

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta



Kvízový přístupový systém pro studenty

Bakalářská práce

Martin Jůda

Vedoucí práce: Mgr. Jakub Geyer

České Budějovice 2021

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Martin Jůda
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika

Katedra/ústav PŘF JU: Ústav aplikované informatiky

Školitel: Mgr. Jakub Geyer
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PŘF JU:
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma bakalářské práce: Kvízový přístupový systém pro studenty

Cíle práce:


Cílem práce je vytvořit a nasadit systémové řešení, rozšiřující stávající přístupový systém univerzity o další způsob autentizace. Tento systém umožní studentům přístup na základě identifikace (stávajícím RFID čipem) a dalším mechanismem zajišťujícím více-faktorovou autentizaci (zodpovězení kvízové otázky ze studijního programu studenta, se zohledněním ročníku či absolvovaných předmětů). Systém bude napojen přes webové API na studijní agendu STAG, ze které budou na základě studijního čísla studenta zjišťovány podrobnosti o studentovi (včetně podkladů pro výběr kvízové otázky). Systém bude fungovat paralelně se stávajícím přístupovým systémem a umožní nahrání potřebných dat z přístupového systému při zajištění bezpečnosti přenosu a jejich uchování. Systém bude rovněž zahrnovat centrální databázi otázek a odpovědí z jednotlivých předmětů a umožňovat jejich aktualizaci na cílovém přístupovém zařízení. Student navrhne vhodné hardwarové řešení, vytvoří příslušný software a realizuje nasazení 1 pilotního zařízení. Výsledný systém umožní rozšíření stávajícího přístupového systému o vlastní řešení s více-faktorovou autentizací (s možností budoucího rozšíření o další technologie), který z pedagogického hlediska nabídne studentům „hravou formou“ možnost procvičování látky absolvovaných předmětů, prostřednictvím začlenění kvízu do běžné činnosti (integrací do přístupového systému).

Součástí práce budou bezpodmínečně následující dokumenty:

- *Analýza funkčních a nefunkčních požadavků*
- *Analýza a výběr vhodných hardwarových a softwarových prostředků*
- *Dokumentace kódu*
- *Testovací dokumentace*
- *Uživatelská dokumentace*

Základní doporučená literatura:

Financování práce UA1 - 101

Školitel práce podpis: 

U externích vedoucích fakultní garant práce podpis:

Garant oboru bak. studia (nepožaduje se u oboru biologie) podpis:

Vedoucí katedry/ústavu PŘF JU, kde proběhne obhajoba podpis:

Případný souhlas vedoucího ústavu AV podpis:

V Českých Budějovicích dne 13. 11. 2020 Podpis studenta: 

Bibliografické údaje

Jůda M. 2021: Kvízový přístupový systém pro studenty. [The quiz access control system for the students. Bc. Thesis, in Czech.] – 61 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato bakalářská práce pojednává o vytvoření přístupového systému umožňující přístup studentům za využití více-faktorové autentizace. Více-faktorová autentizace využívá RFID čip a zodpovězení otázek z absolvovaných předmětů. Systém získává informace o studentech z IS/STAG.

Cílem práce bylo vybrání hardwaru, vytvoření softwaru a nasazení jednoho pilotního zařízení. Výsledkem této bakalářské práce je funkční přístupový systém.

Klíčová slova

IS/STAG, Python, Flask, přístupový systém, více-faktorová autentizace, Raspberry Pi, RFID

Abstract

This thesis deals with the creation of an access control system allowing students access using multi-factor authentication. Multi-factor authentication uses an RFID chip and answering a question from the subjects that a student has passed. The system acquires information about students from IS/STAG.

The goal of this thesis was to select hardware, create software and deploy one pilot device. The result of this thesis is a functional access control system.

Keywords

IS/STAG, Python, Flask, access control system, multi-factor authentication, Raspberry Pi, RFID

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval(a) pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

v České Budějovice dne 7.4.2021

Podpis autora

jiří

Poděkování

Rád bych poděkoval vedoucímu práce panu Mgr. Jakubu Geyerovi za jeho expertní vedení práce. Také bych rád poděkoval panu Václavu Kubákovi za jeho pomoc a ochotu při seznamování s již existujícím přístupovým systémem.

Obsah

Úvod	1
Cíl práce	2
1 Analýza	3
1.1 Funkční požadavky	3
1.2 Nefunkční požadavky	4
1.3 Metodika vývoje	4
1.4 Aktuální přístupový systém	5
1.5 Hardwarové prostředky stávajícího přístupové systému využité novým systémem	5
1.5.1 Karty vydávané kartovým centrem Jihočeské univerzity	6
1.5.2 Zámek ve dveřích	6
1.6 IS/STAG	6
1.6.1 Webové služby IS/STAG	8
1.6.2 Použité webové služby IS/STAG	10
1.7 Více-faktorová autentizace v přístupových systémech	12
1.7.1 Přístupové systémy využívané na vysokých školách	12
1.7.2 Obdobné diplomové práce	13
2 Výběr hardwarového vybavení	14
2.1 Průzkum místa nasazení pilotního zařízení	14
2.2 Řídící jednotka	16
2.3 Displej	18
2.4 Relé	18
2.5 Čtečka karet	18
2.6 Kabeláž	20
3 Návrh	21
3.1 Architektura systému	21
3.2 Technologie	23
3.2.1 Platforma	23
3.2.2 Databáze	23
3.2.3 Webová aplikace	24
3.2.4 Technologie řídicí jednotky	25
3.2.5 Technologie API obsluhující požadavky řídicí jednotky	26
3.3 Wireframes	26

4 Implementace	29
4.1 Databáze	29
4.1.1 Otázky a předměty	29
4.1.2 Uživatelé a jejich práva	31
4.1.3 Logování, správa zařízení a povolené přístupy skupin studentů	32
4.2 Server	34
4.2.1 Webová aplikace na zadávání otázek	34
4.2.2 Backend REST API	36
4.3 Klient	41
4.3.1 Konfigurace klienta	41
4.3.2 Aplikace	43
4.4 Testování	48
5 Budoucí možný rozvoj přístupového systému	50
Závěr	51
Seznam použité literatury	52
Přílohy	61

Seznam použitých zkratk

JCU - Jihočeská univerzita v Českých Budějovicích.

API - Application Programming Interface (rozhraní pro aplikace a jejich programování).

IS/STAG - Informační systém studijní agentury pro vysoké školy a univerzity.

UID - User Identifier (identifikátor uživatele)

JIS - Jednotný Identifikační Systém

ISIC - International Student Identity Card (mezinárodní studentský doklad)

ITIC - International Teachers Identity Card (mezinárodní identifikační karta učitelů a zaměstnanců škol)

RFID - Radio Frequency Identification (radiofrekvenční identifikace)

REST - Representational State Transfer

SOAP - Simple Object Access Protocol

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

MIME - Multipurpose Internet Mail Extensions („víceúčelová rozšíření internetové pošty“)

XML - Extensible Markup Language (rozšiřitelný značkovací jazyk)

JSON - JavaScript Object Notation

YAML - Ain't Markup Language

URL - Uniform Resource Locator („jednotná adresa zdroje“)

JAMU - Janáčkova akademie múzických umění v Brně

VŠPJ - Vysoká škola polytechnická Jihlava

UMPRUM - Vysoká škola uměleckoprůmyslová v Praze

RAM - Random Access Memory (paměť s náhodným přístupem)

GPIO - General Purpose Input/Output

GB - Gigabyte

Kč - Koruna česká

V - Volt

LED - Light-Emitting Diode (elektroluminiscenční dioda)

GND - earth ("ve smyslu uzemnění v rámci obvodu")

VCC - Voltage Common Collector

SPI - Serial Peripheral Interface (sériové periferní rozhraní)

I^2C - Inter-Integrated Circuit

UART - Universal asynchronous receiver-transmitter („univerzální asynchronní přijímač-vysílač“)

HDMI - High-Definition Multimedia Interface

USB - Universal Serial Bus (univerzální sériová sběrnice)

CAT - category (katogire)

STP - Shielded Twisted Pair (stíněná kroucená dvojlinka)

CRUD - Create, Read, Update, Delete (vytvoření, čtení, aktualizace, odstranění)

SFTP - SSH File Transfer Protocol

SSH - Secure Shell

GNU - General Public License

REST - Representational State Transfer

CSV - Comma-separated values (hodnoty oddělené čárkami)

ORM - Object Relational Mapping (objektově relační mapování)

VNC - Virtual Network Computing

HPND - Historical Permission Notice and Disclaimer

CAT - Category (kategorie)

IDM - Identity Management (správa identit)

Úvod

Tato bakalářská práce pojednává o vytvoření přístupového systému pro studenty. Na Jihočeské univerzitě existují oblasti s omezeným přístupem. Vstup do těchto oblastí chrání dosavadní přístupový systém. Tento systém umožňuje přístup za využití autentizace skrze identifikační kartu. Aktuálně je softwarová část systému licencovaná soukromou společností. O modifikaci přístupového systému je nutné společnost zažádat a zaplatit servisní poplatek.

Dosavadní přístupový systém nenabízel více-faktorovou autentizaci. Z tohoto důvodu bylo rozhodnuto, že nově navržené řešení pro studenty bude obsahovat identifikaci na bázi přístupové karty a zodpovězení kvízové otázky z povinných a povinně volitelných předmětů, které student již absolvoval. Tímto způsobem se dosáhne opakování látky nejen v době výuky, ale bude začleněno i do interaktivní praktické podoby. Výsledný přístupový systém bude ve vlastnictví Jihočeské univerzity. Tímto bude zajištěn případný budoucí rozvoj Jihočeskou univerzitou dle potřeb a nebude nutné platit servisní poplatky.

Navržený způsob více-faktorové autentizace byl zvolen, protože z didaktického hlediska je vhodné, aby si student průběžně opakoval naučenou látku. Dochází tak k ukládání poznatků do dlouhodobé paměti. V dlouhodobé paměti se ukládají významné zkušenosti, například poznatky nutné k vykonávání určité činnosti. Pro neefektivnější zapamatování naučených znalostí je třeba využít krátkodobou a dlouhodobou paměť. K tomu je nutné opakování dané látky [[1], s. 28].

Možným následkem této autentizace bude snadnější získávání nových znalostí, které stavějí na základě již naučené látky. Zároveň tento přístup může vést ke zjednodušení studentovy přípravy na státní závěrečnou zkoušku. Hlavním přínosem této práce je poznatek o tom, jak lze navrhnout přístupovým systémem využívající podklady z IS/STAG.

Práce je rozdělena do 5 kapitol. V kapitole 1 jsou popsány funkční a nefunkční požadavky. V této kapitole je také popsána metodika vývoje a jsou představeny hardwarové prvky, které byly přeneseny do navrženého přístupového systému ze stávajícího přístupového systému. V poslední části této kapitoly jsou představeny využití funkce IS/STAG a je provedena analýza stavu přístupových systémů na ostatních veřejných vysokých školách. Kapitola 2 popisuje výběr nového hardwaru. V kapitole 3 je představena architektura systému a použité technologie, včetně návrhu grafického rozhraní pro aplikaci, kterou studenti využívají v rámci přístupového systému. Kapitola 4 popisuje implementaci částí přístupového systému a obsahuje popis testování jednotlivých částí systému. Kapitola 5 obsahuje popis možného budoucího rozšíření vyvinutého systému.

Cíl práce

Cílem práce je vytvořit a nasadit systémové řešení, které rozšíří stávající přístupový systém univerzity o další způsob autentizace. Tento systém umožní studentům přístup na základě identifikace současným čipem karet, vydávaných kartovým centrem JCU, a zodpovězením kvízové otázky z absolvovaných předmětů studenta. Systém bude zjišťovat informace o studentovi přes webové API IS/STAG, včetně podkladů pro výběr kvízové otázky. Výsledný produkt této práce bude fungovat paralelně se stávajícím přístupovým systémem a umožní nahrání potřebných dat z přístupového systému při zajištění bezpečnosti přenosu a jejich uchování. Systém bude rovněž zahrnovat centrální databázi otázek a odpovědí z jednotlivých předmětů a umožňovat jejich aktualizaci na cílovém přístupovém zařízení. Výsledný systém umožní rozšíření stávajícího přístupového systému o vlastní řešení s více-faktorovou autentizací s možností budoucího rozšíření o další technologie.

Součástí práce bude vybrání vhodného hardwarového řešení pro tento projekt. Bude vytvořen příslušný software a realizováno jedno pilotní zařízení na Ústavu aplikované informatiky Přírodovědecké fakulty Jihočeské univerzity v Českých Budějovicích.

1 Analýza

Tato kapitola pojednává o funkčních a nefunkčních požadavcích, které byly obdrženy od Ústavu aplikované informatiky. Kapitola obsahuje metodiku vývoje a jsou zde popsány prostředky, na kterých nový přístupový systém staví. Dále v této kapitole jsou uvedeny informace o stavu přístupových systémů na ostatních veřejných vysokých školách a obdobné projekty.

1.1 Funkční požadavky

1. Klient

- Přečtení UID karty.
- Vzdálené nahrávání souborů do zařízení.
- Možnost vizuálně zobrazit otázku a možné odpovědi.
- Po zvolení odpovědi vždy ukázat správnou odpověď.
- Při zvolení správné odpovědi odemknout dveře.
- Při vybrání špatné odpovědi uživateli zobrazit další otázku, pokud nebylo dosaženo předem nastaveného maximálního počtu otázek. U poslední otázky musí vždy dojít k odemknutí dveří, neohledně na správnost odpovědi.
- Možnost použití obrázku k otázce.
- V případě výpadku systému musí být uživateli sdělena nouzová zpráva.
- Volba odpovědi k otázce pomocí dotykového displeje.

2. Server

- Omezení vstupu do chráněných oblastí dle studijního programu studenta.
- Získání předmětů konkrétního studenta z IS/STAG za celou dobu studování určitého studijního programu.
- Systém bude pracovat pouze s povinnými a povinně volitelnými předměty.
- Předměty, pro které se budou vybírat otázky, budou vybrány náhodně.
- V případě, že student neabsolvoval žádný předmět, se vyberou otázky ze základní sady otázek.
- V případě výskytu chyby se vrátí typ chyby zařízení.
- Uložení záznamu o užití přístupového systému studentem.
- Management přístupového systému.
 - Správa otázek

- Správa uživatelů.
- Správa předmětů.
- Přiřazení studijních programů ke konkrétnímu zařízení.
- Prohlížení záznamů obsahující informace o pokusu užití zařízení konkrétním uživatelem.
- Správa zařízení, včetně nastavení, které určuje, jaké studijní programy smí zařízení pustit.
- Správa studijních programů.

1.2 Nefunkční požadavky

1. Kompatibilita s již používanými hardwarovými prvky aktuálního přístupového systému na Ústavu aplikované informatiky.
2. Otázky uchovávané v databázi.
3. Záznamy o pokusu užití přístupového systému studentem uložené v databázi.
4. Rozdělení aplikace na klientskou a serverovou část.
5. Česká lokalizace grafické podoby aplikace.

1.3 Metodika vývoje

Pro vývoj přístupového systému byla zvolena metodika vývoje typu Rapid Application Development. RAD model je flexibilní a přizpůsobivý změnám, protože jeho vyvíjecí cykly jsou krátké. Díky těmto vlastnostem mohou klienti vidět RAD produkt v krátkém časovém intervalu a zhodnotit jeho stav. Tento model může být aplikován na širokou škálu projektů, jako je vytváření nových systémů a vylepšování existujících systémů [2]. V průběhu vývoje byly jednotlivé prototypy konzultovány s vedoucím práce, který zastupuje Ústav aplikované informatiky. Konzultace probíhaly nejen s vedoucím práce, ale i s kartovým centrem, které je zodpovědné za jednotný identifikační a přístupový systém.

Ověřování s klientem by mělo probíhat ve fázích specifikace, plánování a designu [3]. Z těchto jednotlivých kroků by mělo vyplynout, že konečné specifikace povedou k vhodnému naplnění cílů práce.

1.4 Aktuální přístupový systém

Momentální přístupový systém byl dodán společností ekoTIP ID s.r.o. Hardware systému je ve vlastnictví Jihočeské univerzity. Softwarové vybavení bylo bezúplatně zapůjčeno skrze licenci. V případě modifikací je nutné jejich vyžádání a zaplacení servisního poplatku.

Přístupový systém provozovaný na univerzitě se nazývá Jednotný Identifikační Systém. Tento systém pracuje s čipy karet, které operují na 125 kHz a 13,56 MHz. Systém využívá autentizaci pouze na bázi vlastnictví přístupové karty, která se přikládá ke čtečce karet. Po autentizaci dojde k odemknutí zámku autorizovanému uživateli.

Nyní je systém integrován i do jiných systémů na JCU a to do:

- Stravovacího systému
- Studijní agendy
- Knihovního systému
- Správy tiskových a kopírovacích služeb

Systém umí pracovat s několika druhy čipů:

- H4002
- e5560
- e5561
- MIFARE
- DESFire

Aplikace systému JIS funguje na platformě Linux a instaluje se na vyčleněném IBM kompatibilním serveru [4].

1.5 Hardwarové prostředky stávajícího přístupového systému využité novým systémem

V návaznosti na nefunkční požadavek byla udělána analýza hardwaru využívaného JIS. Důvodem této analýzy bylo zajištění kompatibility a možného propojení už existujících částí s nově pořízeným vybavením.

1.5.1 Karty vydávané kartovým centrem Jihočeské univerzity

Na Jihočeské univerzitě se používají 3 typy karet, které jsou studentům a pracovníkům vydávány za licenční poplatek. Jedná se o karty:

- ISIC
- ITIC
- JU

Typ ISIC je určen studentům. ITIC je vydáván učitelům. Karta JU je určená studentům a zaměstnancům Jihočeské univerzity [5]. Z informací od kartového centra bylo zjištěno, že aktuálně vydávané karty fungují na technologii RFID.

Dle veřejně dostupné informace bylo zjištěno, že v kartách ISIC a ITIC se používají dva čipy, a to [6]:

- Mifare 1Kb
- EM-Marine [EM4102]

Od kartového centra JCU byl tento údaj potvrzen. Zároveň byla poskytnuta informace o kartách typu JU. V tomto typu karet se nachází jeden čip Mifare 1Kb.

1.5.2 Zámek ve dveřích

Na Ústavu aplikované informatiky Jihočeské univerzity se u vstupních dveří používá zámek FAB BeFo PROFI Standard [11211] od společnosti Assa Abloy. Jedná se o elektromechanický zámek umožňující odemknutí dveří při přivedení napětí o 12V [[7], s. 95].

1.6 IS/STAG

Plným názvem se jedná o Informační systém studijní agendy pro vysoké školy a univerzity. Systém byl vytvořen a je nadále vyvíjen Centrem informatizace a výpočetní techniky - Střediskem informačních systémů na Západočeské univerzitě v Plzni. Systém je určený pro administraci studia, nikoliv jeho podporu. Systém je využíván na 14 vysokých a vyšších odborných školách v České republice k jaru roku 2021 [8].

Do systému jsou ukládána osobní i univerzitní data o studentech a zaměstnancích dané školy. Jednotlivá data jsou přiřazena ke konkrétní osobě na základě unikátního identifikátoru. Tímto identifikátorem je řetězec skládající se ze znaku a čísel.

V této studijní agendě jsou vedena data, která lze využít pro přístupový systém. Zejména zajímavými údaji jsou:

- Absolvované předměty studenta.
- Řetězec, který slouží jako unikátní identifikátor studenta.
- Unikátní hexadecimální číslo karty.
- Kód studijního programu, který student studuje.

Číslo karty je uživateli přidáno do systému v momentě převzetí karty od kartového centra Jihočeské univerzity. Předměty nesou informaci, který semestr a rok byly zapsány. Zároveň je uvedeno, o jaký typ předmětu se jedná.

Na JCU existují tři typy předmětů:

- A - Povinné předměty
- B - Povinně volitelné předměty
- C - Volitelné předměty

Absolvování povinných předmětů je podmínkou pro úspěšné dokončení daného studijního programu. U povinně volitelných předmětů musí student získat určitý počet kreditů ze skupiny povinně volitelných předmětů daného studijního programu [[9], článek 6 bod 3]. Volitelné předměty si student zapisuje dle svého uvážení a jejich úspěšné zakončení není vyžadováno pro splnění studijního programu.

Pracoviště / Zkratka	UAI / 714	Akademický rok	2020/2021
Název	Ochrana osobních údajů (GDPR)	Způsob zakončení	Zápočet
Akreditováno / Kredity	Ano, 3 Kred.	Forma zakončení	-
Rozsah hodin	Seminář 30 [HOD/SEM]	Zápočet před zkouškou	NE
Obs/max	Statut A Statut B Statut C	Automatické uznávání zápočtu před zkouškou	Ne
Letní semestr	8 / - 0 / - 0 / -	Počítán do průměru	NE
Zimní semestr	0 / - 0 / - 0 / -	Opakovaný zápis	NE
Rozvrh	Ano	Vyučovaný semestr	Letní semestr

Obrázek 1: Pohled studenta na předmět v IS/STAG

Na obrázku 1 je znázorněno, jak student vidí konkrétní předmět v IS/STAG. Červeným čtvercem jsou znázorněny dříve zmíněné informace. Zkratka pracoviště se zkratkou předmětu slouží jako identifikátor konkrétního předmětu. Nelze používat pouze zkratku předmětu pro vyhledávání předmětů, protože existují předměty s duplicitní zkratkou.

Pro přístup k IS/STAG lze využít tyto přístupy:

- Nativní klient
- Portálový přístup
- Webové služby

Prvním možným přístupem je nativní klient pro operační systém Windows. Tento přístup umožňuje dělat administrativní úkony a je spíše vyhraněn specializovaným uživatelům [10]. Portálový přístup slouží primárně učitelům a studentům. Uživatelům zpřístupňuje jejich data dle role. Role zastupuje skupinu oprávnění, které uživatel má v rámci IS/STAG [11]. Dalším přístupem jsou webové služby. Tyto služby umožňují přístup k informacím z IS/STAG jiným systémům [12]. Webové služby jsou ideálním přístupem a zdrojem informací pro externí systémy.

1.6.1 Webové služby IS/STAG

Tento přístup byl vybrán pro získávání podkladů přístupovým systémem. Komunikaci lze realizovat dvěma rozhraními. Prvním rozhraním je SOAP a druhým REST. SOAP rozhraní je běžně využíváno velkými informačními systémy, zatímco REST slouží pro snadný přístup k webovým službám [[13], s. 7].

Veškeré webové služby komunikují protokolem HTTPS, který zajišťuje šifrovaný přenos dat mezi klientem a serverem. Část služeb je přístupná veřejnosti, bez nutné autentizace. Služby, které poskytují citlivá data, jako je jméno studenta, identifikační číslo a další osobní údaje, vyžadují autentizaci. Služby, které nevyžadují autentizaci, mohou nabízet další informace po autentizování uživatele. Pro přenos přihlašovacích údajů u obou typů rozhraní se používá HTTP mechanismus HTTP BASIC [[13], s. 15].

- **HTTP BASIC**

Tento způsob autentizace je zabudován do HTTP protokolu. V prvním kroku klient odešle HTTP požadavek. V požadavku se nachází hlavičky. Hlavička určující způsob autentizace nese název *Authorization*. Zadává se v následujícím formátu: *Authorization: Basic*. Následuje řetězec kódovaný pomocí Base64, který obsahuje přihlašovací údaje. Názorná ukázka:

```
Authorization: Basic dGVzdDp0ZXN0
```

Problém v tomto způsobu autentizace spočívá v kódování Base64. Base64 lze snadno dekodovat. Pro zabezpečení takového způsobu autentizace je nutné využívat kromě mechanismu kódování i zašifrovaný přenos pomocí protokolu HTTPS [14].

Výstupní formát dat z webových služeb IS/STAG lze zvolit dvěma způsoby. Prvním způsobem je využití hlaviček v HTTP požadavku. Konkrétně se jedná o hlavičku `Accept`, v níž se může uvést i více formátů za sebou. Druhá forma žádání o konkrétní formát probíhá formou parametru. Tento parametr se přidává do URL volané služby. Parametr se nazývá *outputFormat* [[13], s. 10].

Aktuálně podporované výstupní formáty s těmito MIME typy jsou [[13], s. 14-15]:

- XML
text/xml, application/xml, application/*+xml
- JSON
*/json, */x-json, */javascript, */x-javascript
- YAML
text/yaml
- CSV
text/csv
- XLSX
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

Pro využívání webových služeb IS/STAG jinými systémy je doporučeno vyžadovat datové formáty typu XML, JSON a YAML. Nedoporučuje se využívat ostatní formáty. Jedná se totiž o tabulkové formáty, které jsou především určeny pro čtení lidmi. Hlavní zápor tabulkových formátů spočívá v negarantované neměnnosti datové struktury do budoucna. Generování těchto formátů na straně serveru spotřebovává více prostředků. Toto vede k delšímu zpracování a odpovědi ze strany serveru [[13], s. 11].

IS/STAG obsahuje různé role uživatelů. Každý uživatel může mít více rolí. Všechny role mají specifická oprávnění v systému a určité možnosti. Z tohoto důvodu je při každé operaci v systému třeba zcela jasně vědět, z jaké role chce uživatel danou funkci provést [[13], s. 15-16].

Ve vyvinutém systému se používá pro komunikaci s IS/STAG Jihočeské univerzity rozhraní typu REST. Toto rozhraní bylo vybráno, protože umožňuje jednoduchou implementaci tohoto způsobu komunikace přes HTTPS protokol. Dokumentace všech endpointů rozhraní REST je k dispozici ve formátu WADL a experimentálně ve formátu OpenAPI [[13], s. 7]. Dokumentaci jednotlivých služeb rozhraní REST na Jihočeské univerzitě lze nalézt na tomto odkazu: https://stag-ws.jcu.cz/ws/web?pp_locale=cs&selectedTyp=REST&pp_reqType=renderer&pp_page=serviceList

Rozhraní REST na Jihočeské univerzitě nese název REST2, protože se jedná o druhou verzi poskytovaných služeb tímto rozhraním. Webové služby lze volat následujícím způsobem:

<https://stag-ws.jcu.cz/ws/services/rest2/help/testService?outputFormat=json>

Obrázek 2: Ukázka URL adresy při volání služby IS/STAG

- Červená - Adresa serveru, který poskytuje webové služby.
- Hnědá - Rozhraní, přes které se ke službě přistupuje.
- Modrá - Tato barva vyjadřuje z jaké skupiny služeb se služba volá.
- Fialová - Název konkrétní služby.
- Oranžová - Při použití parametru se za název služby doplní otazník a název parametru. V případě kombinování parametrů se za parametr s jeho hodnotou doplní znak & a zapíše se další parametr.

1.6.2 Použité webové služby IS/STAG

Po prozkoumání webových služeb byly vybrány tři služby, které byly využity v přístupovém systému. Dvě z těchto služeb vyžadují roli LOGGED. Tato role je přiřazena uživateli po přihlášení do IS/STAG. Z tohoto vyplývá, že tyto informace nejsou veřejnosti dostupné. Pro zavolání služby je nutné využít způsob autentizace HTTP BASIC. Tento způsob autentizace byl popsán v podsekcí 1.6.1.

- *getStudiaByCisloKarty*

Toto je jediná služba, která umožňuje získat informace o studentovi na základě zadání hexadecimálního čísla karty ve formátu String. Parametr, který se zadává do URL dané služby je: *cisloKarty*. Pro otestování služby bylo třeba si vyžádat číslo karty od kartového centra. Ukázka odpovědi služby ve formátu JSON:

```
{ "studentShort": [{
    "osCislo": "B18175",
    "jmeno": "Martin",
    "prijmeni": "JÚDA",
    "titulPred": null,
    "titulZa": null,
    "stav": "S",
    "userName": "judama00"
}]}
```

Odpovědi služby je seznam studií, které student má uvedené v systému. Student může mít studijních programů více. Ke každému studiu má student přiřazený jiný identifikační řetězec. V odpovědi této služby jsou důležité dva objekty. Jedná se o *osCislo* a *stav*.

Hodnota objektu *osCislo* je identifikátor konkrétního studia u konkrétního studenta. Tato hodnota se následně dá využít jinými službami. Například pro zjištění předmětů studenta. Druhým důležitým objektem je *stav*. Stav obsahuje hodnotu S, či N. Stav S indikuje, že student studuje. Tohoto lze využít pro ověření platnosti karty v přístupovém systému.

Služba *getStudentPredmetyAbsolvovalRef* byla původně zvažována pro využití navrženým přístupovým systémem. Tato služba vyžaduje speciální roli účtu. Jedná se o role *Prorektor*, *Administrátor*, *Zahraniční referentka*, *Tajemník fakulty* a *Studijní referentka*. Bylo požádáno o vytvoření účtu s jakoukoliv z dříve uvedených rolí. Zároveň bylo požádáno o vytvoření nové role pro zobrazení hodnot této služby, v případě, že by nemohlo být vyhověno první žádosti. Obě žádosti byly zamítnuty. První žádost byla zamítnuta z bezpečnostních důvodů. Druhá žádost nebyla akceptována z důvodu technické náročnosti na implementaci.

- *getPredmetyByStudent*

Tato služba byla vybrána jako alternativa k dříve zamítnuté službě. Služba vrací seznam předmětů, které si student zapsal v daném akademickém roce pro konkrétní studijní program. Vračené předměty nenesou informaci, jestli student dané předměty úspěšně dokončil. Důležitými parametry této služby jsou: *osCislo*, *semestr*, *rok*. Je nutné použít dva povinné parametry pro zavolání této služby. Jedná se o parametr *osCislo* a *semestr*. Původní implementace této služby byla pro účely přístupového systému nevyhovující. V dřívější implementaci bylo nutné specifikovat konkrétní semestr hodnotou *LS*, nebo *ZS*. V případě parametru *rok*, bylo nutno vyplnit počáteční rok akademického roku, ve kterém byly předměty zapsány. V případě, že byl parametr *rok* nevyplněn, přiřadila se hodnota reprezentující aktuální akademický rok. Takto by při každém volání služby došlo k vrácení předmětů pouze za jeden semestr v daném akademickém roce. Tento způsob používání služby by vedl k velkému počtu dotazů, což by zvýšilo zátěž na serveru. Zároveň by se prodloužil čas, který by systém potřeboval pro získání předmětů studenta.

Byla vyžádána úprava parametrů této služby. Požadavek na úpravu se týkal možnosti vrácení všech předmětů, za celou dobu studia konkrétního studijního programu. Žádosti bylo vyhověno a byla zavedena implementace akceptování nové hodnoty. Tato nová hodnota může být přiřazena parametru *semestr* i *rok*. Pokud se při volání použijí parametry *semestr* a *rok* s novou hodnotou, dojde k vrácení všech zapsaných předmětů za celou dobu studia studijního programu. Implementovanou novou hodnotou obou parametrů je *%*. Při používání hodnoty *%* je nutné mít na paměti, že do URL se hodnota zapisuje jako *%25* dle platné syntaxe URL [[15], s. 14].

Zkrácená ukázka výstupu funkce ve formátu JSON:

```
"predmetStudenta ": [
  {
    "zkratka ": "910",
    "nazev ": "Bakalářská zkouška z angličtiny",
    "katedra ": "OJZ",
    "rok ": "2020",
    "kredity ": 8,
    "uznано ": "N",
    "statut ": "A" },
```

Vráceno je pole objektů reprezentující předměty. Objekty využívané přístupovým systémem jsou: *zkratka*, *název*, *katedra* a *statut*.

- `getStudentInfo`

Tato webová služba vrací informace o zadaném studentovi. Služba vyžaduje roli LOGGED. Pro zavolání služby je nutné vyplnit parametr *osCislo* s hodnotou identifikačního čísla studenta. Z výsledků služby lze získat kód studijního oboru studenta. Tento kód lze následně využít v přístupovém systému pro ověření, zda uživatel smí použít daného klienta. Informace o kódu oboru nese klíč *"kodSp"*.

1.7 Více-faktorová autentizace v přístupových systémech

Tato sekce je rozdělena na dvě části. V první části je prozkoumáno, zda přístupové systémy na vysokých školách využívají více-faktorovou autentizaci. Ve druhé části jsou zmíněny diplomové práce s podobným zaměřením.

1.7.1 Přístupové systémy využívané na vysokých školách

V této podsekcí jsou popsány zjištěné metody autentizace u přístupových systémů vysokých škol. Školy jsou rozděleny podle toho, zda používají IS/STAG, či ne. Toto rozdělení je důležité z hlediska zjištění, zda existuje přístupový systém, který by byl napojený na IS/STAG a využíval více-faktorovou autentizaci.

Vysoké školy využívající IS/STAG

IS/STAG momentálně využívá 11 veřejných vysokých škol. U 8 vysokých škol je možné zjistit informaci o části přístupového systému z veřejně dostupných zdrojů. Všechny tyto vysoké

školy využívají přístupový systém, který pracuje s identifikační kartou jako prvkem autentizace. Více-faktorová autentizace u přístupových systémů následně zmíněných vysokých škol nebyla zjištěna. Zároveň nebylo zjištěno napojení na IS/STAG za účelem ověření identity člověka v rámci přístupového systému.

Vyjma Jihočeské univerzity [16] se jedná o:

- Západočeskou univerzitu v Plzni [17]
- Univerzitu Hradec Králové [18]
- Ostravskou univerzitu [19]
- Univerzitu Palackého v Olomouci [20]
- Technickou univerzitu v Liberci [21]
- Univerzitu Jana Evangelisty Purkyně v Ústí nad Labem [22]
- Univerzita Pardubice [23]

Vysoké školy nevyužívající IS/STAG

Podle seznamu vydaného Ministerstvem školství, mládeže a tělovýchovy České republiky byly prozkoumány školy nevyužívající IS/STAG [24]. Kromě vysokých škol JAMU, VŠPJ a UMPRUM bylo zjištěno, že přístupový systém na těchto školách využívá identifikační kartu jako součást autentizace. Používání více-faktorové autentizace v přístupových systémech daných škol nebylo zjištěno.

1.7.2 Obdobné diplomové práce

Diplomové práce s podobným zaměřením byly vyhledávány přes portál Theses. Řada prací se zabývá vytvořením přístupového systému pouze na bázi jedno-faktorové autentizace. Ve většině prací se objevovaly tyto řídicí jednotky: *Arduino*, *NetAXS-123*, *Raspberry Pi* a *ESP32*. Nebyl nalezen systém, který by byl napojen na IS/STAG a využíval více-faktorovou autentizaci pomocí kvízové otázky, která by se zjišťovala na základě absolvovaných předmětů studenta. Byla nalezena jedna práce, která se zabývá vytvořením přístupového systému s tří-faktorovou autentizací. V práci se pojednává o autentizaci pomocí RFID karty, biometrického otisku prstu a zadáním hesla známé uživateli [25].

2 Výběr hardwarového vybavení

V návaznosti na cíle práce je v této kapitole popsáno hardwarové vybavení, které bylo zakoupeno za účelem sestrojení přístupového systému. Jsou představeny konkrétní modely hardwaru, které byly pro systém zvažovány. Závěrem je popsáno, proč byla daná součástka vybrána. Bylo nutno zakoupit: řídicí jednotku, čtečku karet, relé pro ovládání dveří, dotykový displej a kabeláž na propojení a napájení. Nákup hardwarových prostředků zafinancoval Ústav aplikované informatiky Jihočeské univerzity.

2.1 Průzkum místa nasazení pilotního zařízení

Před výběrem a následným nákupem bylo prohlédnuto místo, kde má být nasazeno pilotní zařízení. Tento průzkum byl proveden za účelem zjištění, jaké vybavení by se do těchto míst dokázalo umístit a jak by bylo zabezpečené.

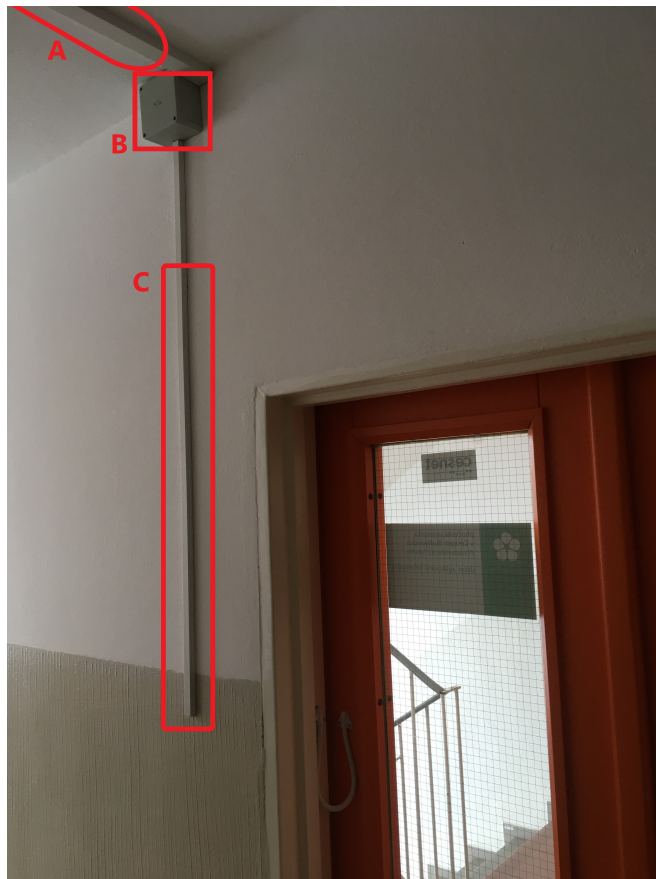


Obrázek 3: Vchod přístupný veřejnosti na Ústavu aplikované informatiky Jihočeské univerzity

Na obrázku 3 jsou vyznačena důležitá místa, kde bude fyzické nasazení realizováno.

- A - Aktuální čtečka karet. Tuto čtečku pro nový přístupový systém nelze použít, protože dokáže přenášet signál ze čtení pouze do jedné řídicí jednotky. Kabeláž čtečky je vedena z druhé strany zdi skrze vyvrtanou díru. Toto je možné si prohlédnout na obrázku 4.
- B - Zvonek na sekretariát. Tohoto zvonku lze využít v případě výpadku systému pro otevření dveří.
- C - Zamýšlené místo pro umístění nové čtečky karet.
- D - V návaznosti na funkční požadavek bylo toto místo vybráno pro umístění dotykového displeje. Bylo dbáno, aby zařízení bylo výškově dostupné uživatelům.

Po průchodu vchodem je možné vyzorovat, kde je aktuální přístupový systém pro studenty umístěn. Jednotlivá umístění si lze prohlédnout na obrázku 4.



Obrázek 4: Východ z Ústavu aplikované informatiky Jihočeské univerzity

- A - V této liště je vedeno napájení k řídicí jednotce.
- B - Bezpečnostní krabička, ve které se nachází řídicí jednotka aktuálního přístupového systému. Do této krabičky bude umístěna i nová řídicí jednotka. Toto vede ke stejné úrovni zabezpečení jako u předchozího systému.

- C - V liště je vedeno napájení pro čtečku karet. Také je zde položena komunikační linka mezi čtečkou karet a řídicí jednotkou.

2.2 Řídicí jednotka

Při výběru komponenty bylo nutné vzít v potaz určitá kritéria. Hlavními kritérii bylo, aby jednotka poskytovala rozhraní pro připojení dotykového displeje, čtečky karet a relé pro ovládání dveří. Dalším kritériem bylo, aby tato komponenta měla dostatečný výkon pro provoz aplikace. Dále jednotka musí nabízet možnost budoucího rozšíření o další hardwarové prvky.

Raspberry Pi

Jedná se o nízkonákladový jednodeskový počítač vyvinutý nadací Raspberry Pi Foundation. Je nabízen v několika modelových variantách. V závislosti na požadovaných kritériích připadaly v úvahu tyto modely: Raspberry Pi 4 Model B, Raspberry Pi 3 Model A+, Raspberry Pi 3 Model B+, protože modely jsou levné a v České republice dobře dostupné.

Z těchto modelů se do zvažovacího procesu dostal model: Raspberry Pi 4 B, protože tato varianta je nejnovější generace a pořizovací cena se příliš neliší od starších modelů. Výhodou je nový procesor, který nabízí vyšší výkon. U modelu je možné zakoupit různé velikosti RAM paměti, která je nabízena ve formě 2GB, 4GB a 8GB [[26], s. 5]. K vlastnostem GPIO pinů byly přidány alternativní periferie, které umožňují větší flexibilitu pro připojení dalších hardwarových komponent [[26], s. 9]. Výroba tohoto modelu je plánovaná nejméně do ledna roku 2026 [27].

Pro tento model jsou uvedeny specifikace přímo na stránkách výrobce. Ty se následně porovnávaly s jinými řídicími jednotkami. Na Raspberry Pi lze nainstalovat různé operační systémy. Typicky se jedná o linuxové distribuce, ale lze nainstalovat i operační systém Windows 10 IoT Core. Doporučuje se instalovat operační systém Raspberry Pi OS [dříve známý jako Raspbian] [27].

LattePanda

LattePanda je jednodeskový počítač vyvinutý firmou Zhiwei Robotics Corp. Společnost nabízí modely s označením LattePanda, LattePanda Alpha a LattePanda Delta. V úvahu byl brán pouze model s označením LattePanda, protože ostatní modely byly předimenzované pro tento druh projektu.

Hlavní výhodou tohoto jednodeskového počítače je, že dokáže poskytnout dostatečný výkon

pro zprovoznění plného operačního systému Windows 10. Nabízí se v provedení 2GB nebo 4GB RAM paměti. S tímto je spojeno i dostupné datové úložiště, které je nabízeno s kapacitou 32GB nebo 64GB. Na desce počítače lze nalézt GPIO piny. Předností tohoto počítače je nabízený výkon, který vychází ze specifikací od výrobce [28]. Zvažována varianta byla s velikostí 2GB RAM paměti a s 32 GB datovým úložištěm, protože tyto parametry byly považovány za dostatečné pro nový přístupový systém.

Orange Pi

Jedná se o jednodeskový počítač, který je vyvíjen ve formátu open-source. Na tomto počítači lze provozovat různé operační systémy. Mezi nejznámějšími jsou například Debian, Android 4.4 a Ubuntu.

Při výběru možných modelů bylo třeba brát v úvahu dostupnost tohoto počítače na trhu v České republice v době, kdy byl tento hardware vybírán. Model, který byl v dané době dostupný a splňoval kritéria, byl Orange Pi Plus2E. I na desce tohoto počítače se nachází GPIO piny. Jeho specifikace lze nalézt na webové stránce komunity, která daný počítač vyvíjí [29].

Tinker Board

Tento mikropočítač je vyvinutý společností Asus. Momentálně lze na tomto počítači provozovat pouze dva linuxové operační systémy. Jedná se operační systém Android a výrobcem doporučený a vyvinutý TinkerOS [30]. V době výběru hardwaru byl k dispozici v České republice pouze jeden model: ASUS Tinker Board. Model nabízí 2GB RAM paměti a sadu GPIO pinů. Veškeré specifikace, které daný model nabízí, se nachází na webové stránce výrobce [31].

Vybraná řídicí jednotka

Vybranou řídicí jednotkou se stal mikropočítač *Raspberry Pi 4 B*. Tento jednodeskový počítač byl vybrán, protože poskytuje dostatečný výkon pro přístupový systém a vyhovuje požadovaným kritériím, které byly zmíněny v úvodu této kapitoly. Dalším důvodem pro zakoupení byla pořizovací cena, která je nižší oproti zvažovaným konkurenčním řešením. V době nákupu se naskytl možnost zakoupit sadu výrobků určených pro Raspberry Pi společně s mikropočítačem. Jednalo se o microSD kartu microSDHC Class 10 UHS-I U1 A1 s kapacitou 32GB, která obsahovala předinstalovaný program NOOBS. Pomocí tohoto programu lze nainstalovat různé operační systémy se základními balíčky programů. Součástí sady byl i napájecí zdroj Zonepi USB-C 5,25V₌₌4A.

2.3 Displej

Dle nefunkčního požadavku byl vybrán a zakoupen dotykový displej. Pomocí tohoto displeje student může vidět otázku s odpověďmi a zvolit odpověď fyzickým dotykem. V návaznosti na vybranou řídicí jednotku bylo nutné zajistit, aby vybraný hardware byl kompatibilní. Další kritérium vycházelo z prozkoumání místa, kde má být daný hardware umístěn. Bylo nutné zajistit, aby vybraný displej mohl být nainstalován na zeď.

Byl vybrán dotykový displej od společnosti Waveshare. Jedná se o model LCD [H] s rozlišením 1024×600 [32]. Menší rozlišení by vyústilo v méně přívětivé uživatelské rozhraní aplikace, proto bylo od takových modelů při výběru upuštěno. Tento model byl vybrán, protože je plně kompatibilní s Raspberry Pi, odkud může být i napájen a jeho pořizovací cena je nižší oproti jiným řešením. Další výhodou je, že tento model nemá ochranný kryt. Tímto bylo umožněno vytisknutí obalu na 3D tiskárně, který podporuje montáž displeje na zeď.

2.4 Relé

Z analýzy už používaných hardwarových prostředků vyplynulo, že na Jihočeské univerzitě se ve většině případech používají elektromechanické zámky. Tyto zámky vyžadují určité napětí, při kterém se odemykají. Na Ústavu aplikované informatiky je pro odemčení takového zámku potřeba dodat 12V. Takové napětí nedokáže dodat řídicí jednotka sama o sobě, proto je nutné použít spínací relé, které dokáže dané napětí dodat.

Jako vhodné relé byl vybrán model JQC-3FF-S-Z, protože může být napojen na GPIO piny Raspberry Pi a pořizovací cena je nízká. Potřebné napájecí napětí je 5V [33]. Na relé se nachází tři piny pro připojení řídicí jednotky:

- GND - Místo pro připojení uzemnění
- IN - Vstup řídicího signálu
- VCC - Vstup napětí ze řídicí jednotky

2.5 Čtečka karet

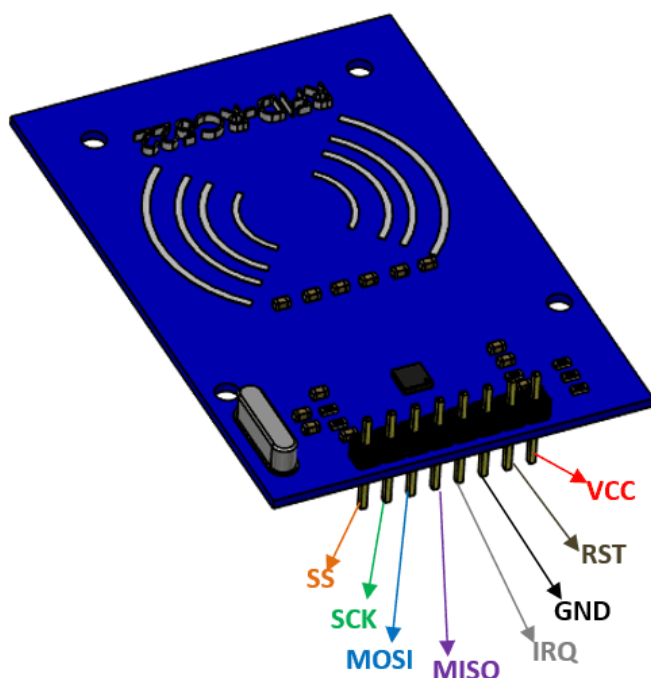
Díky analýze používaných hardwarových prostředků byla zjištěna informace, že používané karty jsou typu RFID a obsahují čip MIFARE 1Kb. Tyto čipy spadají do kategorie MIFARE Classic. Fungují na frekvenci 13,56 MHz. Data uložená na RFID kartách jsou rozdělena do 16 sektorů. Každý sektor má 4 bloky a jeden blok má velikost 16 bajtů. Pro přístup k jednotlivým sektorům je nutné se autentizovat klíčem pro daný sektor [34].

Od kartového centra byla získána informace, že první sektor čipů, vydávaných kartovým centrem JCU, je veřejně dostupný a není pro jeho čtení nutné používat autentizaci. Na prvním sektoru je uloženo unikátní sériové číslo od výrobce karet. Toto číslo je následně přiřazeno ke studentovi do IS/STAG.

Karty neobsahují interní zdroj pro své napájení. Jedná se o takzvaný pasivní RFID systém. Tento systém funguje na principu, kdy je karta napájena magnetickým, či elektromagnetickým polem čtečky karty, které je vysíláno přes anténu [[35], sekce 2.4].

Jako vhodná čtečka RFID karet byl vybrán model RC522, protože tento model vychází z modelu MFRC522 od společnosti NXP semiconductors a pořizovací cena je nízká oproti konkurenčním řešením. Společnost NXP semiconductors zároveň vyvinula čipy MIFARE. Díky tomuto je zajištěna funkčnost komunikace s danými čipy od této společnosti. Čtečka podporuje ISO/IEC 14443 A/MIFARE a NTAG [[36], s. 1].

Model RC522 podporuje komunikaci skrze sběrnice *SPI*, *I²C* a *UART*. Tento model vyžaduje napájení 3,3V a je schopen číst karty do vzdálenosti 5 centimetrů [37]. Komunikaci s danou sběrnicí a napájení lze navázat s řídicí jednotkou skrze sadu GPIO pinů. Model jednotky RC522 je přiložen jako obrázek 5 [37]. Popis pinů čtečky je zmíněn v tabulce 1 [37].



Obrázek 5: Model čtečky RC522

Číslo pinu	Název pinu	Popis
1	Vcc	Pin sloužící pro přivedení napájení. Typicky se používá 3,3V.
2	RST	Resetující pin. Tento pin se používá k vyresetování modelu nebo vypnutí.
3	Ground	Pin pro uzemnění systému.
4	IRQ	Pin pro přerušení. Slouží pro zapnutí modulu v momentě, kdy přijde zařízení do vzdálenosti čtení.
5	MISO/SCL/Tx	MISO pin je použit pro komunikaci se sběrnici SPI. Slouží jako SCL pro I2C sběrnici a jako Tx pro UART sběrnici.
6	MOSI	Master out slave in pin pro komunikaci se sběrnici SPI.
7	SCK	Pin dodávající zdroj času.
8	SS/SDA/Rx	Slouží jako sériový vstup pro komunikaci se sběrnici SPI. SDA pro I2C a Rx pro UART.

Tabulka 1: Popis pinů RC522 čtečky

2.6 Kabeláž

Pro propojení řídicí jednotky a displeje za účelem přenosu obrazu byl zvolen kabel HDMI. Tento kabel byl zakoupen o délce tří metrů s koncovkami HDMI-A a micro HDMI. Daná délka byla zvolena kvůli předpokládanému fyzickému umístění hardwarových součástí systému. Tento průzkum byl uveden v podsekcí 2.1. Pro přenos napájení a signálu o dotyku mezi displejem a řídicí jednotkou, byly použity dva kabely USB s koncovkami USB-A a Micro-USB B.

Pro propojení čtečky karty s řídicí jednotkou se očekává využití kabelu CAT6 STP. Toto provedení bylo zvoleno z důvodu minimalizace možného rušení v přenosu signálu.

3 Návrh

Tato kapitola se zabývá návrhem fungování přístupového systému. V jednotlivých částech jsou vysvětleny vybrané technologie, zvolená databáze, způsob managementu přístupového systému a jsou představeny zjednodušené návrhy grafického rozhraní aplikace provozované na Raspberry Pi.

3.1 Architektura systému

V této sekci je popsána zamýšlená architektura softwarové části přístupového systému. Dle nefunkčního požadavku byla softwarová část systému rozdělena na client-side a server-side. Takto jsou rozdělené dva prvky systému. Jako první prvek lze označit řídicí jednotku a API obsluhující její požadavky. Druhým prvkem je aplikace umožňující management přístupového systému.

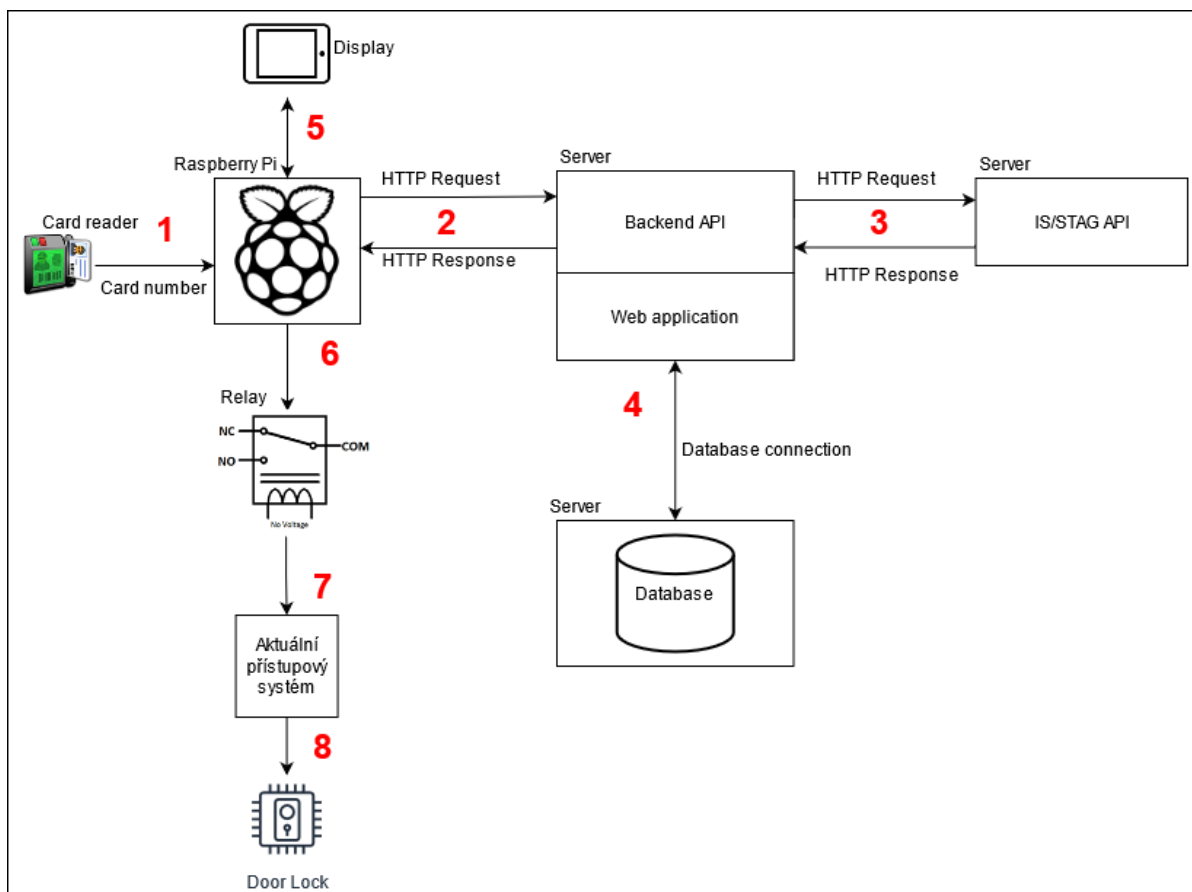
Client-side nese odpovědnost za:

- Načtení karty.
- Zobrazení otázek s odpověďmi a případnými obrázky uživateli.
- Vyhodnocení odpovědi na otázku.
- Ovládání relé pro odemknutí dveří.
- Reakci na výpadek části systému během pokusu o přístup.

Server-side odpovídá za:

- Autentizaci a autorizaci uživatele.
- Získání listu absolvovaných předmětů z IS/STAG.
- Získání listu otázek k vybraným předmětům z databáze.
- Odeslání konkrétních otázek client-side.
- Management přístupového systému.

Navrženou architekturu systému si lze prohlédnout na obrázku 6.



Obrázek 6: Popsané zamýšlené fungování systému

1. Načtení karty řídicí jednotkou.
2. Výměna informací mezi client-side a server-side.
3. Výměna informací mezi backend API a IS/STAG.
4. Komunikace databáze s backend částí.
5. Zobrazení otázky a přijmutí odpovědi.
6. Signál pro sepnutí relé .
7. Připojení k obvodu aktuálního přístupového systému.
8. Signál pro odemknutí zámku.

V produkčním nasazení se předpokládá komunikace mezi řídicí jednotkou, backend částí a IS/STAG pomocí protokolu HTTPS.

3.2 Technologie

V této sekci jsou popsány technologie, které byly použity pro vývoj přístupového systému. Před zahájením vývoje je nutné vybrat platformu, pro kterou budou jednotlivé části systému vyvinuty. Je zde popsán i výběr databáze pro přístupový systém.

3.2.1 Platforma

Raspberry Pi může provozovat několik operačních systémů. Jako operační systém byl vybrán Raspberry Pi OS, protože je oficiálním výrobcem doporučen a je nabízen zdarma. Tento operační systém je založený na architektuře operačního systému Debian a je optimalizovaný pro Raspberry Pi hardware [38]. Pro instalaci tohoto operačního systému lze využít program NOOBS. Program NOOBS umožňuje snadnou instalaci operačních systémů na Raspberry Pi.

3.2.2 Databáze

Dle nefunkčního požadavku je nutné uchovávat otázky k předmětům v databázi. K tomuto účelu byl zvolen typ relační databáze. Tento typ databáze byl zvolen, protože se jedná o jeden z nejpoužívanějších typů databází a pro účely tohoto projektu je tento typ vhodný. V úvahu připadaly databáze nabízené zdarma.

PostgreSQL

PostgreSQL je objektově-relační databáze. Tato databáze je vyvíjena jako open source projekt. Databáze má vlastní licenci zvanou PostgreSQL License. PostgreSQL je multiplatformní. Do databáze lze přidávat své vlastní datové typy a vytvářet nové funkce. PostgreSQL nabízí možnost komunikace pomocí protokolu SSL, který umožňuje zašifrovaný přenos dat [39].

MySQL

Tato databáze nabízí různé edice. Edice se liší v rámci nabízených funkčních vlastností databáze. Pro tento projekt byla zvažována komunitní edice. Komunitní edice je vydána pod GPL licencí a je vyvíjena komunitou jako open source projekt. Databáze je multiplatformní. Lze ji provozovat na více jak dvaceti platformách [40].

Microsoft SQL

Microsoft SQL nabízí dvě varianty, které jsou zdarma. Jedná se o variantu Vývojář a Express. V potaz byla brána verze Express, protože umožňuje vývoj a provozování desktopových, webových a malých serverových aplikací. Tento server lze provozovat na operačním systé-

mu Windows a Linux [41].

Vybraná databáze

Výběr zvažovaných databází vycházel ze seznamu nejlépe hodnocených databází [42]. Vybranou databází se stala PostgreSQL. Tato databáze byla vybrána, protože na tuto databázi lze využít nástroje pro vygenerování webové aplikace, pomocí které lze provozovat management přístupového systému. Další výhodou této databáze je výborná dokumentace a spousta návodů, které uživatel může využít.

3.2.3 Webová aplikace

Pro management přístupového systému byla navržena webová aplikace. V rámci návrhu bylo požadováno, aby aplikace byla přístupná pouze oprávněným osobám a bylo možné ji používat z různých zařízení. Funkcionalitu webové aplikace lze shrnout jako CRUD operace. Pro tvorbu webové aplikace s těmito funkcemi lze využít generátory aplikací, které dokážou danou aplikaci vytvořit na základě struktury databáze.

Portofino

Generátor je vyvíjen v podobě open source projektu pod licencí LGPL společností Many-Designs. Vygenerovaná aplikace tímto generátorem využívá jako front-end technologii Angular a Javascript. Backend technologie je napsána v jazyce Groovy [43]. K tomuto generátoru existuje dokumentace a návody pro jeho užití [44]. Nevýhodou je zpoplatněná podpora. Další nevýhodou jsou nefunkční odkazy na wikipedii generátoru a špatně čitelné uživatelské fórum.

Laravel Generator

Jedná se o generátor vyvíjený jako open source projekt od společnosti InfyOmLabs. Generátor je vydán pod licencí MIT. Vyvinutá aplikace tímto generátorem využívá jako backend technologii jazyk PHP. Momentálně lze využít více jak 14 balíčků, které umožňují generování různých částí aplikace [45]. Na stránkách společnosti lze nalézt dokumentaci a návody pro generování nejen samotné aplikace, ale i například test cases a další prvky usnadňující vývoj aplikace [46]. V současné době je vyvíjena možnost použít generátor jako "wizard" nástroj pro snazší vývoj aplikace uživateli. Nástroj se aktuálně nachází v beta verzi.

PostgreSQL PHP Generator

Tento generátor je vyvinut společností SQL Maestro Group. Generátor je nabízen ve dvou

variantách a to komerční a nekomerční. Komerční verze umožňuje využití všech funkcionalit. Kód generátoru je ve vlastnictví skupiny a nelze ho modifikovat. Momentálně nejnovější verze generátoru je 20.5.0.5, který generuje aplikaci do jazyka PHP 8.0.0. Společnost nabízí bezplatnou podporu uživatelům. Dále je nabízena dokumentace jednotlivých částí generátoru s následnými ukázkami možné implementace dalších funkcionalit do vygenerované aplikace. Hlavní výhodou generátoru je dokumentace vygenerovaného kódu. Generátor funguje jako "wizard" nástroj, který umožňuje uživatelům přívětivou cestou nastavit zabezpečení aplikace, vzhled vygenerovaných stránek, validaci zadávaných dat uživateli a mnoho dalších funkcionalit [47].

Vybraný generátor aplikace

Jako generátor aplikace byl použit PostgreSQL PHP Generator. Tento generátor byl vybrán, protože oproti ostatním poskytuje uživatelsky přívětivé rozhraní a větší sadu funkcí. V rámci tohoto rozhraní lze navolit požadované vlastnosti aplikace bez nutnosti implementování dalších funkcí. Další výhodou je obsáhlá dokumentace funkcí generátoru a ukázky kódu pro možnou budoucí implementaci dalších funkcionalit aplikace. Byla využita verze PostgreSQL PHP Generator Professional Edition, která umožňuje vygenerovat aplikaci za možnosti použití veškerých funkcí generátoru. Tato verze je nabízena zdarma na 30 dní od její instalace. Následně vygenerovaná aplikace je v plném vlastnictví uživatele a vypršení licence generátoru nemá vliv na funkcionalitu vygenerované aplikace.

3.2.4 Technologie řídicí jednotky

Pro vývoj aplikace na řídicí jednotce byl vybrán jazyk Python. Tento programovací jazyk byl vybrán, protože je jednoduché se ho naučit a je jeden z nejpoužívanějších programovacích jazyků na Raspberry Pi OS. Pro vytvoření GUI aplikace byla vybrána knihovna Tkinter, protože tato knihovna se instaluje společně s jazykem Python [48]. Tímto je tato knihovna dostupná všude, kde je tento jazyk nainstalovaný.

Pro zabezpečené nahrání souborů do řídicí jednotky dle funkčního požadavku, lze využít protokol SFTP. Tento protokol využívá protokol SSH, pomocí kterého je řešena bezpečnost přenosu informací a autentizace klienta. SFTP protokol chrání proti odposlechu hesla a útoku typu man-in-the-middle. Chrání integritu dat pomocí šifrovacích a kryptografických hash funkcí [49].

Pro připojení k SFTP serveru a následnému přenášení souborů lze využít klienta s názvem FileZilla. FileZilla je open source projekt vydávaný pod licencí GNU. Klient je multiplatformní a lze ho provozovat na operačním systému Windows, MacOS a Linux [50].

Pro přenos informací mezi čtečkou karet a řídicí jednotkou bude využita sběrnice SPI, protože tento typ sběrnice využívá vybraná čtečka karet. Jako maximální délka přenosu pomocí této sběrnice bez užití zesilovače, se uvádí 10 metrů [51]. Tímto je umožněno tuto sběrnici použít v návaznosti na sekci 2.1.

3.2.5 Technologie API obsluhující požadavky řídicí jednotky

Pro vývoj této části systému byl zvolen microframework Flask, protože tento microframework je jednoduchý na pochopení a je často využíván pro tvorbu API. Flask je napsaný v jazyce Python a využívá Jinja template engine a Werkzeug WSGI toolkit [52]. Vydáván je pod licenci BSD-3-Clause.

Výhoda Flask spočívá ve snaze zachovat jádro, co nejjednodušší a v případě potřeby ho rozšiřovat pomocí modulů. Toto umožňuje snazší vývoj API, které bude obsluhovat příchozí požadavky od klienta.

API je typu REST. Tento typ byl zvolen, protože se jedná o moderní přístup pro vytváření webových API. Typicky se používá protokol HTTP pro webová API. Toto je přínosné pro vývojáře, protože nemusí instalovat žádný přídatný software a knihovny při vytváření REST API. REST API poskytují flexibilitu a dokážou obsluhovat několik typů zavolání [53].

3.3 Wireframes

Jedná se o zjednodušené GUI návrhy, které nezobrazují konečnou grafickou podobu aplikace. V návrhu se aplikace skládá ze 7 obrazovek. Tyto si lze prohlédnout na obrázku 7.

MainPage

Toto je úvodní obrazovka, kterou uživatel při prvotním pohledu uvidí. Uživatel je vyzván k přiložení karty, čímž je zahájen proces přístupu. Po přiložení karty je uživatel přesměrován na stránku *Loading*.

Loading

Na této obrazovce je uživatel vyzván k vyčkání. Během čekání systém autentizuje klienta a uživatele, zjišťuje studijní podklady a vybírá příslušné otázky. V případě výskytu chyby je uživateli zobrazena obrazovka *Error*. Pokud proces proběhne v pořádku, je uživateli zobrazena obrazovka *QuestionsAndAnswers*.

QuestionsAndAnswers

Jedná se o obrazovku zobrazující otázku se čtyřmi možnými odpověďmi. Horní lišta zobrazuje jméno studenta a název předmětu, ze kterého je otázka vybrána. Otázka je studentovi zobrazena v textovém poli. Pod otázkou se může nacházet případný obrázek přiložený k otázce. Jednotlivé odpovědi jsou vloženy do tlačítek. Při zvolení odpovědi je uživateli zobrazena obrazovka *ShowAnswers*.

ShowAnswers

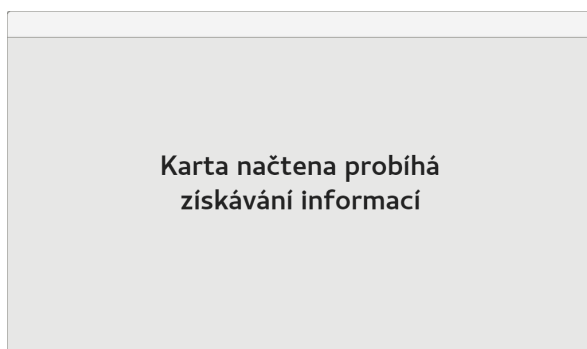
Tato obrazovka se může nacházet ve třech stavech. Stav *RightAnswer* se zobrazí, pokud uživatel zvolí správnou odpověď. Dalším stavem je *WrongAnswerNext*. Tento stav se zjeví v případě zvolení špatné odpovědi a nebylo dosaženo poslední otázky. Poslední stav je *WrongAnswerLast*. Tento stav se uživateli zobrazí v případě zvolení špatné odpovědi u poslední nabízené otázky. V této obrazovce je uživateli zobrazena správná odpověď společně se špatnými odpověďmi. Správnost odpovědí je zobrazena podbarvením tlačítek. Nad tlačítky je textové pole, které informuje uživatele, zda odpověděl správně.

Error

Obrazovka informuje uživatele o typu chyby, která se vyskytla. Nabízí informace o postupu, jak na danou chybu reagovat. Následuje informace o kontaktu na osobu, která může dané dveře odemknout v případě nemožného použití přístupového systému.



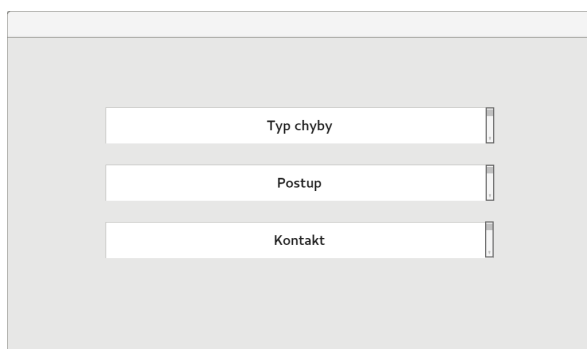
(a) MainPage



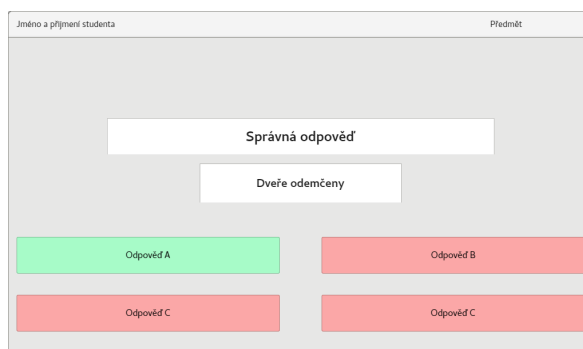
(b) Loading



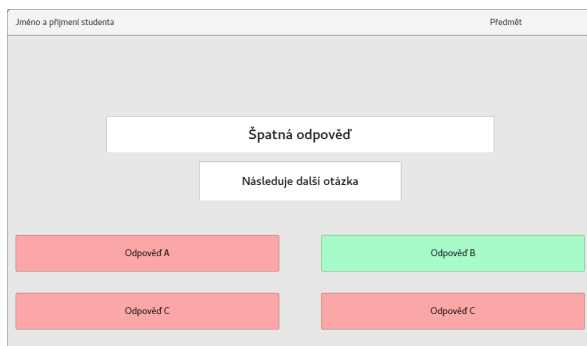
(c) QuestionsAndAnswers



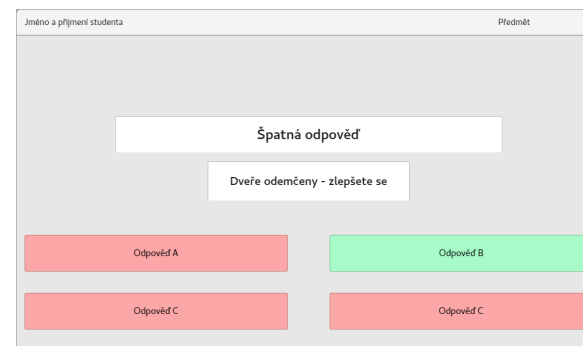
(d) Error



(e) RightAnswer



(f) WrongAnswerNext



(g) WrongAnswerLast

Obrázek 7: Wireframes zamýšlené aplikace

4 Implementace

V této kapitole se lze dočíst o implementaci a popisu jednotlivých částí přístupového systému. Implementace vychází z návrhu popsaného v předchozí kapitole.

4.1 Databáze

Pro management databáze PostgreSQL byl využit nástroj pgAdmin. Jedná se o multiplatformní open source nástroj, který umožňuje spravovat databázi skrze grafické uživatelské rozhraní.

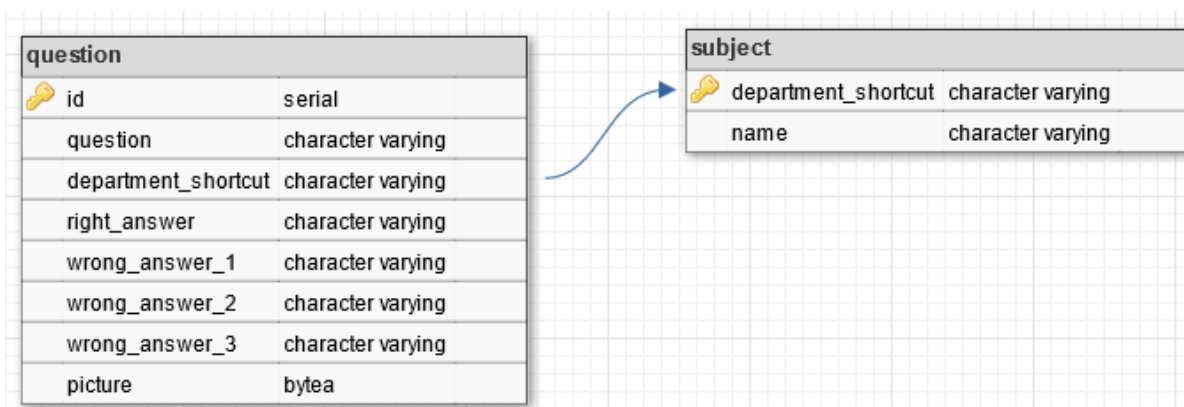
Po instalaci databáze je nutné povolit vzdálené připojení k databázi. Tato konfigurace se provádí v souborech *postgresql.conf* a *pg_hba.conf*. V souboru *postgresql.conf* je nutné odkomentovat řádek obsahující *listen_addresses = 'localhost'* a hodnotu *localhost* změnit na ***. V souboru *pg_hba.conf* je nutné na konec souboru přidat řádek s následujícími hodnotami:

```
host    all         all         0.0.0.0/0      scram-sha-256
host    all         all         :::/0          scram-sha-256
```

Tímto dojde k povolení připojení k databázi ze všech IPv4 a IPv6 adres. Nastavení lze změnit a je možné povolit připojení klientů k databázi pouze z určitého rozsahu IP adres [54]. Databáze implicitně obsahuje administrátorský účet, který má uživatelské jméno *postgres*. Po nastavení hesla k tomuto účtu, je možné se k databázi připojit vzdáleně skrze nástroj pgAdmin.

4.1.1 Otázky a předměty

Pro ukládání otázek byly vytvořeny dvě entity. Entita *question* a entita *subject*.



Obrázek 8: Implementace entit pro ukládání otázek a předmětů

subject

Tato entita slouží pro uložení názvu předmětu a přiřazené zkratky. Zkratka se skládá ze zkratky katedry, pod kterou je předmět vyučován a zkratky předmětu. Formát kombinace zkratky katedry a zkratky předmětu vypadá příkladově v této podobě: UAI/601. Tímto je zajištěna jednoznačná identifikace konkrétního předmětu, dle vzoru IS/STAG. Údaje v této entitě slouží k přiřazení otázky ke konkrétnímu předmětu.

question

V této entitě se ukládají otázky. Atribut *id* je typu SERIAL. Vytvořením atributu s tímto datovým typem pomocí pgAdmin, má za následek vytvoření i funkce, která automaticky inkrementuje hodnotu atributu při vložení nového záznamu. Atribut *picture* slouží pro ukládání binárních řetězců. Tento atribut je využit pro ukládání obrázků.

Implementace dat z IS/STAG

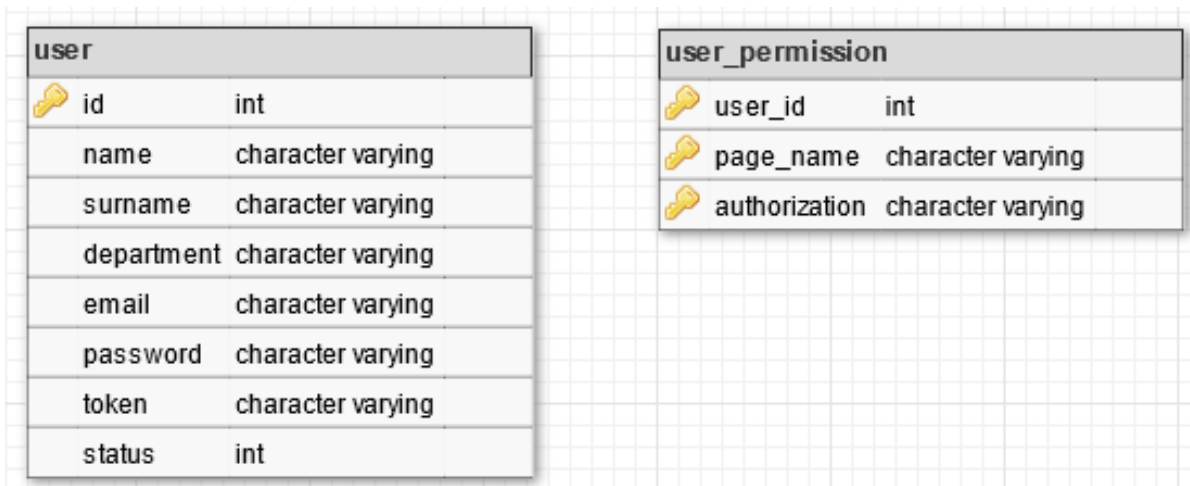
Pro získání všech předmětů z IS/STAG Jihočeské univerzity bylo využito webové služby *getPredmetyByFakulta* ze skupiny služeb *predmety*. Vzor pro použití služby si lze prohlédnout na obrázku 2. Služba vyžaduje roli LOGGED. Při volání služby byl využit parametr *outputFormat* s hodnotou *csv*. Získaný csv soubor lze upravit v programu Excel od společnosti Microsoft. V programu Excel lze využít nástroje Power Query pro upravení datového setu do žádané podoby. Byly odstraněny a sloučeny sloupce do formátu korespondujícího se strukturou entity *subject*. IS/STAG Jihočeské univerzity obsahuje nečistá data. Ukázkou nežádoucích dat si lze prohlédnout na obrázku 9. Datový set byl od takovýchto dat očištěn. Některé předměty nesou stejný název, proto nelze přiřazení otázky realizovat pouze na základě jména předmětu. Nástroj pgAdmin umožňuje importování dat z csv souboru do entit. Tohoto přístupu bylo využito. Při importování dat je nutné zvolit kódování dat a oddělovač. Soubor csv získaný z IS/STAG Jihočeské univerzity má znakovou sadu Windows-1250 a oddělovačem je středník.


7112	Obhajoba kvalifikační práce	KPE/7OKP	
7113	Pedagogika I	KPE/7PE1	
7114	A	- Teo. a met. vých. č. u."/	
7115	Pedagogika II	KPE/7PE2	
7116	N	- Soc. as. vých. a prev."/	
7117	Psychohygiena	KPE/7PH	
7118	Psychologie I	KPE/7PS1	
7119	A	Psychologie osob. uč."/	
7120	Psychologie II	KPE/7PS2	
7121	N	Vývojová psy. žáka na ZŠ"/	
7122	Pedagogika a psychologie pro SŠ	KPE/7Q1	
7123	Pedagogika a psychologie pro 2. st. ZŠ	KPE/7Q2	




Obrázek 9: Ukázka nečistoty dat předmětů z IS/STAG

4.1.2 Uživatelé a jejich práva

Pro implementaci uživatelů webové aplikace a možného nastavení jejich práv byly vytvořeny dvě entity s názvem *user* a entita *user_permission*



user	
 id	int
name	character varying
surname	character varying
department	character varying
email	character varying
password	character varying
token	character varying
status	int

user_permission	
 user_id	int
 page_name	character varying
 authorization	character varying

Obrázek 10: Implementace entit pro uložení uživatelů a jejich práv k webové aplikaci na zadávání otázek

user

V této entitě jsou uloženi uživatelé webové aplikace pro správu přístupového systému.

user_permission

Tato entita obsahuje informace o právech uživatelů ke konkrétním stránkám webové aplikace. Entita byla vytvořena skrze generátor webové aplikace [více popsáno v sekci 4.2.1]. Propojení mezi entitami není realizováno, protože vygenerovaná aplikace propojení mezi entitami neumožňuje.

Implementace dat z IS/STAG

Pro získání kompletního seznamu uživatelů byla využita webová funkce IS/STAG *getUcitele-Katedry* ze skupiny služeb *ucitel*. Webová služba nevyžaduje autentizaci a informace získané z této služby jsou takto dostupné veřejnosti. Bylo využito tří parametrů při volání této služby. Parametr *katedra* slouží pro získání seznamu pracovníků z konkrétních pracovišť nebo celé fakulty. K volání byl připojen parametr *jenAktualni* s hodnotou *true*. Tento parametr zajistí vrácení seznamu aktuálních pracovníků z konkrétního pracoviště. Posledním připojeným parametrem byl parametr *outputFormat* s hodnotou *csv*. IS/STAG nenabízí webovou službu pro získání všech pracovníků ze všech fakult pomocí jednoho volání. Proto byla tato funkce použita pro každou fakultu Jihočeské univerzity. Zkratky jednotlivých fakult Jihočeské univerzity

jsou následující: *FBI, FEK, FFI, FPE, FRO, FTE, FZE, FZS*.

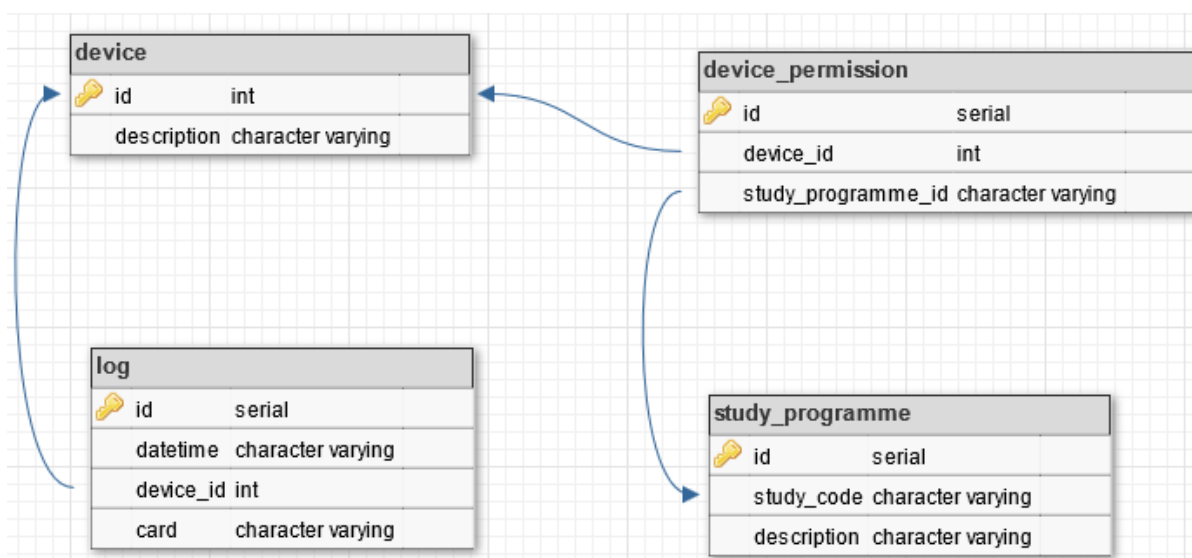
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	ucitIdno	jmeno	prijmeni	titulPred	titulZa	platnost	zamestnanec	katedra	pracovisteDalsi	email	telefon	telefon2	url
2	4281	Ladislav	Beránek	doc. Ing.	CSc.	A	A	KMI	UAI	beranek@ef.jcu.cz			

Obrázek 11: Ukázka dat získaných voláním služby `getUciteleKatedry`

Jednotlivé csv soubory byly spojeny do jednoho datového setu pomocí programu Excel, za využití nástroje Power Query. Datový set byl zbaven nepotřebných atributů a záznamy neobsahující emailovou adresu nebyly do seznamu uživatelů zahrnuty. Výsledný datový set obsahoval atributy *ucitIdno*, *jmeno*, *prijmeni*, *katedra*, *email*. Pomocí nástroje pgAdmin byl do entity *user* tento datový set importován.

4.1.3 Logování, správa zařízení a povolené přístupy skupin studentů

Pro ukládání logů, správu zařízení a nastavení přístupových práv skupin uživatelů k jednotlivým zařízením byly vytvořeny 4 entity. Entity si lze prohlédnout na obrázku 12.



Obrázek 12: Implementace entit pro logy a správu zařízení

device

V této entitě je uložen identifikátor zařízení a jeho popis. V popisu se očekává uvedení lokace, kde je zařízení umístěno.

log

Do této entity jsou ukládány záznamy obsahující pokus uživatele o přístup pomocí přístupového systému skrze validní zařízení. Záznam obsahuje datum a čas, kdy se uživatel pokusil

skrze zařízení vstoupit do chráněného areálu. K záznamu je připojeno hexadecimální číslo karty uživatele, čímž je zajištěna možná identifikace konkrétní osoby.

study_programme

V této entitě jsou uloženy studijní programy studentů. V atributu *description* je uveden název studijního programu.

device_permission

V této entitě jsou jednotlivým zařízením přiřazeny případné studijní programy. Z této entity se následně zjišťuje, jakým skupinám studentů smí dané zařízení udělit přístup.

Implementace dat z IS/STAG

Pro získání všech studijních programů byla vyžita služba IS/STAG *getStudijniProgramy*. Služba umožňuje získat veškeré studijní programy Jihočeské univerzity. Služba byla zavolána s parametrem *pouzePlatne* s hodnotou *TRUE*. Tímto je ovlivněno získání pouze aktuálních studijních programů. Pro získání souboru do csv formátu bylo využito parametru *outputFormat* s hodnotou *csv*. V datovém setu byly ponechány pouze sloupce nesoucí kód programu a název programu. Výsledný datový set byl za pomoci pgAdmin importován do entity *study_programme*.

Shrnutí

Výsledná databáze je přiložena jako příloha A. V době vývoje byla databáze nasazena na Ubuntu Server. Server byl provozován ve virtuálním prostředí pomocí nástroje VirtualBox. Přiložená databáze obsahuje vytvářecí skript databáze a data, čímž je databáze připravena pro nasazení do produkčního prostředí. Databázi lze do produkčního prostředí nasadit následujícím způsobem:

- Pomocí nástroje pgAdmin je třeba na serveru vytvořit databázi pojmenovanou *Questions_Database*.
- Vyvoláním kontextového menu na databázi se zobrazí nabídka operací. Z této nabídky je nutné vybrat položku *Restore*. V zobrazeném menu je nutné vybrat přiloženou databázi a v kartě *Restore options* je možné vybrat zda databáze se má vytvořit i s obsaženými daty. Zároveň v této kartě je nutné vybrat položku *Include CREATE DATABASE statement*. Po nastavení je nutné kliknout na tlačítko *Restore*. Po těchto krocích proběhne vytvoření databáze do struktury, se kterou pracují vyvinuté softwarové části přístupového systému. Závěrem databázi lze přejmenovat dle preferencí správce.

4.2 Server

4.2.1 Webová aplikace na zadávání otázek

Generování aplikace

Aplikace generátoru umožňuje uložení konfigurace pro generování aplikace do konfiguračního souboru. Tento soubor je přiložen ve složce *Generátor* přílohy B. Současně je zde přiložena uživatelská příručka generátoru. Tato příručka je vytvořena společností vydávající generátor. V této příručce jsou popsány komponenty a veškeré funkcionality generátoru, včetně ukázek kódu, které lze využít pro doplnění výsledné aplikace. Také je zde popsána dokumentace kódu, který byl vygenerován.

Před zahájením generování je nutné v databázi vytvořit uživatele. Tento uživatel musí mít nastavená práva pro schéma databáze, entity a potřebné sekvence.

Úpravy vygenerovaných souborů

Ve vygenerované aplikaci byly upraveny následující soubory:

- `phpgen_settings.php`

Byl vymazán obsah funkcí *GetPagesHeader* a *GetPagesFooter*. Generátor zde generuje odkaz na stránky společnosti SQLMaestro. Zmínku o společnosti SQLMaestro není třeba na stránkách aplikace uvádět. Tato informace byla získána z emailové komunikace se společností. Funkce *GetGlobalConnectionOptions* obsahuje informace pro připojení aplikace k databázi. Funkce *GetMailer* slouží pro nastavení emailové služby aplikace. Tato služba je využívána pro nastavení hesla uživatele.

- `components/templates/mail/recovering_password_body.tpl`
- `components/templates/mail/recovering_password_subject.tpl`

V souborech byla upravena zpráva a název předmětu emailové zprávy, která informuje uživatele o požadavku na změnu hesla.

- `components/languages/lang.php`

Soubor obsahuje konfiguraci textu, který aplikace zobrazuje. Využité části byly upraveny do české lokalizace.

Výsledná aplikace

Pomocí této aplikace mohou uživatelé zadávat otázky ke konkrétním předmětům. Uživatel s rolí správce může kromě managementu otázek provádět následující operace:

- Management uživatelů této aplikace.
- Management předmětů.
- Přiřazení studijních programů k zařízením.
- Prohlížení záznamů obsahující informace o pokusu použití zařízení konkrétním uživatelem.
- Management zařízení.
- Management studijních programů.

Uživatelé aplikace jsou uloženi v entitě *user*. Aplikace vyžaduje autentizaci uživatele pro její používání. Heslo uživatele je přenášeno a uchováváno v databázi ve formě hash. Hash je implementován hashovací funkcí SHA-256. Politika hesla vyžaduje od uživatele nastavení hesla o minimální velikosti 12 znaků. Heslo musí obsahovat alespoň jedno velké písmeno, jedno malé písmeno a jedno číslo. Tato část kódu byla převzata z uživatelské příručky generátoru ze strany 259. Kód byl doplněn v souboru *authorization.php* do funkce *VerifyPasswordStrength*.

Výsledná aplikace využívá ORM přístup pro práci s databází. Jednotliví uživatelé mají nastavená práva k jednotlivým stránkám v entitě *user_permission*. Pokud je ke stránce přiřazeno id s hodnotou 0, znamená to, že toto právo platí pro všechny uživatele aplikace. Ukázkou dat v entitě *user_permission* lze vidět na obrázku 13. Tímto způsobem bylo přiřazeno právo všem uživatelům k manipulaci s daty v entitě *question*. Do této entity budou všichni uživatelé vkládat své otázky. Uživatelé vidí otázky ze všech předmětů. Této implementace lze využít u předmětů, které na sebe navazují a porovnat, zda není stejná otázka položena už v jiném předmětu. Správci aplikace byly nastaveny veškerá práva ke všem entitám, s výjimkou práv na vkládání, mazání a upravování záznamu v entitě *log* z bezpečnostních důvodů.

user_id [PK] integer	page_name [PK] character varying	authorization [PK] character varying
0	public.question	DELETE

Obrázek 13: Ukáзка dat z entity *user_permission*

Z důvodu zobrazovacích kapacit displeje je omezena maximální možná délka otázek a odpovědí na 255 znaků. Cílem není dávat studentům dlouhé a komplexní otázky, ale krátké, úderné a snadno srozumitelné. Maximální velikost obrázků, které mohou uživatelé k otázce připojit,

je stanovena na 2 MB.

Emaily z aplikace jsou zasílány pomocí protokolu SMTP s využitím SLL protokolu. Webová aplikace využívá mailový server Přírodovědecké fakulty Jihočeské univerzity. Tímto je zajištěna bezpečnost přenášených zpráv. Mailový server lze změnit dle potřeby správce webové aplikace. O používání aplikace, včetně nastavení hesla, se lze dočíst v příložené uživatelské dokumentaci ve složce *Webová aplikace* přílohy B.

Při vývoji byl využit webový server Apache, který byl nainstalován pomocí nástroje XAMPP. Tento nástroj umožňuje nainstalování distribuce Apache serveru s podporou jazyka PHP. Pro umožnění nasazení aplikace je nutné v souboru *php.ini* povolit ovladač *pgsql*. Pro nasazení vygenerované aplikace je nutné, aby uživatel webového serveru měl povoleno zapisovat do složky *templates_c*. Informace, o nasazení vygenerované aplikace, se lze dočíst z příložené uživatelské příručky ze stránky 21.

Pro zajištění bezpečné komunikace se očekává nasazení vygenerované aplikace na webový server, který bude využívat certifikát. Tímto bude realizována zabezpečená komunikace pomocí HTTPS protokolu. Pro nejrychlejší a nejbezpečnější komunikaci mezi aplikací a databází je vhodné nasazení na stejné zařízení.

4.2.2 Backend REST API

Dle návrhu bylo pomocí microframeworku Flask vytvořeno REST API. Skrze toto API byly implementovány dva koncové body pro klienty. Jednotlivé koncové body jsou: */questions* a */images/<int:image_id>*. Z bezpečnostních důvodů bylo implementováno zabezpečení HTTP BASIC, které bylo popsáno v podsekcí 1.6.1. Pro implementaci tohoto zabezpečení je využito rozšíření *flask_httpauth*, protože toto rozšíření implementaci této autentizace usnadňuje. REST API je implementováno pomocí rozšíření *flask_restful*, protože toto rozšíření umožňuje snadné vytvoření API typu REST. Práce s databází je implementována pomocí ORM. ORM je implementováno za použití rozšíření *SQLAlchemy*, protože se jedná o jedno z nejpoužívanějších řešení pro tento druh implementace. API využívá konfiguračního souboru s názvem *constants.py*. V tomto souboru lze nastavit následující konfigurace:

- Připojení do databáze.
- Uživatelské jméno a heslo pro zařízení.
- Základní počet otázek, které se vrací klientovi.
- Maximální možný počet otázek, které API umožňuje vracet.
- Základní odkaz na webové služby IS/STAG.

- Přihlašovací údaje pro využití služeb IS/STAG.
- Zkratka předmětu, ze kterého se vybírají otázky v případě nedostupnosti dostatečného počtu otázek z vybraných předmětů, či pokud uživatel nemá žádné předměty zapsané z minulých let.
- Název souboru, do kterého se ukládají záznamy v případě výpadku části systému.
- Kód studijního programu, který reprezentuje všechny studenty.

/questions

- Obecný popis koncového bodu

Tento koncový bod přijímá tři parametry. Povinným parametrem je *card_number* a *device_id*. Nepovinným parametrem je *question_count*. Parametr *card_number* slouží pro zaslání hexadecimálního čísla karty uživatele. Parametr *device_id* slouží pro identifikaci klienta. Parametrem *question_count* si může klient specifikovat, kolik otázek si přeje od API obdržet. Při nezaslání parametru *question_count*, API vrací počet otázek specifikovaný ve svém konfiguračním souboru. Koncový bod vrací uživateli odpověď ve formátu JSON.

- Autentizace uživatele

Po přijmutí požadavku proběhne autentizace klienta a verifikace správného formátu zaslaných parametrů. Následně proběhne pokus o napojení do databáze. Z databáze je zjištěno, zda existuje klient s id, které bylo uvedeno v parametru dotazu. V případě, že takové id neexistuje v databázi, je klientovi vrácena informace o zamítnutí přístupu. Pokud takovéto id v databázi existuje, jsou zjištěny přiřazené studijní programy k tomuto klientu. Následně dojde k uložení záznamu do entity *log*, o pokusu užití klienta uživatelem. Záznam obsahuje identifikátor klienta, kód karty uživatele a datum s časem, kdy k pokusu o přístup došlo.

V návaznosti na popsané služby v podsekcí 1.6.2 se odešle dotaz na API IS/STAG. Využívá se služby *getStudiaByCisloKarty*. Z výsledků služby se ověřuje platnost karty. Karta je platná v případě, že služba vrátí studium, které obsahuje klíč *stav* s hodnotou *S*. Student může mít takovýchto studií více. Z těchto studií jsou uložena identifikační čísla uživatelů. Pokud uživatel nemá studium se stavem *S*, je jeho karta považována za neplatnou a klientovi vrácena informace o odmítnutí přístupu.

Jednotliví klienti mají přiřazeno, jaké skupiny uživatelů smí pouštět na základě studijního programu. Po získání osobních čísel studenta dojde k volání služby *getStudentInfo* ze skupiny služeb *student* systému STAG. Služba je volána opakovaně, pokud v předchozím kroku bylo získáno více studií se stavem *S*. Do parametru služby je uvedeno osobní číslo studenta.

Výsledkem služby jsou informace o studentovi, včetně kódu studijního programu ke konkrétnímu studiu studenta. Následně dojde k porovnání získaného seznamu studijních programů studenta se seznamem povolených studijních programů klienta. Pokud se nevyskytuje alespoň jedna hodnota ze seznamu studijních programů studenta v seznamu povolených studií klienta, je klientovi vráceno odmítnutí přístupu. V případě, že klient má přiřazený studijní program odpovídající programu nastaveného v konfiguračním souboru, dojde k udělení přístupu uživateli. Studijní program nastavený v konfiguračním souboru reprezentuje skupinu, která zahrnuje všechny studenty Jihočeské univerzity. Tímto umělým studijním programem je implementována možnost, použití klienta pro udělení přístupu všem studentům s platnou identifikační kartou.

- Získání předmětů studenta

Ze seznamu získaných aktivních studií uživatele je vybráno první studium a z tohoto studia je vybráno osobní číslo studenta. Následně dojde k volání služby IS/STAG *getPredmetyByStudent* ze skupiny služeb *predmety*. Pro volání služby se využívají parametry s hodnotou vracející seznam všech předmětů, které si student za celou dobu studia zapsal. Následně z obdrženého seznamu předmětů jsou vypuštěny předměty, které student studuje v aktuálním akademickém roce, protože tyto předměty student ještě nemusel absolvovat. Poté jsou dle funkčního požadavku ze seznamu vybrány pouze předměty se statusem *A* a *B* [popsáno v sekci 1.6]. Tímto jsou studentovi nabízeny pouze předměty, které je nutné splnit v rámci studijního programu a znalost těchto předmětů je esenciální k úspěšnému složení státní zkoušky. Je možné, že student si daný předmět v průběhu studia zapsal dvakrát. Tento případ může nastat v momentě, kdy se studentovi nepodařilo předmět napoprvé úspěšně absolvovat. Větší počet zapsání stejného předmětu se statusem *A* nebo *B* není na Jihočeské univerzitě povolen. Toto je ošetřeno a v seznamu jsou zachovány pouze předměty s unikátní zkratkou, což vede k odstranění duplicit v seznamu předmětů.

- Výběr otázek

Ze seznamu předmětů je náhodně vybráno tolik předmětů, kolik se má vrátit otázek. Z těchto předmětů je vytvořen nový seznam. Následně dojde k dotazu do databáze pro sadu otázek k předmětům z nového seznamu. Vždy z každé sady otázek vztahující se ke konkrétnímu předmětu je vybrána náhodně jedna otázka. Může nastat případ, kdy předmět nemá v entitě *question* žádnou otázku přiřazenou. V případě nedosažení požadovaného počtu otázek, dojde k dalšímu dotazu pro získání chybějícího počtu otázek. Tento dotaz vrací sadu otázek přiřazených k předmětu se zkratkou *DEFAULT*. Takovýto předmět je v databázi vytvořen pro tyto případy. Tento předmět obsahuje obecné informace o Jihočeské univerzitě. Tímto je zároveň ošetřeno, kdy student nemá zapsané žádné předměty z předchozích let [student v prvním ročníku]. V případě, že sada otázek z předmětu *DEFAULT* nemá dostatečný počet otázek pro doplnění chybějícího počtu otázek, je vrácen maximální počet otázek, které bylo možné doplnit.

Ukázka odpovědi koncového bodu:

```
{
  "questions": [
    {
      "question": "g elit. Etiam neque. Praesen ,
      "department_shortcut": "UAI/702",
      "subject_name": "Administrace UNIXu,
      "right_answer": "Správná odpověď",
      "wrong_answer_1": "tiam neque. Praesent d",
      "wrong_answer_2": "tiam neque. Praesent d",
      "wrong_answer_3": "tiam neque. Praesent d",
      "picture_id": 186
    },
    {
      "question": "Lorem ipsum dolor sit am,
      "department_shortcut": "UAI/682",
      "subject_name": "Architektura počítačů II",
      "right_answer": "Správná odpověď",
      "wrong_answer_1": "Lorem ipsum dolor sit ame",
      "wrong_answer_2": "Lorem ipsum dolor sit ame",
      "wrong_answer_3": "Lorem ipsum dolor sit ame",
      "picture_id": null
    }
  ],
  "student_name": "MARTIN JÚDA" }
```

Klíč *picture_id* reprezentuje, zda otázka má v databázi obrázek, či nikoliv. Obrázek není vrácen přímo v rámci tohoto koncového bodu. Je vráceno pouze *id* otázky, která obrázek obsahuje. Tato implementace zrychluje odezvu na klientův požadavek a jedná se o správný přístup implementace API typu REST. Klient po obdržení otázek se doptá pro obrázek ke každé otázce, která nese neprázdný klíč *picture_id*. V případě výpadku systému, tedy nedostupnosti IS/STAG nebo databáze, se tato událost zapíše do lokálního logu na serveru a klientovi je vrácen patřičný HTTP kód.

/images/<int:image_id>

Tento koncový bod slouží pro zaslání obrázku klientovi. Parametrem je celé kladné číslo. Toto číslo reprezentuje *id* otázky, ke které chce klient obdržet obrázek.

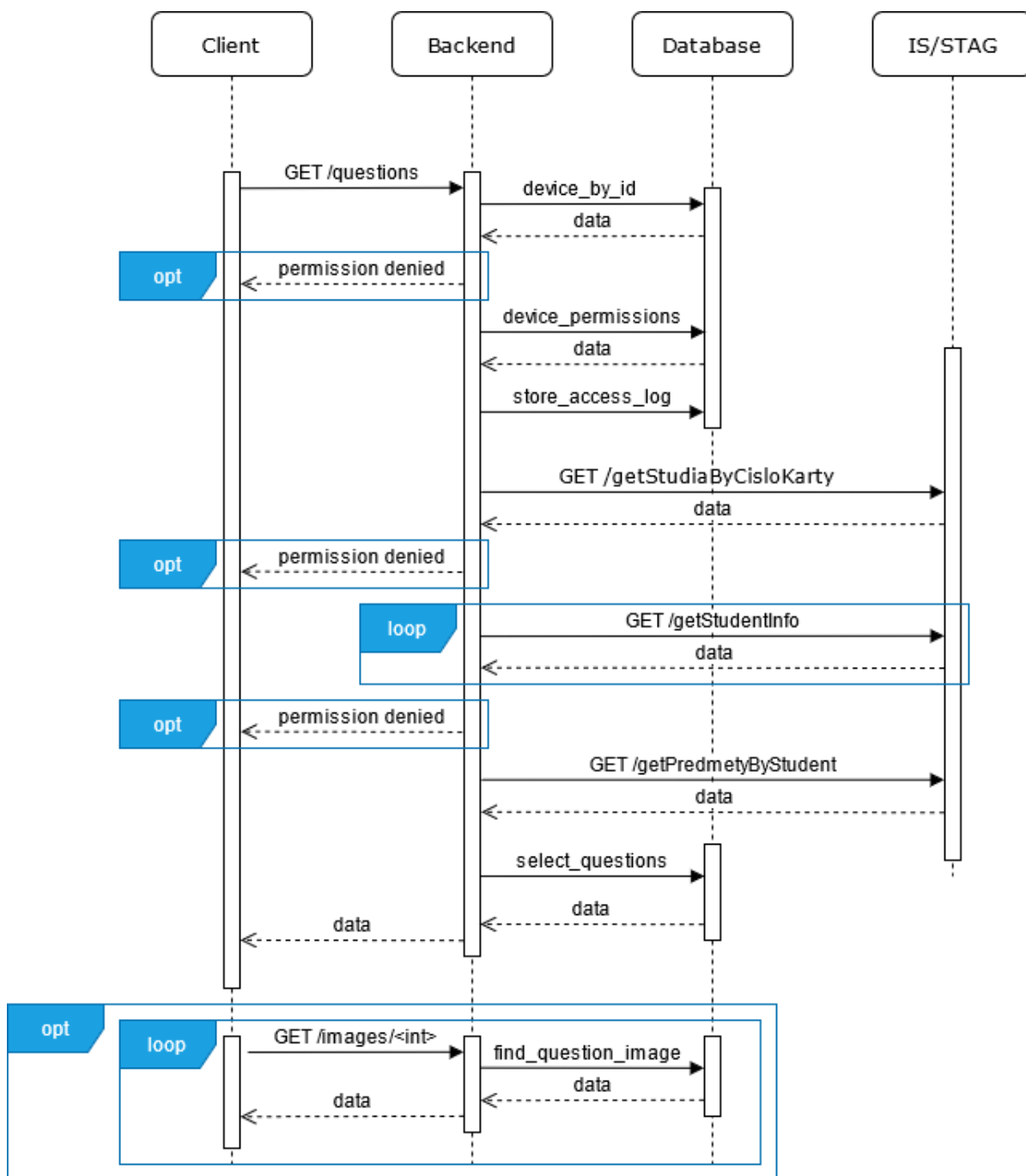
Prvotně proběhne autentizace klienta a verifikace formátu parametru. Po autentizování klienta

se vytvoří spojení s databází. Z databáze je získána otázka a proběhne kontrola, zda otázka obrázků obsahuje. Pokud otázka obrázků obsahuje, je získána reprezentace bitů obrázku a zakóduje se pomocí Base64. Následně se tento formát dekoduje do kódování UTF-8. Takto vznikne lidsky čitelný řetězec znaků reprezentující bity obrázku. Tento výsledek je vrácen klientovi.

Zabezpečená komunikace, doporučené nasazení API a dokumentace

Zabezpečená komunikace s klientem byla realizována pomocí certifikátu typu *on-the-fly* pro možnou komunikaci pomocí HTTPS protokolu. Jedná se o self-signed typ certifikátu, který je implementován rozšířením *pyopenssl*. V produkčním prostředí se nedoporučuje nasadit zabudovaný *WSGI* server, který flask obsahuje. Flask aplikace je objektem *WSGI* aplikace. Dokumentace flask nabízí návody pro nasazení do produkčního prostředí [55].

Pro dokumentaci kódu byly použity komentáře ve stylu *Google* [56]. Pro vygenerování dokumentace bylo využito rozšíření *pdoc3*. K zprovoznění tohoto REST API je doporučeno vytvořit virtuální prostředí jazyka Python. Veškeré potřebné externí knihovny lze najít v souboru *requirements.txt*. Pomocí příkazu *pip install -r requirements.txt* lze nainstalovat všechny potřebné knihovny pro zprovoznění tohoto REST API. Vytvořené REST API společně s dokumentací a unit testy je přiloženo jako příloha C. Diagram komunikace si lze prohlédnout na obrázku 14.



Obrázek 14: Komunikace mezi klientem a API

4.3 Klient

4.3.1 Konfigurace klienta

Pomocí programu NOOBS, který byl nainstalován na zakoupené SD kartě, proběhla instalace operačního systému Raspberry Pi OS. Implementace SSH protokolu byla realizována pomocí open source serveru OpenSSH, protože tento server zároveň nabízí SFTP server implementaci ve formě out-of-the-box. OpenSSH server je implicitně nainstalován v rámci operačního systé-

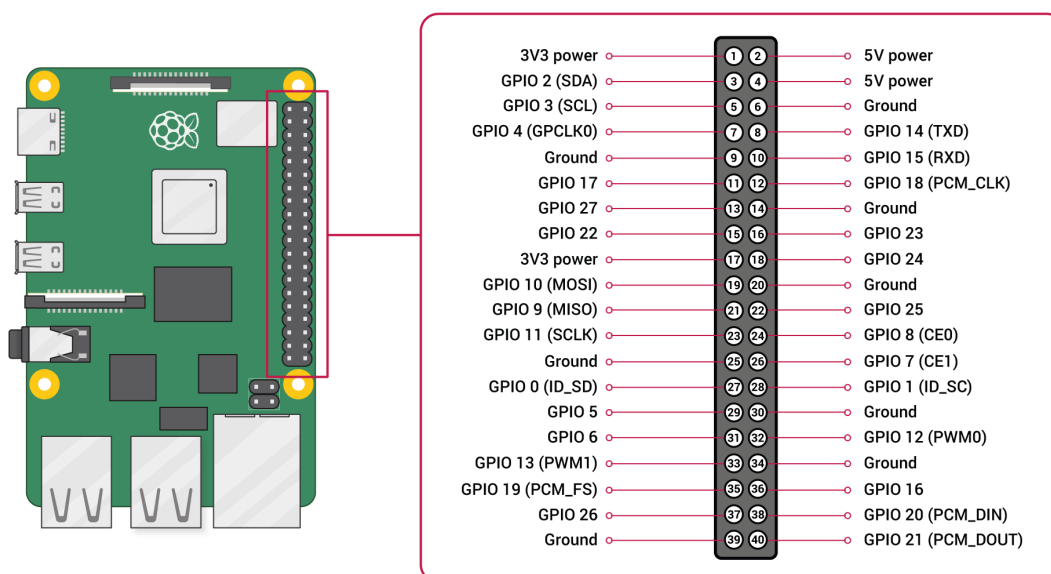
mu Raspberry Pi OS. Pro umožnění grafického připojení ke vzdálené ploše byl využit program VNC Connect. Tento program je implicitně nainstalován s operačním systémem Raspberry Pi OS.

Pomocí příkazu `sudo raspi-config` bylo povoleno rozhraní SPI, SSH a VNC. Zakoupený displej má atypické rozlišení 1024x600. Přímou toto rozlišení neumožňuje Raspberry Pi OS nastavit v grafickém nastavení. Je nutné upravit hodnotu rozlišení v konfiguračním souboru. Pro konfiguraci tohoto souboru je nutné zadat příkaz `sudo nano /boot/config.txt`. V souboru je nutné provést následující nastavení [57]:

- Odkomentování: `disable_overscan=1`
- Pod výše zmíněný příkaz doplnit tento příkaz: `hdmi_cvt=1024 600 60 3 0 0 0`
- Odkomentování a přepsání hodnoty: `framebuffer_width=1024`
- Odkomentování a přepsání hodnoty: `framebuffer_height=600`

Pro propojení čtečky karet a relé je využito GPIO pinů řídicí jednotky. Označení pinů si lze prohlédnout na obrázku 15 [58]. Každý pin má přiřazenou vlastnost určující jeho možné vstupy a výstupy.

Pro deaktivaci vypínání obrazovky po časové nečinnosti, je nutné provést změnu v nastavení Raspberry. Tato změna se provádí v kartě *Volby -> Raspberry Pi Configuration -> Display -> Screen Blanking*.



Obrázek 15: Sada GPIO pinů Raspberry Pi 4B

Propojení GPIO pinů Raspberry Pi 4B se zakoupenými komponentami

Čtečka karet	Číslo pinu Raspberry Pi
SDA	24
SCK	23
MOSI	19
MISO	21
GND	6
RST	22
3.3V	1

Tabulka 2: Propojení pinů mezi čtečkou karet a Raspberry Pi

Relé	Číslo pinu Raspberry Pi
IN	16
VCC	2
GND	9

Tabulka 3: Propojení pinů relé a Raspberry Pi

4.3.2 Aplikace

Pro implementaci komunikace se sběrnicí SPI je využito knihovny *SPI-Py*. Tuto knihovnu vytvořil Louis Thiery a je následně vyvíjena Connorem Wolfem. Knihovna je vydána pod licencí General Public License V2. Po stažení knihovny je nutné zadat příkaz `sudo python3 setup.py install` [59]. Pro implementaci čtení karet z kartové čtečky je využita knihovna *MFRC522-python* vyvinutá Mario Gomezem. Tato knihovna je vydána pod licencí GNU Lesser General Public License v3.0. Propojení pinů řídicí jednotky se čtečkou karet si lze prohlédnout v tabulce 2 [60]. Problém této knihovny je její neaktuálnost a bylo využito aktualizace od Lucase Arcanjo [61]. Aktualizace upravila knihovnu do syntaxe jazyka Python verze 3 a upravila funkční vlastnosti pro kompatibilitu s nynější verzí knihovny *SPI-Py*. Při testování této knihovny bylo odhaleno přetečení paměti při dlouhodobém používání. Tento problém jsem opravil dle souladu s licencí. Opravená verze knihovny je přiložená v příloze D a umožňuje bezproblémové použití.

Aplikace klienta obsahuje dva konfigurační soubory: *config.py* a *constants.py*.

config.py

V tomto konfiguračním souboru lze nastavit tato nastavení:

- Číslo pinu, který je využíván pro zaslání signálu relé. Po dobu přenosu signálu zůstává relé sepnuto.
- URL na server, kde je provozováno vyvinuté REST API.
- Autentizační údaje, kterými je klient v API autentizován.
- Počet otázek, které si klient přeje obdržet.
- Čas v sekundách, po který má být vyslán signál do relé pro odemknutí zámku.
- Čas v sekundách pro přepínání mezi otázkami, po zvolení špatné odpovědi.
- Čas v sekundách, během kterého uživatel musí zodpovědět otázku.
- Nastavení, zda klient má ověřovat certifikáty serveru vůči certifikační autoritě.
- ID zařízení.
- Text, který se zobrazí v závislosti na typu obdržené chyby od API.

constants.py

V tomto konfiguračním souboru lze nastavit tato nastavení:

- Základní rozlišení obrazovky aplikace. Toto nastavení funguje v případě, že aplikace není zobrazena v režimu celé obrazovky.
- URL na koncové body vyvinutého REST API.

Implementace aplikace

Aplikace vyžaduje pro zahájení procesu přístupu interakci uživatele. Uživateli je prvně prezentována obrazovka, která ho vyzývá k přiložení karty ke čtečce karet. Při čtení karty jsou čteny první čtyři adresové bloky. Načtené hodnoty jsou převedeny do hexadecimální reprezentace. Tímto je zajištěna stejná reprezentace hodnot, kterou přijímá REST API obsluhující požadavky klienta a webová služba IS/STAG vracející seznam studií uživatele.

Po přiložení karty je uživateli zobrazena nová obrazovka. V této obrazovce je uživatel vyzván k vyčkání, než systém vyhodnotí získané informace. Během této doby je zaslán dotaz na koncový bod REST API */question* pro autentizaci uživatele a získání otázek. Po obdržení odpovědi je vyhodnoceno, zda některá otázka nese klíč *picture_id* s neprázdnou hodnotou. V případě, že taková otázka existuje, proběhne na pozadí další dotaz na koncový bod REST API */images/<int:image_id>*. Do URL za */images* se vkládá hodnota, kterou klíč *picture_id* obsahuje. Tento proces je zopakován pro všechny otázky, které mají neprázdnou hodnotu klíče

picture_id. Pokud se nepodaří k otázce obrázek získat, je otázka přeskočena a nebude uživateli nabídnuta. V momentě, kdy by nastala kritická situace a všechny obdržené otázky obsahovaly obrázek a dané obrázky nebylo možné získat, dojde k zobrazení chyby uživateli a vyzvání k novému pokusu o přístup.

Pokud načtení otázek proběhne úspěšně, je uživateli zobrazena nová obrazovka nesoucí tyto informace:

- Jméno a příjmení uživatele.
- Název předmětu, ze kterého byla získána otázka.
- Otázka s případným obrázkem.
- Čtyři možné odpovědi.
- Čas, ve kterém musí uživatel zvolit odpověď na otázku.

V případě výskytu chyby, odmítnutí přístupu nebo výpadku systému, je uživateli zobrazena obrazovka popisující konkrétní událost. Na konkrétní událost je zobrazena instruktáž s postupem a kontaktem na odpovídající osobu. Pro určení místa problému je uživateli sděleno ID klienta. Pomocí tohoto ID může správce dohledat v databázi místo umístění zařízení. Obrazovku může uživatel zavřít tlačítkem a pokusit se o nový přístup. V případě, že uživatel tuto obrazovku nezavře, je uzavřena po časovém intervalu nastaveného v konfiguračním souboru. Následně je opět zobrazena uvítací obrazovka.

Rozložení odpovědí v tlačítkách probíhá náhodně. Tímto je zajištěno, že správná odpověď nebude vždy na stejném místě. V případě nezvolení žádné odpovědi v časovém limitu, je proces přístupu ukončen. Tímto je ošetřena možnost, kdy uživatel zahájí proces přístupu a od terminálu odejde.

V případě, že výška obrázku přesahuje hranici 320 pixelů, dojde ke zmenšení obrázku při zachování poměru stran. Velikost obrázků je změněna za využití knihovny *Pillow*. Knihovna je vydána pod licencí HPND. Implementace tímto způsobem je provedena, protože displej má omezený prostor vycházející z rozlišení displeje.

Při zvolení odpovědi je uživateli vždy zobrazena správná odpověď. Při zobrazování pravdivosti odpovědí jsou tlačítka podbarvena. Zelená barva indikuje správnou odpověď a červená barva informuje o špatné odpovědi. V případě zvolení nesprávné odpovědi, je uživateli nabídnuta další otázka, nejedná-li se o poslední otázku. V případě zvolení špatné odpovědi u poslední otázky, je uživateli udělen přístup dle funkčního požadavku. V momentě zvolení správné odpovědi, je uživateli udělen přístup. Po zvolení správné odpovědi, je vysílán signál do relé.

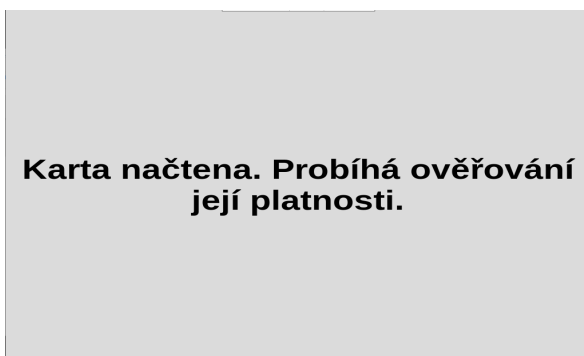
Signál pro relé je vysílán po časový interval stanovený v konfiguračním souboru. Během této doby je relé sepnuto, čímž dochází k odemknutí zámku. Po těchto krocích je proces přístupu zresetován a připraven na dalšího uživatele.

Aplikaci společně s dokumentací kódu a uživatelskou dokumentací lze nalézt v příloze D. GUI aplikace si lze prohlédnout na obrázku 16.

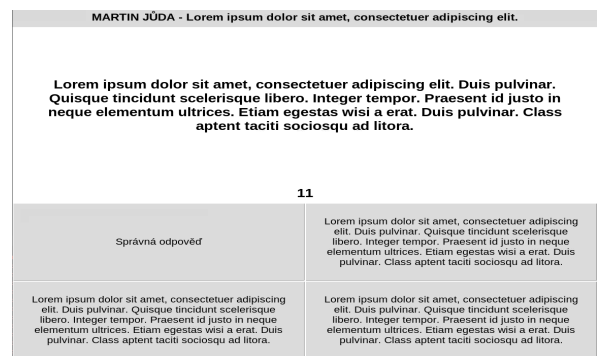
GUI aplikace



(a) Úvodní obrazovka



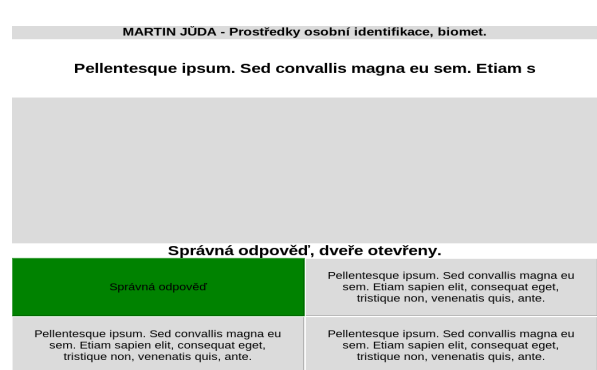
(b) Obrazovka vyzývající k vyčkání uživatele



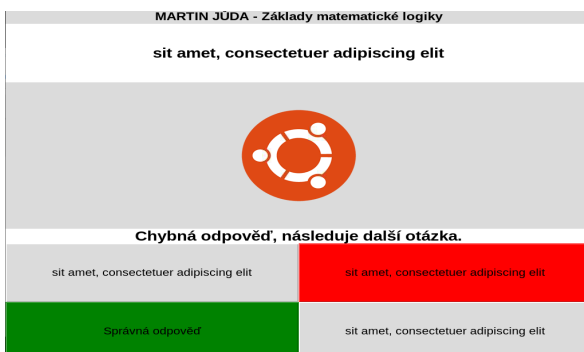
(c) Otázky a odpovědi



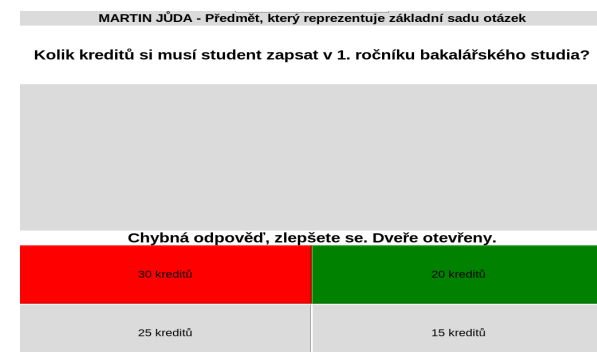
(d) Obrazovka informující uživatele o neúspěšném pokusu o přístup



(e) Zvolení správné odpovědi



(f) Zvolení špatné odpovědi



(g) Zvolení špatné odpovědi u poslední otázky

Obrázek 16: GUI aplikace

Spuštění aplikace po startu zařízení

Pro ošetření výpadku elektřiny a následné obnovení funkčnosti systému, je nutné aplikaci spustit po přihlášení uživatele. Pro spuštění aplikace byl napsán skript. Tento skript lze nalézt v příloze D ve složce *Spuštění aplikace po zapnutí zařízení*.

Pro umožnění spuštění skriptu je nutné přiřadit souboru obsahující skript práva pro spuštění. Pro zajištění spuštění po zapnutí zařízení, je nutné vytvořit soubor typu *DESKTOP*. Tento soubor je nutné vložit do adresáře */etc/xdg/autostart/*. V souboru tohoto typu byly nastaveny následující konfigurace:

```
[ Desktop Entry ]
Type= Application
Name= Client for student access control system
NoDisplay=false
Exec=/usr/bin/lxterminal -e /home/pi/run_app.sh
```

Do parametru *Exec* se udává příkaz, který se má vykonat. Prvním příkazem dochází ke spuštění terminálu. Parametrem *-e* a následujícím příkazem se udává, jaký příkaz se má v terminálu vykonat.

4.4 Testování

Fyzické propojení čtečky karet a řídicí jednotky bylo otestováno pomocí UTP kabelu standardu Cat6. Tento kabel měřil 10 metrů. Tímto bylo otestováno možné nasazení těchto dvou komponent v rozsahu dané vzdálenosti. Odemknutí zámku bylo simulováno rozsvícením diody.

Navržené REST API bylo manuálně testováno a byly napsány Unit testy. Manuální testování spočívalo v:

- Posílání špatného formátu parametrů.
- Zkoumání chování při nedostupnosti služeb IS/STAG.
- Zkoumání chování při nedostupnosti databáze.
- Zaslání požadavků s neplatnou RFID kartou.
- Zaslání požadavku uživatele z klienta, který nemá povolené studijní programy uživatele.
- Zkoumání chování při zaslání požadavku na větší počet otázek, než je povoleno v konfiguračním souboru.

- Zkoumání chování při obdržení požadavku na počet otázek, které nebylo možné získat z databáze.
- Zkoumání chování v momentě, kdy databáze neobsahuje žádné otázky.

Otestována byla komunikace mezi klientem a REST API. Komunikace byla zachycena pomocí programu Wireshark pro ověření komunikace přes HTTPS protokol.

Webová aplikace byla manuálně otestována. Testování probíhalo s administrátorským účtem. Bylo otestováno:

- Management otázek. Zejména pokus o vložení otázky a obrázku přesahující povolené limity.
- Management klientů. Zvláště změna přiřazených studijních programů.
- Prohlížení záznamů klienta.
- Management uživatelů aplikace.

Pro otestování aplikace klienta proběhlo manuální testování. Aplikace byla testována v nepřetržitém provozu po dobu 48 hodin. Během této doby proběhlo 50 pokusů o přístup uživatelem. Ke čtečce karet byly přikládány různé RFID karty pro ověření platnosti karet. Byly otestovány kritické situace:

- Nemožné získání žádné otázky z databáze.
- Výpadek databáze.
- Výpadek IS/STAG.
- Nemožné získání obrázku k otázce, která měla obrázek obsahovat.

Z důvodu koronavirové pandemie a státních opatření k jaru roku 2021, nebylo možné nabídnout fyzické testování přístupového systému studentům Jihočeské univerzity. Pěti studentům Jihočeské univerzity bylo fungování klienta virtuálně představeno. Dle instrukcí těchto studentů byla simulována jejich interakce s klientem a proběhlo získání zpětné vazby na vzhled GUI aplikace.

5 Budoucí možný rozvoj přístupového systému

Navržené hardwarové řešení umožňuje rozšíření klienta o další bezpečnostní prvky. Příkladem může být použití kamer pro sledování zabezpečených objektů. Pomocí API lze rozšířit funkcionality systému. Systém by takto mohl být rozšířen nejen pro studenty, ale i zaměstnance školy.

V budoucnu lze navrhnout i tento systém pro zaměstnance Jihočeské univerzity. Pomocí systému lze zavést více-faktorovou autentizaci do míst vyhraněných pouze určité skupiny zaměstnanců Jihočeské univerzity. Po přiložení karty by byl uživatel vyzván k zadání jednorázového klíče. Tento klíč by uživatel obdržel na jeho fakultní emailovou adresu. Toto rozšíření lze implementovat i pomocí zaslání jednorázového klíče na telefonní číslo zaměstnance. Pro zavedení tohoto typu autentizace by bylo nutné požádat o novou funkci IS/STAG. Tato funkce by fungovala obdobně jako funkce *getStudiaByCisloKarty*. Nová funkce by na základě parametru obsahující hodnotu hexadecimálního čísla RFID karty vrátila stav, který by informoval o platnosti karty. Dále by musela funkce vrátit emailovou adresu nebo telefonní číslo uživatele. Poslední nutnou položkou pro fungování tohoto přístupového systému by byl seznam všech pracovišť uživatelů. Jednotlivá pracoviště lze využít jako reprezentaci skupiny uživatelů. Na základě pracovišť by byl omezen přístup do prostor pouze vyhraněným skupinám zaměstnanců.

Webovou aplikaci pro správu přístupového systému lze do budoucna začlenit do IDM systému. Tímto by nebylo nutné vést uživatele v databázi a jednalo by se o uživatelsky přívětivější variantu používání webové aplikace.

Možnou doplňující funkcionalitou v budoucnu může být odemknutí zámku na dálku skrze webovou aplikaci. Tímto bude možné odemknout na požádání zámek veřejnosti nebo studentovi, který nebude mít platnou RFID kartu.

V budoucnu lze navrhnout tří-faktorovou autentizaci. Tato autentizace by se skládala z vlastnictví RFID karty, zadání jednorázového hesla a biometrického ověření identity. Při žádání o RFID kartu musí žadatel dodat svoji fotografii. Tato fotografie je uložena v univerzitním systému. Přístupový systém by musel být hardwarově rozšířen o kameru, pomocí které by snímal fotografii uživatele. Muselo by být vytvořeno rozhraní umožňující získání fotografie z univerzitního systému. Obě fotografie by byly porovnány umělou inteligencí pro ověření identity.

Závěr

Cílem této práce bylo vytvoření přístupového systému pro studenty za využití více-faktorové autentizace. Práce zahrnovala seznámení se s funkcemi IS/STAG, výběr hardwaru a vytvoření softwaru. Výsledkem této bakalářské práce je funkční přístupový systém. Tento přístupový systém pro studenty může být nasazen kdekoliv v rámci univerzitní sítě Jihočeské univerzity. Pilotní nasazení systému bude zrealizována na Ústavu aplikované informatiky Jihočeské univerzity.

System umožňuje studentům přístup na základě identifikace RFID čipem a zodpovězením kvízové otázky z absolvovaných předmětů. Verifikace uživatele a získávání studijních podkladů pro výběr otázky probíhá na základě informací ze systému STAG. Klientům v systému lze nastavit, jaké studijní programy mají povolený přístup.

Otázky k jednotlivým předmětům jsou uchovávány v centrální databázi. Pro management systému byla vygenerována webová aplikace. Tato aplikace je vytvořena v programovacím jazyce PHP. Pro zpracování požadavků klienta bylo vytvořeno REST API. Toto API byly vyvinuto pomocí microframeworku Flask. Koncové body API jsou zabezpečené a vyžadují autentizaci klienta pro jejich použití. Pomocí tohoto API dochází k autentizaci uživatele a vybrání sady otázek pro klienta. Uživatel s klientem interaguje pomocí čtečky karet a dotykového displeje. Pro klienta byla napsána aplikace v programovacím jazyce Python. Pomocí displeje volí uživatel odpovědi na položené otázky.

Ke klientovi je možné se připojit pomocí SSH protokolu a přenášení souborů je realizováno pomocí SFTP protokolu. Komunikace mezi prvky systému probíhá pomocí HTTPS protokolu. Tímto je zajištěna bezpečnost přenosu dat.

Z důvodu koronavirových opatření k jaru roku 2021 bude fyzické nasazení pilotního zařízení realizováno po odevzdání této práce. Pro uživatele klientů i webové aplikace je přiložena uživatelská dokumentace. Na základě popsanych skutečností lze konstatovat cíle práce za splnění.

Díky této práci jsem se seznámil s jednodeskovými počítači a zjistil, jaké možnosti nabízí jejich užití. V budoucnu bych se rád této oblasti věnoval více a tuto bakalářskou práci případně dále rozvíjel. Prvky této práce mohou převzít ostatní školy užívající IS/STAG a použít je jako stavební kámen pro tvorbu nových přístupových systémů.

Seznam použité literatury

- [1] ALEXOVÁ, Šárka. *Analýza stylů učení a jejich možné využití při výuce odborných předmětů na střední škole*. [online]. Brno, 2013 [cit. 2021-02-6]. Dostupné z: <https://is.muni.cz/th/wgcu7/>. Diplomová práce. Masarykova univerzita, Pedagogická fakulta. Vedoucí práce Jiří Strach.
- [2] SABALE, Rajendra. Comparative Study of Prototype Model For Software Engineering With System Development Life Cycle. *IOSR Journal of Engineering* [online]. IOSR Journal of Engineering, 2012, 2012, 2(7), 23 [cit. 2021-02-6]. ISSN 2250-3021. Dostupné z: [http://www.iosrjen.org/Papers/vol2_issue7%20\(part-2\)/D0272124.pdf](http://www.iosrjen.org/Papers/vol2_issue7%20(part-2)/D0272124.pdf)
- [3] BHUVANESWARI, T a S PRABAHARAN. A Survey on Software Development Life Cycle Models. *International Journal of Computer Science and Mobile Computing* [online]. 2013, 2(5), 264 [cit. 2021-02-6]. ISSN 2320-088X. Dostupné z: <https://www.ijcsmc.com/docs/papers/May2013/V2I5201384.pdf>
- [4] *Používané bezpečnostní systémy: JIS* [online]. Plzeň: ekoTIP ID, 2009 [cit. 2021-02-15]. Dostupné z: <http://www.ekotip.cz/index.php?n=Systemy.HomePage>
- [5] Kartové centrum. *Helpdesk: Všechny IT služby / IT Services* [online]. České Budějovice: Jihočeská univerzita v Českých Budějovicích, [2021] [cit. 2021-02-6]. Dostupné z: <http://helpdesk.jcu.cz/sluzby/idkarta>
- [6] ISIC školám. *Projekt ISIC ŠKOLÁM* [online]. Praha: GTS Alive s.r.o, c2020 [cit. 2021-02-15]. Dostupné z: https://www.isicskolam.cz/faq_category/03-cipy-a-cipove-systemy-ss/
- [7] *Elektromechanické uzamykací systémy* [online]. Praha: ASSA ABLOY Czech & Slovakia, 2013 [cit. 2021-02-15]. Dostupné z: <https://www.assaabloyopeningsolutions.cz/PageFiles/131179/Katalog%20Elektromechanick%c3%bdch%20uzamykac%c3%adch%20syst%c3%a9m%c5%af%202013.pdf>
- [8] Základní informace. *IS/STAG Informační systém studijní agentury: Zájemci* [online]. Plzeň: Západočeská univerzita v Plzni, 2020 [cit. 2021-02-16]. Dostupné z: <https://is-stag.zcu.cz/zajemci/>
- [9] MACHULA, Tomáš. Studijní a zkušební řád Jihočeské univerzity v Českých Budějovicích: STUDIUM VE STUDIJNÍM PROGRAMU. *Jihočeská univerzita v Českých Budějovicích: O univerzitě: Dokumenty: Vnitřní předpisy JU: Studijní a zkušební řád Jihočeské univerzity v Českých Budějovicích* [online]. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2017 [cit. 2021-02-25]. Dostupné z: <https://www.jcu.cz/o>

-univerzite/dokumenty/internal_doc/studijni-a-zkusebni-rad-ju-v-ceskych-budejovicich

- [10] Nativní klient pod MS Windows. *IS/STAG Informační systém studijní agenty: Zájemci: Přístupy k systému: Windows klient* [online]. Plzeň: Západočeská univerzita v Plzni., 2017 [cit. 2021-02-16]. Dostupné z: https://is-stag.zcu.cz/zajemci/pristupy/tlusty_klient.html
- [11] PORTÁLOVÝ PŘÍSTUP K IS/STAG. *IS/STAG Informační systém studijní agenty: Přístupy k systému: Portálové rozhraní* [online]. Plzeň: Západočeská univerzita v Plzni, 2017 [cit. 2021-02-16]. Dostupné z: <https://is-stag.zcu.cz/zajemci/pristupy/portal.html>
- [12] Webové služby. *IS/STAG Informační systém studijní agenty: Zájemci: Přístupy k systému: Webové služby* [online]. Plzeň: Západočeská univerzita v Plzni, 2017 [cit. 2021-02-16]. Dostupné z: <https://is-stag.zcu.cz/zajemci/pristupy/webservices.html>
- [13] Webové služby nad IS/STAG: Referenční příručka. *IS/STAG Informační systém studijní agenty: Náповěda: Tištěná příručka portálového rozhraní* [online]. Plzeň: Západočeská univerzita v Plzni, ©2010-2014 [cit. 2021-02-25]. Dostupné z: https://is-stag.zcu.cz/NapovedyBuild/PortalNapovedy/build/pdf-zcu/Web-Services_IS-STAG.pdf
- [14] Basic Authentication. *API Development for Everyone: Resources: Docs: Specification V3.0: Authentication: Basic Authentication* [online]. Somerville: SmartBear Software, c2021 [cit. 2021-02-16]. Dostupné z: <https://swagger.io/docs/specification/authentication/basic-authentication/>
- [15] BERNERS-LEE, T., R. FIELDING a L. MASINTER. *Uniform Resource Identifier (URI): Generic Syntax: RFC3986*. San Jose: Adobe Systems, c2005.
- [16] *SBÍRKA ROZHODNUTÍ A OPATŘENÍ JIHOČESKÉ UNIVERZITY V ČESKÝCH BUDĚJOVICÍCH: Opatření rektora k uplatnění a zavádění jednotného identifikačního a přístupového systému na Jihočeské univerzitě v Českých Budějovicích*. R 340. České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2016, s. 1-2. Dostupné také z: https://www.jcu.cz/o-univerzite/dokumenty/rectors_proceedings/platna-opatreni/2016/r_340_jednotny_ident_a_prist_system.pdf
- [17] JIS karta. *Stránky uživatelské podpory: JIS karta* [online]. Plzeň: Západočeská univerzita v Plzni, 2020, 3. září 2020, 3.9.2020 [cit. 2021-02-17]. Dostupné z: <https://support.zcu.cz/index.php/Kategorie:JIS>

- [18] Přístupový systém. *Univerzita Hradec Králové: Celouniverzitní pracoviště: Oddělení informačních technologií UHK: IT poradna + návody: Učebny: Přístupový systém* [online]. Hradec Králové: Univerzita Hradec Králové, c2021 [cit. 2021-02-17]. Dostupné z: <https://www.uhk.cz/cs/univerzita-hradec-kralove/uhk/celouniverzitni-pracoviste/oddeleni-informacnich-technologii-uhk/it-poradna/ucebny/pristupovy-system>
- [19] Kartové centrum. *Ostravská univerzita: O univerzitě: Struktura univerzity: Zázemí univerzity: Centrum informačních technologií: Služby* [online]. Ostrava: Ostravská univerzita, 2021, 4. 1. 2021 [cit. 2021-02-17]. Dostupné z: <https://cit.osu.cz/20446/kartove-centrum/>
- [20] *Centrum výpočetní techniky Univerzity Palackého Olomouc* [online]. Olomouc: Univerzita Palackého v Olomouci, 2010 [cit. 2021-02-17]. Dostupné z: <http://cvt-publikace.upol.cz/mesicnik1004.pdf>
- [21] Identifikační průkazy. *Technická univerzita v Liberci* [online]. Liberec: Technická univerzita v Liberci, [2021] [cit. 2021-02-17]. Dostupné z: <https://www.tul.cz/intranet-studenti/identifikacni-prukazy/identifikacni-prukazy>
- [22] Průkaz studenta. *Univerzita Jana Evangelisty Purkyně v Ústí nad Labem: Studium: Pro studenty: Průkaz studenta* [online]. Ústí nad Labem: Univerzita Jana Evangelisty Purkyně v Ústí nad Labem, [2021] [cit. 2021-02-17]. Dostupné z: <https://www.ujep.cz/cs/prukaz-studenta>
- [23] Identifikační systém na univerzitě. *Univerzita Pardubice: Studium* [online]. Pardubice: Univerzita Pardubice, c2021 [cit. 2021-02-17]. Dostupné z: <https://www.upce.cz/studium/pro-uchazece/info-stud/is.html>
- [24] Přehled vysokých škol v ČR: Veřejné vysoké školy. *Ministerstvo školství, mládeže a tělovýchovy České republiky: Vzdelávání: Vysoké školství: Přehled vysokých škol v ČR* [online]. Praha: Ministerstvo školství, mládeže a tělovýchovy České republiky, c2013-2021 [cit. 2021-02-17]. Dostupné z: <https://www.msmt.cz/vzdelavani/vysoke-skolstvi/prehled-vysokych-skol-v-cr-3>
- [25] KONEČNÝ, Richard. *Návrh přístupového systému pomocí platformy Arduino* [online]. Zlín, 2020 [cit. 2021-02-27]. Dostupné z: <https://theses.cz/id/ed4f6y/>. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Ing. Stanislav Kovář.
- [26] *Raspberry Pi 4 Model B* [online]. 1. Cambridge: Raspberry Pi (Trading), c2019 [cit. 2021-02-18]. Dostupné z: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf

- [27] Raspberry Pi 4 Tech Specs. [*Raspberry Pi*] [online]. Cambridge: Raspberry Pi (Trading), c2021 [cit. 2021-02-18]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>
- [28] LattePanda 2G/32G: SKU: DFR0444. *LATTEPANDA* [online]. Šanghaj: Zhiwei Robotics, 2015 [cit. 2021-02-18]. Dostupné z: <https://www.lattepanda.com/products/1.html>
- [29] What's Orange Pi Plus 2E? [*Orange Pi*] [online]. Shenzhen: Xunlong Software CO., Limited, c2016 [cit. 2021-02-18]. Dostupné z: <http://www.orangepi.org/orangepiplus2e/>
- [30] Tinker Board: tinker your way to the future. *Asus* [online]. Tchaj-pej: ASUSTeK Computer, [2017] [cit. 2021-02-18]. Dostupné z: <https://www.asus.com/us/Single-Board-Computer/Tinker-Board/>
- [31] Tinker Board: ARM-based SBC with market-leading performance. *Asus* [online]. Tchaj-pej: ASUSTeK Computer, [2017] [cit. 2021-02-18]. Dostupné z: <https://tinker-board.asus.com/product/tinker-board.html>
- [32] 7inch Capacitive Touch Screen LCD (H), 1024×600, HDMI, IPS, Various Devices & Systems Support. *Waveshare Electronics* [online]. Shenzhen: Waveshare Electronics, [2017] [cit. 2021-02-20]. Dostupné z: <https://www.waveshare.com/7inch-hdmi-lcd-h.htm>
- [33] *Modul relé s úrovnovým měničem -5 V* [online]. Havlíčkův Brod: ECLIPSERA, 2018 [cit. 2021-03-27]. Dostupné z: <https://dratek.cz/docs/produkty/0/576/1517435132.pdf>
- [34] MIFARE Classic® EV1 1K - 4K: Features. *NXP Semiconductors: Products: RFID/NFC: MIFARE (HF): MIFARE Classic: MIFARE Classic EV1 1K-4K* [online]. Eindhoven: NXP Semiconductors, c2006-2021 [cit. 2021-02-20]. Dostupné z: https://www.nxp.com/products/rfid-nfc/mifare-hf/mifare-classic/mifare-classic-ev1-1k-4k:MF1S50YYX_V1
- [35] FINKENZELLER, Klaus. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication* [online]. 3rd ed. Chichester: John Wiley, c2010 [cit. 2021-02-24]. ISBN 978-0-470-69506-7. Dostupné z: https://books.google.cz/books?hl=cs&lr=&id=jAszZEQYa9wC&oi=fnd&pg=PT6&dq=rfid&ots=2KjAgk00Rk&sig=s8RxTvb5Y0WoRHBXQH69GZy0rnE&redir_esc=y#v=onepage&q=rfid&f=false

- [36] *MFRC522: Standard performance MIFARE and NTAG frontend* [online]. Eindhoven: NXP Semiconductors, 2016 [cit. 2021-02-20]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>
- [37] RC522 RFID Module. *Components101* [online]. Delhi: Components101, c2020 [cit. 2021-02-20]. Dostupné z: <https://components101.com/wireless/rc522-rfid-module>
- [38] Raspberry Pi OS. *[Raspberry Pi]* [online]. Cambridge: Raspberry Pi (Trading), [2021] [cit. 2021-03-03]. Dostupné z: <https://www.raspberrypi.org/documentation/raspbian/>
- [39] What is PostgreSQL? *PostgreSQL: The World's Most Advanced Open Source Relational Database: About* [online]. California: The PostgreSQL Global Development Group, c1996-2021 [cit. 2021-03-03]. Dostupné z: <https://www.postgresql.org/about/>
- [40] MySQL Community Edition. *MySQL: Products* [online]. Austin: Oracle, c2021 [cit. 2021-03-03]. Dostupné z: <https://www.mysql.com/products/community/>
- [41] Vyzkoušejte SQL Server v místním prostředí nebo v cloudu. *Datová platforma: Produkty: SQL Server* [online]. Praha: Microsoft, c2021 [cit. 2021-03-03]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-downloads>
- [42] *Top Rated Relational Databases Products* [online]. Austin: TrustRadius, c2013-2021 [cit. 2021-03-27]. Dostupné z: <https://www.trustradius.com/relational-databases>
- [43] *ManyDesigns Portofino 5* [online]. San Francisco: GitHub, c2021 [cit. 2021-03-04]. Dostupné z: <https://github.com/ManyDesigns/Portofino/>
- [44] *Portofino* [online]. Genoa: ManyDesigns S.r.l., c2005-2019 [cit. 2021-03-04]. Dostupné z: <https://portofino.manydesigns.com/en>
- [45] *InfyOm Laravel Generator* [online]. San Francisco: GitHub, c2021 [cit. 2021-03-04]. Dostupné z: <https://github.com/InfyOmLabs/laravel-generator>
- [46] Generator: Introduction. *Quality & Innovation: Open Source* [online]. Surat: InfyOm Technologies, c2015-2021 [cit. 2021-03-04]. Dostupné z: <https://www.infyom.com/open-source/laravelgenerator/docs/8.0/introduction>
- [47] PostgreSQL PHP Generator. *Database Management and Web Development tools: Products: PostgreSQL* [online]. New York City: SQL Maestro Group, c2002-2021 [cit. 2021-03-04]. Dostupné z: <https://www.sqlmaestro.com/products/postgresql/phpgenerator/>

- [48] *Tkinter — Python interface to Tcl/Tk*. 3.7.10. Wilmington [Delaware]: Python Software Foundation, c2001-20021. Dostupné také z: <https://docs.python.org/3.7/library/tkinter.html>
- [49] SFTP File Transfer Protocol - get SFTP client & server. *We protect your critical data: SSH Academy: SFTP* [online]. New York City: Communications Security, c2020 [cit. 2021-03-05]. Dostupné z: <https://www.ssh.com/ssh/sftp/>
- [50] FileZilla SSH and FTP Client. *We protect your critical data: SSH Academy: SSH Clients* [online]. New York City: Communications Security, c2020 [cit. 2021-03-05]. Dostupné z: <https://www.ssh.com/academy/ssh/filezilla>
- [51] BRAND, Thomas. Isolated SPI Communication Made Easy. *ADI RELEASES INDEPENDENT RESEARCH ON INDUSTRIAL INNOVATION: Education: Technical Articles* [online]. Norwood: Analog Devices, c1995-2021 [cit. 2021-03-05]. Dostupné z: <https://www.analog.com/en/technical-articles/isolated-spi-communication-made-easy.html#>
- [52] *Flask: web development, one drop at a time* [online]. Pallets, c2010 [cit. 2021-03-05]. Dostupné z: <https://flask.palletsprojects.com/en/1.1.x/>
- [53] What is a RESTful API? *Connect data from any system faster: Resources: Articles: API Integration* [online]. San Francisco: MuleSoft, c2021 [cit. 2021-03-12]. Dostupné z: <https://www.mulesoft.com/resources/api/restful-api>
- [54] The `pg_hba.conf` File. *PostgreSQL: The World's Most Advanced Open Source Relational Database: Documentation: 12: Client Authentication: The pg_hba.conf File* [online]. California: The PostgreSQL Global Development Group, c1996-2021 [cit. 2021-04-12]. Dostupné z: <https://www.postgresql.org/docs/12/auth-pg-hba-conf.html>
- [55] Deployment Options. *[Flask]* [online]. Pallets, c2010 [cit. 2021-03-16]. Dostupné z: <https://flask.palletsprojects.com/en/1.1.x/deploying/>
- [56] Google Python Style Guide. San Francisco: GitHub, 2018.
- [57] TAHIROVIČ, Erol. *How to make Raspberry Pi work with 1024×600 resolution* [online]. Erol Tahirovič, c2007-2017 [cit. 2021-03-16]. Dostupné z: <https://www.erol.name/how-to-make-raspberry-pi-work-with-1024x600-resolution/>
- [58] GPIO. *[Raspberry Pi]: Documentation: Usage: GPIO* [online]. Cambridge: Raspberry Pi (Trading), [2021] [cit. 2021-03-16]. Dostupné z: <https://www.raspberrypi.org/documentation/usage/gpio/>

- [59] THIERY, Louis. *SPI-Py: Hardware SPI as a C Extension for Python* [online]. San Francisco: GitHub, 2019 [cit. 2021-03-17]. Dostupné z: <https://github.com/lthiery/SPI-Py>
- [60] GOMEZ, Mario. *MFRC522-python* [online]. San Francisco: GitHub, 2018 [cit. 2021-03-17]. Dostupné z: <https://github.com/mxgxw/MFRC522-python>
- [61] ARCANJO, Lucas. *MFRC522-python* [online]. San Francisco: GitHub, 2018 [cit. 2021-03-17]. Dostupné z: <https://github.com/mxgxw/MFRC522-python/pull/80>

Seznam obrázků

1	Pohled studenta na předmět v IS/STAG	7
2	Ukázka URL adresy při volání služby IS/STAG	10
3	Vchod přístupný veřejnosti na Ústavu aplikované informatiky Jihočeské univerzity	14
4	Východ z Ústavu aplikované informatiky Jihočeské univerzity	15
5	Model čtečky RC522	19
6	Popsané zamýšlené fungování systému	22
7	Wireframes zamýšlené aplikace	28
8	Implementace entit pro ukládání otázek a předmětů	29
9	Ukázka nečistoty dat předmětů z IS/STAG	30
10	Implementace entit pro uložení uživatelů a jejich práv k webové aplikaci na zadávání otázek	31
11	Ukázka dat získaných voláním služby getUciteleKatedry	32
12	Implementace entit pro logy a správu zařízení	32
13	Ukázka dat z entity user_permission	35
14	Komunikace mezi klientem a API	41
15	Sada GPIO pinů Raspberry Pi 4B	42
16	GUI aplikace	47

Seznam tabulek

1	Popis pinů RC522 čtečky	20
2	Propojení pinů mezi čtečkou karet a Raspberry Pi	43
3	Propojení pinů relé a Raspberry Pi	43

Přílohy

Přílohy lze nalézt v přiloženém komprimovaném souboru.

A Databáze

B Webová aplikace pro management přístupového systému

C REST API

D Aplikace klienta