



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Vylepšení webového serveru pro humanoidní roboty na FM

Bakalářská práce

Studijní program: B0714A270001 – Mechatronika
Studijní obor: B0714A270001ME – Mechatronika
Autor práce: **Jan Svárovský**
Vedoucí práce: Ing. Miroslav Holada Ph.D.





Zadání bakalářské práce

Vylepšení webového serveru pro humanoidní roboty na FM

Jméno a příjmení: Jan Svárovský
Osobní číslo: M19000141
Studijní program: B0714A270001 Mechatronika
Zadávající katedra: Ústav informačních technologií a elektroniky
Akademický rok: 2021/2022

Zásady pro vypracování:

1. Seznamte se se stávajícím webovým portálem o humanoidních robotech NAO a Pepper na pracovišti školitele. Dále proveďte stručnou rešerši webových stránek výrobce těchto robotů (SoftBank Robotics), případně dalších relevantních webových portálů.
2. Navrhněte inovaci stávajících webových stránek o robotech na pracovišti školitele. Stránky budou zabezpečené se snadnou správou a aktualizací bezpečnostních certifikátů. Navrhněte i strukturu oprávnění přístupu k obsahu stránek.
3. Vytvořený návrh realizujte a naplňte relevantním obsahem. Jedná se především o databázi již realizovaných projektů v souvislosti s roboty na pracovišti školitele a o interaktivní přístup k již realizovaným rozhraním i k vlastním robotům přes VPN.
4. Navržený portál odladte a otestujte.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

podle potřeby dokumentace
30 – 40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] KESLER, Václav: Interaktivní ovládání humanoidních robotů NAO a Pepper přes webové rozhraní: Interactive control of humanoid robots NAO and Pepper via web interface. Liberec: Technická univerzita v Liberci, 2021. Bakalářská práce.
- [2] NANCHEV, Dragomir N. a Atsushi KONNO. Humanoid Robots. 1. Berlin, SRN: Elsevier – Health Sciences Division, 2016. ISBN 9780128045602.
- [3] Softbank Robotics [online]. Paris, France: SoftBank Robotics Europe, [c2005-2021] [cit. 2021-9-9]. Dostupné z: <https://www.softbankrobotics.com/>
- [4] World Wide Web Consortium [online]. c2021 [cit. 2021-9-9]. Dostupné z: <https://www.w3.org/>

Vedoucí práce:

Ing. Miroslav Holada, Ph.D.
Ústav informačních technologií a elektroniky

Datum zadání práce:

12. října 2021

Předpokládaný termín odevzdání:

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

prof. Ing. Ondřej Novák, CSc.
vedoucí ústavu

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

15. 5. 2022

Jan Svárovský

Vylepšení webového serveru pro humanoidní roboty na FM

Abstrakt

Původní webové stránky <http://nao.ite.tul.cz/>, běžící na virtuálním stroji na TUL, byly už nevyhovující. Tato práce měla za cíl vytvořit zabezpečenou a intuitivní webovou aplikaci s možností snadné administrace. Řešení je výsledkem kombinace webového serveru na platformě IIS, databázového serveru Microsoft SQL Server a vlastních skriptů PHP. Vzhled webového rozhraní je založen na HTML 5 a CSS stylech frameworku Bootstrap. Jedná se hlavně o systém oprávnění a uživatelských rolí, správu uživatelů a jejich oprávnění. V uživatelském rozhraní je možné upravovat obsahy stránek pomocí jednoduchého editoru. Na webu je také přehled a podrobnosti o dokončených studentských pracích týkajících se humanoidních robotů a pro uživatele s příslušným oprávněním také jejich správa. Dále jsou na webu fotografie a nechybí ani základní dokumentace k robotům a jejich programování. Dále jde o vylepšení databáze řešených projektů a zpřístupnění již realizovaných webových rozhraní a aplikací pro ovládání a programování robotů, ke kterým je přístup z lokální sítě.

Klíčová slova: Nao, Robot, IIS, PHP, SSL, Shibboleth

Abstract

Existing website on <http://nao.ite.tul.cz/>, hosted on virtual machine on UL was no longer sufficient. This work targeted making of a secured and intuitive web application with ability of easy administration. The result is a combination of a web server on the IIS platform, Microsoft SQL Server and own PHP scripts. Design of user interface is based on HTML 5 and CSS styles from the Bootstrap framework. It is mainly about system of user permissions and roles, management of users and their permissions. There is an option for easy editation of the content of pages in the user interface using simple editor. On the web there are also list and details about previously completed student works about humanoid robots and, for users with specific permission, also their management. Next, there are photos and basic documentation about robots and their programming. It is also improvement of database of completed projects and making available already realised web interfaces and applications for cotroll and programming of the robots, to which is access from local network.

Keywords: Nao, Robot, IIS, PHP, SSL, Shibboleth

Poděkování

Rád bych poděkoval vedoucímu práce panu **Ing. Miroslavu Holadovi Ph.D.** za ukázkou práce s roboty, konstruktivní konzultace a připomínky a návrhy k samotnému návrhu a realizaci práce. Dále bych rád poděkoval panu **Ing. Jiřímu Jěničkoví Ph.D.** za konzultaci ohledně implementace certifikátů zabezpečení, pánům **Ing. Igoru Kopetschke** a **Petru Adamcovi** za konzultaci a pomoc s implementací přihlašovacího systému SSO prostřednictvím univerzitních účtů LIANE. Dále bych chtěl poděkovat panu **Bc. Václavu Keslerovi** za pomoc se zpřístupněním jeho webového vývojového prostředí. V neposlední řadě bych rád poděkoval také své rodině a kamarádům za podporu během vytváření této práce a celého studia.

Obsah

Seznam zkratk	11
1 Původní web a možnosti zlepšení	12
2 Webové aplikace výrobce robotů	13
3 Základní nastavení serveru	16
3.1 Instalace a konfigurace PHP	16
3.2 Možnosti zabezpečeného spojení	17
3.3 Automatizovaná správa certifikátů	18
3.4 Integrace certifikátu do webového serveru	19
3.5 Přesměrování na zabezpečený protokol	20
3.6 Konfigurace Shibboleth Service Provider	21
3.7 Spuštění aplikace webového vývojového prostředí	23
4 Návrh webové aplikace	25
4.1 Webový server	25
4.2 Využití databáze	25
4.3 Vzhled uživatelského rozhraní	27
4.4 Návrh přihlašování	28
4.5 Realizace systému přihlášení	29
4.6 Systém uživatelských oprávnění	29
4.7 Struktura webu	30
5 Další webová rozhraní	33
5.1 Webové vývojové prostředí	34
5.2 Integrované webové rozhraní robota	34
5.3 Textová komunikace s roboty	35
6 Realizace vlastního kódu webu	37
6.1 Připojení databáze a definice konstant	38
6.2 Skript hlavičky a zpracování přihlášení	38
6.3 Společné funkce webové aplikace	38
6.4 Realizace nabídky hlavního menu	39
6.5 Zápatí stránek a propojení skriptů uživatelského rozhraní	41
6.6 Stránka přihlášení do aplikace	41
6.7 Proces odhlášení	42

6.8	Úvodní stránka a carousel	43
6.9	Statické články a editor úprav	44
6.10	Studentské práce a podrobnosti	44
6.11	Programy a přiložené soubory	45
6.12	Stránky dokumentace	45
6.13	Diskuse	46
6.14	Profil uživatele a úprava údajů	46
6.15	Správa funkcí a vlastností částí webu	47
6.15.1	Uživatelé	48
6.15.2	Správa robotů	50
6.15.3	Správa příspěvků	51
6.15.4	Správa médií	52
6.15.5	Správa programů	52
6.15.6	Studentské práce	52
6.16	Aplikace pro komunikaci s roboty	53
7	Závěr	55
8	A Obsah přiloženého CD	58

Seznam obrázků

1.1	Úvodní stránka původního webu.	12
2.1	Softbank Robotics - My Robots.	13
2.2	Softbank Robotics - My Missions.	14
2.3	Softbank Robotics - My Team - Users.	14
2.4	Softbank Robotics - My Team - Roles.	15
3.1	Konfigurace nově nainstalovaného PHP Manageru.	17
3.2	Správa SSL certifikátů služby IIS a self-signed certifikát.	18
3.3	Ukázka úvodní konfigurace klientské aplikace WIN-ACME pro správu SSL certifikátů.	19
3.4	Postup přiřazení SSL certifikátu k webovému serveru pomocí vazeb.	20
3.5	Seznam existujících pravidel modulu URL Rewrite a jejich pořadí.	21
4.1	Diagram struktury tabulky uživatelů.	27
4.2	Přihlašovací stránka Shibboleth v systému centrálního přihlášení LIANE.	28
4.3	Struktura jednotlivých stránek webu a jejich závislost na přihlášení uživatele.	30
6.1	Struktura složek webu na serveru, včetně souborů php a dalších částí HTML a skriptů.	37
6.2	Přihlašovací stránka. Na levé straně centrální přihlášení, napravo lokální účet.	41
6.3	Profil uživatele v režimu úprav vybraných údajů.	47
6.4	Správa vlastností webu.	48
6.5	Stránka správy stránek dokumentace a formulář na přidání nové stránky.	51
6.6	Seznam studentských prací a rozbalené podrobnosti jedné vybrané práce.	53

Seznam úryvků kódů

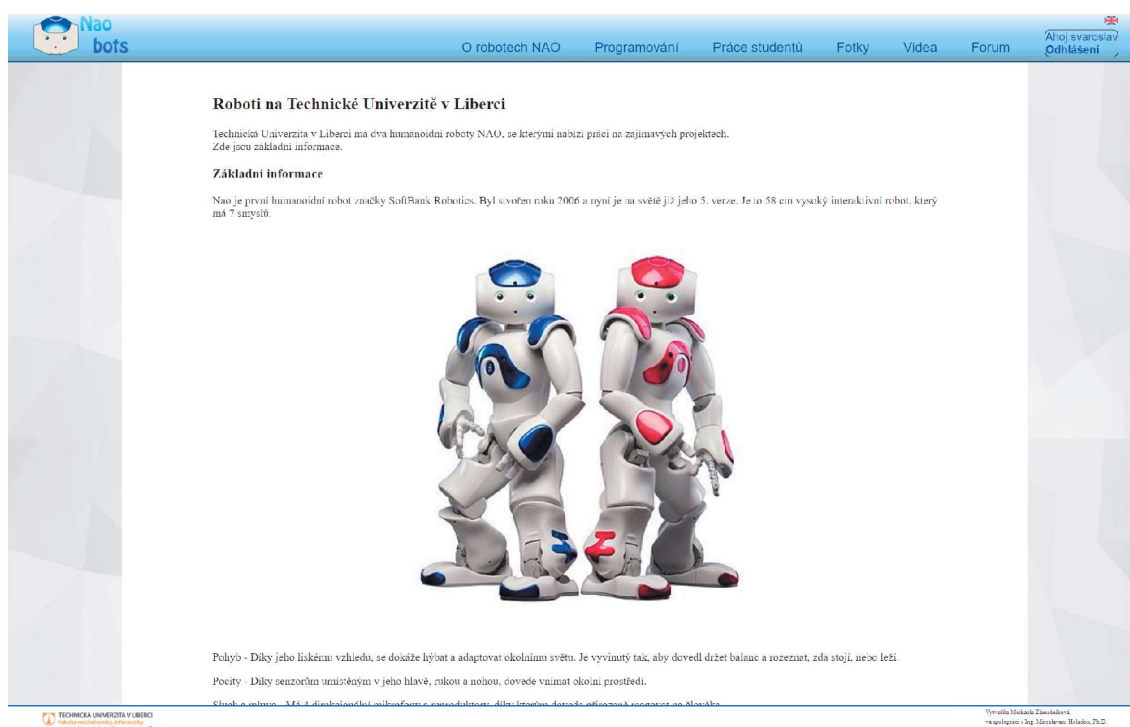
3.1	Ukázka konfigurace modulu Rewrite	20
3.2	Definice poskytovatele metadat v konfiguračním souboru Shibboleth	22
4.1	Ukázka propojení souboru se styly CSS v hlavičce stránky	28
5.1	Zprostředkování komunikace s integrovaným webovým serverem robota	36
6.1	Ukázka skriptu pro připojení k MySQL databázi	38
6.2	Část sestavení hlavního navigačního menu.	40
6.3	Zpracování dat z přihlašovacího formuláře na stránce přihlášení uživatele	42
6.4	Funkce ověření přihlášeného uživatele a provedení jeho odhlášení.	43
6.5	Navigační podokno stránek dokumentace	46
6.6	JavaScriptová funkce pro filtrování tabulek	49
6.7	Část funkce pro seřazení řádků v tabulce pomocí JavaScriptu	50

Seznam zkratek

WWW	World Wide Web
HTML	HyperText Markup Language
CSS	Cascading Style Sheet
PHP	PHP: Hypertext Preprocessor
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
SSL	Secure Socket Layer
IIS	Internet Information Services
ACME	Automated Certificate Management Environment
CA	Certificate Authority
JSON	JavaScript Object Notation
SSO	Single Sign On
SP	Service Provider
IDP	IDentity Provider

1 Původní web a možnosti zlepšení

Původní web, vytvořený studentkou Michaelou Zámečnickovou v roce 2018 jako ročníkový projekt, byl provozován na univerzitním serveru pomocí nezašifrovaného HTTP protokolu. Web umožňoval zobrazení základních informací o robotech NAO a odkazů na podrobnější dokumentaci. Dále zde bylo možné zobrazit krátké úryvky kódů určených pro programování robotů. V další záložce webu bylo po přihlášení možné zobrazit/stáhnout vybrané práce studentů. Na webu bylo také možné prohlížet nahrané fotografie a videa. Bylo zde i jednoduché fórum pro psaní vzkazů. Frontend webu byl napsaný v jazyce HTML a vzhled upravený styly CSS, backend byl vytvořený ve scriptovacím jazyce PHP, který zajišťoval určitou dynamiku webu a umožňoval použití uživatelských účtů.



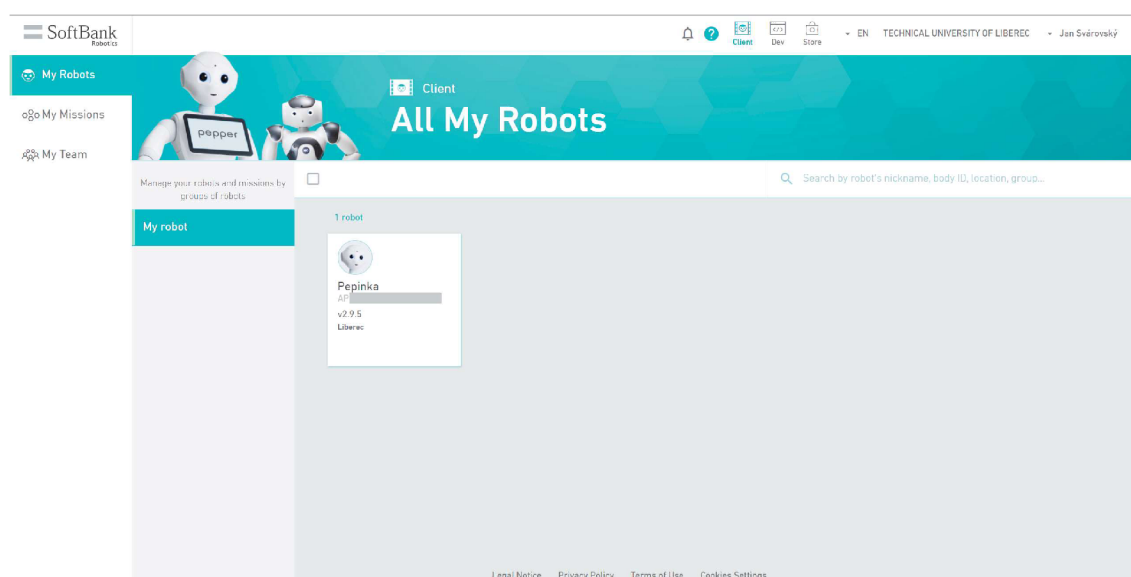
Obrázek 1.1: Úvodní stránka původního webu.

Obrázek 1.1 zobrazuje úvodní stránku původního webu od Michaely Zámečnickové.

2 Webové aplikace výrobce robotů

Ve webové aplikaci společnosti SoftBank Robotics je k dispozici přehledná správa vlastněných robotů (viz obrázek 2.1, knihoven doplňků (My Missions) a také uživatelů (viz obrázek 2.3 a uživatelských rolí (viz obrázek 2.1. [3]

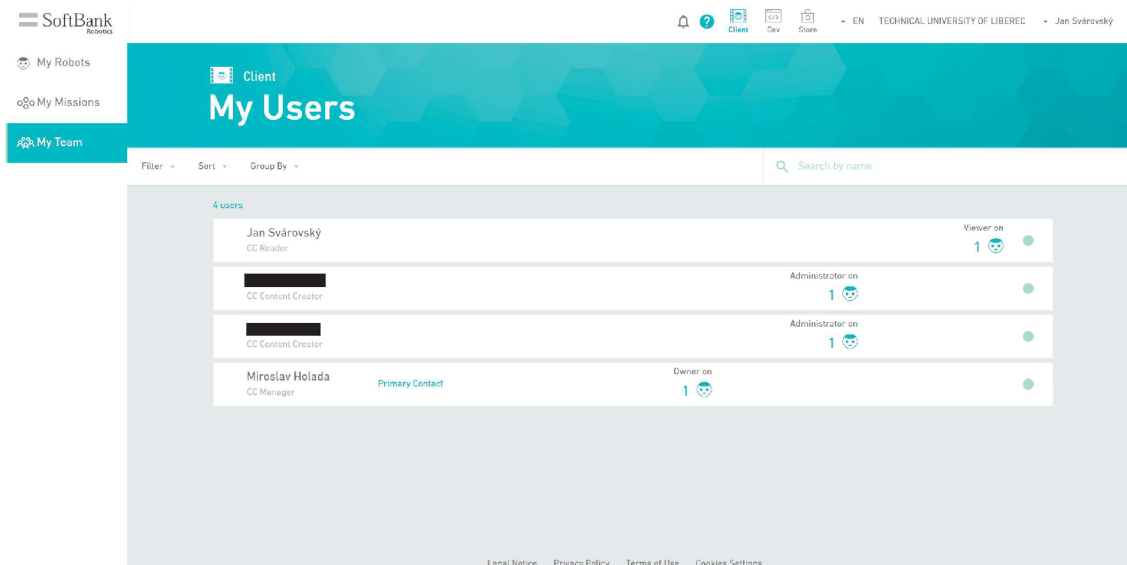
Tento web byl základní inspirací pro návrh mé webové aplikace.



Obrázek 2.1: Softbank Robotics - My Robots.

Na stránce *My robots* je možné prohlížet a spravovat přidané - propojené roboty. Jsou tu k dispozici základní informace o daných robotech a základní možnosti správy.

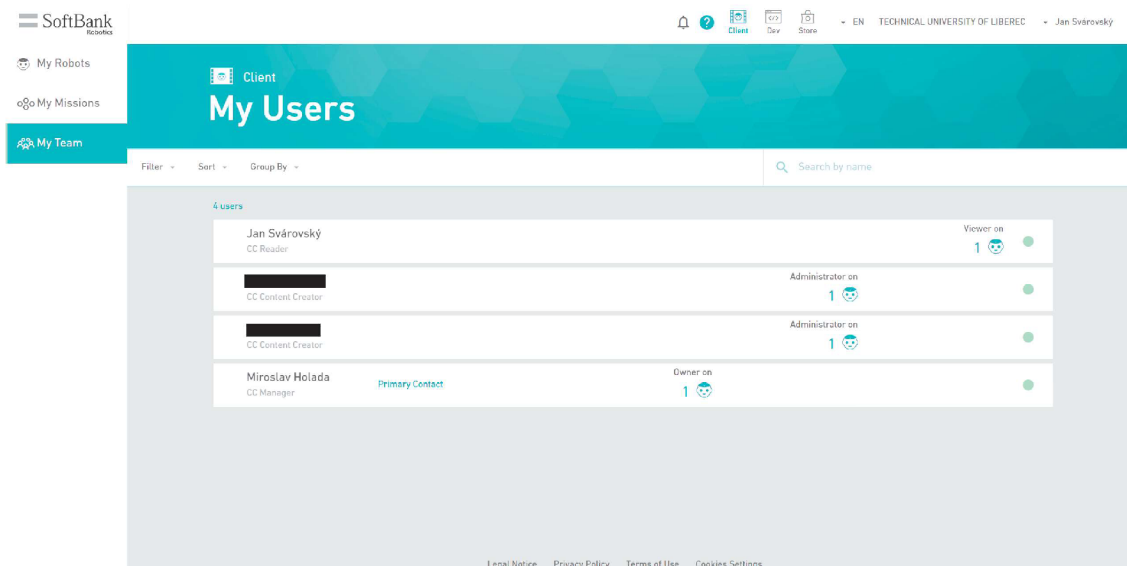
Po kliknutí na konkrétního robota je možné prohlížet i podrobnější informace včetně uživatelů s uděleným přístupem k danému robotovi.



Obrázek 2.2: Softbank Robotics - My Missions.

Dále je na tomto webu k dispozici přehled použitých balíčků, konkrétně jsou zde nazývány *Missions*. Ty mohou zahrnovat nějaká rozšíření pro aplikace robotů, jako další funkce nebo jazykové balíčky. Náhled stránky je možné si prohlédnout na obrázku 2.2.

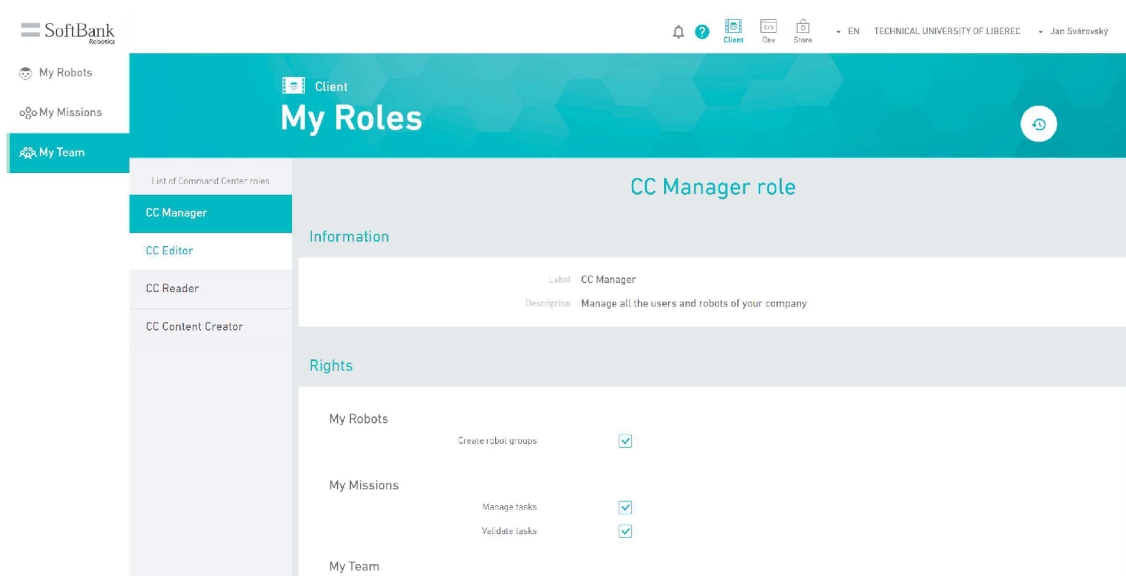
Položky v seznamu je možné vyhledávat, řadit a filtrovat. Po kliknutí na vybranou položku se zobrazí více podrobností.



Obrázek 2.3: Softbank Robotics - My Team - Users.

Web také umožňuje správu uživatelů a uživatelských rolí. Na stránce správy uživatelů (*My Team - Users*) je možné prohlížet a spravovat přidané uživatele dané organizace.

Uživatelé mohou mít přiřazenu jednu z několika definovaných uživatelských rolí. Na základě přiřazené role pak mají oprávnění používat některé funkce webu a práce s roboty.



Obrázek 2.4: Softbank Robotics - My Team - Roles.

Vzhledem k tomu, že tento web je společný pro všechny uživatele robotů od Softbank Robotics, je navržený poměrně univerzálně, ale neumožňuje přílišnou personalizaci pro danou organizaci, ani propojení s dalšími vlastními aplikacemi. Další nevýhodou je provozování webu v cloudu, protože vyžaduje připojení robotů k internetu.

3 Základní nastavení serveru

Strukturu backendu webu jsem volil s ohledem na strukturu stávajícího webu, změny v požadavcích na funkce i na své dřívější zkušenosti s vývojem webových aplikací.

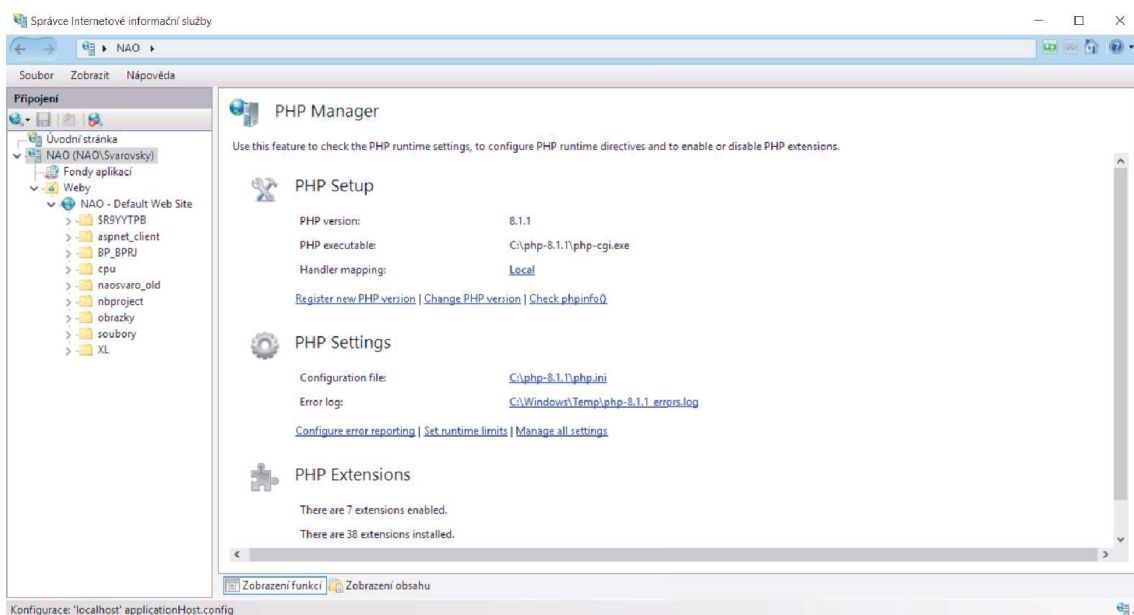
Cílem vypracování vlastního webu by tedy měla být možnost snadné správy uživatelů a robotů, a hlavně umožnění integrace dalších aplikací.

3.1 Instalace a konfigurace PHP

Jako server slouží virtuální stroj se systémem Windows Server, na kterém jsou spuštěny služby IIS a PHP. Na původním webu nebyly tyto služby propojeny a nebylo tudíž možné spravovat nastavení PHP jinak než pomocí příkazového řádku. Toto bylo potřeba změnit z důvodu ulehčení správy.

Jako první bylo třeba doinstalovat službu PHP Manager. Ta umožňuje správu nainstalovaných verzí i jejich konfiguraci přímo z ovládacího panelu služby IIS. PHP Manager byl nainstalován pomocí instalačního balíčku nejnovější vydané verze z GitHub stránky vývojářů[6]. Tento balíček byl zvolen na základě návodu v dokumentaci IIS na webu Microsoftu[7].

Dále byla provedena aktualizace instalovaného PHP (verze 7.0) na aktuálně nejnovější stabilní verzi (verze 8.1). Instalační balíček byl získán z webu vývojářů PHP v sekci ke stažení. Konkrétně se jedná o verzi x64 Non Thread Safe, jak je doporučeno na tomto webu [8]. Snímky obrazovky z průběhu instalace balíčku na obrázcích 3.1.



Obrázek 3.1: Konfigurace nově nainstalovaného PHP Manageru.

3.2 Možnosti zabezpečeného spojení

Vzhledem k tomu, že na webu je možné se přihlašovat a tudíž i používat hesla, je téměř nezbytné používat zabezpečené spojení. Na rozdíl od původního webu, který využíval protokol **HTTP**, tento umožňuje zabezpečené spojení pomocí **HTTPS**. Zabezpečení je docíleno další transportní vrstvou, **SSL**, která ke svému fungování využívá bezpečnostní certifikáty. Protože na serveru nebyl dosud žádný certifikát použit, bylo třeba získat nový.

V průběhu testování byla zvolena jednodušší, avšak méně bezpečná, varianta tzv. **Self-Signed** certifikátu. Jde o certifikát vystavený sám sebou, proto se nedá ověřit u certifikační autority zda je pravý a bezpečný. Tento způsob je sice bezpečnější než spojení bez certifikátu, protože využívá šifrovanou komunikaci, ale klient nemůže spolehlivě ověřit, zda je certifikát pravý a zda komunikuje opravdu se zamýšleným serverem.

Vzhledem k tomu, že bezpečnostní certifikáty jsou vydávány tzv. certifikační autoritou, je potřeba od některé z nich o vystavení certifikátu zažádat. V tuto chvíli byly k dispozici dvě možnosti.

První z nich bylo využít certifikační autority **CA CESNET, z. s. p. o.**, jejíž certifikáty využívá i většina dalších univerzitních služeb. Postup pro vystavení a následné obnovování těchto certifikátů však zatím nelze plně automatizovat, protože CESNET nedisponuje podporou technologie **ACME**. Veškeré žádosti jak o vystavení, tak i o obnovení již existujícího certifikátu je potřeba odesílat z vyhrazené, oficiální, emailové adresy. Již zpracovanou žádost je však možné dokončit připraveným automatickým skriptem. Ve výsledku je správa certifikátů od CESNETu částečně automatizovatelná.

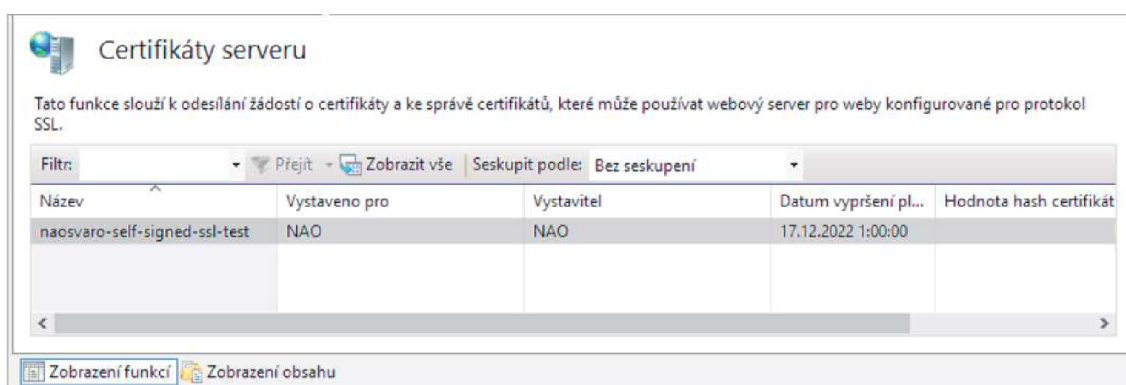
Druhou možností bylo využití open-source služby *Let's Encrypt*. Ta na rozdíl od CESNETu disponuje podporou technologie ACME a je tedy možná plná automatizace. Tato varianta zahrnuje instalaci klientské aplikace, která se stará o správu instalovaných certifikátů podporujících technologii ACME. Tato aplikace automaticky hlídá expiraci certifikátu a včas požádá o jeho obnovení. Postará se také o proces ověření ACME a následnou instalaci obnoveného certifikátu do webového serveru. Klientských aplikací je na výběr velké množství, a liší se jak doplňujícími službami, tak kompatibilitou s použitým operačním systémem a aplikací webového serveru. Seznam a podrobnosti o některých klientských aplikacích například na webu společnosti *ACME Client Implementations - Let's Encrypt*.

3.3 Automatizovaná správa certifikátů

Cílem Let's Encrypt a protokolu ACME je umožnění zprovoznění zabezpečených serverů pomocí HTTPS a automatizovatelné získávání důvěryhodných certifikátů bez jakéhokoliv zásahu uživatele. Toho je docíleno spuštěním klientské aplikace pro správu certifikátů na webovém serveru. Tento proces má dva kroky. Zaprvé, klientská aplikace dokáže certifikační autoritě, že má kontrolu nad požadovanou doménou a webovým serverem. Poté může klientská aplikace požádat, prodloužit a stornovat certifikáty pro danou doménu. Let's Encrypt identifikuje webový server podle veřejného klíče. Při první interakci klientské aplikace s certifikační autoritou Let's Encrypt vygeneruje klientská aplikace pár veřejného a privátního klíče. Následně prokáže certifikační autoritě, že má kontrolu nad danou doménou nebo doménami. [9]

Zpočátku nastavování nového webu byl použit dočasný SSL certifikát podepsaný tzv. sám sebou (viz obrázek 3.2). Po zvážení dalších možností a jejich konzultace se zástupcem správy univerzitní sítě a vedoucím práce, byla pro účely této práce zvolena varianta certifikátu vystaveného autoritou Let's Encrypt a klientské aplikace *WIN-ACME*, a to vzhledem k použitému operačnímu systému a webovému serveru.

(viz obrázek 3.3)



Obrázek 3.2: Správa SSL certifikátů služby IIS a self-signed certifikát.

```
C:\win-acme\wacs.exe
host names or identifiers (comma-separated) or filter them using one of the
options from the menu.

P: Pick bindings based on a search pattern
A: Pick *all* bindings

Binding identifiers(s) or menu option: a

1: nao.ite.tul.cz (Site 1)

Continue with this selection? (y*/n) - yes

Source generated using plugin IIS: nao.ite.tul.cz

Terms of service: C:\ProgramData\win-acme\acme-v02.api.letsencrypt.org\LE-SA-v1.2-November-15-2017.pdf

Open in default application? (y/n*) - yes

Do you agree with the terms? (y*/n) - yes

Enter email(s) for notifications about problems and abuse (comma-separated): pepper.robot@tul.cz

[nao.ite.tul.cz] Authorizing...
[nao.ite.tul.cz] Authorizing using http-01 validation (SelfHosting)
[nao.ite.tul.cz] Authorization result: valid
Downloading certificate [IIS] NAO - Default Web Site, (any host)
Store with CertificateStore...
Installing certificate in the certificate store
Adding certificate [IIS] NAO - Default Web Site, (any host) @ 2022.2.5 18:14:59 to store WebHosting
Installing with IIS...
Updating existing https binding nao.ite.tul.cz:443 (flags: 0)
Committing 1 https binding changes to IIS
Adding Task Scheduler entry with the following settings
- Name win-acme renew (acme-v02.api.letsencrypt.org)
- Path C:\win-acme
- Command wacs.exe --renew --baseuri "https://acme-v02.api.letsencrypt.org/"
- Start at 09:00:00
- Random delay 04:00:00
- Time limit 02:00:00
Adding renewal for [IIS] NAO - Default Web Site, (any host)
Next renewal scheduled at 2022.4.1 19:12:40
Certificate [IIS] NAO - Default Web Site, (any host) created

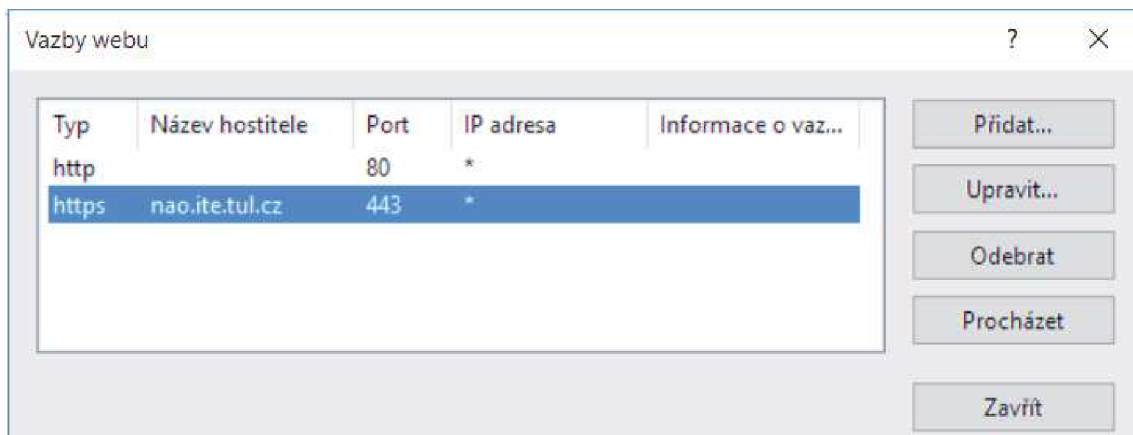
N: Create certificate (default settings)
M: Create certificate (full options)
R: Run renewals (0 currently due)
A: Manage renewals (1 total)
O: More options...
Q: Quit

Please choose from the menu: _
```

Obrázek 3.3: Ukázka úvodní konfigurace klientské aplikace WIN-ACME pro správu SSL certifikátů.

3.4 Integrace certifikátu do webového serveru

Pak už stačilo jen službě IIS nastavit, k čemu nový certifikát používat. To se nastavuje přes *Vazby* (tzv. Bindings) v okně správy služby (viz obrázek 3.4). Na stávajícím webu byla nastavena pouze jedna vazba, a to na port 80, což je standardní port pro přístup na web pomocí HTTP. Aby byl možný i zabezpečený přístup na web, byla zde vytvořena nová vazba na port 443 pomocí protokolu HTTPS. Port 443 je opět standardní port pro zabezpečenou komunikaci s webovým serverem.



Obrázek 3.4: Postup přiřazení SSL certifikátu k webovému serveru pomocí vazeb.

Další zásadní změnou v nastavení serveru byl modul **URL Rewrite**. Ten umožňuje úpravy adres v **HTTP** požadavcích podle přednastavených pravidel. V případě tohoto webu je modul použit pro skrytí koncovek **.php** v adresním řádku webového prohlížeče. Adresy tak vypadají přehledněji a je možné psát srozumitelně i do adresního řádku a zkrátit si tak cestu na často používané podstránky.

Modul Rewrite bylo také potřeba nastavit tak, aby upravoval adresy s cestami k souborům **.php**. Vzhledem k tomu, že během vývoje webu byl používán lokální server Apache, byla tato pravidla nastavena ve formátu, který používají webové servery Apache, tj. soubor **.htaccess**. Naštěstí modul Rewrite služby IIS umožňuje tato pravidla importovat a je tedy snadné je beze změny přenést. Po importu pravidel už není třeba psát do adresního řádku prohlížeče na konec adresy příponu **.php** a webový server si adresu sám upraví a vrátí uživateli správný výsledek.

```

RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^((([A-Za-z0-9\ -]+/)*[A-Za-z0-9\ -]+)?$) $1.php

```

Kód 3.1: Ukázka konfigurace modulu Rewrite

3.5 Přesměrování na zabezpečený protokol

V této fázi již bylo možné se serverem komunikovat jak pomocí nezabezpečeného protokolu **HTTP**, tak i pomocí šifrovaného **HTTPS**. Protože ale uživatelé standardně neuvádí v adresním řádku svého internetového prohlížeče jaký protokol má pro komunikaci použít (například výslovným uvedením "https://")

před názvem domény webu), a většina prohlížečů v současné době automaticky nevyužívá zabezpečené připojení jako výchozí možnost, bylo potřeba vynutit využívání zabezpečeného protokolu na straně webového serveru. K tomu je využito již zmíněné rozšíření **URL Rewrite**. V jeho konfiguraci bylo vytvořeno nové pravidlo, které při přijetí jakéhokoliv požadavku nevyužívajícího protokol **HTTPS** provede jeho přesměrování (stavový kód protokolu HTTP - 301) na protokol zabezpečený **HTTPS**, a to včetně zbylých parametrů z původního dotazu. Ukázka výsledné konfigurace na obrázku 3.5.

Name	Input	Match	Pattern	Action Type	Action URL	Stop Proce...
Redirect HTTP to HTTPS	URL path after '/'	Matches	(*)	Redirect	https://{HTTP...	True
Rewrite imported from htaccess	URL path after '/'	Matches	^([A-Za-z0-9\-\-]+/)*[A-Za-z0...	Rewrite	{R:1}.php	False

Obrázek 3.5: Seznam existujících pravidel modulu URL Rewrite a jejich pořadí.

3.6 Konfigurace Shibboleth Service Provider

Do webové aplikace je možné se přihlásit dvěma způsoby. Jedním je standardní přihlášení pomocí uživatelského jména/emailu a hesla. Tyto uživatelské účty se vytváří pomocí registračního formuláře a následného ověření některým ze správců webu.

Druhou možností, jak se přihlásit, je využití služby **SSO** pomocí Shibboleth. Tato metoda funguje tak, že webová aplikace si ověřuje identitu uživatele u serveru poskytovatele identity (Identity Provider). V případě univerzity je to server LIANE, který slouží pro jednotné přihlašování v rámci většiny služeb na univerzitě.

Nejprve bylo nutné nainstalovat na server aplikaci Shibboleth **SP** a během instalace vybrat možnost integrace do služby IIS. Následně bylo potřeba provést samotnou konfiguraci služby[14].

Většina konfigurace aplikace Shibboleth **SP** se provádí v konfiguračním souboru `shibboleth2.xml`, kde je potřeba upravit hned několik položek. První z nich je `entityID`, která slouží jako identifikátor nastavované služby. Zde byl upraven pouze DNS název serveru. Položka `Sessions` zůstala beze změny. Upravit ale bylo potřeba **SSO**, a to změnou hodnoty `entityID` na DNS název serveru univerzitního poskytovatele identity - LIANE (`https://shibbo.tul.cz/idp/shibboleth`) [15].

Dále bylo třeba Shibbolethu určit, jak má vypadat struktura souboru `metadata.xml`, který slouží k tomu, aby služba poskytovatele identity věděla, na jakých adresách má kontaktovat aplikaci Shibboleth na tomto webovém serveru. Tento soubor také obsahuje klíče

pro šifrování probíhající komunikace. To bylo provedeno tak, že do položky `<Handler type="MetadataGenerator" Location="/Metadata" signing="false" template="metadata-template.xml"/>` byla přidána hodnota `template` s cestou k souboru obsahujícího šablonu pro generování metadat. Finální metadata jsou pak dostupná na adrese <https://localhost/Shibboleth.sso/Metadata> na webovém serveru. Pak bylo potřeba vytvořit zmíněnou šablonu pro metadata - `metadata-template.xml`. V ní jsou uvedeny konkrétní hodnoty položek, jako například název služby (`<mdui:DisplayName xml:lang="cs">nao.ite.tul.cz</mdui:DisplayName>`) a její stručný popis (`<mdui:Description xml:lang="cs">Webová aplikace k humanoidním robotům na Technické univerzitě v Liberci.</mdui:Description>`), nebo kontaktní informace správce (`<md:ContactPerson contactType="technical">`).

Po restartování procesu `shibd` služby Shibboleth bylo možné pokračovat v konfiguraci. Bylo provedeno stažení metadat poskytovatele identity a ověření jejich pravosti pomocí veřejného klíče. Po úspěšném ověření bylo možné metadata nastavit jako zdroj v konfiguračním souboru `shibboleth2.xml` přidáním položky `<MetadataProvider type='XML' .../>`, kde je potřeba uvést adresu zdroje metadat a cestu k souboru veřejného klíče, pomocí kterého jsme metadata ověřovali v předchozím kroku.[10]

Nakonec bylo třeba vygenerovat pár klíče a certifikátu, které jsou potřeba pro zabezpečenou komunikaci mezi touto aplikací Shibboleth SP a serverem IDP. Cestu k vytvořeným souborům je nutné nastavit opět v konfiguraci Shibbolethu v položce `<CredentialResolver type="File" key="sp-key.pem" certificate="sp-cert.pem"/>`.

Konfigurace byla zakončena restartováním procesu `shibd` a předáním vygenerovaných metadat správci IDP.

```
<MetadataProvider type="XML" validate="true"
  url="https://metadata.eduid.cz/entities/eduid+idp"
  backingFilePath="eduid.xml" maxRefreshDelay="900">

  <MetadataFilter type="RequireValidUntil"
    maxValidityInterval="2419200"/>
  <MetadataFilter type="Signature"
    certificate="metadata.eduid.cz.crt.pem"
    verifyBackup="false"/>
</MetadataProvider>
```

Kód 3.2: Definice poskytovatele metadat v konfiguračním souboru Shibboleth

Nakonec byl nastaven i extraktor atributů, ten slouží k získání dat o přihlášeném uživateli z hodnot obdržných od IDP. V něm se používají standardizované identifikátory jednotlivých hodnot, které chceme získat, a to například

`<Attribute name="email" nameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" id="mail"/>` pro hodnotu uživatelské emailové adresy, nebo `<am:Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">` pro jedinečný identifikátor uživatele. Takto bylo nastaveno několik hodnot, které jsou užitečné pro používání webové aplikace. Tyto hodnoty jsou pak dostupné v serverových proměnných, pro `php` je to konkrétně `$_SERVER["eppn"]` a další hodnoty podle nastavených názvů.

3.7 Spuštění aplikace webového vývojového prostředí

Součástí realizace webu je i zpřístupnění již existujících webových rozhraní. Jedním z nich je i webová aplikace vývojového prostředí (`IDE`), kterou vytvořil jako výsledek bakalářské práce Bc. Václav Kesler. Aplikace je naprogramována v `Pythonu` a postavena na frameworku `Django`. Navíc využívá knihovnu `Naoqi` ke komunikaci s roboty Nao a Pepper.[1]

Aplikace má ke svému běhu poměrně konkrétní požadavky na verze jednotlivého softwaru. Konkrétně je potřeba Python ve verzi 2.7.16, knihovna Django pro Python ve verzi 1.11.29 a `pynaoqi` ve verzi 2.1.4.13. Posledním požadavkem je knihovna `pytz` ve verzi 2022.1, což je v době psaní tohoto dokumentu aktuální verze. Vzhledem k tomu, že z důvodu použití knihovny `pynaoqi` je nutné používat verzi Pythonu 2.7.16, která je v době vzniku této práce již nepodporovaná, (aktuální verze Pythonu je 3.10) bylo poměrně obtížné nalézt a správně nainstalovat všechny závislosti. Jako největší problém se ukázala nekompatibilita s 64-bit verzí Pythonu, která byla dost překvapivá. Projevem této nekompatibility byly různé chybové hlášky při pokusu o spuštění projektu, převážně se týkající nefunkčních nebo nekompatibilních knihoven `DLL` systému Windows. Problémy se však podařilo vyřešit přeinstalací Pythonu na 32-bit verzi.

Nejprve byl na server nainstalován Python, a to v konkrétní verzi 2.7.16. Vzhledem k zachování možnosti pozdějšího využití nainstalovaného Pythonu byla zvolena souběžná instalace této verze a nejnovější verze 3.10. Aby toto bylo možné, bylo potřeba nejprve nainstalovat starší verzi pomocí instalačního balíčku z webu vývojářů. Důležitým krokem při instalaci byla změna instalační složky, v tomto případě na `C:\Python\Python27`. Následně byla nainstalována novější verze, také do jiné instalační složky - `C:\Python\Python310`. Při instalaci byla deaktivována volba *Přidat do cesty (Add to PATH)*. Ta je důležitá, protože tyto cesty budeme upravovat manuálně v dalším kroku. Po dokončení instalace bylo nutné z ovládacích panelů Windows otevřít nabídku proměnného prostředí a upravit cestu `path`. Zde bylo nutné přidat položky `C:\Python\Python27`, `C:\Python\Python27\Scripts`, `C:\Python\Python310` a `C:\Python\Python310\Scripts`. V případě instalace více než dvou verzí by byly přidány cesty i pro další verze.

Dalším krokem byla instalace virtuálních prostředí `venv` pro všechny nainstalované verze Pythonu. Tyto moduly byly nainstalovány prostřednictvím nástroje `pip`.

Následovalo vytvoření virtuálního prostředí pro spuštění konkrétní apli-

kace. Jednalo se o vytvoření prostředí s názvem *venv-ide*, a to v pythonu verze 2.7.16. Po vytvoření bylo prostředí aktivováno pomocí skriptu *venv-ide\Scripts\activate*.

Pak již bylo možné pro toto konkrétní prostředí doinstalovat potřebné knihovny. To bylo provedeno opět pomocí nástroje *pip*. Knihovna *pynaoqi* však tímto nástrojem instalovat nelze. Je nutné ji tedy přidat manuálně, a to zkopírováním jejího obsahu do složky virtuálního prostředí.

Po dokončení instalace bylo možné projekt spustit v Pythonu příkazem *py venv-ide\manage.py runserver*. Všechny výše uvedené příkazy z postupu byly prováděny v terminálu Powershell systému Windows.

Aplikace vývojového prostředí obsahuje vlastní webový server a je tedy kompletně samostatná. Aby ji bylo možné integrovat do webové aplikace, bylo nutné upravit její konfiguraci. V souboru *venv-ide website*

settings.py byla upravena cesta a typ používané databáze. Při spouštění aplikace bylo také potřeba poskytnout IP adresu a port, na kterých má být server spuštěn, jako argumenty příkazu *runserver*, a to ve standardní syntaxi pro Django.

Posledním krokem ke zpřístupnění aplikace byla definice pravidel reverzní proxy v systému IIS. Toho bylo docíleno přidáním pravidel modulu Url Rewrite typu Reverse proxy. Pro tento krok je nutné mít aktivní i modul Application Request Routing.

4 Návrh webové aplikace

Webová aplikace je, podobně jako na předchozím webu, postavena na **PHP**, který se stará o většinu dynamických funkcí, přístup k datům a uživatelské účty. Pro uživatele viditelná část webu je složena ze základní struktury **HTML**, konkrétně ve verzi **HTML 5**, a vzhled je definován pomocí stylů **CSS**, a to převážně založených na frameworku **Bootstrap**.

Web je hodně provázaný se systémem oprávnění, který definuje, k jakým funkcím a informacím mají jednotliví uživatelé přístup. Na základě těchto oprávnění jsou například skryté prvky uživatelského rozhraní, ke kterým daný uživatel přístup udělen nemá.

4.1 Webový server

Nový web je, stejně jako původní, provozován na virtuálním stroji s **Windows Server 2019**. Na něm je spuštěna služba **IIS** a na ní několik rozšíření. Asi nejdůležitějším rozšířením je **URL Rewrite**, který zajišťuje převádění url požadavků z webového prohlížeče klienta na konkrétní soubor na webovém serveru[?]. Je to z toho důvodu, že pro větší uživatelské pohodlí a přehlednost odkazů bylo zvoleno "skrytí" přípon souborů **.php** v adresách url. Jako další důležité rozšíření byl přidán **PHP Manager**, který slouží ke snadné správě nainstalovaných balíčků **PHP** na serveru a k výběru verze používané k běhu webových aplikací [11]. V menu tohoto rozšíření je také možné měnit některá nastavení balíčku **PHP**, anebo prohlížet soubory záznamů přístupu a chyb. Umožňuje také vypnutí/zapnutí jednotlivých modulů balíčku **PHP**.

4.2 Využití databáze

Webová aplikace pro svůj běh využívá **MySQL** databázi. Ta je provozována na stejném serveru jako webová aplikace a je založena na softwaru **Microsoft SQL Server** [5]. Databáze slouží k ukládání většiny často používaných dat webové aplikace kromě souborů příloh a obsahů programů, a samotného obsahu příspěvků a článků. Ty jsou ukládány jako samostatné soubory v příslušných složkách serveru. Tato varianta byla zvolena vzhledem k optimalizaci množství informací načítaných z databáze. Samotné soubory je sice možné ukládat do databáze také, ale toto použití výrazně navyšuje objem dat načítaných z data-

báze, a zvyšuje režii v datovém toku i využití procesoru než při načítání souboru přímo z disku.

Jednotlivé objekty databáze a jejich stručný popis:

- **Uživatelé (users)** – všechna data o uživateli včetně uživatelského jména, hashovaného hesla a dalších osobních údajů.
- **Roboti (robots)** – definice robotů včetně jednoznačného identifikátoru, zobrazovaného jména a IP adresy a portu pro komunikaci prostřednictvím webového rozhraní.
- **Role (roles)** – definice uživatelských rolí.
- **Oprávnění (permissions)** – definice existujících oprávnění a jejich krátký popis, tyto oprávnění jsou následně přidělovány uživatelským rolím a konkrétním uživatelům.
- **Oprávnění rolí (role_permissions)** – přiřazení jednotlivých oprávnění k rolím. Pouze propojuje tabulky jako relace N:N.
- **Oprávnění uživatelů (user_permissions)** – přiřazení jednotlivých oprávnění ke konkrétním uživatelům. Pouze pro specifická oprávnění, která již nezahrnuje uživatelskou roli.
- **Média (media)** – seznam a podrobnosti o mediálních souborech. Samotné soubory jsou uloženy ve složce serveru.
- **Studentské programy (programs)** – data o studentských programech kromě příloh. Přílohy jsou uloženy jako soubory v příslušné složce na webovém serveru.
- **Studentské práce (works)** – data o studentských pracích. Případné přílohy jsou uloženy jako soubory v příslušné složce na webovém serveru.
- **Příspěvky (articles)** – informace o příspěvcích v diskusi. Samotný obsah příspěvků je uložen ve formátu **JSON** ve složce.
- **Komentáře (comments)** – případné komentáře k některému z příspěvků v diskusi. Struktura dat v této tabulce umožňuje přidávání komentáře i k jinému komentáři místo k příspěvku.
- **Přihlášení (login)** – záznamy akcí týkajících se přihlašování a odhlašování uživatelů.
- **Klíče sezení (user_keys)** – klíče pro identifikaci přenosu sezení na server třetí strany, slouží k identifikaci uživatele a zachování jeho přihlášení, nebo předání nějakých uživatelských dat.
- **Statistika přístupů (tracker)** – záznamy o proběhlých **HTTP** dotazech na webový server včetně identifikátoru sezení připojeného klienta a adresy dotazu včetně argumentů.

Diagram části databáze týkající se uživatelských dat na obrázku 4.1, kompletní diagram databáze je k dispozici v příloze.

users	
PK	<u>id</u> (UINT, 5) AI
K	email (VARCHAR, 256)
K	firstname (VARCHAR, 32)
K	secondname (VARCHAR, 32) NULL
K	lastname (VARCHAR, 32)
U	username (VARCHAR, 32) NULL
K, FK	role_id (UINT, 3) 1
	password (VARCHAR, 256) NULL
	register_at (TIMESTAMP) CURRENT_TIMESTAMP
	register_ip (VARCHAR, 40)
	profile_image (VARCHAR, 40) default.png
	web_theme (VARCHAR, 8) auto
	liane (INT, 1) 0
	verified (INT, 1) 0
	verify_code (VARCHAR, 16) NULL
	verify_time (TIMESTAMP) NULL
	2fa_key (VARCHAR, 32) NULL
	active (INT, 1) 1
	last_activity (TIMESTAMP) CURRENT_TIMESTAMP

Obrázek 4.1: Diagram struktury tabulky uživatelů.

4.3 Vzhled uživatelského rozhraní

Pro zobrazovanou část webu uživateli je zvolena standardní kombinace struktury webových stránek **HTML** a definice stylů **CSS**.

Co se týče základního **HTML**, jsou použity prvky verze HTML 5, jako je například záhlaví “header” nebo zápatí “footer” a v neposlední řadě prvky pro zobrazování obrázků a videa.

Grafické styly webu jsou definovány pomocí kaskádových stylů **CSS** a to buď přímo definovaných v konkrétním **HTML** tagu nebo v odděleném souboru stylů **.css** propojeném v hlavičce stránky. Pro snazší úpravy vzhledu webu je použit framework **Bootstrap**, konkrétně ve verzi 3.3.7. Většina grafických úprav webu je proto definována pomocí standardních prvků tohoto frameworku.

```
<!-- dark theme css -->  
<link rel="stylesheet" href="assets/styles/dark-mode.css">
```

Kód 4.1: Ukázka propojení souboru se styly CSS v hlavičce stránky

4.4 Návrh přihlašování

Jednou ze základních funkcí webu jsou uživatelské účty. Aby bylo možné uživatele jednoznačně identifikovat, je potřeba aby se přihlásil pod svým uživatelským účtem. K tomu slouží systém přihlášení.

V původním návrhu byly nejprve dvě možnosti přihlášení. První z nich, vlastní uživatelské účty pouze pro tuto webovou aplikaci, vytvářené pomocí registrace. Jako druhá možnost bylo využití jednotného systému přihlášení metodou SSO pomocí služby Shibboleth. V této variantě by se uživatel přihlásil pomocí svých údajů centrálního přihlašování LIANE. Webová aplikace by si pak u autentizačního serveru této služby kontrolovala, zda je uživatel přihlášen či ne. Z pohledu uživatelského pohodlí by byla tato možnost výhodnější a jednodušší. V ideálním případě by fungovaly obě tyto možnosti přihlášení současně.



› LIANE › eduID

Obrázek 4.2: Přihlašovací stránka Shibboleth v systému centrálního přihlášení LIANE.

4.5 Realizace systému přihlášení

Na webu byly realizovány výše popsané metody přihlášení. Pokud je uživatel členem Technické univerzity, je vhodnější využít metody centrálního přihlášení LIANE. Tento způsob nevyžaduje registraci, jen při prvním přihlášení je nutné povolit přístup k údajům z centrálního systému. Uživatelský účet na tomto webu je následně založen automaticky. Další přihlášení už probíhá automaticky bez nutnosti souhlasu s poskytnutím informací.

Druhým způsobem přihlášení je využití lokálního účtu. Ten je nejprve nutné založit pomocí registrace. Při ní je nutné uvést základní údaje jako je emailová adresa, jméno a příjmení. Dále je nutné si vytvořit silné heslo, které pak bude sloužit k dalšímu přihlašování. Z důvodu kontroly nad nově registrovanými uživateli je nově založený účet po registraci neaktivní, a není možné se k němu přihlásit. Je nutné, aby některý ze správců webu tento nový účet schválil pomocí tlačítka aktivace ve správě uživatelů. Po schválení je účet aktivní a lze se pomocí něj přihlašovat.

Pokud by se uživatel po registraci stal členem Technické univerzity, nebo naopak přestal být jejím členem, je možné již existující účet převést na druhý typ. Převod je možný pouze s dostatečným oprávněním, a může být proveden správcem. Pokud je proveden převod z lokálního účtu na účet LIANE, je z uživatelových dat v databázi odstraněn záznam o jeho heslu. Pokud se jedná o převod z účtu LIANE na lokální, je nutné naopak heslo vytvořit. Pokud změnu provádí uživatel, může si heslo zvolit sám, pokud převod provedl správce, je heslo vytvořeno automaticky a zobrazeno v oznámení danému správci. Ten ho pak musí bezpečným způsobem sdělit uživateli a požádat ho o následnou změnu hesla. Tu už může uživatel provést sám.

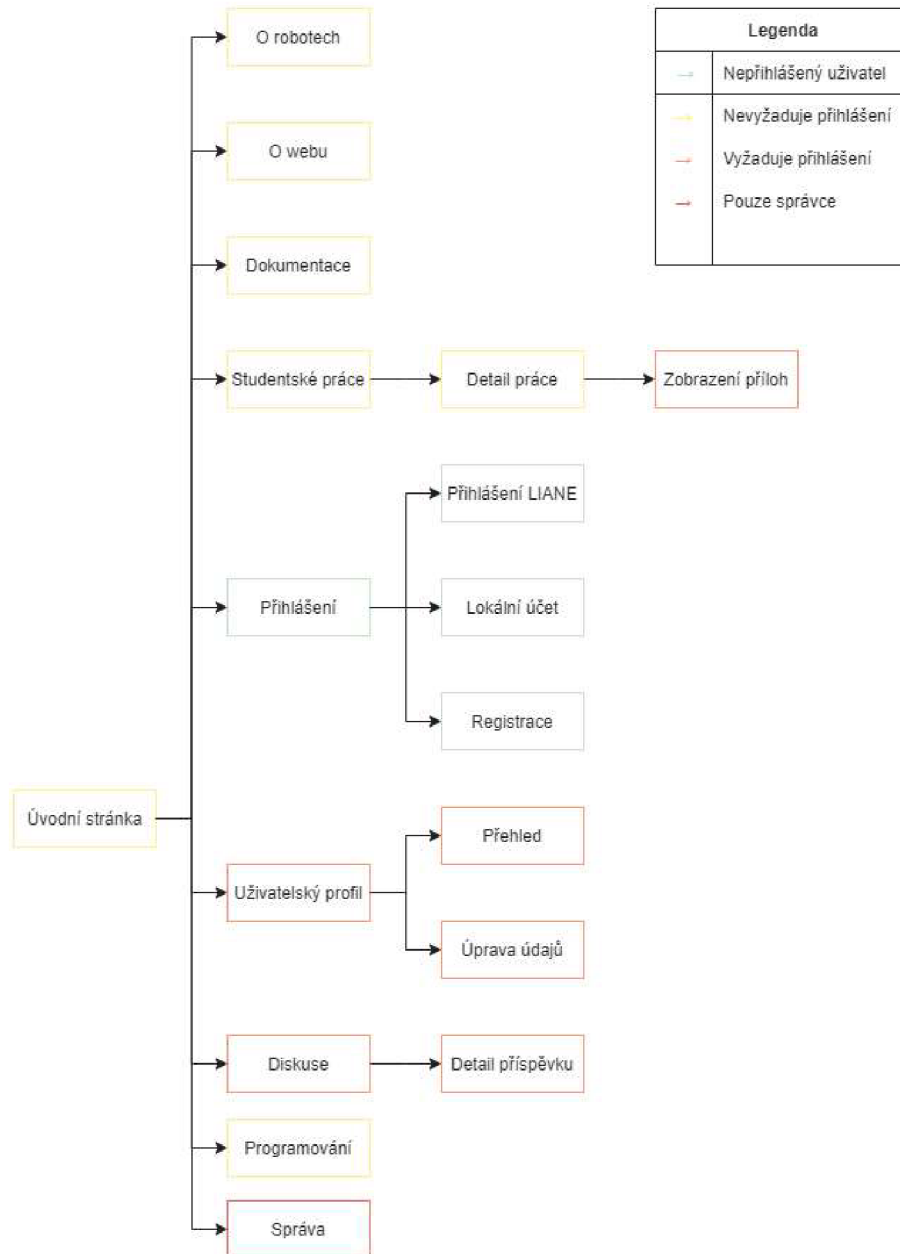
Při pokusu o přihlášení k účtu jinou metodou přihlášení než je daný účet, je uživateli zobrazeno upozornění, že je potřeba nejprve změnit typ účtu. V případě, že je uživatel přihlášen lokálním účtem, a zároveň si vyžádá přihlášení centrálním účtem, má centrální přihlášení přednost a původně přihlášený uživatel je v danou chvíli automaticky odhlášen z lokálního účtu.

4.6 Systém uživatelských oprávnění

Na webu funguje systém oprávnění, a kromě uživatelů samotných, jsou tu ještě uživatelské role. Každý uživatel má přiřazenu jednu z rolí, které jsou k dispozici, a to **Správce**, **Programátor**, **Zaměstnanec**, **Student** a **Host**. Nově registrovaný uživatel má automaticky přiřazenu roli hosta. Ke každé roli se vztahují konkrétní oprávnění, která automaticky přebírají všichni uživatelé s danou rolí. Navíc lze, z důvodu větší flexibility, přiřazovat oprávnění i individuálně. Takový uživatel má pak přiřazena všechna oprávnění své role a k tomu navíc individuálně nastavená oprávnění.

4.7 Struktura webu

Web je rozdělen na několik částí, které jsou přístupné v závislosti na aktuálním stavu sezení uživatele. Některé části (např. Přihlášení a Registrace) jsou přístupné pouze pro nepřihlášeného uživatele. Pro zpřístupnění některých stránek je potřeba být přihlášen uživatelským účtem. Množství dostupných stránek webu se pak liší v závislosti na konkrétních oprávněních daného uživatele.



Obrázek 4.3: Struktura jednotlivých stránek webu a jejich závislost na přihlášení uživatele.

Další stránky, jako je správa částí webu, jsou dostupné pouze pro uživatele s dostatečnými oprávněními. Pokud má uživatel příslušná oprávnění, má v hlavním menu také položku Správa. V té se nachází rozbalovací nabídka s jednotlivými kategoriemi správy, které jsou opět přístupné na základě uživatelských oprávnění.

Úvodní stránka je to první, co uživatel vidí po příchodu na web. Najdete zde:

- **Navigační lištu** – menu u horního okraje, slouží k navigaci po webu. Menu je stejné pro všechny stránky webu, liší se však podle toho zda je uživatel přihlášen, případně jaká má konkrétní oprávnění.
- **Carousel** – galerie vybraných obrázků rotujících pořád dokola. V případě tohoto webu je vždy při načítání stránky vybráno několik náhodných obrázků z galerie, ty se pak s nastaveným časovým intervalem postupně střídají, nebo lze mezi nimi listovat pomocí navigačních šipek po stranách obrázku a nebo pomocí šipek na klávesnici. Ke každému obrázku je v jeho dolní části připraven název a krátký popis.
- **Informace** – pod galerií s obrázky je na úvodní stránce také krátký článek se základními informacemi.
- **Patička** – se nachází vždy na dolním okraji stránky a je stejná pro všechny stránky webu. Najdete zde například užitečné odkazy, informace o autorovi a v poslední řadě možnost změny jazyka webu.

O robotech je stránka se základními informacemi o humanoidních robotech, které zde na fakultě máme:

- **Navigační lištu** – opět stejně jako na všech stránkách.
- **Článek** – upravovatelný příspěvek, k zobrazení a úpravám slouží knihovna `Editor.js`, je tedy možné článek upravit a používat v něm například odrážkové seznamy, podnadpisy nebo obrázky.
- **Patička** – u dolního okraje stránky, opět stejně jako na ostatních stránkách.

O webu je stránka s informacemi o této webové aplikaci, jejím vzniku a dalších detailech:

- **Článek** – upravovatelný příspěvek pomocí editoru založeného na knihovně `Editor.js`.

Rozhraní – zde se nachází propojení s dalšími webovými rozhraními pro programování a ovládání robotů.

Programy je stránka věnována programům vytvořených studenty. Po kliknutí na vybraný program je možné prohlížet jeho detaily a přiložené soubory.

Studentské práce je stránka věnována pracím našich studentů. Nachází se zde seznam se základními údaji o pracích s možností jednoduchého řazení a filtrování. Po kliknutí na vybranou práci se zobrazí další podrobnosti a seznam příloh.

Dokumentace je stránka s různou dokumentací buď přímo k robotům, nebo k vytváření příslušných programů a obsluze dalších rozhraní.

Diskuse – zde najdete diskusní fórum. Uživatelé s dostatečným oprávněním zde mohou přidávat příspěvky a ostatní pod ně mohou přidávat komentáře.

Přihlášení – zde se může již registrovaný uživatel přihlásit. K dispozici jsou dvě metody přihlášení. Centrální přihlášení LIANE mohou využít všichni členové Technické univerzity. Tato metoda přihlášení nevyžaduje registraci, pouze stačí povolit přístup k informacím z centrálního systému. Pro uživatele mimo univerzitu je zde druhá metoda, a to lokální uživatelské účty. Takový účet je nejprve vytvořit pomocí registračního formuláře na stránce Registrace.

Registrace – slouží k registraci nových uživatelů na náš web.

Odhlášení – slouží k odhlášení aktuálně přihlášeného uživatele, jako jedna z mála stránek nemá vůbec žádnou grafickou podobu, ihned po vykonání kódu je uživatel přesměrován na úvodní stránku

Profil je stránka s podrobnostmi o aktuálně přihlášeném uživateli. Uvedené údaje je možné prohlížet, a v režimu úprav i měnit. S dostatečnými oprávněními je zde možné převést typ účtu, nebo změnit heslo. Změna hesla je ovšem možná pouze pro lokální účty, u účtů LIANE se heslo mění centrálně na webu správy sítě.

Správa je skupina stránek určených pro správu webu, podle konkrétních oprávnění může uživatel spravovat například ostatní uživatele nebo příspěvky.

5 Další webová rozhraní

Jedním z hlavních úkolů nového webu bylo zpřístupnění dalších již existujících webových rozhraní. Ty jsou buď výsledkem předchozí práce některého ze studentů, nebo jsou již součástí některého z robotů.

V průběhu realizace tohoto projektu byly zvoleny dvě metody zpřístupnění těchto prací. První z nich je přesměrování na jiný webový server pomocí funkce `Application Routing`, což je modul služby IIS, který zajišťuje funkci proxy. Ve výsledku to znamená, že uživatel komunikuje ze svého internetového prohlížeče s jediným webovým serverem, a to s tím, na kterém je provozován tento web. Modul `Application Routing` pomocí nastavených pravidel rozhodne, zda má být požadavek zpracován webovým serverem služby IIS, nebo má být předán na jiný definovaný webový server. Jeho adresa je definována v příslušném pravidlu. V případě této práce jsou pravidla nastavena tak, že při obdržení požadavku na adresu například `/app/IDE` bude požadavek uživatele předán na server s adresou například `192.168.20.10`, na kterém běží webový server aplikace rozhraní pro grafické programování robotů Nao. Tato metoda přesměrování funguje pro kompletní komunikaci s cílovým serverem, ale neposkytuje žádnou kontrolu nad přenášeným obsahem. Veškerá komunikace je službou IIS pouze předávána dál.

Druhou metodou přesměrování je vlastní implementace `proxy` ve skriptu `php`. Ta je založena na funkci `curl`, která se stará o získání dat ze vzdáleného serveru pomocí protokolu `HTTP`. Tato metoda umožňuje plnou kontrolu nad přenášeným obsahem. Hlavní výhodou je také možnost opatřit takto přesměrovanou službu ověřením uživatele a případně i jeho oprávnění. Z tohoto důvodu byla metoda zvolena pro zpřístupnění webových rozhraní, která nedisponují vlastním ověřením uživatele, nebo je toto ověřením nedostatečné. Konkrétně je metoda použita pro přesměrování na webové rozhraní jednotlivých robotů. Při požadavku z uživatelova prohlížeče na adresu například `/robotui/nao-modry-1`, bude nejprve ověřena identita uživatele a příslušná oprávnění, a následně bude proveden požadavek z `php` skriptu na webový server vybraného robota. Adresa robota je získána z databáze na základě uvedeného identifikátoru. Po obdržení odpovědi od robota je obsah této odpovědi přeposlán uživateli. Tato metoda přesměrování má však i svou nevýhodu. A to, že je možné přesměrovat pouze komunikaci pomocí protokolu `HTTP`, případně `HTTPS`. Tato metoda přesměrování navíc zajišťuje i zabezpečenou komunikaci pomocí `HTTPS`, a to bez nutnosti, aby server, na který je komunikace přesměrována, podporoval zabezpečené spojení.

Obě uvedené metody mají tu výhodu, že cílový server nemusí být přístupný z internetu. To znamená, že může být pouze v lokální síti, jen je potřeba, aby do této sítě měl přístup i server, na kterém je tento web a služba IIS.

5.1 Webové vývojové prostředí

Dalším webovým rozhraním je také již realizovaná webová aplikace studenta Václava Keslera - *Interaktivní ovládání humanoidních robotů NAO a Pepper přes webové rozhraní*. Tato webová aplikace, provozovaná na frameworku Django v Pythonu, je samostatně funkční a dostupná na vlastní webové adrese. Úkolem zpřístupnění aplikace v rámci webu bylo tedy jen přesměrování veškeré odpovídající komunikace na server Djanga. Toho je dosaženo použitím modulu Application Routing ve službě IIS, a nastavení pravidel přesměrování. Přesměrovány jsou veškeré požadavky směřující na adresu <https://nao.ite.tul.cz/app/IDE/> a to včetně případné další cesty a parametrů. Problém by neměl nastat ani v případě použití dalšího protokolu, například WebSocket.

5.2 Integrované webové rozhraní robota

Pro komunikaci s integrovaným webovým serverem v robotu je použita vlastní implementace běžné HTTP proxy v jednom z php skriptů. To je hlavně z důvodu, že webové rozhraní robota umožňuje pouze základní metodu omezení přístupu, a to Base HTTP Auth. Jde o poměrně slabé zabezpečení, navíc není v nijak rozumné podobě možné využívat zabezpečené komunikace. Pro přihlášení k rozhraní je také nutné zadat přihlašovací jméno a heslo. To znamená, že každý uživatel webu by musel tyto údaje vyplnit, a tudíž i znát. Měl by tak k dispozici přístup odkudkoli vždy, kdy by byl konkrétní robot aktivní a připojen k síti.

Byla tedy zvolena varianta využívající proxy, aby bylo možné pomocí webové aplikace přebrat plnou kontrolu nad ověřením identity uživatele a následným přihlášením k rozhraní robota. Další výhodou této implementace je to, že rozhraní robota nemusí být přístupné z internetu, stačí aby mohla probíhat komunikace mezi konkrétním robotem a serverem webové aplikace. V tomto případě prostřednictvím lokální sítě, ve které se nacházejí pouze jednotliví roboti a zmíněný server.

Veškerá komunikace uživatelevo internetového prohlížeče probíhá pouze se serverem webové aplikace, která tyto požadavky následně zpracovává. Nejprve se kontroluje, zda je uživatel přihlášen a případně je společně s upozorněním přesměrován na přihlašovací stránku webu. Pokud kontrola proběhne v pořádku, začne se vyhodnocovat adresa dotazu. Ta může vypadat třeba takto: “<https://nao.ite.tul.cz/robotui/nao-modry-1/status?show=camera>”, kde “robotui” odkazuje na skript proxy a “nao-modry-1” je identifikátor robota. Proxy násled-

ně na základě těchto parametrů vyhledá v databázi robotů jednoho konkrétního. V databázi je také uložena IP adresa a port webového rozhraní robota. Tyto údaje jsou však před běžným uživatelem skryty. Zbylé parametry z požadavku se v nezměněné podobě předávají robotovi. Proxy takto zpracovaný dotaz pošle na adresu webového rozhraní robota a čeká na odpověď. Jakmile odpověď dorazí, přeposílá ji zpět uživateli a ten vidí už finální podobu integrovaného webového rozhraní robota.

S touto implementací je tu ovšem jeden problém. Pokud jsou v HTML dokumentu rozhraní robota propojeny další soubory, například styly nebo obrázky, je nutné, aby tyto soubory byly uvedeny pomocí relativní cesty, jinak budou z pohledu uživatelského prohlížeče nedostupné.

5.3 Textová komunikace s roboty

Zatím posledním implementovaným rozhraním je možnost textové komunikace přímo s roboty. Roboti Nao a Pepper podporují komunikaci prostřednictvím protokolu **TCP** se vzdáleným serverem. V našem případě je to TCP server pro základní textovou komunikaci, kdy jakýkoliv textový řetězec přijatý od jednoho z připojených klientů rozešle všem ostatním. V praxi to umožňuje komunikaci mezi všemi připojenými roboty a jejich vzájemnou koordinaci plněných úkolů. Dosud byla jedinou možností sledování a zásahu do této komunikace konzole klienta TCP, nejčastěji program **PUTTY**. Tato možnost byla samozřejmě k dispozici pouze v lokální síti, a to i z důvodu, že systém neměl žádnou kontrolu uživatelů. Cílem tedy bylo zpřístupnit tuto funkcionalitu uživatelům webu a to na základě jejich oprávnění.

Je tu ovšem ještě jeden malý problém. Komunikace mezi roboty probíhá pomocí přenosu textových řetězců prostřednictvím protokolu **TCP**. K této komunikační technologii je však nepraktické připojování z webového prohlížeče, a navíc by byla potřeba řešit problém přesměrování komunikace přes server přístupný z internetu. Dalším problémem by byla i absence ověření uživatelů. V takovém případě by se tedy mohl ke komunikaci připojit kdokoli, a to by nebylo úplně vhodné.

Byla tedy zvolena opět možnost proxy serveru. Ten je řešen v rámci konzolové aplikace pro windows naprogramované v **C++** a frameworku **Qt**. Součástí této aplikace je **Websocket** server, ke kterému se připojuje uživatel prostřednictvím svého internetového prohlížeče. Zde také probíhá kontrola uživatele a jeho oprávnění. Pak je uživateli umožněno zasílání a příjem textové komunikace. V opačném případě je uživateli zobrazeno upozornění a je s ním ukončeno spojení. Druhou částí aplikace je klient TCP. Pomocí něhož se proxy připojuje ke zmiňovanému TCP serveru pro komunikaci mezi roboty. Proxy následně přeposílá veškeré zprávy z TCP spojení všem klientům připojeným k **Websocket** a opačně.

```

// Use path as specific robot identifier and translate it
// to ip address based on database data about
// selected robot
$pathList = explode('/', $path);
$robotIdentifier = filter_var($pathList[0],
    FILTER_SANITIZE_STRING);
unset($pathList[0]);
$path = implode('/', $pathList);
if ($loggedIn) {
    $robotData = getRobotData($robotIdentifier);
    if ($robotData == (object)NULL) {
        appendMessage('warning', tr(
            $_SESSION['placeholder.warning']),
            tr($_SESSION['lang'], 'robotui.unknown-robot-0').
            $robotIdentifier.tr($_SESSION['lang'],
                'robotui.unknown-robot-1'));
        header('location:⌘'.__ROOT__.'/interfaces');
        exit;
    }
    if (!inArrayPermissions(
        'robotui.'. $robotData->type.'.connect',
        $currentUserPermissions)) {
        appendMessage('warning', tr(
            $_SESSION['placeholder.warning']), tr(
            $_SESSION['lang'], 'robotui.no-permission.robot'));
        header('location:⌘'.__ROOT__.'/interfaces');
        exit;
    }
    $robotAddress = $robotData->ip.(empty($robotData->port)
        ? ':80' : ':'.$robotData->port);
    $url = 'http://'.$robotAddress.'/'.$path;
    if (!empty($params)) {
        $url .= '?' . $params;
    }
}
}

```

Kód 5.1: Zprostředkování komunikace s integrovaným webovým serverem robota

6 Realizace vlastního kódu webu

Hlavní částí webové aplikace je kód na straně serveru, tzv. backend. Některé funkce webu jsou sice realizovány prostřednictvím jiných projektů (například úpravy a vykreslení editovatelných článků na webu), ale i tyto části potřebovaly dodělat nějakou část v backendu. Celý vlastní kód backendu je poměrně rozsáhlý, a proto byl rozdělený do několika souborů skriptu php. Na obrázku 6.1 je ukázka části stromové struktury složky webu.

Název	Velikost	Obsazeno	Souborů	Složek
C:\UwAmp\www\naosvaro\	51,8 MB	56,1 MB	2 387	114
assets	19,2 MB	19,3 MB	70	11
images	19,1 MB	19,2 MB	51	5
[35 souborů]	9,7 MB	9,8 MB	35	0
carousel	9,3 MB	9,3 MB	11	0
user	65,3 KB	72,0 KB	3	1
uploads	52,7 KB	56,0 KB	1	0
[2 souborů]	12,6 KB	16,0 KB	2	0
robot	29,0 KB	32,0 KB	2	1
[2 souborů]	29,0 KB	32,0 KB	2	0
uploads	0 Bajtů	0 Bajtů	0	0
lang	51,0 KB	56,0 KB	2	0
cs.json	26,8 KB	28,0 KB	1	0
en.json	24,2 KB	28,0 KB	1	0
styles	22,0 KB	24,0 KB	1	0
dark-mode.css	22,0 KB	24,0 KB	1	0
texts	10,1 KB	24,0 KB	6	0
about-robots-cs.json	1,0 KB	4,0 KB	1	0
about-robots-en.json	1,5 KB	4,0 KB	1	0
about-web-cs.json	2,7 KB	4,0 KB	1	0
about-web-en.json	2,7 KB	4,0 KB	1	0
main-page-cs.json	1,1 KB	4,0 KB	1	0
main-page-en.json	1,1 KB	4,0 KB	1	0
help	8,7 KB	16,0 KB	7	0
manage-articles.html	3,3 KB	4,0 KB	1	0
manage-robots.html	898 Bajtů	4,0 KB	1	0
manage-users.html	2,0 KB	4,0 KB	1	0
manage-works.html	2,6 KB	4,0 KB	1	0
manage-media.html	0 Bajtů	0 Bajtů	1	0
manage-programs.html	0 Bajtů	0 Bajtů	1	0
manage-settings.html	0 Bajtů	0 Bajtů	1	0
scripts	8,0 KB	8,0 KB	3	0
sortTableColumn.js	3,5 KB	4,0 KB	1	0
sortTableColumn2.js	4,0 KB	4,0 KB	1	0
navTabHashLink.js	517 Bajtů	0 Bajtů	1	0

Obrázek 6.1: Struktura složek webu na serveru, včetně souborů php a dalších částí HTML a skriptů.

6.1 Připojení databáze a definice konstant

V tomto skriptu je provedeno připojení k MySQL databázi a jsou zde definovány konstantní proměnné, které určují vybrané funkce webu. Tento skript je použit při každém požadavku na webový server a určuje základní funkce. V úryvku kódu 6.1 je propojení s databázovým serverem.

```
define( 'DB_SRV' , 'localhost:3306' );
define( 'DB_USR' , 'naosvaro' );
define( 'DB_PASS' , '<PASSWORD>' );
define( 'DB_NAME' , 'naosvaro' );
// Attempt to connect to MySQL database
$link = mysqli_connect(DB_SRV, DB_USR, DB_PASS, DB_NAME);
// Check connection
if($link === false){
    die("Could not connect to database. (Err0010)");
}
$link->set_charset("utf8");
```

Kód 6.1: Ukázka skriptu pro připojení k MySQL databázi

6.2 Skript hlavičky a zpracování přihlášení

Skript obsažený v hlavičce (header) se provádí při každém dotazu na webový server. Obsahuje definice základních proměnných a ověření přihlášení pomocí `PHP SESSION`. Pokud je uživatel přihlášen, probíhá zde také kontrola, zda nevypršela platnost sezení (`SESSION`) daného uživatele, a případně provede jeho odhlášení. V hlavičce je také realizováno ověřování přihlášení pomocí Shibbolethu.

6.3 Společné funkce webové aplikace

Toto je soubor PHP skriptu, který obsahuje většinu funkcí webové aplikace, všechny přehledně na jednom místě. Na tento soubor je odkázáno v hlavičce, a proto je jeho obsah k dispozici při každém dotazu na webový server. Funkce jsou uloženy v tomto souboru z důvodu přehlednosti, ale i proto, že některé z nich jsou potřeba na různých místech webu, a bylo by zbytečné je mít víckrát ve stejné podobě. Součástí definic funkcí jsou i komentáře s popisem dané funkce a jejich parametrů a návratových hodnot. Tyto komentáře jsou ve vývojovém prostředí dostupné jako nápověda.

6.4 Realizace nabídky hlavního menu

Pro hlavní nabídku v záhlaví webu byl zvolen standardní vzhled frameworku bootstrap. V levé části hlavní nabídky se nachází logo a název webu. Postupně směrem doprava následují odkazy na jednotlivé stránky webu. Zobrazení některých těchto podstránek je závislé na konkrétních oprávněních daného uživatele, a proto se ne vždy zobrazují všechny možnosti. Na pravé straně se nachází tlačítko pro přihlášení. V případě, že už je uživatel přihlášen, je zde uvedeno jeho jméno a profilový obrázek. Tato položka odkazuje na profil aktuálního uživatele a je zde v rozbalovací nabídce možnost odhlášení.


```

<?php
if($loggedIn)
{
    ?>
    <nav class="navbar
<?php echo $currentUserData->theme=="dark"?
'navbar-inverse ':'navbar-default';?> navbar-fixed-top">
<?php
}
else
{
    ?>
    <nav class="navbar navbar-default navbar-fixed-top">
    <?php
}
?>
<div class="container">
    <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed"
data-toggle="collapse" data-target="#navbar"
aria-expanded="false" aria-controls="navbar">
            <span class="sr-only">Toggle navigation </span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="<?php echo __ROOT__."/";?>">
            <span class="pull-left"
style="padding: 10px 10px 0 10px;">
                </span>
                <span class="pull-left" style="font-size: 20px;
color: #0ec90ff; padding-top: 15px;">
                    <?php echo tr($_SESSION['lang'], 'navbar.web-name'); ?>
                </span>
            </a>
        </div>
    </div>
</nav>

```

Kód 6.2: Část sestavení hlavního navigačního menu.

V závislosti na oprávněních konkrétního uživatele může být k dispozici i nabídka správy webu. Ta je řešena prostřednictvím rozbalovací nabídky v pravé části hlavního menu, vedle tlačítka uživatele. Položky v této rozbalovací na-

bídce se mohou lišit rovněž podle konkrétních oprávnění uživatele.

6.5 Zápatí stránek a propojení skriptů uživatelského rozhraní

V patičce je převážně obsah zápatí stránky, který je na většině stránek stejný. Obsahuje Kontaktní informace a základní informace o webu. Najdeme zde také užitečné odkazy a v neposlední řadě také přepínač jazyka webu.

6.6 Stránka přihlášení do aplikace

Na přihlašovací stránce webu jsou k dispozici dvě možnosti přihlášení. Je možné využít centrálního přihlášení prostřednictvím účtu LIANE. Pro tuto možnost je tu tlačítko *Centrální přihlášení*, po kliknutí na něj bude uživatel přesměrován na přihlašovací formulář LIANE, případně je rovnou provedeno přihlášení, a je mu zobrazena rovnou úvodní stránka webu již v přihlášeném režimu. Konkrétně pro tuto stránku bylo zvoleno trochu odlišné barevné schéma, kdy tlačítka vedena jako *primary* ve frameworku bootstrap, jsou v univerzitní fialové barvě, místo původní jasně modré.

Pro přihlášení prostřednictvím lokálního účtu je zde klasický formulář pro vyplnění uživatelského jména nebo emailu a hesla.

Obrázek 6.2: Přihlašovací stránka. Na levé straně centrální přihlášení, napravo lokální účet.

Na úryvku kódu 6.3 je část struktury webové stránky přihlášení, konkrétně dva sloupce, kdy v prvním je pouze tlačítko pro přihlášení prostřednictvím centrálního systému, a ve druhém klasický formulář pro jméno a heslo. Součástí kódu jsou i popisky s upozorněním pro případ nějaké chyby v zadaných

údajích. Například chybného hesla nebo neexistujícího účtu se zadaným identifikátorem.

```
<div class="col-xs-12 col-md-6 col-lg-6"
    style="text-align:center;">
  <a type="button" class="btn btn-primary btn-lg"
    href="Shibboleth.sso/Login"><?php echo tr(
    $_SESSION['lang'], 'login.button.shibbo.login'); ?></a>
</div>
<div class="col-xs-12 col-md-6 col-lg-6">
  <form class="form-signin" action="..." method="post">
    ...
  </form>
</div>
```

Kód 6.3: Zpracování dat z přihlašovacího formuláře na stránce přihlášení uživatele

6.7 Proces odhlášení

Odhlášení uživatele může proběhnout, stejně jako přihlášení, dvěma způsoby. Pokud byl uživatel přihlášen pomocí centrálního přihlašování, je během procesu odhlášení přeměrován na stránku odhlášení z centrálního přihlašovacího systému. Po dokončení odhlášení v centrálním systému už uživatel nemusí být přeměrován zpět na web a odhlášení je považováno za dokončené. Pokud byl uživatel přihlášen lokálním účtem, je odhlášení provedeno vyčištěním uživatelských proměnných z globální proměnné `$_SERVER` a přidáním záznamu o odhlášení do databáze. Tento proces je znázorněn na úryvku kódu 6.4, kde je pouze znázorněna podmínka, zda je uživatel právě přihlášen, a není znázorněna část kódu, která toto ověření vykonává.

```

if ($loggedIn) {
    // Check Shibboleth session
    if (isset($_SERVER['AUTH_TYPE'])
        && $_SERVER['AUTH_TYPE'] == 'Shibboleth') {
        // Redirect to login page
        header("location:␣" . __ROOT__ . "/Shibboleth.sso/Logout");
        // Must be before session_unset()
        logLoginAction('logout', 'shibbo');
    } else {
        // Must be before session_unset()
        logLoginAction('logout', 'manual');
        // Redirect to login page
        header("location:␣" . __ROOT__ . "/" );
    }
    // Temporarily store user's language
    $tmpLang = $_SESSION['lang'];
    $tmpTheme = $_SESSION['system_theme'];
    // Unset all of the session variables
    session_unset();
    // Destroy the session.
    session_destroy();
    // Redefine user's language to new, clean, session
    session_start();
    $_SESSION['lang'] = $tmpLang;
    $_SESSION['system_theme'] = $tmpTheme;
}

```

Kód 6.4: Funkce ověření přihlášeného uživatele a provedení jeho odhlášení.

6.8 Úvodní stránka a carousel

Na úvodní stránce webu byl vytvořen tzv. carousel, což je skupina obrázků, které se postupně střídají stále dokola. Konkrétně byly zvoleny některé fotografie z práce s roboty na univerzitě. Obrázky pro toto předvádění jsou uloženy ve specifické složce webu, a protože ke každému z obrázků je na webu zobrazen i název a krátký popis, patří k obrázkům i textový soubor s těmito informacemi. Konkrétně je použit soubor ve formátu **JSON**. V tomto souboru je seznam objektů, kdy pro každý obrázek v galerii je jeden objekt. Každý objekt obrázku obsahuje jeho identifikátory pro vybraná rozlišení, pokud je v nich k dispozici, a další podrobnosti, jako zmíněný název a krátký popis.

Skript **php**, který se spustí při každém zobrazení úvodní stránky, vybere vždy náhodně čtyři obrázky z nabídky a spolu s jejich názvem a popisem je zobrazí uživateli. Celkový počet zobrazovaných obrázků je možné nastavit v konfiguračním souboru. Vybrané obrázky jsou automaticky zbaveny duplicit, které

mohou vzhledem k náhodnému výběru nastat. Pokud je k dispozici velmi malé množství obrázků, je možné, že se uživateli zobrazí nižší počet obrázků, než je nastavený.

Obrázky jsou při vykreslování stránky prohlížečem načítány asynchronně v tzv. *lazy* režimu, tzn. že jsou načítány až v případě potřeby, třeba při průběžném listování mezi vybranými obrázky.

6.9 Statické články a editor úprav

Pro úpravu delších textů na webu byl zvolen framework Editor.js [17]. Ten umožňuje jednoduchou editaci textových příspěvků pomocí editoru integrovaného přímo ve stránce webu. Framework se rovněž stará o vykreslení již existujícího článku v rámci webu.

Celý editor je založený na Javascriptu, a funguje tedy zcela na straně webového prohlížeče. Server pouze editoru poskytuje čistá data o vykreslovaném příspěvku, ta jsou uložena ve formátu JSON v samostatných souborech na webovém serveru.

Framework editoru ovšem nedisponuje kompletním systémem ukládání změn, ten bylo nutné dodělat přímo na míru vytvářené webové aplikaci. Po přepnutí editoru do režimu úprav je možné editovat existující text, přidávat text další a také vkládat základní struktury jako jsou například tabulky, odkazy či obrázky. Všechny funkce jsou však celkem základní, a není u nich příliš možností bližší specifikace.

Vkládání obrázků do příspěvku se provádí například vložením odkazu na soubor vybraného obrázku, který musí být přístupný z internetu, a zároveň to musí být odkaz přímo na soubor s obrázkem ve standardně čitelném formátu - např. s příponou *.png*.

Pro zjednodušení vkládání obrázků do příspěvků je v rohu stránky zobrazena nabídka mediálních souborů nahraných na webovém serveru. Vybraný obrázek pak stačí pomocí funkce *Drag and Drop* přetáhnout na konkrétní místo v editoru.

Pro uložení změn v příspěvku je nutné kliknout na tlačítko *Uložit*, kdy se provede export obsahu příspěvku z editoru do formátu JSON a takto zpracovaný obsah se následně asynchronně odešle na server, kde proběhne jeho uložení. Při dalším zobrazení daného příspěvku už jsou zobrazeny provedené změny.

6.10 Studentské práce a podrobnosti

Na této stránce je možné prohlížet dokončené práce studentů týkající se humanoidních robotů na Technické univerzitě. Přehled je řešen vlastním stylem zobrazení kombinovaném s filtrovatelnou tabulkou jako na jiných stránkách

webu. V seznamu jsou na řádcích jednotlivé práce a základní podrobnosti. Po kliknutí na vybraný řádek se zobrazí více podrobností o práci.

6.11 Programy a přiložené soubory

Tato stránka funguje obdobně jako přehled studentských prací, jen jsou zde uvedené přidané programy a jejich podrobnosti. Více detailů o daném programu se zobrazí po kliknutí na tlačítko *Více podrobností*, zde je možné například prohlížet přiložené soubory.

6.12 Stránky dokumentace

Na pravé straně obrazovky je zobrazena vybraná stránka dokumentace. Na levé straně je umístěno navigační podokno, zde je seznam všech dostupných stránek dokumentace. Po kliknutí na vybranou položku je zobrazena konkrétní stránka dokumentace. Pokud není žádná stránka specifikována, je vybrána první v seznamu. Pokud uživatel vybere k zobrazení stránku, která neexistuje, je mu zobrazena chybová hláška.

Na konci vybrané stránky dokumentace je nabídka akcí, která obsahuje tlačítka pro ovládání editoru stránky (Povolit/zakázat úpravy, Uložit, Odstranit) nebo pro přepnutí režimu zobrazení. Vzhledem k tomu, že web podporuje tmavý režim aplikace, tuto funkcionalitu automaticky přebírá i editor příspěvků. Toto může být využito i v režimu úprav třeba pro kontrolu přístupnosti.

V úryvku je uvedena funkce pro načtení obsahu konkrétní stránky dokumentace na straně serveru. Vzhledem k tomu, že identifikátory stránek mohou obsahovat i znaky mimo základní `ASCII` a jsou obdrženy prostřednictvím UR'L adresy, je nutné při jejich zpracování použít standardní PHP funkci `urldecode`.

```

// Handle selection of specific documentation article
$selectedArticle = urldecode(trim(filter_input(INPUT_GET,
    'id', FILTER_SANITIZE_STRING)));
// Load content of selected documentation article
$articleContent = getDocuArticleContent($selectedArticle);
if (empty($articleContent)) {
    // Show warning message that requested article not exist
    appendMessage('warning', tr(
        $_SESSION['lang'], 'placeholder.warning'), tr(
        $_SESSION['lang'], 'documentation.message.no-exist-0')
        . $selectedArticle . tr(
        $_SESSION['lang'], 'documentation.message.no-exist-1'),
    false);
}

```

Kód 6.5: Navigační podokno stránek dokumentace

6.13 Diskuse

Tato stránka je dostupná pouze pro přihlášené uživatele. Jsou zde zobrazené jednotlivé příspěvky v přehledném seznamu, kde je vždy úryvek obsahu příspěvku a podrobnosti o vytvoření a autorovi a také titulek. Po kliknutí na vybraný příspěvek se zobrazí celý jeho obsah.

Příspěvky do diskuse mohou přidávat pouze uživatelé s příslušným oprávněním, správci mají navíc možnost upravovat i příspěvky jiných uživatelů. Tyto úpravy jsou dostupné po otevření konkrétního příspěvku, a to buď přímo z nástěnky diskuse nebo ze správy příspěvků.

Původně bylo v plánu umožnit i přidávání komentářů pod jednotlivé příspěvky v diskusi, tato funkcionality však není zatím implementována. Struktura databáze a práce s daty ve skriptech PHP však s přidáním této možnosti počítají. Součástí možnosti přidávání komentářů je i možnost přidávat komentáře pod jiné komentáře.

6.14 Profil uživatele a úprava údajů

Zde může uživatel spravovat své uživatelské údaje. Na úvodní stránce uživatelského profilu je přehled aktuálních údajů o daném uživateli, mezi které patří uživatelské jméno, email uvedený při registraci a jméno a příjmení uživatele. Jako další údaj je zde uveden i preferovaný vzhled webové aplikace.

Na záložce *Zabezpečení* je možné například změnit heslo, nebo zkontrolovat nedávnou historii přihlášení. Pokud má uživatel dostatečná oprávnění, může iniciovat i převedení typu svého účtu z lokálního na typ LIANĚ a naopak.

Pokud k tomu má uživatel oprávnění, má k dispozici ještě záložku *Oprávnění*, kde je možné prohlížet, jaká konkrétní oprávnění má udělena.

The screenshot shows a web interface for editing a user profile. At the top, there is a navigation bar with the Nao TUL logo and menu items: 'O robotech', 'Programování', 'Studentské práce', 'Dokumentace', and 'Diskuse'. On the right, there is a 'Správa' dropdown menu and a user profile icon for 'jan.svarovsky'. Below the navigation bar, there are two tabs: 'Základní' (selected) and 'Zabezpečení'. The main content area is titled 'Základní' and contains several form fields: 'Profilový obrázek' with a 'Vybrat soubor' button and 'Soubor nevybrán' text; 'Uživatelské jméno' with the value 'jan.svarovsky'; 'Email' with the value 'jan.svarovsky@tul.cz'; and 'Režim zobrazení' with a dropdown menu set to 'Světlý'. A red 'Zrušit' button is located below the form fields. At the bottom of the page, there is a footer with three columns: 'PŘIDEJ SE K NÁM' with a link to 'robotech'; 'UŽITEČNÉ ODKAZY' with links to 'ITE tul.cz' and 'FM tul.cz'; and 'O NÁS' with text about the university's humanoid robots and a link to 'Change language'. The footer also includes the logo of 'TECHNICKÁ UNIVERZITA V LIBERCI' and 'Fakulta mechatroniky, informatiky a mezioborových studií'. The copyright notice at the bottom reads '© 2021 Jan Svárovský @ TUL | webu | 0.1.2'.

Obrázek 6.3: Profil uživatele v režimu úprav vybraných údajů.

Úprava některých údajů o uživateli je možná pomocí tlačítka *Upravit údaje*, kdy po kliknutí se zobrazí některé hodnoty ve formě editovatelného pole. Jak vypadá stránka profilu uživatele v režimu úprav některých údajů je znázorněno na obrázku 6.3. Vzhledem k návrhu webu není možné upravit registrační email a v době dokončení práce ani uživatelské jméno. Tyto údaje je však možné upravit prostřednictvím správce webu přímo v databázi.

6.15 Správa funkcí a vlastností částí webu

Na obrázku 6.4 je stránka správy vlastností webu. V současné verzi zde sice není možné žádné položky měnit, stránka však poskytuje přehled pro správce, v jakém stavu se web zrovna nachází. V první části nalezneme přehled využití úložného místa na disku serveru. Je zde jak celkové využití systémového disku, tak i přehled využití prostoru webem, rozděleném podle jednotlivých částí webu. Například je zde vidět, kolik místa zabírají přílohy studentských prací, kolik je to celkem souborů a jakou procentuální část z celkového využití úložného místa využívá web.

Nao TUL O robotech Programování Studentské práce Dokumentace Diskuse Správa jan.svarovsky

Uložiště

Využití systémového disku

Celkem disk 42.1 %

Rozdělení využitého uložště podle typu souboru

Statické položky 27.4 % Média Studentské práce 68 %

Využití uložště podle typu souboru

Statické položky - 72 souborů, 26 121 kB
Média - 5 souborů, 4 365 kB
Příspěvky - 1 souborů, 0 kB
Nahrané soubory - 1 souborů, 0 kB
Studentské práce - 16 souborů, 64 751 kB
Studentské programy - 1 souborů, 0 kB
Stránky dokumentace - 3 souborů, 2 kB

Konstanty

Kořenová složka webu	
Interval automatického odhlášení	28800 s
Vyžadovat při registraci TUL email	Ano
Role přiřazená nově registrovaným uživatelům	Host (ID = 1)
CIDR rozsah univerzitních IP	147.230.0.0/16
Regulérní výraz pro kontrolu síly hesla	^(?=.*d)[^a-zA-Z0-9]){8,}\$
Velikost jednoho bloku pro nahrávání souborů po částech	8 MB
Maximální velikost jednoho souboru pro nahrávání po částech	16384 MB
Cesta ke složce, kde jsou uložena data studentských programů	<ROOT>/programs-data/

Obrázek 6.4: Správa vlastností webu.

Dále tu je přehled konstant, kterými se řídí některé funkce webu. Je zde například možné definovat, jakou roli bude mít nově registrovaný uživatel v případě registrace pomocí lokálního účtu, nebo pokud nebude poskytnut údaj o vztahu k univerzitě ze systému centrálního přihlášení. Také je zde možnost, která určuje, zda je možná registrace z jiné než univerzitní emailové adresy. Mimo dalších několika položek tu najdeme i regulérní výraz pro kontrolu síly hesla, který se využívá jak při registraci, tak i při případné změně hesla uživatelem.

6.15.1 Uživatelé

Tabulka uživatelů obsahuje informace o všech existujících uživateli webu. Je zde sloupec s nástroji, kde je možné zobrazit profil daného uživatele a dělat případné úpravy.

V tabulce uživatelů je možné použít funkci vyhledávání/filtrování. Filtr se aktivuje zadáním hledaného výrazu do pole vyhledávače nad danou tabulkou. Vyhledávání je provedeno po každém stisku klávesy ve vyhledávacím poli. Funkce hledání je dostupná u většiny tabulek na webu a všude funguje stejně. Na ukázce níže je uvedena funkce, která zajišťuje filtrování řádků dané tabulky. Filtrování je provedeno tak, že pokud se na daném řádku tabulky vyskytuje

hledaný výraz, je řádek ponechán viditelný, pokud se zde hledaný výraz nevyskytuje, je řádek tabulky skryt.

```
// Define function on page load
$(document).ready(function() {
  // Make event listener to keyup in search input field
  $("#filterTableInput").on("keyup", function() {
    var value = $(this).val().toLowerCase();
    // Call filter function on specified table with value
    // to search for
    $("#filterTable tr").filter(function() {
      // Check whether the table's row contains searched
      // value and toggle it's visibility
      $(this).toggle($(this).text().toLowerCase()
        .indexOf(value) > -1)
    });
  });
});
```

Kód 6.6: JavaScriptová funkce pro filtrování tabulek

Většina tabulek na webu rovněž umožňuje řazení podle hodnot ve vybraném sloupci. Podle kterého sloupce je aktuálně tabulka seřazena je označeno pomocí šipky v hlavičce tabulky nad příslušným sloupcem. Každý sloupec je možné řadit vzestupně i sestupně. V některých tabulkách jsou však i sloupce, které řazení neumožňují. Většinou se jedná o sloupec s nástroji, kde není řazení relevantní.

```

// Function to clear already visible sorting arrows
function clearArrows() {
  console.log("clearing sort arrows");
  var arrUp=document.querySelector("#sortTableobject from current iteration
      callback.call(scope, i, array[i]);
    }
  };
  // call the forEach function and set style for each of
  // the array's elements
  forEach(arrUp, function(index, value) {
    value.style = "display:none";
  });
  forEach(arrDown, function(index, value) {
    value.style = "display:none";
  });
}
// Function to set the correct sorting arrow to visible
function setArrow(n, direction) {
  console.log("setting sort arrow");
  if (direction == "asc") {
    var arr=document.querySelector("#sortTable

```

Kód 6.7: Část funkce pro seřazení řádků v tabulce pomocí JavaScriptu

6.15.2 Správa robotů

V této nabídce je možné prohlížet přidané roboty, a po kliknutí na vybraného robota i upravovat jeho vlastnosti. Pro přidané roboty je pak podle jejich konkrétního typu dostupné například otevírání integrovaného webového rozhraní.

Každý z robotů má nastavenou vlastní IP adresu a port, na kterých je možné s ním komunikovat. Má také přiřazen typ, který určuje, zda se jedná o robota Nao nebo Pepper. Na základě těchto vlastností mohou být v budoucnu zpřístupněny další funkce nebo metody komunikace.

Vlastnosti robota mohou být poskytnuty další aplikaci na základě požadavku na API webu, podobně jako je to řešeno u aplikace pro textovou komunikaci s roboty. Ta využívá API k ověření uživatele a získání základních uživatelských dat.

6.15.3 Správa příspěvků

Zde jsou přehledy všech textových příspěvků na webu, a to na jednotlivých záložkách podle typu daného příspěvku. Na první záložce s popisem *Statické stránky* se nachází seznam statických textů na některých stránkách webu. Například je zde text zobrazovaný na úvodní stránce, nebo na stránce *O robotech*. Tyto texty je možné pouze upravovat, není možné žádný z nich odebrat, nebo naopak přidat nějaký další. Množství či názvy textových souborů jsou pevně nastaveny ve struktuře webu a změny v tomto uspořádání jsou možné pouze úpravou vlastního kódu na webovém serveru. Uvedené stránky jsou k dispozici ve všech jazykových mutacích webu. Tyto jazykové verze je nutné upravovat každou zvlášť, protože fungují nezávisle na sobě.

Druhá záložka správy příspěvků je určena pro stránky dokumentace. Je zde přehledná tabulka se seznamem jednotlivých stránek a jejich základními údaji. Pomocí jednoduchého formuláře je možné přidávat další stránky a existující je možné naopak mazat. Každá stránka dokumentace je po vytvoření prázdná a skrytá před běžnými uživateli webu. Skryté stránky mohou zobrazit pouze uživatelé s příslušným oprávněním.

Screenshot of the 'Správa příspěvků' (Manage posts) page on the Nao TUL website. The page shows a navigation bar with 'Nao TUL' and various menu items. Below the navigation, there are tabs for 'Statické stránky', 'Stránky dokumentace', and 'Příspěvky v diskuzi'. A form for adding a new page is visible, with fields for 'Název' (Name) and 'Jedinečný identifikátor' (Unique identifier). Below the form is a table listing existing pages with columns for 'Nástroje', 'Velikost', 'Název', 'Přidal', 'Naposledy upravil', 'Přidáno', 'Naposledy upraveno', and 'Viditelné'. The footer contains contact information for the Technical University of Liberec and copyright information for Jan Svárovský.

Obrázek 6.5: Stránka správy stránek dokumentace a formulář na přidání nové stránky.

Poslední záložka je vyhrazena pro příspěvky v diskusi. Zde je opět přehledná tabulka se všemi příspěvky v diskusi a základními nástroji jejich správy. V tabulce lze opět filtrovat položky a podle některých sloupců tabulky i jednotlivé položky řadit.

6.15.4 Správa médií

Zde bylo vytvořeno základní rozhraní pro správu mediálních souborů na webu. V přehledné tabulce je možné prohlížet, filtrovat a řadit seznam souborů nahraných na server a podrobnosti o nich. V hlavičce stránky je možné nahrávat další soubory, a to asynchronně. Je tedy možné nahrávat větší množství souborů najednou a to i celkem velké soubory.

Pro usnadnění asynchronního nahrávání souborů byla zvolena knihovna `PIUpload` [18], která se stará o rozdělení souborů na části, tzv. *chunks*, na straně uživatelského prohlížeče, a na straně serveru se stará o jejich opětovné složení dohromady. Celkové limity pro velikost nahrávaných souborů a velikost jednotlivých částí, na které jsou soubory rozdělovány, jsou definovány v konfiguračním souboru webu, a je možné je v tomto souboru upravit.

Uživatel má možnost sledovat průběh nahrávání souborů prostřednictvím ukazatelů průběhu nahrávání. Pro jejich vykreslení byl využit standardní design frameworku `bootstrap`. O aktualizaci průběhu v grafu se stará funkce z knihovny `PIUpload`, která přidá odpovídající procenta průběhu v závislosti na odeslaných datech během nahrávání souboru.

Pro nahrávání souborů potřebuje uživatel příslušná oprávnění, která se vztahují i na konkrétní typy souborů, ty jsou určeny jejich příponou.

6.15.5 Správa programů

Tato stránka webu je velice podobná správě studentských prací. Je zde opět přehledný seznam existujících programů, ve kterém je možné filtrovat a řadit podle vybrané hodnoty. Přidání programu se provádí pomocí tlačítka, kdy je po kliknutí vytvořen nový program, který je následně možné upravovat. Po otevření podrobností vybraného programu je možné přidávat soubory příloh a upravovat základní informace o daném programu.

6.15.6 Studentské práce

V tomto přehledném seznamu je možné nalézt všechny předešlé studentské práce týkající se humanoidních robotů na Technické univerzitě.

Po kliknutí na řádek s vybranou prací se zobrazí o něco více podrobností včetně abstraktu. Zobrazí se také tlačítko *Více podrobností*, které odkazuje na stránku konkrétní práce. Na té jsou k dispozici další detaily a přílohy.

Nao TUL O robotech Programování Studentské práce Dokumentace Diskuse Správa jan.svarovsky

Studentské práce

Filtrovat a seřadit:

Hledat...

Typ práce	Název	Autor	Vypracováno	Přílohy
Bakalářská práce	Spolupráce robotů NAO	Autor práce: Pavel Váner, Datum vypracování: 14.05.2018 Nahráno: 07.02.2022, Naposledy upraveno: 28.02.2022, Nahráno uživatelem: svaroslav		
Ostatní	Využití dostupných senzorů robota Nao pro detekci objektů	Autor práce: Miroslav Eichler, Datum vypracování: 14.05.2018		
Bakalářský projekt	Webové stránky o humanoidních robotech Nao na ITE FM TUL	Autor práce: Michaela Zámečnicková, Datum vypracování: 03.01.2018		
Ostatní	Možnosti rozpoznávacích algoritmů v robotu NAO	Autor práce: Jan Vacek, Datum vypracování: 15.05.2017		
Ostatní	Interaktivní ovládání humanoidního robota Nao pomocí SDK	Autor práce: Václav Jiše, Datum vypracování: 15.05.2017		
Bakalářský projekt	Ovládání mobilním telefonem humanoidního robota Nao od firmy	Autor práce: Jan Vacek, Datum vypracování: 01.01.2016		
Ostatní	Humanoidní robot Nao od firmy Aldebaran - Interaktivní předčítá	Autor práce: Václav Jiše, Datum vypracování: 01.01.2016		
Ostatní	Návrh mobilního vozidla pro robota NAO	Autor práce: Šimon Škapik, Datum vypracování: 02.06.2015		
Ostatní	Pokročilé metody ovládání humanoidního robota NAO	Autor práce: Šimon Škapik, Datum vypracování: 01.01.2014		
	RPP I Interaktivní ovládání humanoidních robotů NAO			

Obrázek 6.6: Seznam studentských prací a rozbalené podrobnosti jedné vybrané práce.

6.16 Aplikace pro komunikaci s roboty

Tato aplikace, naprogramovaná v jazyce C++ na framewroku Qt, se stará o zprostředkování real-time textové komunikace mezi roboty a uživateli webu.

První součástí proxy je WebSocket server, ke kterému se připojují uživatelé webu ze svého webového prohlížeče.

Další částí je TCP server, který může být aktivován volitelně pomocí argumentů při spouštění aplikace. Ten slouží k připojení jednotlivých robotů z lokální sítě prostřednictvím protokolu TCP a následné textové komunikace.

Proxy se však dokáže i připojit k jinému TCP serveru. K tomu slouží část TCP klienta. V této konfiguraci se proxy připojí k jinému TCP serveru a přeposílá veškeré od něj přijaté zprávy do dalších komunikačních kanálů. Například všem připojeným klientům na svém WebSocket serveru.

K ověřování uživatelů a práci s uživatelskými daty využívá aplikace API rozhraní webové aplikace. Když uživatel webu navazuje spojení s WebSocket serverem proxy, je zároveň předán i klíč *sessionKey*, který je uložen v databázi webu a umožňuje identifikovat, o jakého uživatele se jedná. Tyto klíče mají jen velmi krátkou platnost, ale to ničemu nevádí. Proxy klíč použije ihned po

připojení uživatele k WebSocket spojení, a to k obstarání potřebných uživatelských dat z webové aplikace. K těmto datům patří uživatelské jméno a zda má uživatel oprávnění používat tuto metodu komunikace. Po použití již není sessionKey potřeba a je z databáze odstraněn, aby nemohlo dojít k jeho opětovnému použití.[13]

7 Závěr

Vývoj práce probíhal postupně a od určitého bodu byla webová aplikace již přístupná. Postupně byly přidávány další funkce a odstraňovány nalezené chyby. Většinu práce se podařilo dokončit i úspěšně otestovat. Některé části, jako například finální zpřístupnění webových rozhraní robotů Nao, nebylo možné zcela otestovat, protože prostředky pro komunikaci s roboty závisely na dalším projektu, který nebyl v době dokončení práce k dispozici. Tato část byla otestována jen v dočasné lokální síti.

Druhou, téměř dokončenou částí je zpřístupnění webového rozhraní vytvořeného panem Václavem Keslerem. Zde byl zatím nevyřešený problém s kompatibilitou použitého softwaru, hlavně vzhledem k nutnosti použití již nepodporované verze Pythonu potřebné k fungování knihovny Naoqi. Část této práce, která se stará o komunikaci s tímto webovým rozhraním, však byla dokončena a odladěna na jiném, testovacím, projektu v Pythonu. Pro tyto externí aplikace je vytvořeno i jednoduché api, které slouží k ověření přesměrovaného uživatele a získání základních údajů o něm. Toto api je využito pro přihlášení k proxy serveru vytvořenému pro textovou komunikaci s roboty pomocí TCP spojení.

Zbylé části práce jsou plně funkční a připravené na případnou aktualizaci a doplnění dalších částí. Na webu je možné spravovat uživatele, jejich role i konkrétní oprávnění. Dále je možné upravovat příspěvky v jednoduchém webovém editoru, upravovat a přidávat stránky dokumentace i nahrávat mediální soubory. Je k dispozici také přehled a podrobnosti o dokončených studentských pracích týkajících se humanodních robotů.

Aplikace je připravena i pro budoucí rozšíření, a to pomocí zmíněného api, propojení dalších uživatelských rozhraní nebo stránek dokumentace.

Literatura

- [1] KESLER, Václav: Interaktivní ovládání humanoidních robotů NAO a Pepper přes webové rozhraní: Interactive control of humanoid robots NAO and Pepper via web interface. Liberec: Technická univerzita v Liberci, 2021. Bakalářské práce.
- [2] NENCHEV, Dragomir N. a Atsushi KONNO. Humanoid Robots. 1. Berlin, SRN: Elsevier – Health Sciences Division, 2016. ISBN 9780128045602.
- [3] Softbank Robotics [online]. Paris, France: SoftBank Robotics Europe, [c2005-2021] [cit. 2021-11-28]. Dostupné z: <https://www.softbankrobotics.com/>
- [4] World Wide Web Consortium [online]. c2021 [cit. 2022-01-18]. Dostupné z: <https://www.w3.org/>
- [5] SQL Server 2019 | Microsoft [online]. [cit. 2022-02-24]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2019>
- [6] GitHub - phpmanager/phpmanager - Release 2,7 RTW [online]. [cit. 2022-02-12]. Dostupné z: <https://github.com/phpmanager/phpmanager/releases/tag/v2.7>
- [7] Using PHP Manager for IIS setup and configure PHP | Microsoft Docs [online]. [cit. 2022-02-12]. Dostupné z: <https://docs.microsoft.com/en-us/iis/application-frameworks/install-and-configure-php-on-iis/using-php-manager-for-iis-to-setup-and-configure-php>
- [8] Php.net - PHP for Windows [online]. [cit. 2022-01-12]. Dostupné z: <https://windows.php.net/download/>
- [9] How It Works - Let's Encrypt [online]. [cit. 2022-02-02]. Dostupné z: <https://letsencrypt.org/how-it-works/>
- [10] XMLAttributeExtractor - Service Provider 3 - Confluence [online]. [cit. 2022-03-15]. Dostupné z: <https://shibboleth.atlassian.net/wiki/spaces/SP3/pages/2065334421/XMLAttributeExtractor#Attributes>

- [11] PHP Manager 1.5.0 for IIS 10 [online]. [cit. 2022-02-07]. Dostupné z: <https://www.iis.net/downloads/community/2018/05/php-manager-150-for-iis-10>
- [12] URL Rewrite : The Official Microsoft IIS Site [online]. [cit. 2022-02-07]. Dostupné z: <https://www.iis.net/downloads/microsoft/url-rewrite>
- [13] Qt Framework - One framework to rule them all! [online]. [cit. 2022-04-05]. Dostupné z: <https://www.qt.io/product/framework>
- [14] Install on Windows - Service Provider 3 - Confluence [online]. [cit. 2022-03-28] Dostupné z: <https://shibboleth.atlassian.net/wiki/spaces/SP3/pages/2065335545/Install+on+Windows>
- [15] Shibboleth SP 3 (eduID.cz) [online]. [cit. 2022-03-21] Dostupné z: <https://www.eduid.cz/cs/tech/sp/shibboleth>
- [16] Ijason - Dark mode for Bootstrap [online]. [cit. 2021-12-01] Dostupné z: <https://gist.github.com/ijason/cf32446780c944d86f1c9cf71648056d>
- [17] CodeX - Editor.js [online]. [cit. 2021-12-01] Dostupné z: <https://codex.so/editor>
- [18] Plupload: Multi-runtime File-Uploader [online]. [cit. 2021-12-01] Dostupné z: <https://www.plupload.com/>
- [19] Bootstrap - The world's most popular responsive front-end framework [online]. [cit. 2021-12-03] Dostupné z: <https://getbootstrap.com/docs/3.3/>

8 A Obsah přiloženého CD

- Bakalářská práce ve formátu .pdf
- Zdrojové kódy webové aplikace v adresářové struktuře
- Struktura databáze ve formátu .sql
- Snímky obrazovky finální verze webu ve formátu .png
- Úvodní verze aplikace proxy pro websocket ve formě spustitelné aplikace .exe s příloženými DLL
- Metadata pro komunikaci klienta Shibboleth s poskytovatelem identity