské práce byla zpracována základní rešerše problematiky a úvod do fungování chatbota. Nejprve byla popsána historie chatbotů a jejich postupný vývoj až do doby, jak je známe dnes. Popsáno bylo také základní teoretické rozdělení chatbotů tak, jak jej uvádí většina literárních zdrojů. V poslední teoretické části bylo provedeno porovnání chatbotů na klíč a byly popsány jejich výhody a nevýhody.

V praktické části této práce byly navrženy a vytvořeny jednotlivé funkcionality, které byly předem konzultovány s vedoucím Programming Support Centra. Dále zde byla vybrána konkrétní technologie a platforma Facebook Messenger. Právě díky výběru této platformy bylo pro uživatele vytvořeno vhodné uživatelské rozhraní.

Následně zde byla vytvořena hlavní funkcionalita, a to schopnost rezervace studentů na jednotlivé termíny skrze chatbota. Součástí práce byl také návrh a vytvoření fyzického modelu databáze, který obsahuje nezbytné informace pro správné fungovaní chatbota. Dále se také do fyzického modelu ukládá email a ID jednotlivých uživatelů proto, aby při dalších rezervacích uživatel měl jednoduší postup při zadávání rezervace. Také se pomocí Facebook id zjišťuje, zda se ten samý uživatel nesnaží přihlásit na stejný termín vícekrát

Součástí práce je také částečný návrh chatbota, a to pomocí programu Enterprise Architect včetně případu užití a entitně relačního diagram. V bakalářské práci byly také zdokumentovány jednotlivé zdrojové kódy včetně uvedení aktivitních diagramů k důležitým funkcím".

V předposlední části bakalářské práce bylo popsáno vytváření aplikace na platformě Facebook messenger. Následně bylo popsáno i schvalování aplikace. Kvůli situaci ohledně COVID 19 nicméně nastal problém, kdy společnost Facebook pozastavila individuální schvalování aplikací na platformě Messenger. Tento problém byl naštěstí vyřešen tím, že v dubnu roku 2021 bylo toto omezení zrušeno.

Poslední část bakalářské práce byla zaměřena na nasazení aplikace na vlastní linuxový server distribuce Fedora. Při nasazování aplikace na server bylo využito populární technologie Docker, ve které je také spuštěna celá aplikace včetně serveru MS SQL. Právě z důvodu této technologie nastal problém a musel být zvolen dražší server s větší RAM pamětí. Dále bylo potřeba kontaktovat Linode Support a vyřešit automaticky zablokované porty na posílání emailů z důvodu prevence proti spamu.

V bakalářské práci je také velký prostor pro další rozvoj této aplikace, a to například implementací jednotlivých „her“ pro seznámení studentů FAI s projektem Programming Support Centre. Zcela jistě je také potřeba dodělat systém pro správu jednotlivých tutorů a dat, které se aktuálně editují přímo v databázi. Dále se zde nabízí také možnost rozšíření tohoto projektu na všechna doučovací centra na fakultě a sjednotit tak tyto centra na jedno místo.

SEZNAM POUŽITÉ LITERATURY

[1] Facebook Revenue and Usage Statistics (2021). Businessofapps [online]. MANSOOR IQBAL, 2021 [cit. 2021-5-6]. Dostupné z: https://www.businessofapps.com/data/facebook-statistics/

[2] MULDOWNEY, Oisin.Chatbots: An Introduction And Easy Guide To Making Your Own. Dublin, Ireland: Curses & Magic, 2017. ISBN 978-1-9998348-0-7

[3] TURING, A. M. COMPUTING MACHINERY AND INTELLIGENCE. The imitation game: based onthe incredible true story of Alan Turing [online]. [London]: Studiocanal, [2015], s. 1-22 [cit. 2021-5-8]. ISBN Turing. Dostupné z: https://www.csee.umbc.edu/courses/471/papers/turing.pdfZEMČÍK, Tomáš. A Brief History of Chatbots [online]. 2019 [cit. 2021-5-6]. Dostupné z: doi:10.12783/dtcse/aicae2019/31439

[4] LOKVENCOVÁ, Iva. Argument čínského pokoje – včera, dnes a zítra. Brno, 2014. Bakalářská diplomová práce. MASARYKOVA UNIVERZITA FILOZOFICKÁ FAKULTA. Vedoucí práce Mgr. Martina Ivičičová.

[5] G. Neff, P. Nagy, Talking to Bots: Symbiotic Agency and the Case of Tay, International Journa of Communication. 10 (2016) 4915-31.

[6] ELIZA. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation,2001-[cit.2021-5-6]. Dostupné z: https://en.wikipedia.org/wiki/ELIZAhttps://onlim.com/en/the-history-of-chatbots/

[7] Chatbot History: What is Dr. Sbaitso. Yakbots [online]. Sean B, 2020 [cit. 2021-5-6]. Dostupné z: https://yakbots.com/chatbot-history-what-is-dr-sbaitso/https://www.chatbots.org/chatterbot/smarterchild/

[8] A Computer Called Watson. IBM [online]. Dr. David Ferrucci, 2009 [cit. 2021-5-6]. Dostupné z: https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/

[9] A History of Chatbots [online]. ChatBot Pack, 2019 [cit. 2021-5-6]. Dostupné z: https://www.chatbotpack.com/a-history-of-chatbots/

[10] A brief history of Chatbots. Chatbotslife [online]. Mitusha Arya, 2019 [cit. 2021-5-6]. Dostupné z: https://chatbotslife.com/a-brief-history-of-chatbots-d5a8689cf52f

[11] Chatbot History: The Mitsuku Chatbot. Yakbots [online]. Sean B, 2020 [cit. 2021-5-6]. Dostupné z: https://yakbots.com/chatbot-history-the-mitsuku-chatbot/

[12] Virtual Assistants: Microsoft Cortana Chatbot. Yakbots [online]. Sean B, 2020 [cit. 2021-5-6]. Dostupné z: https://yakbots.com/virtual-assistants-microsoft-cortana-chatbot/

[13] Virtual Assistants: Amazon Alexa Chatbot. Yakbots [online]. Sean B, 2020 [cit. 2021-5-6]. Dostupné z: https://yakbots.com/virtual-assistants-amazon-alexa-chatbot/

[14] Develop Skills in Multiple Languages [online]. USA: Amazon.com, 2020 [cit. 2021-5-8]. Dostupné z: https://developer.amazon.com/en-US/docs/alexa/custom-skills/develop-skills-in-multiple-languages.html

[15] Chatbot History: Facebook Messenger Chatbots. Yakbots [online]. Sean B, 2020 [cit. 2021-5-6]. Dostupné z: https://yakbots.com/chatbot-history-facebook-messenger-chatbots/

[16] The Complete Guide to Using Facebook Messenger Bots for Business. Hootsuite [online]. Paige Cooper, 2019 [cit. 2021-5-6]. Dostupné z: https://blog.hootsuite.com/facebook-messenger-bots-guide/

[17] Facebook for Developers [online]. Facebook, 2021 [cit. 2021-5-8]. Dostupné z: https://developers.facebook.com/docs/messenger-platform//

[18] Já, chatbot. Casopis.fit.cvut [online]. Tomáš Nováček, 2018 [cit. 2021-5-6]. Dostupné z: https://casopis.fit.cvut.cz/deni-na-fit/ja-chatbot/

[19] The Complete Beginner's Guide To Chatbots. Chatbotsmagazine [online]. Matt Schlicht, 2016 [cit. 2021-5-6]. Dostupné z: https://chatbotsmagazine.com/the-complete-beginner-s-guide-to-chatbots-8280b7b906ca

[20] Co je chatovací robot? [online]. Microsoft, 2021 [cit. 2021-5-8]. Dostupné z: https://powervirtualagents.microsoft.com/cs-cz/what-is-a-chatbot/

[21] Is voice activated chatbot better than the text-based chatbot? Chatbotsmagazine [online]. James grills, 2019 [cit. 2021-5-6]. Dostupné z: https://chatbotsmagazine.com/is-voice-activated-chatbot-better-than-the-text-based-chatbot-7230e9161620

[22] Why Voice-Enabled Chatbot Is The Future Of Internet? Chatbotsjournal [online]. Mohit Dua, 2019 [cit. 2021-5-6]. Dostupné z: https://chatbotsjournal.com/why-voice-enabled-chatbot-is-the-future-of-internet-29326af705d3

[23] The ManyChat Review – What is it and is it Worth Using? Automationagency [online]. Carl Taylor, 2019 [cit. 2021-5-6]. Dostupné z: https://automationagency.com/the-manychat-review-what-is-it-and-is-it-worth-using/

[24] Choose a ManyChat plan that's right for you [online]. ManyChat, 2021 [cit. 2021-5-8]. Dostupné z: https://manychat.com/pricing

[25] WINGBOT [online]. wingbot.ai, 2021 [cit. 2021-5-8]. Dostupné z: https://wingbot.ai/about/

[26] Increase sales, reduce costs, and automate support [online]. chatfuel, 2021 [cit. 2021-5-8]. Dostupné z: https://chatfuel.com/

[27] Start free or go Pro if you're ready to GROW [online]. chatfuel, 2021 [cit. 2021-5-8]. Dostupné z: https://chatfuel.com/pricing

[28] For companies big and small [online]. LiveChat, 2021 [cit. 2021-5-8]. Dostupné z: https://www.chatbot.com/pricing/

[29] Introduction to the Messenger Platform [online]. USA: Facebook, 2021 [cit. 2021-5-8]. Dostupné z: https://developers.facebook.com/docs/messenger-platform/introduction/

[30] Pymessenger. Github [online]. davidchua, 2019 [cit. 2021-5-6]. Dostupné z: https://github.com/davidchua/pymessenger

[31] Wrapper. Techterms [online]. Sharpened Productions, 2019 [cit. 2021-5-6]. Dostupné z: https://techterms.com/definition/wrapper#:~:text=In%20computer%20science%2C%20a%20wrapper,function%20wrappers%2C%20and%20driver%20wrappers.

[32] Buttons [online]. Facebook, 2021 [cit. 2021-5-8]. Dostupné z: https://developers.facebook.com/docs/messenger-platform/send-messages/buttons

[33] Generic Template [online]. Facebook, 2021 [cit. 2021-5-8]. Dostupné z: https://developers.facebook.com/docs/messenger-platform/send-messages/template/generic/

[34] Quick Replies [online]. Facebook, 2021 [cit. 2021-5-8]. Dostupné z: https://developers.facebook.com/docs/messenger-platform/send-messages/quick-replies/

[35] Media Template [online]. Facebook, 2021 [cit. 2021-5-8]. Dostupné z: https://developers.facebook.com/docs/messenger-platform/send-messages/template/media/

[36] Button Template [online]. Facebook, 2021 [cit. 2021-5-8]. Dostupné z: https://developers.facebook.com/docs/messenger-platform/send-messages/template/button/

[37] What is Python? Executive Summary [online]. Python Software Foundation, 2021 [cit. 2021-5-8]. Dostupné z: https://www.python.org/doc/essays/blurb/

[38] What is MSSQL? [online]. Atlantic.Net, 2021 [cit. 2021-5-8]. Dostupné z: https://www.atlantic.net/vps-hosting/what-is-mssql/

[39] Vyzkoušejte SQL Server v místním prostředí nebo v cloudu [online]. Microsoft, 2021 [cit. 2021-5-8]. Dostupné z: https://www.microsoft.com/cs-cz/sql-server/sql-server-downloads

[40] What is ngrok? [online]. PubNub, 2021 [cit. 2021-5-8]. Dostupné z: https://www.pubnub.com/learn/glossary/what-is-ngrok/

[41] What is ngrok? [online]. NGROK, 2021 [cit. 2021-5-8]. Dostupné z: https://ngrok.com/product

[42] User's Guide [online]. Pallets, 2010 [cit. 2021-5-8]. Dostupné z: https://flask.palletsprojects.com/en/1.1.x/

[43] What is Flask Python [online]. https://pythonbasics.org, 2021 [cit. 2021-5-8]. Dostupné z: https://pythonbasics.org/what-is-flask-python/

[44] SMTP protocol Explained (How Email works?). Afternerd [online]. Karim, 2019 [cit. 2021-5-6]. Dostupné z: https://www.afternerd.com/blog/smtp/

[45] Green Unicorn (Gunicorn). Fullstackpython [online]. Matt Makai, 2021 [cit. 2021-5-6]. Dostupné z: https://www.fullstackpython.com/green-unicorn-gunicorn.html

[46] Pyodbc. Pypi [online]. mkleehammer, 2020 [cit. 2021-5-6]. Dostupné z: https://pypi.org/project/pyodbc/

[47] ENGEL, David, Rothja, Craigg MSFT a Carl RABELER. What Is ODBC? Docs.microsoft [online]. Microsoft, 2017 [cit. 2021-5-6]. Dostupné z: https://docs.microsoft.com/en-us/sql/odbc/reference/what-is-odbc?view=sql-server-ver15

[48] ENGEL, David, Saisang, Craigg MSFT, PRMerger20, arob98, LowlyDBA a Rekojeht. Step 3: Proof of concept connecting to SQL using pyodbc. Docs.microsoft [online]. Microsoft, 2021 [cit. 2021-5-6]. Dostupné z: https://docs.microsoft.com/en-us/sql/connect/python/pyodbc/step-3-proof-of-concept-connecting-to-sql-using-pyodbc?view=sql-server-ver15

[49] What is NGINX? [online]. F5, 2021 [cit. 2021-5-8]. Dostupné z: https://www.nginx.com/resources/glossary/nginx/

[50] What Is A Reverse Proxy? | Proxy Servers Explained. Cloudflare [online]. Cloudflare [cit. 2021-5-6]. Dostupné z: https://www.cloudflare.com/learning/cdn/glossary/reverse-proxy/

[51] A nonprofit Certificate Authority providing TLS certificates to 260 million websites. [online]. USA: Internet Security Research Group (ISRG)., 2021 [cit. 2021-5-8]. Dostupné z: https://letsencrypt.org/

[52] Welcome to Freedom. [online]. Red Hat, 2021 [cit. 2021-5-8]. Dostupné z: https://getfedora.org/en/

[53] Nishanil a Olprod. Úvod do kontejnerů a Docker. Docs.microsoft [online]. Microsoft, 2020 [cit. 2021-5-6]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/architecture/containerized-lifecycle/introduction-to-containers-and-docker

[54] Nishanil a Olprod. Co je Docker? Docs.microsoft [online]. Microsoft, 2020 [cit. 2021-5-6]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/architecture/containerized-lifecycle/what-is-docker

[55] The good, the bad, and Let's Encrypt. Catalyst2 [online]. The catalyst2 team [cit. 2021-5-8]. Dostupné z: https://www.catalyst2.com/blog/good-bad-lets-encrypt/

[56] After Data Breach Exposes 530 Million, Facebook Says It Will Not Notify Users. Npr [online]. EMMA BOWMAN, 2021 [cit. 2021-5-8]. Dostupné z: https://www.npr.org/2021/04/09/986005820/after-data-breach-exposes-530-million-facebook-says-it-will-not-notify-users?t=1620478600770

[57] Turingův test. Wikipedia [online]. [cit. 2021-5-11]. Dostupné z: https://en.wikipedia.org/wiki/Turing_test#/media/File:Turing_test_diagram.png

[58] Chinese-room. Wikipedia [online]. [cit. 2021-5-11]. Dostupné z: https://cs.wikipedia.org/wiki/Argument_%C4%8D%C3%ADnsk%C3%A9ho_pokoje#/media/Soubor:2-chinese-room.jpg

[59] Eliza. Wikipedia [online]. [cit. 2021-5-11]. Dostupné z: https://cs.wikipedia.org/wiki/ELIZA#/media/Soubor:GNU_Emacs_ELIZA_example.png

[60] When Parry met Eliza. Theatlantic [online]. [cit. 2021-5-11]. Dostupné z: https://www.theatlantic.com/technology/archive/2014/06/when-parry-met-eliza-a-ridiculous-chatbot-conversation-from-1972/372428/

[61] The Mitsuku Chatbot. Yakbots [online]. [cit. 2021-5-11]. Dostupné z: https://yakbots.com/chatbot-history-the-mitsuku-chatbot/

[62] Introduction to the Messenger Platform. Developers.facebook [online]. [cit. 2021-5-11]. Dostupné z: https://scontent-frt3-1.xx.fbcdn.net/v/t39.8562-6/64382845_2370704119653345_4919414098698960896_n.png?_nc_cat=102&ccb=1-3&_nc_sid=6825c5&_nc_ohc=eI90so0iEhgAX-juLT9&_nc_ht=scontent-frt3-1.xx&oh=f4834efed4efa3a3a4769733b38f4e12&oe=60BFD7FB

[63] Ngrok_schema. Atwix [online]. [cit. 2021-5-11]. Dostupné z: https://www.atwix.com/wp-content/uploads/2017/11/ngrok_schema.png

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PSC	Programming Support Centre
API	Application Programming Interface
URL	Uniform Resource Locator
MSSQL	Microsoft Structured Query Language
SMTP	Simple Mail Transfer Protocol
FAQ	Frequently asked questions
ERD	Entity Relationship Diagram
WSGI	Web Server Gateway Interface
OBDC	Open Database Connectivity
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
AIML	Artificial Intelligence Markup Language
XML	Extensible Markup Language
HTTPS	Hypertext Transfer Protocol Secure

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: OBSAH CD

Obsah CD:

- BC_SOUBORY_KOD – Obsahuje jednotlivé soubory s kódem Messenger aplikace
- BC_PSC_Diagram_DB – Obsahuje EA soubor s kompletním návrhem chatbota