

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**Webové rozhranie laboratória merania na báze LabVIEW
NXG**

Diplomová práca

2020

Bc. Maroš Mihal'

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Webové rozhranie laboratória merania na báze LabVIEW
NXG

Diplomová práca

Študijný program: Automobilová elektronika
Študijný odbor: Elektrotechnika
Školiace pracovisko: Katedra elektroniky a multimediálnych komunikácií
Školiteľ: prof. Ing. Ján Šaliga, Phd.
Konzultant:

Abstrakt v SJ

Práca sa zaoberá návrhom a realizáciou inovovaného riešenia laboratória merania s prístupom cez internet a správou klientov použitím vývojového prostredia LabVIEW NXG a jeho doplnkom LabVIEW NXG Web Module. Riešenie bolo realizované vytvorením webovej služby, ktorá zabezpečuje komunikáciu klienta so vzdialeným panelom pracoviska pomocou HTTP protokolu a platformy SystemLink. Prístup na vzdialený panel pracoviska je možný cez webový prehliadač bez nutnosti inštalovania zásuvného modulu. Aplikácia webovej služby ponúka nástroje na správu pracovísk, rezervácií a používateľov. Súčasťou práce je nové dynamické webové rozhranie, ktoré využíva webové technológie ako HTML, CSS a JavaScript.

Kľúčové slova v SJ

LabVIEW, LabVIEW NXG, LabVIEW NXG Web Modul, webové rozhranie, vzdialené laboratórium

Abstrakt v AJ

This thesis deals with the design and implementation of an innovative solution for measurement laboratory with remote access and client management using the development environment LabVIEW NXG and its software add-on LabVIEW NXG Web Module. The solution was implemented by creating a web service that ensures the client's communication with the remote panel of the remote laboratory workplace using the HTTP protocol and the SystemLink platform. Access to the remote panel is possible via a web browser without the need to install a plug-in. The web service application offers tools for managing workplaces, reservations and users. Part of the work is a new dynamic web interface that uses web technologies such as HTML, CSS and JavaScript.

Kľúčové slova v AJ

LabVIEW, LabVIEW NXG, LabVIEW NXG Web Module, web interface, remote laboratory

50451

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra technológií v elektronike

ZADANIE DIPLOMOVEJ PRÁCE

Študijný odbor: **Elektrotechnika**
Študijný program: **Automobilová elektronika**

Názov práce:

Webové rozhranie laboratória merania na báze LabVIEW NXG
Web site of remote laboratory based on LabVIEW NXG

Študent: **Bc. Maroš Mihaľ**
Školiteľ: **prof. Ing. Ján Šaliga, PhD.**
Školiace pracovisko: **Katedra elektroniky a multimediálnych telekomunikácií**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie diplomovej práce:

- zoznámte sa s existujúcim laboratóriom merania s prístupom cez internet
- zoznámte sa s vývojovým prostredím LabVIEW NXG a web modulom
- navrhnete inovované riešenie web stránok a web servera laboratória s využitím LabVIEW NXG
- prepracujte súčasné web stránky na inovovanú verziu a odskúšajte
- navrhnete a implementujete jednoduchú správu klientov pre prístup k pracoviskám laboratória
- vypracujte dokumentáciu k postupu konverzie stránok
- stručne porovnajte Vaše riešenie s inými možnými riešeniami využívajúcimi iné vývojové a komunikačné prostriedky

Jazyk, v ktorom sa práca vypracuje: slovenský
Termín pre odovzdanie práce: 04.05.2020
Dátum zadania diplomovej práce: 31.10.2019



12.

prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry.

Košice, 04. mája 2020

.....

vlastnoručný podpis

PodĎakovanie

Ďakujem môjmu vedúcemu diplomovej práce prof. Ing. Jánovi Šaligovi, PhD. za odborné rady a trpezlivosť pri tvorbe práce.

Obsah

Zoznam obrázkov	1
Zoznam symbolov a skratiek	4
Úvod	5
1. Teoretický úvod.....	6
1.1. LabVIEW a grafické programovanie	6
1.1.1. LabVIEW NXG	8
1.1.2. LabVIEW NXG Web Module	9
1.2. NI SystemLink	10
1.2.1. Dátová komunikácia	11
1.2.2. API	11
1.3. Počítačová sieť.....	11
1.3.1. Adresovanie.....	15
1.3.2. IP Adresa.....	15
1.4. API a REST architektúra	15
1.5. HTTP Protokol.....	16
1.5.1. HTTP Správy.....	17
1.5.2. Požiadavka.....	17
1.5.3. HTTP Metódy.....	18
1.5.4. URI	18
1.5.5. Odpoveď.....	19
1.6. Webová stránka	20
1.7. HTML	21
1.7.1. Základná štruktúra HTML dokumentu	23
1.8. CSS.....	24
1.8.1. Umiestnenie CSS	24
1.8.2. Štruktúra jazyka CSS	25
1.9. JavaScript.....	27

1.9.1.	Umiestnenie JS kódu	28
1.9.2.	Príklad kódu JavaScript programu.....	28
1.9.3.	DOM	29
1.9.4.	AJAX.....	30
1.9.5.	JSON	30
1.10.	Databázový systém a SQL.....	31
1.10.1.	SRDB	32
1.10.2.	Relačná databáza	33
1.10.3.	SQL.....	33
2.	Analýza súčasného laboratória merania	35
2.1.	Vzdialený panel	37
2.2.	Analýza vzdialeného panela	39
3.	Návrh	43
3.1.	Úvod	43
3.2.	Návrh riešenia	44
3.2.1.	Webové rozhranie	44
3.2.2.	Webová služba	45
3.2.3.	Databáza.....	46
3.2.4.	SystemLink server.....	46
3.2.5.	Pracovisko	46
4.	Realizácia navrhnutého riešenia	47
4.1.	Úvod	47
4.2.	Webové Rozhranie	47
4.2.1.	Opis funkcií	50
4.2.2.	Štruktúra hlavného priečinka	51
4.3.	Webová služba	52
4.3.1.	Opis funkcií	53
4.4.	Databáza.....	57

4.4.1.	Tabuľka - sessions.....	57
4.4.2.	Tabuľka - applications	57
4.4.3.	Tabuľka - app_{appid}_tags.....	58
4.4.4.	Tabuľka - control_requests	58
4.4.5.	Tabuľka - users	58
4.4.6.	Tabuľka - user_data.....	59
4.4.7.	Tabuľka - reservations.....	59
4.5.	SystemLink server.....	59
4.6.	Pracovisko	60
4.6.1.	Štruktúra priečinka vzdialeného pracoviska	65
5.	Iné možné riešenia	66
5.1.	Komunikácia	66
5.2.	Webová služba	66
	Záver.....	67
	Zoznam použitej literatúry	68
	Prílohy	70
	Príloha B:	71
6.	Úvod	72
7.	Inštalácia a spustenie	73
8.	Konfigurácia webovej služby	79
8.1.	Záložka – Server	79
8.1.1.	Záložka - Rozhranie	81
8.1.2.	Záložka - Ostatné.....	82
8.1.3.	Záložka – SMTP.....	84
9.	Aplikačné rozhranie webovej služby	85
9.1.	Relácia	85
9.2.	Pracovisko	85
9.3.	Používateľ.....	88

9.4.	Rezervácie	90
9.5.	Zoznam chybových správ	93
9.5.1.	Všeobecné	93
9.5.2.	Kontrola.....	93
9.5.3.	Rezervacia	93
9.5.4.	Prihlasovanie	94
9.5.5.	Pracovisko	94
9.5.6.	Relácie	94

Zoznam obrázkov

Obr. 1 Predný panel.....	6
Obr. 2 Bloková schéma.....	7
Obr. 3 Predný panel v LabVIEW NXG	8
Obr. 4 Bloková schéma v LabVIEW NXG.....	9
Obr. 5 Webové rozhranie vytvorené pomocou web modulu	10
Obr. 6 OSI model a TCP/IP model.....	13
Obr. 7 Klient - server	14
Obr. 8 Peer-to-peer	15
Obr. 9 HTTP správy	17
Obr. 10 HTTP požiadavka	17
Obr. 11 Syntax URI.....	18
Obr. 12 HTTP odpoveď	19
Obr. 13 Dynamické generovanie obsahu na strane servera	21
Obr. 14 Dynamické generovanie obsahu na strane klienta	21
Obr. 15 HTML element.....	22
Obr. 16 Štruktúra HTML dokumentu.....	23
Obr. 17 Pravidlo CSS	25
Obr. 18 DOM štruktúra.....	29
Obr. 19 Formát JSON	31
Obr. 20 Databázový systém.....	32
Obr. 21 Webová stránka laboratória merania	35
Obr. 22 Stránka pracoviska	36
Obr. 23 Front panel na webe	37
Obr. 24 Možnosti strany servera	38
Obr. 25 Správca pripojení vzdialeného panela.....	38
Obr. 26 Možnosť limitovania kontroly	38
Obr. 27 Pred uplynutím	39
Obr. 28 Po uplynutí	39
Obr. 29 Front panel na webe	39
Obr. 30 Zdrojový kód stránky	40
Obr. 31 Zobrazenie komunikácie v programe Wireshark	40
Obr. 32 Zaslanie požiadavky o zmenu protokolu	40
Obr. 33 Odpoveď na požiadavku.....	41

Obr. 34 Zaslania mena pouzivatela pocitaca	41
Obr. 35 Odpoved servera	41
Obr. 36 Zaslania nazvu VI subuoru	41
Obr. 37 Prenos VI subuoru	42
Obr. 38 Prenesená hodnota panela	42
Obr. 39 Možnosti nastavenia v NI Web Server	44
Obr. 40 Navrhnuté riešenie	44
Obr. 41 Návrh rozloženia stránky	45
Obr. 42 Webová stránka laboratória merania	47
Obr. 43 Navigačná lišta	48
Obr. 44 Vrchná časť stránky a aplikačná lišta	48
Obr. 45 Okno prihlásenia	49
Obr. 46 Okno zmeny hesla	49
Obr. 47 Okno úpravy profilu	49
Obr. 48 Okno rezervácií	50
Obr. 49 Okno pridania rezervácie	50
Obr. 50 Podprogram pre podstránku a pracovisko	50
Obr. 51 Hlavné okno aplikácie	52
Obr. 52 Okno správy pracovísk a rezervácií	52
Obr. 53 Okno správy pouzivatelov a nastavení	53
Obr. 54 Náhľad do hlavného programu main.gvi	53
Obr. 55 Náhľad do podprogramu handleClient.gvi	54
Obr. 56 Náhľad do podprogramu handleHttpRequest.gvi	54
Obr. 57 Tabuľky databázy	57
Obr. 58 Konfiguračné okno SystemLink servera	60
Obr. 59 Predný panel pracoviska Multimeter v LabVIEW	60
Obr. 60 Vzdialený panel pracoviska Multimeter	61
Obr. 61 Otváranie tagov na pracovisku	61
Obr. 62 Otvorenie spojenia	62
Obr. 63 Zapisovanie hodnôt do tagov	62
Obr. 64 Čítanie hodnôt tagov vzdialeného panela v pracovisku Multimeter	62
Obr. 65 Správa pracovísk	63
Obr. 66 Blokový diagram vzdialeného panela	63
Obr. 67 Časť blokového diagramu	64
Obr. 68 Posielanie hodnôt ovládacích prvkov	64

Obr. 69 Pohľad na prenášané hodnoty panela	64
Obr. 70 Okno ODBC.....	73
Obr. 71 Pridať nový zdroj	74
Obr. 72 Nastavenie údajov.....	74
Obr. 73 Nastavenie connection pooling.....	75
Obr. 74 Okno UDL súboru	75
Obr. 75 Nastavenie hesla	76
Obr. 76 Záložka HTTPS.....	77
Obr. 77 Záložka Remote	77
Obr. 75 Okno nastavenia – záložka Server.....	79
Obr. 76 Okno nastavenia – záložka Rozhranie	81
Obr. 77 Okno nastavenia – záložka Ostatné	82
Obr. 78 Okno nastavenia – záložka SMTP	84

Zoznam symbolov a skratiek

HTTP	Hypertext Transfer Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
NI	National Instruments
SQL	Structured Query Language
JS	JavaScript
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
API	Application programming interface
SAPI	Server Application Programming Interface
REST	Representational state transfer
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
VI	Virtual Instruments
DAQ	Data Acquisition
GPIB	General Purpose Interface Bus

Úvod

V dnešnej dobe je prístup na internet neodmysliteľnou súčasťou bežného života. Laboratórium nemusí predstavovať fyzické miesto, v ktorom študenti môžu získať prvé praktické skúsenosti s meracími prístrojmi a overiť si naučenú teóriu. Vybavenie skutočného laboratória je nákladné a riziko poškodenia prístrojov predstavuje ďalšie náklady na opravu či výmenu prístrojov. Aj to je jeden z dôvodov prečo začali vznikať virtuálne pracoviská, ktoré umožňujú vykonávať vzdialené meranie v reálnom čase na skutočných zariadeniach pomocou webových technológií.

E-LAB je projekt zameraný na tvorbu virtuálnych a vzdialených pracovísk, ktoré umožňujú demonštrovať vybrané základné senzory, metódy snímania a spracovania neelektrických veličín. Pomocou webového prehliadača sa tak študenti môžu aj mimo vyučovania oboznámiť so štandardnými prístrojmi a ich použitím pri elektronických meraniach. Nevýhodou však je to, že na prístup k pracovisku je potrebné mať nainštalovaný zásuvný modul, ktorý v súčasnosti nie je podporovaný webovými prehliadačmi. Práca sa zaoberá riešením aj tohto problému.

V prvej časti je teoretický úvod do technológií použitých v tejto práci. Ďalej sa nachádza analýza súčasného laboratória merania. Táto časť opisuje rozloženie, štruktúru a stav v akom sa nachádza súčasná webová stránka laboratória merania a obmedzenia súčasného riešenia vzdialeného panela pracoviska.

Ďalej sa nachádza návrh nového riešenia, ktoré opisuje návrh webového rozhrania a webovej služby, ktorá realizuje komunikácie klienta so vzdialeným panelom pracoviska pomocou HTTP protokolu a platformy SytemLink. Ďalej sa nachádza realizácia navrhnutého riešenia. Táto časť prezentuje nové webové rozhranie laboratória merania so správou klientova popisuje vybrané funkcie riešenia.

1. Teoretický úvod

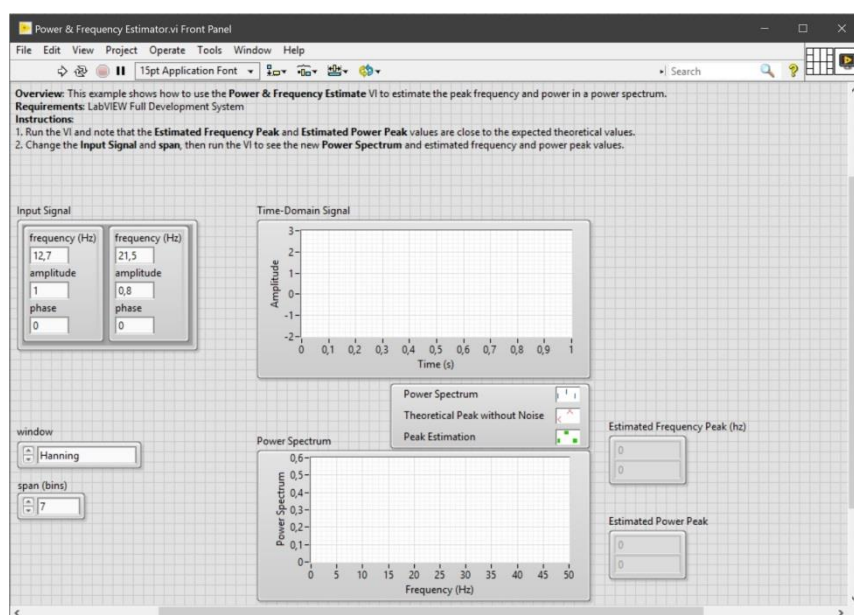
1.1. LabVIEW a grafické programovanie

LabVIEW (skratka pre Laboratory Virtual Instrument Engineering Workbench) je platforma pre návrh systémov a vývojové prostredie pre vizuálny programovací jazyk od firmy National Instruments, založené na grafickom programovaní. Zmyslom tohto grafického programovania je jednoduché, rýchle a efektívne programovanie. V tomto ohľade sa líši od tradičných programovacích jazykov ako C, C++ alebo Java, v ktorých sa programuje textom.

Programovací jazyk nesie názov G (Graphical language). G jazyk využíva na beh programu tok dát. Spájajú sa v ňom uzly (ikony) pomocou vodičov (čiar), cez ktoré sa prenášajú (tečú) údaje. Vykonávanie je určené štruktúrou grafického blokového diagramu, na ktorý programátor pripája rôzne funkčné uzly kreslením čiar. Tieto čiary predstavujú premenné a každý uzol sa vykoná hneď ako sú prístupné vstupné dáta. Toto platí aj ak je pripojených viac uzlov a umožňuje to paralelné riadenie, ktoré je potom možné upraviť na viacprocesorové, či viacvláknové.

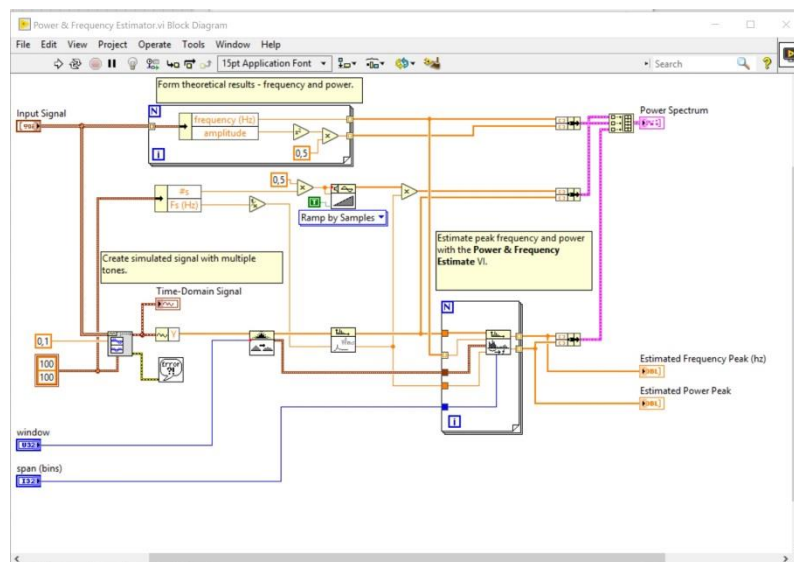
Programy LabVIEW sa nazývajú virtuálne nástroje alebo VI (Virtual Instruments), pretože ich vzhľad a fungovanie napodobňujú fyzické nástroje, ako sú osciloskopy a multimetre. [1]

Každý VI má tri komponenty: blokovú schému, predný panel (front panel) a panel konektorov (connector pane). Posledný sa používa na znázornenie VI v blokových schémach ostatných volajúcich VI. Predný panel je vyrobený pomocou ovládacích prvkov a indikátorov (Obr. 1).



Obr. 1 Predný panel

Ovládacie prvky sú vstupy - umožňujú užívateľovi poskytovať informácie do VI. Indikátory sú výstupy - indikujú alebo zobrazujú výsledky na základe vstupov poskytnutých VI. Zadný panel, ktorý je blokovou schémou (Obr. 2), obsahuje grafický zdrojový kód. Všetky objekty umiestnené na prednom paneli sa zobrazia na zadnom paneli ako svorky. Zadný panel tiež obsahuje štruktúry a funkcie, ktoré vykonávajú operácie s ovládacími prvkami a dodávajú údaje indikátorom. Štruktúry a funkcie sa nachádzajú na palette Funkcií a môžu byť umiestnené na zadnom paneli. Spoločne sa ovládacie prvky, indikátory, štruktúry a funkcie označujú ako uzly. Uzly sú navzájom spojené pomocou vodičov, napr. dva ovládacie prvky a indikátor môžu byť zapojené do prídavnej funkcie tak, že indikátor zobrazuje súčet oboch ovládacích prvkov. VI tak môže byť spustený buď ako program s predným panelom slúžiacim ako užívateľské rozhranie, alebo, ak je vložený ako uzol na blokovej schéme, predný panel definuje vstupy a výstupy pre uzol cez panel konektorov. To znamená, že každé VI sa dá ľahko testovať predtým, ako sa vloží ako podprogram do väčšieho programu.



Obr. 2 Bloková schéma

Grafický prístup tiež umožňuje neprogramátorom vytvárať programy pretiahnutím a vložením virtuálnych reprezentácií laboratórnych zariadení, s ktorými sú už oboznámení. Vývojové prostredie LabVIEW s priloženými príkladmi a dokumentáciou zjednodušuje vytváranie malých aplikácií. LabVIEW ponúka schopnosť vytvárať samostatné aplikácie. Ďalej je možné vytvárať distribuované aplikácie, ktoré komunikujú pomocou modelu klient-server, a preto sa dajú ľahšie implementovať z dôvodu prirodzene paralelnej povahy G.

Výhodou LabVIEW je aj možnosť prepojenia s inými programovacími jazykmi. Takisto má rozsiahle knižnice funkcií a podprogramov, ktoré pomôžu pri väčšine programovacích úloh s problémami,

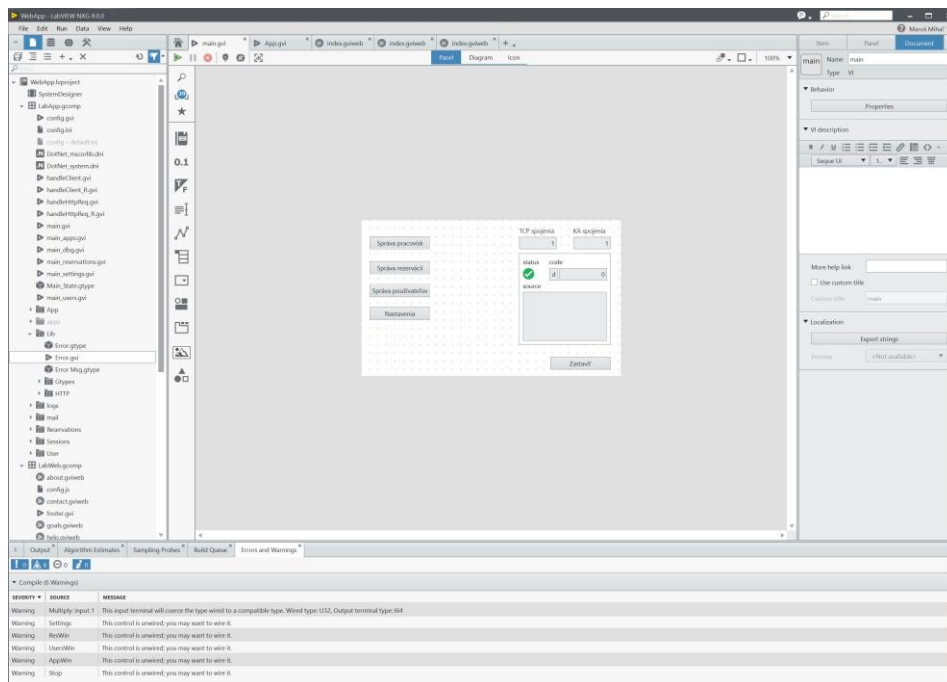
ktoré sa vyskytujú v konvenčných programovacích jazykoch. Obsahuje aj knižnice kódu pre zber dát (DAQ), zbernicu rozhrania na všeobecné použitie (GPIB) a sériového rozhrania, analýzu údajov, prezentáciu údajov, ukladanie údajov a komunikáciu cez internet. Široké využitie má v ovládaní meracích prístrojov alebo taktiež v priemyselnej automatizácii na vizualizáciu a riadenie technologických procesov. LabVIEW funguje na Windows, Mac OS X (Apple) a Linux platformách čím je vhodný pre väčšinu počítačových systémov.

1.1.1. LabVIEW NXG

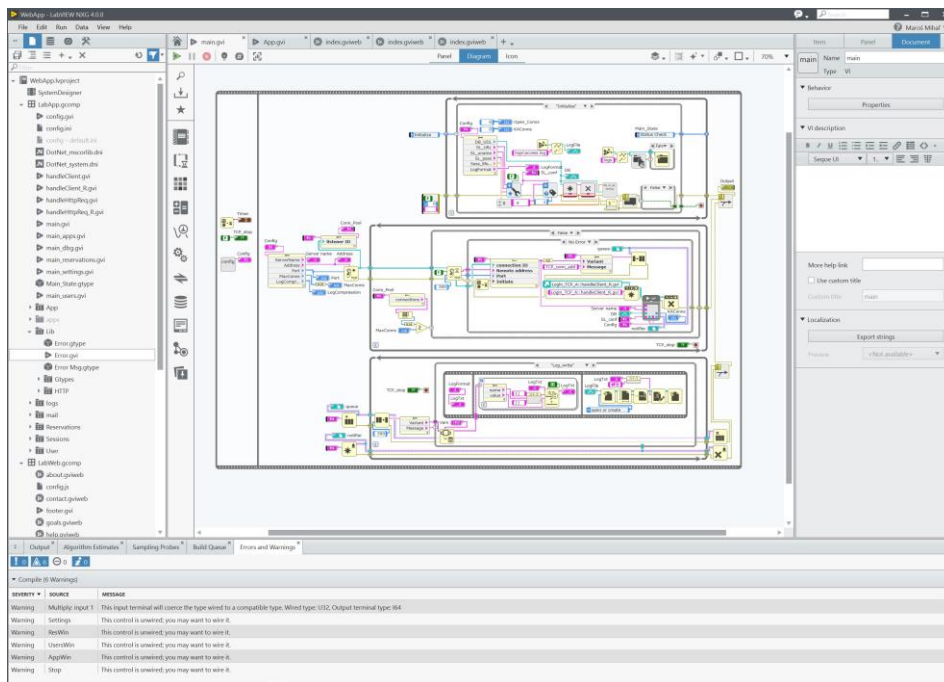
LabVIEW NXG (Next) je novou generáciou vývojového prostredia LabVIEWa obsahuje iba časť funkcií a hardvérovej podpory, ktorá je v súčasnosti k dispozícii v LabVIEW. Má nové grafické rozhranie a umožňuje konfigurovať, automatizovať a vizualizovať výsledky testov, a to všetko v rámci jedného programu. S automatickou identifikáciou nastavbového ovládača (instrument driver) pre mnoho prístrojov možno skrátiť čas na meranie. Výsledky je možné zobrazíť kdekoľvek vytvorením webových používateľských rozhraní, ktoré je možné zobrazíť v ľubovoľnom webovom prehliadači na PC, tablete alebo telefóne – bez inštalácie doplnkov. [2]

LabVIEW NXG je odporúčaný pre nasledujúce oblasti použitia:

- Meranie fyzikálnych systémov pomocou snímačov alebo akčných členov
- Schvaľovanie alebo overovanie elektronických návrhov
- Vývoj systémov testovania výroby



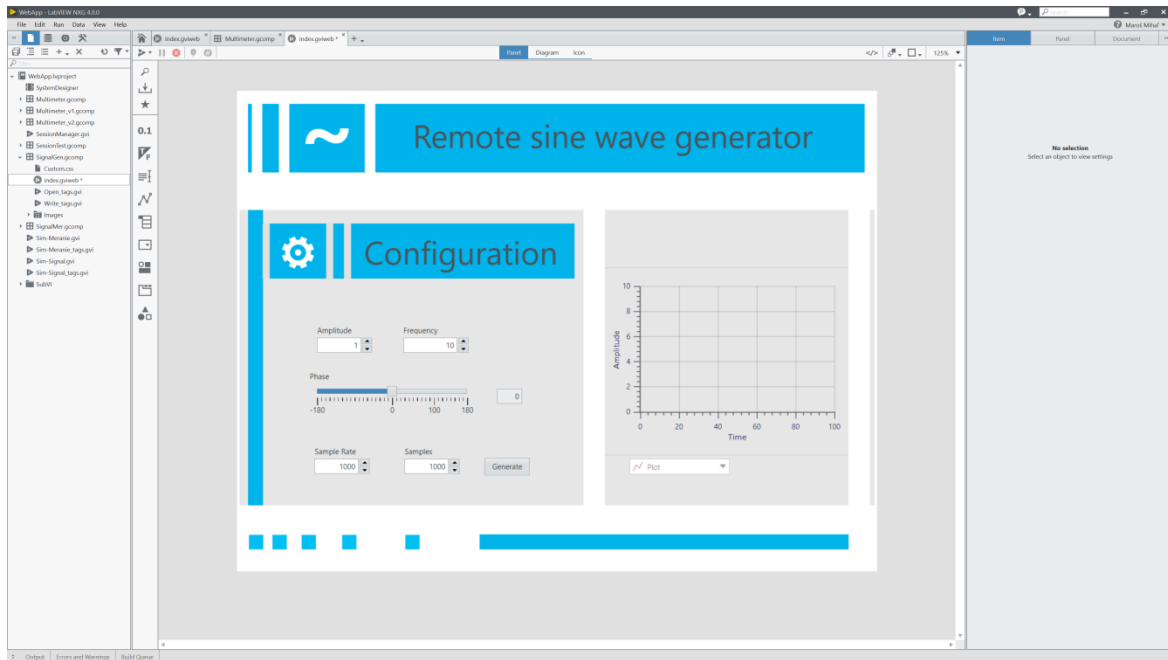
Obr. 3 Predný panel v LabVIEW NXG



Obr. 4 Bloková schéma v LabVIEW NXG

1.1.2. LabVIEW NXG Web Module

LabVIEW NXG Web Module je softvérový doplnok vývojového prostredia LabVIEW NXG, ktorý umožňuje navrhovanie, vytváranie a nasadzovanie webových používateľských rozhraní pre aplikácie, bez toho, aby ste potrebovali zručnosti pre vývoj webových aplikácií. Rozhrania fungujú v akomkoľvek modernom webovom prehliadači bez doplnkov alebo inštalácie programov na základe štandardných technológií, ako sú HTML, CSS a JavaScript. Súčasťou web modulu je aj NI Web Server.



Obr. 5 Webové rozhranie vytvorené pomocou web modulu

1.2. NI SystemLink

SystemLink od National Instruments je platforma, ktorá poskytuje nasadenie softvéru, konfiguráciu systémov, monitorovanie testov a nástroje na správu dát pre testovacie a meracie systémy. SystemLink je navrhnutý pre úzku spoluprácu s produktmi NI, ako sú LabVIEW, TestStand, hardvér a obsahuje nástroje na začlenenie širokého spektra softvérových a hardvérových technológií tretích strán. [3]

SystemLink kombinuje poprednú softvérovú infraštruktúru so softvérovými službami a aplikáciami navrhnutými pre NI, ktoré sa špecificky prispôbujú prípadom použitia testov a meraní. Pri testovacích a meracích aktivitách sa vyžadujú flexibilné nástroje na prácu s veľkým objemom dát, ako aj schopnosť vytvárať užívateľsky definované aplikačné rozhrania na prácu s týmito dátami. SystemLink poskytuje túto flexibilitu a nasledujúce vlastnosti:

- Otvorenosť a rozšíriteľnosť
- Škálovateľnosť a výkonnosť
- Modularita
- Zabezpečenia

Súčasťou platformy je NI Web Server. Pomocou webového servera NI možno konfigurovať nastavenia servera, ako sú napríklad roly a oprávnenia užívateľa, autentifikácia užívateľa a bezpečnosť.

1.2.1. Dátová komunikácia

Architektúra dát SystemLink je navrhnutá na efektívnu a bezpečnú komunikáciu dát medzi pripojenými uzlami. Pomocou tejto dátovej architektúry poskytuje SystemLink dátové služby, ktoré sú prístupné cez API - patria medzi ne File Service a Tag Service.

- Služba súborov (File Service) - pomocou súborovej služby možno zaznamenávať a komunikovať dáta o súboroch z aplikácií, ktoré zapisujú namerané dáta alebo výsledky testov do súborov.
- Služba Tagov (Tag Service) - pomocou služby tagov možno uložiť hodnoty dátového bodu, ktorý čítate a nahlasujete pomocou aplikácie bežiacей v uzle
- Message API – pomocou rozhrania API správ možno publikovať akékoľvek dáta reťazcov pre definované témy (topics). Následne možno použiť ktorýkoľvek uzol v ľubovoľnom riadenom systéme na prijímanie správ prihlásením sa k definovaným témam

1.2.2. API

SystemLink poskytuje aplikačné programovacie rozhrania (API), ktoré umožňujú programový prístup k funkciám servera a klienta. Konkrétne rozhrania sú uvedené nižšie:

- API webových služieb (Web Service API) - funguje prostredníctvom rozhraní API webových služieb, ktoré vyhovuje architektúre REST
- LabVIEW API - dátové služby SystemLink ako napr. súbory a tagy, sú dostupné ako rozhrania LabVIEW API
- Python API

1.3. Počítačová sieť

Počítačová sieť je komplex technických prostriedkov, počítačov a ich softvérových zdrojov, ktoré sú vhodne prepojené a zabezpečujú vzájomnú komunikáciu a prenos dát. Pomocou stanovených pravidiel umožňuje používateľom navzájom komunikovať.

Uzlom siete sa rozumie zariadenie, ktoré je vybavené sieťovým rozhraním - časťou zariadenia, ktorá umožňuje pripojenie do siete (napr. sieťová karta), a je zapojené do počítačovej siete. Prenosovým médium (cestou) sa prenášajú dáta. Táto cesta je realizovaná elektrickým alebo optickým vodičom, no taktiež pomocou elektromagnetických vln.

Architektúra siete definuje spôsob, akým sú jednotlivé prvky siete spájané do funkčného celku a určuje základné princípy prenosu dát medzi uzlami siete.

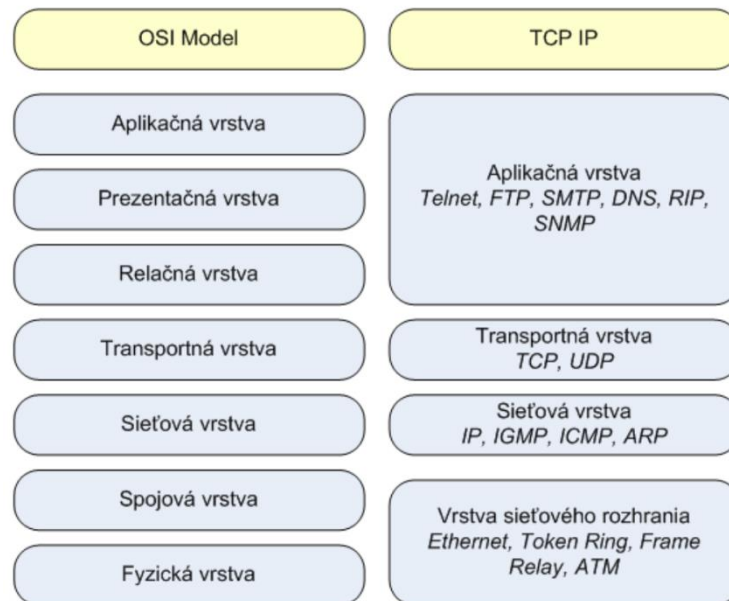
Dátová komunikácia realizovaná medzi sieťovými uzlami vyžaduje vykonanie určitého postupu činností ako nadviazanie spojenia, rozloženie prenášaných informácií do blokov (datagramov), ich prenos a zabezpečenie pred chybami.

Sieťovú architektúru tvorí:

- topológia siete - fyzické usporiadanie prepojení medzi uzlami siete (mapa siete)
- prístupová metóda - pravidlá, podľa ktorých môže uzol pristúpiť na prenosové médium
- komunikačný protokol - predpis, ktorý určuje, podľa akých pravidiel majú byť dáta pri prenose spracované a ako majú byť kódované. Komunikačných protokolov existuje obrovské množstvo a problematika protokolov a ich štruktúry je veľmi široká.

Typické architektúry počítačových sietí

- SNA (System Network Architecture) - architektúra, ktorá slúžila na prepojenie terminálov IBM s centrálnymi počítačmi triedy Mainframe cez komunikačné radiče
- OSI (Open System Interconnection - Reference Model) - referenčný model OSI/ISO, je abstraktný, na vrstvách založený opis návrhu štruktúry komunikačných a počítačových sieťových protokolov, vyvinutý ako súčasť iniciatívy otvorených systémov
- TCP/IP (Transport Control Protocol / Internet Protocol) - je súbor komunikačných protokolov, na ktorých je postavený systém Internet



Obr. 6 OSI model a TCP/IP model

Počítačové siete delíme podľa rozľahlosti na:

- PAN (Personal Area Network) – osobná sieť, ktorá spája mobilné zariadenia
- LAN (Local Area Network) – lokálna sieť, ktoré spájajú jednotlivé počítače v rámci malého územia (napr. dom, skupina budov)
- MAN (Metropolitan Area Network) – metropolitná sieť, ktorá pokrýva väčšiu lokalitu (napr. mesto)
- WAN (Wide Area Network) – rozľahlá sieť, ktorá pokrýva rozľahlé územie (napr. krajina, kontinent)

Počítačové siete delíme podľa vzťahov medzi počítačmi na:

1. Klient - server

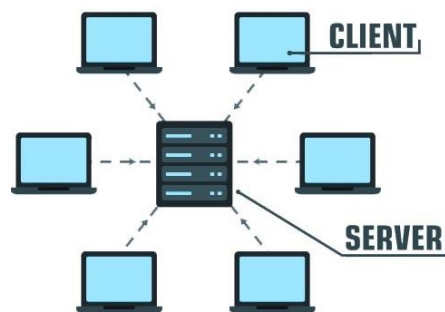
Server poskytuje svoje služby a technické zariadenia počítačom zapojeným do siete. Existuje viacero typov serverov, ktoré možno rozdeliť podľa poskytovaných služieb, čo znamená, že na jednom fyzickom počítači môže existovať viacero serverov. Podľa poskytovaných služieb možno rozdeliť servery na [4]:

- súborové servery (file server), ktoré poskytujú diskový priestor pre prácu so súbormi
- tlačové servery (print server), ktoré organizujú a riadia tlačové požiadavky
- komunikačné servery (napr. poštové - mail server)

- webové servery (web server)
- databázové servery (database server)

Klient je ten subjekt, ktorý služby servera prijíma, najčastejšie aplikáciou na počítači (napr. webový prehliadač), ktorá pracuje ako rozhranie so serverom. Komunikácia so serverom je riadená protokolom (napr. HTTP protokol), ktorý je špecifický pre konkrétnu službu. Zvyčajne komunikáciu vyvoláva klient, server čaká na klientske požiadavky.

Výhodou tohto typu siete je jednoduchšia správa sieťových dát, jednoduchšie zaistenie bezpečnosti dát, ako i celej siete. Nevýhodou je to, že akákoľvek vážna chyba na serveri naruší činnosť celej siete.

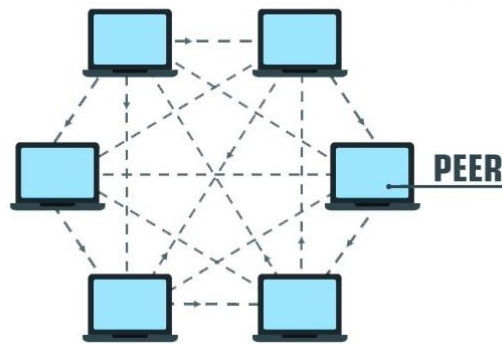


Obr. 7 Klient - server

2. Peer-to-peer

Peer-to-peer (rovný s rovným) - všetky počítače zapojené do siete sú rovnocenné -nemajú centrálnu správu, alebo nadradený počítač. Každý počítač sa spravuje sám a môže poskytnúť svoje služby alebo technické zariadenia ostatným počítačom v sieti, ktoré ich môžu využívať. Komunikácia je iniciovaná alebo prijímaná medzi sebou.

Výhodou tohto typu siete je, že je lacnejšia, pretože neobsahuje žiadny dedikovaný server. Ak jeden počítač prestane fungovať, ostatné počítače neprestanú fungovať. Je jednoduché ho nastaviť a udržiavať, keďže sa každý počítač riadi sám. Nevýhodou tohto typu je, že neobsahuje centralizovaný systém. Preto nemôže zálohovať údaje, pretože sa na rôznych miestach líšia.



Obr. 8 Peer-to-peer

1.3.1. Adresovanie

Aby počítače dokázali spolu komunikovať, musia vedieť, komu majú príslušné dáta poslať. Každý počítač má teda 48-bitovú, tzv. MAC (Media Access Control) adresu. Je to identifikačné číslo sieťového adaptéra slúžiace na jednoznačnú identifikáciu sieťového rozhrania. Každý sieťový adaptér (sieťová karta) má zaručenú jedinečnú MAC adresu. (hardvérová adresa sieťovej karty je pridelená výrobcom)

Vzhľadom na skutočnosť, že moderné sieťové zariadenia majú možnosť MAC adresu zmeniť, nie je zaručená jednoznačná identifikácia zariadenia v lokálnej počítačovej sieti LAN. Pri výskyte zariadenia s rovnakou MAC adresou v rovnakej lokálnej sieti nemusí byť komunikácia medzi niektorými zariadeniami plne funkčná.

1.3.2. IP Adresa

V počítačovej sieti je dôležitá ešte jedna adresa. Je to takzvaná IP adresa - logický číselný identifikátor fyzického sieťového rozhrania (sieťovej karty) daného uzla (najčastejšie počítača) v sieti, ktorý komunikuje s inými uzlami prostredníctvom IP protokolu (napríklad pre Internet). IP adresa musí byť jednoznačná, inak vznikajú konflikty a pripojenie do siete nie je funkčné.

Internet Protokol verzie 4 (IPv4) definuje IP adresu ako 32-bitové číslo, ktoré je rozdelené na štyri 8-bitové čísla (číslo v rozsahu 0-255) zapísané v desiatkovej sústave oddelené bodkou, napr. 192.168.0.1. Avšak z dôvodu rastu internetu a vyčerpania dostupných IPv4 adries, bola štandardizovaná nová verzia IP (IPv6), používajúca 128 bitov pre IP adresu, napr. 20:abc:0:1234:0:567:8:90

1.4. API a REST architektúra

API (Application programming interface) je počítačové rozhranie, ktoré definuje interakcie medzi viacerými sprostredkovateľmi softvéru. Definuje druhy volaní alebo požiadaviek ktoré sa môžu

uskutočniť, ako ich vyrobiť, formáty údajov ktoré by sa mali použiť, konvencie ktoré sa majú dodržiavať. Môže tiež poskytovať mechanizmy rozširovania, aby používatelia mohli rôznymi spôsobmi rozširovať existujúcu funkčnosť a v rôznej miere. Rozhranie API môže byť úplne vlastné, špecifické pre komponent alebo môže byť navrhnuté na základe priemyselného štandardu na zabezpečenie interoperability. Pretože ostatné komponenty sa spoliehajú iba na API, systém, ktorý poskytuje API, môže (v ideálnom prípade) zmeniť svoje interné podrobnosti za rozhraním API bez ovplyvnenia jeho používateľov.

Web API je aplikačné programovacie rozhranie pre webový server alebo webový prehliadač. Ide o koncepciu vývoja webu, ktorá je zvyčajne obmedzená na klientskej strane webovej aplikácie (vrátane akýchkoľvek používaných webových frameworkov), a preto zvyčajne nezahŕňa podrobnosti o implementácii webového servera alebo prehliadača, ako sú SAPI alebo API, pokiaľ nie sú verejne prístupné vzdialenej webovej aplikácii.

REST je softvérový architektonický štýl, ktorý definuje súbor obmedzení, ktoré sa majú použiť na vytváranie webových služieb. Webové služby, ktoré zodpovedajú architektonickému štýlu REST, nazývanému RESTful Web services, poskytujú interoperabilitu medzi počítačovými systémami na internete. RESTful webové služby umožňujú žiadajúcim systémom prístup a manipuláciu s textovými zázorneniami webových zdrojov pomocou jednotnej a preddefinovanej sady operácií bez stavov. Pomocou bezstavových protokolov a štandardných operácií sa systémy RESTful zameriavajú na rýchly výkon, spoľahlivosť a schopnosť rásť opätovným použitím komponentov, ktoré je možné spravovať a aktualizovať bez ovplyvnenia systému ako celku, aj keď je spustený.[5]

1.5. HTTP Protokol

Hypertextový prenosový protokol (Hypertext transfer protocol, HTTP) je hlavným prostriedkom dátovej komunikácie na World Wide Web-e. Protokol je na aplikačnej vrstve a definuje tvar prenášaných informácií, možnosti a náležitosti, požiadavky a odpovede medzi klientom a serverom. HTTP protokol je bezstavový, nerozpoznáva klientov od ktorých chodia požiadavky.

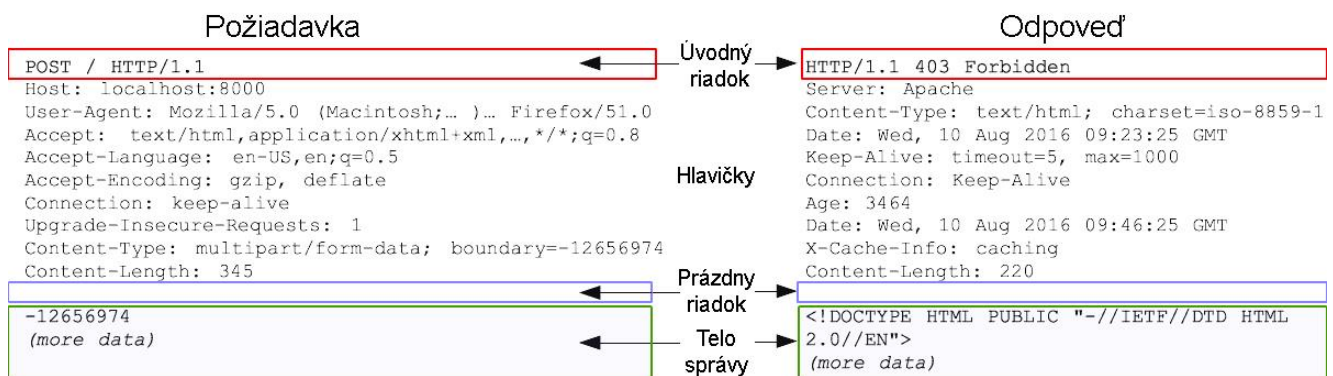
HTTP protokol je nezávislý od typu prenášaných dát (napr. súbor, text, obrázok), preto je dôležité v hlavičke špecifikovať formát obsahu podľa MIME (Multipurpose Internet Mail Extensions) štandardu, aby klient alebo server vedel danú požiadavku spracovať. Vo verzii protokolu HTTP/1.0 server po odpovedaní na požiadavku spojenie uzavrie, no vo verzii HTTP/1.1 vytvára tzv. perzistentné spojenie a preto sa spojenie hneď neuzavrie a čaká určitú dobu na ďalšie požiadavky. Klient tak môže pokračovať v požiadavkách a spojenie ukončiť sám. Čas odozvy servera sa takto zníži, keďže nie je stále potrebné otvárať nové TCP spojenie. [7]

HTTP klientom (označovaný ako user agent) môže byť akákoľvek aplikácia schopná nadviazať HTTP spojenie, vyžiadať si informácie a následne ich prijať. Väčšinou sú to webové prehliadače (Web Browser, ako napr. Chrome, Mozilla Firefox, Opera, atd.).

1.5.1. HTTP Správy

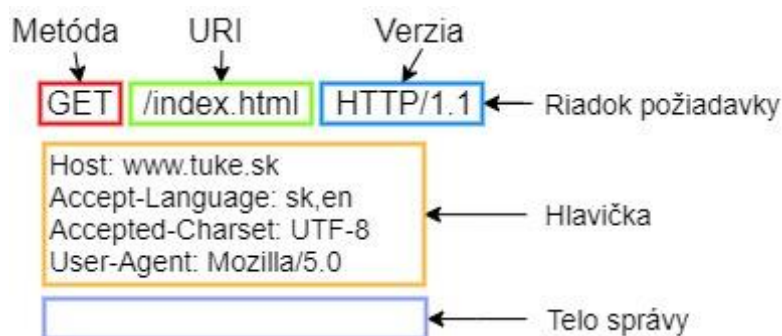
HTTP protokol používa dva typy správ (Obr. 9), požiadavky (HTTP request) od klienta a odpovede (HTTP response) od servera, ktoré majú podobnú štruktúru:

1. Úvodný riadok (start-line).
2. Hlavička (Header), ktorá obsahuje dodatočné informácie pre klienta alebo server
3. Prázdny riadok
4. Nepovinné telo správy (Body)



Obr. 9 HTTP správy

1.5.2. Požiadavka



Obr. 10 HTTP požiadavka

Úvodný riadok požiadavky (Obr.10) pozostáva z:

1. Metóda požiadavky, špecifikuje postup akým sa s informáciami má zaobchádzať
2. Jednotný identifikátor prostriedku (URI)
3. Verzia HTTP protokolu

1.5.3. HTTP Metódy

- GET - najbežnejší typ žiadosti, slúži na vyžiadanie a získanie špecifikovaného zdroja
- HEAD - podobné metóde GET, okrem toho, že sa nepožaduje telo správy, iba hlavičky (získavanie meta-informácií)
- POST– slúži na poslanie dát na server pre vytvorenie alebo aktualizovanie špecifikovaného zdroja
- PUT– podobne ako POST, pričom viacnásobné poslanie rovnakej žiadosti bude mať rovnaký výsledok, pri POST by bol však iný (napr. zakaždým vytvorenie nového objektu)
- PATCH – slúži na čiastočnú úpravu existujúceho špecifikovaného zdroja na základe inštrukcii v odoslanej požiadavke
- DELETE –žiada o zmazanie špecifikovaného zdroja
- TRACE -odošle kópiu obdržanej požiadavky späť klientovi,slúži na testovanie a diagnostiku
- OPTIONS–opisuje komunikačné možnosti (napr. HTTP metódy), ktoré daný server podporuje pre špecifikovaný zdroj
- CONNECT–slúži na vytvorenie TCP tunelu medzi klientom a serverom, používa sa, ak je klient za proxy serverom

1.5.4. URI

Jednotný identifikátor prostriedku alebo zdroja (angl. Uniform Resource Identifier, skratka URI) je jednoducho formátovaný reťazec znakov používaný na identifikáciu alebo pomenovanie zdroja. URI v HTTP môže byť reprezentovaný v absolútnom tvare alebo relatívnom k nejakej známej základnej URI. Všeobecný syntax URI (Obr. 11) [6] [8]:

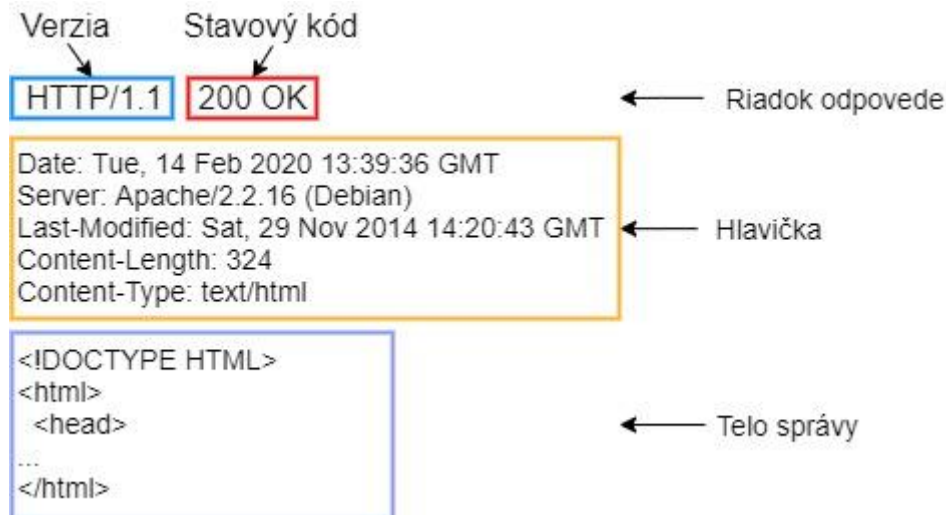
schéma : `[/ [používateľ:heslo@] host'[:port]] [/] cesta [?dopyt] [#fragment]`

Obr. 11 Syntax URI

- schéma identifikuje schému prístupu napr. Internetový protokol (http, ftp) alebo prístup k lokálnemu súboru (file)
- používateľ:heslo – autentizačná časť, ktorá sa používa pri prístupe k zabezpečeným zdrojom
- host' – hosťovské meno alebo IP adresa počítača na ktorom je zdroj umiestnený
- port – port na ktorom beží služba servera, ktorý daný zdroj poskytuje
- cesta – cesta k zdroju, napríklad cesta v súborovom systéme

- dopyt – reťazec dopytu, ktorý sa používa pre dynamický prístup ku zdrojom (napr. krajina=Slovensko&mesto=Presov)
- fragment – identifikuje časť dokumentu, v HTML dokumente napr. sekciu #odsek1

1.5.5. Odpoveď



Obr. 12 HTTP odpoveď

Úvodný riadok odpovede pozostáva z (Obr.12):

1. Verzia HTTP protokolu
2. Stavový kód
3. Textová fráza

Stavový kód je trojciferný identifikátor a upresňuje ako bola odpoveď serverom spracovaná, či bola požiadavka vybavená kladne, záporne alebo došlo k chybe. Prvé číslo stavového kódu určuje triedu odpovede a zvyšné čísla ju bližšie špecifikujú. Kategórie stavových kódov:

- 1XX – Informačné – znamená, že požiadavka bola prijatá a proces pokračuje
- 2XX – Úspešné – znamená, že požiadavka bola úspešne prijatá, spracovaná a akceptovaná (napr. 200 OK)
- 3XX – Presmerovanie – znamená, že potrebné vykonať ďalšiu akciu na dokončenie požiadavky
- 4XX – Chyba klienta – znamená, že požiadavka obsahuje nesprávny syntax alebo nemôže byť splnená (napr. 404 Not Found)

- 5XX – Chyba servera – znamená, že server zaznamenal chybu, alebo nie je schopný spracovať platnú požiadavku

1.6. Webová stránka

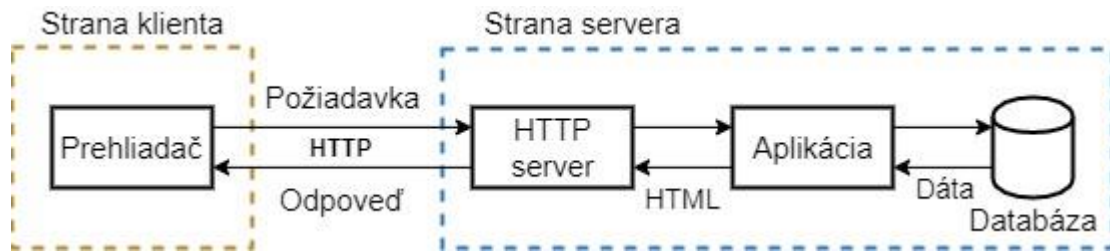
World Wide Web (WWW alebo web) je informačný priestor, v ktorom sú objekty záujmu (zdroje), identifikované globálnymi identifikátormi nazývanými jednotnými identifikátormi zdroja (Uniform Resource Identifiers, URI). Informačný priestor webu tvoria dokumenty rôznych typov, ktoré môžu byť prepojené cez hypertextové odkazy. Webová stránka je dokument, uložený obyčajne na webovom serveri, ktorý je možné klientovi zobrazíť pomocou webového prehliadača. Komunikácia so serverom prebieha pomocou HTTP protokolu. Stránka je obyčajne tvorená textom v značkovom jazyku HTML a môže obsahovať odkazy, obrázky, videá a iné multimediálne prvky. [9]

Webové stránky je možné rozdeliť na:

- Statické – jednoduché stránky, tvorené len ako dokumenty HTML a CSS, ktoré webový server odošle priamo zo súborového systému servera bez modifikovania. Zmena ich obsahu je možná len priamou editáciou dokumentov. Obsah je rovnaký pre všetkých používateľov.
- Dynamické – zložitejšie stránky, ktorých obsah je aktuálne vytvorený na základe požiadaviek zvlášť pre každého používateľa. Dynamické webové stránky sa zvyknú nazývať aj webové aplikácie ak je ich rozhranie funkčne podobné počítačovým aplikáciám. Na serveri sú dáta často uložené v databáze, nie priamo v HTML dokumente, preto pre ich zobrazenie je potrebné dynamicky vygenerovať HTML obsah, ktorý sa zobrazí v prehliadači.

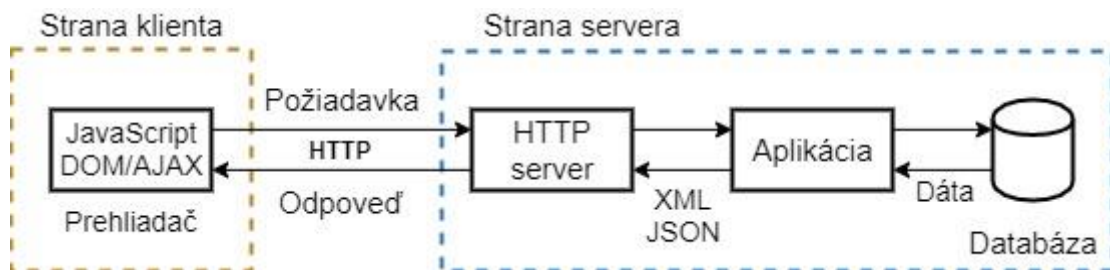
Dynamické stránky možno podľa miesta tvorby obsahu rozdeliť na:

- Dynamické generovanie obsahu na strane servera (Obr. 13) – HTML obsah sa vygeneruje priamo na strane servera, ktorý ju odošle klientovi a ten sa zobrazí v prehliadači. Generovanie obsahu je často zabezpečené pomocou skriptov vytvorených v rôznych jazykoch (napr. PHP, ASP, Perl, Python)



Obr. 13 Dynamické generovanie obsahu na strane servera

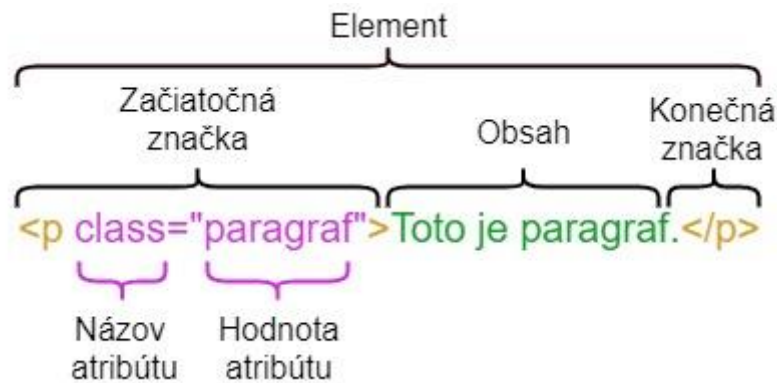
- Dynamické generovanie obsahu na strane klienta (Obr. 14) – na strane servera sú dáta zakódované do vhodného formátu na prenos dát (napr. XML alebo JSON), ktoré sa následne odošlú klientovi. Prehliadač následne pomocou technológií AJAX - JavaScriptu a rozhrania DOM spracuje údaje a dynamicky vygeneruje a zobrazí HTML obsah.



Obr. 14 Dynamické generovanie obsahu na strane klienta

1.7. HTML

Hypertextový značkový jazyk (HyperText Markup Language; HTML), je značkový jazyk určený pre dokumenty prehliadateľné vo webovom prehliadači, no používa sa aj v elektronickej pošte. HTML poskytuje možnosti na vytvorenie štruktúrovaných dokumentov, kde kladie dôraz skôr na obsah (text, hypertextové odkazy, tabuľky, formuláre, skripty, multimediálny a iný obsah) a nie na vzhľad. [10]



Obr. 15 HTML element

HTML elementy (Obr. 15) (prvky) sú stavebnými blokmi HTML stránok, tie sú vymedzené značkami (tzv.tagy), ktoré sú uzavreté v hranatých zátvorkách (<značka>). Značky môžu byť:

- Párové, tie sa skladajú zo začiatočnej značky, obsahu a konečnej značky, ktorá má predponu „/“
- Nepárové (bez obsahu) – reprezentujú tzv. prázdny (void) element (napríklad ,
)

Značky môžu byť do seba vnorované, avšak nesmú sa križovať (<i>kurzívátučné</i>)

Existujú prípady, kedy názov elementu nemôže opísať potrebnú vlastnosť. Pomocou atribútov je možné pridať ďalšie informácie o elemente, ktoré sú umiestnené v začiatočnej značke a skladajú sa z názvu atribútu a hodnoty oddelenej znakom „=“ (Obr. 14). Hodnota atribútu musí byť uzavretá do jednoduchých alebo dvojitéch úvodzoviek. Element môže obsahovať viac rôznych atribútov. Nemôže však obsahovať rovnaký atribút viackrát.

Globálne atribúty, ktoré môžu byť priradené každej značke:

- id – priraďuje elementu jedinečný identifikátor, ktorý umožňuje sa na element odkazovať
- style – priraďuje elementu formátovacie nastavenia kaskádového štýlu (CSS)
- class – zoskupuje elementy do skupín, tzv. tried, všetkým elementom triedy potom môže byť naraz priradené formátovanie CSS, jeden element môže byť zaradený do viacerých tried napr <div class="odsek uvod">
- title – titulok, resp. doplňujúce informácie o elemente
- hidden – ak je atribút uvedený, element sa na stránke nezobrazí, je možné ho zapísať aj skráteno bez hodnoty, napr. <div hidden>, používa sa pri dynamických stránkach

Niektoré znaky sú v HTML vyhradené, preto existuje definovaná sada špeciálnych znakov, ktoré sa označujú aj ako entity. Entity začínajú znakom „&“, nasleduje slovné alebo číselné označenie

entity a sú zakončené znakom „;“. V prípade číselného označenia je na začiatku znak „#“. (&názov; alebo &#cislo;). Príklady špeciálnych znakov:

- & - znak „&“
- < - znak „<“
- > - znak „>“
- - nedeliteľná medzera

1.7.1. Základná Štruktúra HTML dokumentu



Obr. 16 Štruktúra HTML dokumentu

HTML dokument je textový súbor a mal by začínať deklaráciou typu dokumentu (Obr. 16), ktorá určuje, v akej verzii je HTML dokument zapísaný. Obsah celej stránky je v elemente html a ten obsahuje elementy head a body (hlavičku a telo), tieto elementy sa musia vyskytovať iba raz. Hlavička môže obsahovať rôzne metadáta, ktoré sa v okne prehliadača nezobrazujú, napr. element title, ktorý definuje názov dokumentu. Telo obsahuje samotný obsah dokumentu, ktorý nám prehliadač zobrazí. Jeho štruktúru určuje použitie rôznych ďalších elementov.

Elementy sa z pohľadu zobrazovania v dokumente delia na dva základné typy.

- riadkové (inline) - ide o elementy, ktoré obvykle niečo zdôrazňujú či vyznačujú a ovplyvňujú len obsah elementu, nemajú žiadny vplyv na umiestnenie elementu v dokumente a ich výška a šírka je rovná ich obsahu
- blokové (block) – ide o elementy, ktoré ovplyvňujú formátovací model dokumentu, vždy začínajú na novom riadku a zaberajú celú dostupnú šírku

Základnými elementami pre členenie textu sú napr.

- **div** – základný blok textu (je odsadený na novom riadku)
- **span** – vymedzenie textu v bloku
- **br** – zalomenie riadku
- **p** – odstavec
- **h1 - h6** – nadpisy rôznych úrovní

Základnými elementami pre zvýraznenie textu sú napr. *b* pre tučné písmo, *i* pre kurzíva, *sup* pre horný index a *sub* pre dolný index. Elementy, ktoré členia stránku podľa obsahu:

- **<main>** – hlavný obsah dokumentu,
- **<article>** – časť HTML dokumentu, ktorá tvorí samostatne prezentovateľný celok
- **<section>** – časť (kapitola) dokumentu,
- **<header>** – hlavička dokumentu alebo časti
- **<footer>** – päta dokumentu alebo časti
- **<aside>** – vedľajší obsah dokumentu, nesúvisiaci s jeho hlavnou náplňou
- **<nav>** – navigačný panel dokumentu (mal by obsahovať odkazy na hlavné časti stránky)

1.8. CSS

Kaskádové štýly (angl. Cascading Style Sheets, CSS), je jazyk, ktorý slúži na vizuálne formátovanie dokumentov. Umožňuje oddeliť obsah dokumentu (HTML) od jeho vzhľadu (CSS), čím sa získa prehľadnosť a jednoduchosť dokumentov. [11]

1.8.1. Umiestnenie CSS

Existujú tri možné spôsoby priradenia kaskádových štýlov k dokumentu HTML:

- Priama definícia štýlu pre jeden element v hodnote atribútu style

```
<p style="color: red; font-weight: bold;">Text</p>
```

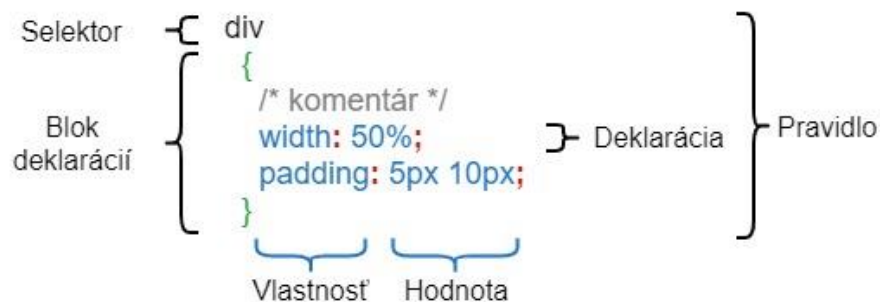
- Vloženie štýlu do elementu style v hlavičke HTML - má väčšiu prioritu ako externý štýl

```
<head>  
<style>  
p { color: red; padding: 5px 5px 10px 10px;}  
</style>  
</head>
```

- Použitie externého štýlu, ktorý sa nachádza v samostatnom súbore .css, pridaním odkazu cez element link v hlavičke HTML. Štýl je tak možné aplikovať na viacero dokumentov (stránok).

```
<head>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
```

1.8.2. Štruktúra jazyka CSS



Obr. 17 Pravidlo CSS

Každé pravidlo (Obr. 17) má dve časti a to selektor, alebo množinu selektorov a jemu prislúchajúci blok deklarácií.

Selektor určuje, na ktorý element bude deklarácia aplikovaná, je možné zvoliť viac elementov, ktoré sa následne oddelia čiarkou.

Deklarácia sa skladá z vlastnosti a hodnoty, ktorá je oddelená dvojbodkou. Deklarácie sa spájajú do skupín (blok), ktoré sú uzavreté do zložených zátvoriek. Jednotlivé deklarácie sú oddelené bodkočiarkou.

Jednotlivé vlastnosti umožňujú upravovať formát elementov (rozmery, pozadie, farba, typ písma a pod.), polohovanie, obtekanie (position, float) a mnoho ďalších.

Druhy selektorov:

- Univerzálny selektor – označuje sa znakom „*“ a vyberá všetky elementy
- Selektor typu – vyberá elementy rovnakého typu

p { text-align: left; } – všetky elementy **<p>**

- Selektor id - vyberá element, na základe hodnoty atribútu id. Má tvar „**#hodnota**“.

#demo { border: red 2px solid; } – element s id=“demo“

p#demo { border: blue 4px solid; } – element <p> s id=“demo“

- Selektor triedy – vyberá elementy, podľa hodnoty atribútu class. Má tvar „**hodnota**“. Element môže byť zaradený vo viacerých triedach, jednotlivé triedy musia byť v atribúte class oddelené medzerami.

.center {text-align: center; } – všetky elementy s atribútom class=“center“

.center.red{ color: red; } - všetky elementy s atribútom class=“center red“

<p class=“center red”>Odsek</p> - element <p> zaradený v dvoch triedach

- Selektor atribútu – vyberá element, podľa výskytu zvoleného atribútu alebo podľa jeho hodnoty

a[title] { color: purple; } – všetky elementy <a> s atribútom **title**

input[type=“text”] { color: green; } – všetky elementy <input> s atribútom **type=“text“**

- Selektor pseudo-triedy - vyberá element, ktorý je v určitom stave. Začína znakom „:“. Pseudo-triedy sú napr.: visited (navštívený odkaz), :hover (element, nad ktorým je práve kurzor).
- Selektor pseudo-elementu – umožňuje formátovať špecifickú časť elementu. Začína znakom „::“. Pseudo-elementy sú napr. ::first-letter (prvé písmeno textu) alebo ::before a ::after, ktoré umožňujú pridať ďalší obsah pred obsahom elementu alebo za obsah elementu.

Selektory je možné kombinovať, na to slúžia určené operátory:

- Zlučovanie selektorov do jedného pravidla – zabezpečené pomocou operátora čiarka (,)

h1, h2, p { text-align: left; } – všetky elementy <p>, <h1> a <h2>

- Operátor „medzera“

div p { } – vyberie všetky elementy <p>, ktoré sú vnorené v elemente <div> v ľubovoľnej hĺbke

- Operátor „>“

div > ul{ } – vyberie všetky elementy , ktoré sú priamo vnorené v elemente <div>

- Operátor „+“

h1 + p { } - vyberie všetky elementy <p>, ktoré priamo nasledujú za elementom <h1> a sú v rovnakej hĺbke

- Operátor „~“

div ~ p{} – vyberie všetky elementy **<p>** ktoré nasledujú za elementom **<div>** a sú v rovnakej hĺbke

Niektoré vlastnosti deklarované pre element môžu byť dedičné, čo znamená že sa automaticky prenesú na všetky elementy, ktoré sú v ňom vnorené. Kaskáda je jednou zo základných funkcií štýlov, ktorý riadi konečný výsledok v situácii, keď sa na rovnaký element uplatňujú odlišné pravidlá. Rozhodovanie o uplatnení vlastnosti je závislé od:

- dôležitosti pravidla (transition – prechody, !important alebo animation - animácie)
- od toho kde bolo pravidlo definované
 - Prehliadač – každý prehliadač má štandardnú sadu štýlov pre rôzne dokumenty
 - Stránka – štýly vytvorené autorom stránky
 - Používateľ – má možnosť prepísať štýly stránky v prehliadači vlastnými štýlmi
- Špecifickosť (váha) - priama definícia štýlu v atribúte style (v riadku - inline) má najväčšiu váhu, ďalej záleží od druhu selektora (zoraďené od najvyššej po najnižšiu)
 - Selektor id
 - Selektor triedy, pseudo-triedy a atribútu
 - Selektor typu a pseudo-elementu
 - Univerzálny selektor a operátory nemajú žiadnu váhu
- Ak je špecifickosť rovnaká, záleží od poradia v akom sú pravidlá v dokumente umiestnené - vyberie sa to, ktoré je najbližšie umiestnené pri zvolenom elemente

1.9. JavaScript

JavaScript (JS) je skriptovací programovací jazyk s podporou objektového programovania, ktorý umožňuje tvorbu dynamických webových stránok. Je štandardizovaný ako jazyk ECMAScript organizáciou ECMA International. JavaScript sa spúšťa na strane klienta, najčastejšie vo webovom prehliadači. JS je intepretovaný jazyk, avšak moderné webové prehliadače obsahujú JavaScript Engine – podprogram, ktorý kód zdrojového programu prekladá do strojového kódu tesne pred vykonaním kódu (just-in-time compilation), čím dochádza k zlepšeniu jeho výkonnosti. [12]

Premenné v JavaScripte priamo neprislúchajú dátovému typu – do jednej premennej možno priradiť napr. číslo a neskôr reťazec. JS je funkcionálny, čo umožňuje s funkciou pracovať ako s ľubovoľnou hodnotou, alebo ju napr. možno predať ako parameter inej funkcii. JS je prototypový, umožňuje definovať objekty, ktoré majú dátové vlastnosti a metódy – funkcie pracujúce nad vlastnosťami. Syntax je podobný jazyku C a Java.

Pre JS existuje množstvo knižníc a frameworkov, ktoré uľahčujú tvorbu webových stránok. JS sa často používa aj na strane servera (Node.js), ale aj v iných aplikáciách mimo webu.

1.9.1. Umiestnenie JS kódu

Existujú dve možné spôsoby pridania JavaScript kódu do HTML dokumentu:

- Vloženie kódu do elementu script v hlavičke alebo tele HTML

```
<head>
<script>
//komentár
function HelloWorld(){
    alert("Ahoj svet!");
}
</script>
</head>
```

- Použitie externého kódu, ktorý sa nachádza v samostatnom súbore .js, pridaním odkazu do atribútu src v elemente script v hlavičke alebo tele HTML. JS kód je tak možné jednoduchšie vložiť na viacero dokumentov (stránok). Externé kódy sa môžu odkazovať na súbor umiestnený na inej stránke alebo pomocou cesty relatívnej k aktuálnej stránke.

```
<script src="/js/externalScript.js" ></script>
<script src="http://ww.google.sk/js/externalScript.js"></script>
```

1.9.2. Príklad kódu JavaScript programu

Nasledujúci program ukazuje základné vlastnosti JavaScriptu – deklaráciu globálnej aj lokálnej premennej, deklaráciu a volanie funkcie s argumentmi a návratovou hodnotou, výrazy, priradenie, cyklus, komentár a výpis správy na obrazovku.

```
var fname = "John";
var lname = "Doe";
function displayFullName(first_name, last_name) {
    var full_name;
    full_name = first_name + last_name;
    for( var i=0; i<10; i++ ) { // vypíš 10 krát
        alert(full_name);
    }
    return full_name;
}
displayFullName(fname,lname);
```

1.9.3. DOM

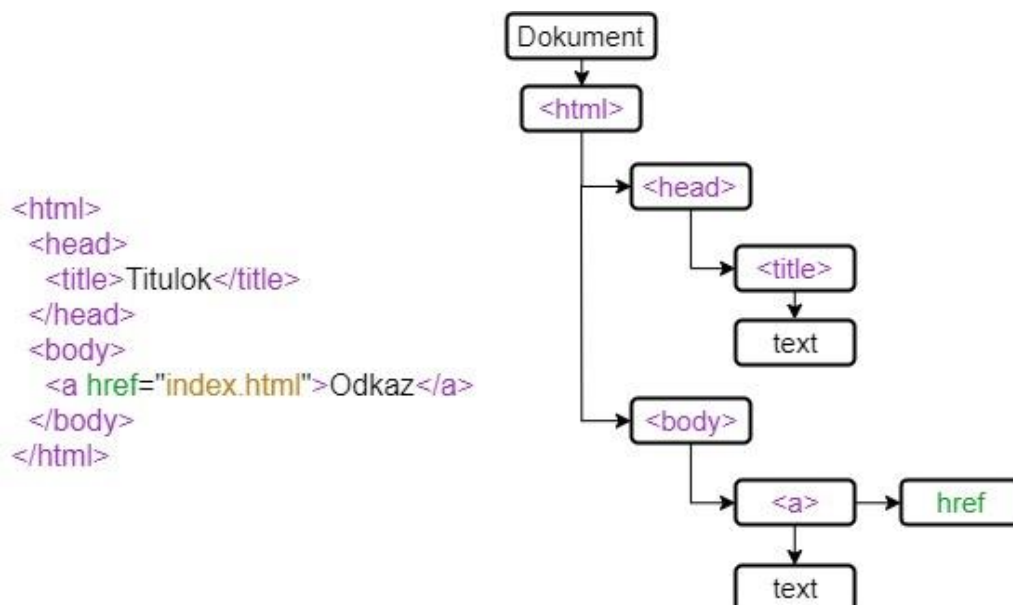
Objektový model dokumentu (DOM – Document Object Model) je štandard definovaný konzorciom W3C (World Wide Web Consortium). Je to rozhranie nezávislé od platformy a jazyka umožňujúce programom a skriptom dynamicky pristupovať a upravovať obsah, štruktúru a štýl dokumentov. DOM reprezentuje dokumenty HTML a XML vo forme stromovej štruktúry (Obr. 18) a umožňuje pristupovať k objektom, ktoré reprezentujú jednotlivé elementy dokumentu (v prípade HTML napríklad formuláre, hyperlinkové odkazy, text, tabuľky či samotný dokument). Pomocou JavaScript-u je tak možné pristupovať k DOM a manipulovať s vlastnosťami a funkciami elementov dokumentu. [13]

```

<html>
<body>
  <p id="demo"></p>
  <script>
    document.getElementById("demo").innerHTML = "Hello World!";
  </script>
</body>
</html>

```

Príklad – zmení obsah elementu <p> s atribútom id="demo".



Obr. 18 DOM štruktúra

1.9.4. AJAX

Asynchrónny JavaScript a XML (Asynchronous JavaScript and XML) je názov pre niekoľko technológií, ktoré umožňujú tvorbu dynamických webových stránok. AJAX kombinuje: [14]

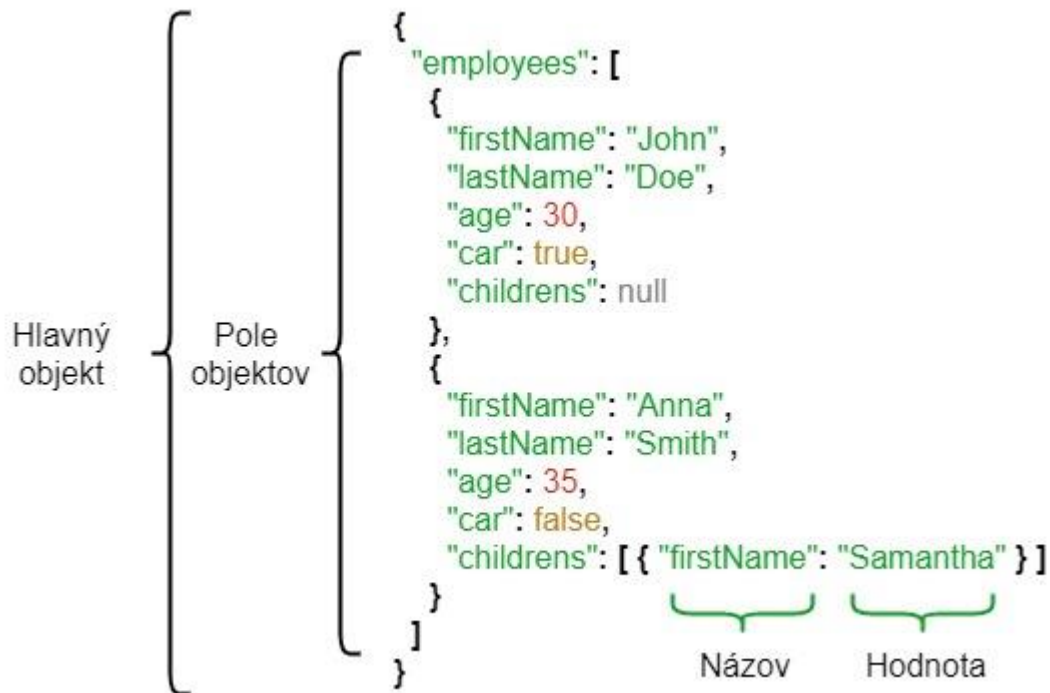
- HTML a CSS pre zobrazenie a formátovanie informácií
- Dynamické zobrazenie a interakciu s prezentovanou informáciou pomocou DOM
- JSON alebo XML pre výmenu dát
- XMLHttpRequest (XHR) objekt pre asynchrónnu komunikáciu so serverom na pozadí
- JavaScript, ktorý všetko spája

AJAX dokáže urýchliť chod webových stránok, keďže nie je potrebné načítať celý obsah stránky po každej akcii používateľa. Dodáva potrebnú interaktivitu – napríklad vytvorením vysúvacích navigačných ponúk, transformáciou textu a obrázkov, dynamickým pridaním elementov do stránky či kontrolou vstupných polí formulárov. JavaScript však negarantuje správnosť a bezpečnosť odosielaných dát, keďže kód sa spúšťa na strane klienta, ktorý ho môže upraviť a zmeniť výsledok než sa odošle na server. Preto je potrebná kontrola vstupov aj na strane servera.

1.9.5. JSON

JavaScript objektový zápis (JavaScript Object Notation, JSON) je jednoduchý, odľahčený textový formát zápisu dát nezávislý na počítačovej platforme, ktorý vychádza priamo z JavaScript-u. Často slúži na prenos dát pomocou AJAX-u. Oficiálny MIME typ pre JSON je "application/json".

JSON je založený na dvoch univerzálnych štruktúrach - objekt a pole (Obr. 19). Jednotlivé štruktúry sú oddelené čiarkou a môžu byť organizované v poliach alebo zahrnuté v objektoch. Zložitosť hierarchie nie je teoreticky nijak obmedzená. [15]



Obr. 19 Formát JSON

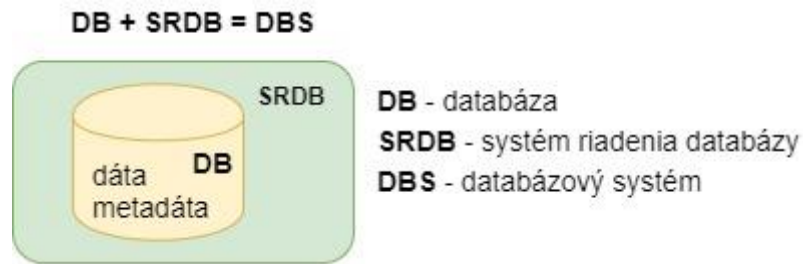
Hodnotou sa rozumie ľubovoľný dátový typ (číslo, reťazec, boolean, null, objekt alebo pole). Iné dátové typy sa musia previesť na niektorý z JSON typov.

Objekt je neusporiadaná množina párov názov/hodnota. Za každým názvom nasleduje dvojbodka a jednotlivé páry sú oddelené čiarkou. Názov musí byť reťazec (ohraničený v úvodzovkách). Ohraničujú ho zložené zátvorky „{“ „}“.

Pole je zoznam zoradených hodnôt, ktoré sú oddelené čiarkou. Ohraničujú ho hranaté zátvorky „[“ a „]“.

1.10. Databázový systém a SQL

Databázový systém je systém zložený z databázy a zo systému riadenia databázy (SRDB) (Obr. 20). Používatelia databázy komunikujú so SRDB prostredníctvom databázových rozhraní a databázových aplikácií (aplikačných programov). [17]



Obr. 20 Databázový systém

Databáza je množina štruktúrovaných dát uložených v počítačovom systéme, ktoré spolu súvisia a tvorí jadro celého databázového systému. Databázový model (dátový model) určuje, akým spôsobom ukladá databázový systém objekty do databázy a vzájomne ich prepája. Najčastejšie je používaný relačný model.

1.10.1. SRDB

Systém riadenia databázy (SRDB) pozostáva zo softvéru, ktorý organizuje ukladanie údajov. Riadi vytváranie, údržbu a používanie štruktúr databázových úložísk organizácií a ich koncových užívateľov. Vo veľkých systémoch umožňuje SRDB používateľom a inému softvéru štruktúrované ukladanie a získavanie údajov. SRDB sa zvyčajne kategorizujú podľa databázového modelu, ktorý podporujú, ako napríklad sieťový, relačný alebo objektový model. Model má tendenciu určovať jazyky dotazov, ktoré sú k dispozícii na prístup do databázy. Jedným z bežne používaných dopytovacích jazykov pre relačnú databázu je SQL, hoci syntax a funkcia SQL sa môžu medzi jednotlivými databázami SRDB líšiť. Veľká časť interného inžinierstva SRDB je nezávislá od údajového modelu a zaoberá sa faktormi riadenia ako je výkon, súbežnosť, integrita a obnova po zlyhaní hardvéru. [16]

Hlavnými cieľmi systému riadenia databáz sú dostupnosť údajov ich integrita, bezpečnosť a nezávislosť. Dostupnosť údajov sa týka skutočnosti, že údaje sa sprístupňujú širokému spektru používateľov v zmysluplnom formáte. Integrita údajov sa týka správnosti údajov v databáze. Inými slovami, údaje dostupné v databáze sú spoľahlivé údaje. Bezpečnosť údajov sa týka skutočnosti, že k údajom majú prístup iba oprávnení používatelia. Zabezpečenie údajov je možné vynútiť pomocou hesiel. Ak dvaja oddelení užívatelia prístupujú k určitým údajom súčasne, SRDB im nesmie dovoliť vykonať protichodné zmeny. Nezávislosť údajov SRDB umožňuje užívateľovi efektívne ukladať, aktualizovať a získavať údaje. SRDB poskytuje „abstraktný pohľad“ na to, ako sú údaje uložené v databáze. Pre efektívne ukladanie informácií sa na reprezentáciu údajov používajú zložité dátové štruktúry. Systém skryje určité podrobnosti o tom, ako sa údaje ukladajú a udržiavajú.

1.10.2. Relačná databáza

Relačná databáza je súbor údajov s vopred definovanými vzťahmi medzi nimi. Tieto položky sú usporiadané ako skupina tabuliek so stĺpcami a riadkami. Tabuľky sa používajú na uchovávanie informácií o objektoch, ktoré majú byť zastúpené v databáze. Každý stĺpec v tabuľke obsahuje určitý druh údajov a pole uchováva skutočnú hodnotu atribútu. Riadky v tabuľke predstavujú kolekciu súvisiacich hodnôt jedného objektu alebo entity. Každý riadok v tabuľke môže byť označený jedinečným identifikátorom nazývaným primárny kľúč a riadky medzi viacerými tabuľkami môžu byť spojené pomocou cudzích kľúčov. K týmto údajom je možné pristupovať rôznymi spôsobmi bez reorganizácie samotných databázových tabuliek.

1.10.3. SQL

Štruktúrovaný dopytovací jazyk (Structured Query Language, SQL), je štandardný jazyk určený na komunikáciu s relačnými systémami riadenia databáz (RDBMS). SQL jazyk je neprocedurálny, čiže opisuje údaje, ktoré chceme získať a nie ako ich chceme získať. [18]

Príkazy SQL jazyka možno rozdeliť do podskupín (jazykov):

- Data Definition Language (DDL) - jazyk na definíciu dát, ktorý umožňuje definovať štruktúry databázy, vytvára a upravuje objekty ako napr. tabuľky, pohľady a funkcie
 - CREATE – vytvorí nový databázový objekt (napr. tabuľku alebo databázu)
 - DROP – vymaže databázový objekt
 - ALTER – upraví databázový objekt (napr. pridanie stĺpca do existujúcej tabuľky)
- Data Manipulation Language (DML) - jazyk na manipuláciu s dátami, ktorý obsahuje príkazy na výber a úpravu dát
 - SELECT - vyberie dáta z databázy
 - INSERT INTO- vloží nové dáta do tabuľky
 - UPDATE- aktualizuje dáta v tabuľke
 - DELETE - vymaže dáta z tabuľky
- Data Control Language (DCL) - jazyk na riadenie prístupu k dátam, ktorý zahŕňa príkazy ako napr.
 - GRANT – prideluje používateľom práva na vykonanie rôznych operácií nad dátami
 - REVOKE – odoberá práva

Štandard ANSI (American National Standards Institute)SQL definuje dátové typy pre databázové systémy, ale ich podpora sa môže v jednotlivých (SRDB) líšiť. Medzi najpoužívanejšie dátové typy v relačných DBMS patria:

- Reťazec znakov (napr. CHAR, VARCHAR, TEXT)
- Bitový reťazec (BIT)
- Číselné hodnoty (napr. INTEGER, FLOAT, REAL)
- Logická hodnota (BOOLEAN)
- Čas a dátum (napr. DATE, TIME, TIMESTAMP, INTERVAL)
- Multimediálne dáta (BLOB)

1.10.3.1. Príklad

```
CREATE TABLE employees
(
  id UNSIGNED INT NOT NULL PRIMARY KEY,
  first_name VARCHAR(40) NOT NULL,
  last_name VARCHAR(40) NOT NULL,
  city VARCHAR(50) NOT NULL,
  position VARCHAR(60) NOT NULL
)
```

Vytvorí tabuľku s názvom „empoyees“ a stĺpcami „ip“, „first_name“, „last_name“, „city“ a „position“. Stĺpec „id“ je primárnym kľúčom a jeho dátový typ je „unsigned integer“. Ostatné stĺpce sú dátového typu „varchar“.

```
SELECT first_name, last_name
FROM employees
WHERE (position='mechanic' AND city='London')
ORDER BY first_name,last_name;
```

Vyberie všetkých zamestnancov z tabuľky „employess“, kde je pozícia „mechanic“ a mesto „London“ a usporiada výsledky podľa stĺpcov „first_name“ meno a „last_name“ priezvisko. To znamená, že ak budú existovať zamestnanci s rovnakým menom, zoradí ich podľa mena.

2. Analýza súčasného laboratória merania

Prvou úlohou práce bolo oboznámiť sa s existujúcim laboratóriom merania s prístupom cez internet. E-LAB predstavuje realizáciu a praktický výstup projektu zameraného na tvorbu pracovísk, ktoré umožňujú demonštrovať vybrané základné senzory, metódy snímania a spracovania neelektrických veličín. Študenti sa tak môžu aj mimo vyučovania oboznámiť so štandardnými prístrojmi a ich použitím pri elektronických meraniach pomocou webového prehliadača. Jednou z výhod je aj zníženie rizika poškodenia meracích prístrojov študentmi s nedostatočnými skúsenosťami.

The screenshot shows the E-LAB website interface. At the top left, the text 'E-LAB' is displayed in large, bold letters. To the right, the university name 'TECHNICKÁ UNIVERZITA V KOŠICIACH' and faculty 'Fakulta elektrotechniky a informatiky' are listed, along with the department 'Katedra elektroniky a multimediálnych telekomunikácií'. A vertical navigation menu on the left contains links for 'Úvod', 'Ciele', 'Pracoviská', and 'Pomoc'. The main content area is titled 'Pracoviská:' and lists various measurement stations under different subjects. The 'Predmet Lekárska elektronika' section includes: 1. Pulzoximeter, 2. Elektrokardiografia - meranie, 3. Elektrokardiografia - generovanie, 4. Vektorová kardiografia, and 5. Meranie EMG. The 'Predmet Signálové a komunikačné rozhrania' section includes: 1. Senzory teploty, 2. Meranie koncentrácie plynov (CO₂), 3. Meranie uhlového natočenia a lineárneho posunu, 4. Meranie kvality vody (Senzor pH v kvapaline), and 5. Experimentálne meracie pracovisko s reálnym fotovoltaickým článkom. Below this is a sub-section 'Spracovanie signálov v meraní neelektrických veličín' with links for digitalization, FFT, Parseval's theorem, filter characteristics, and analog filters. The 'Predmet Základy elektroniky' section includes: 1. Differenčný a darlingtonov zosilňovač (with simulation and experimental links), 2. Stabilizovaný zdroj (with simulation and experimental links), and 3. Operačný zosilňovač (with simulation and experimental links).

Obr. 21 Webová stránka laboratória merania

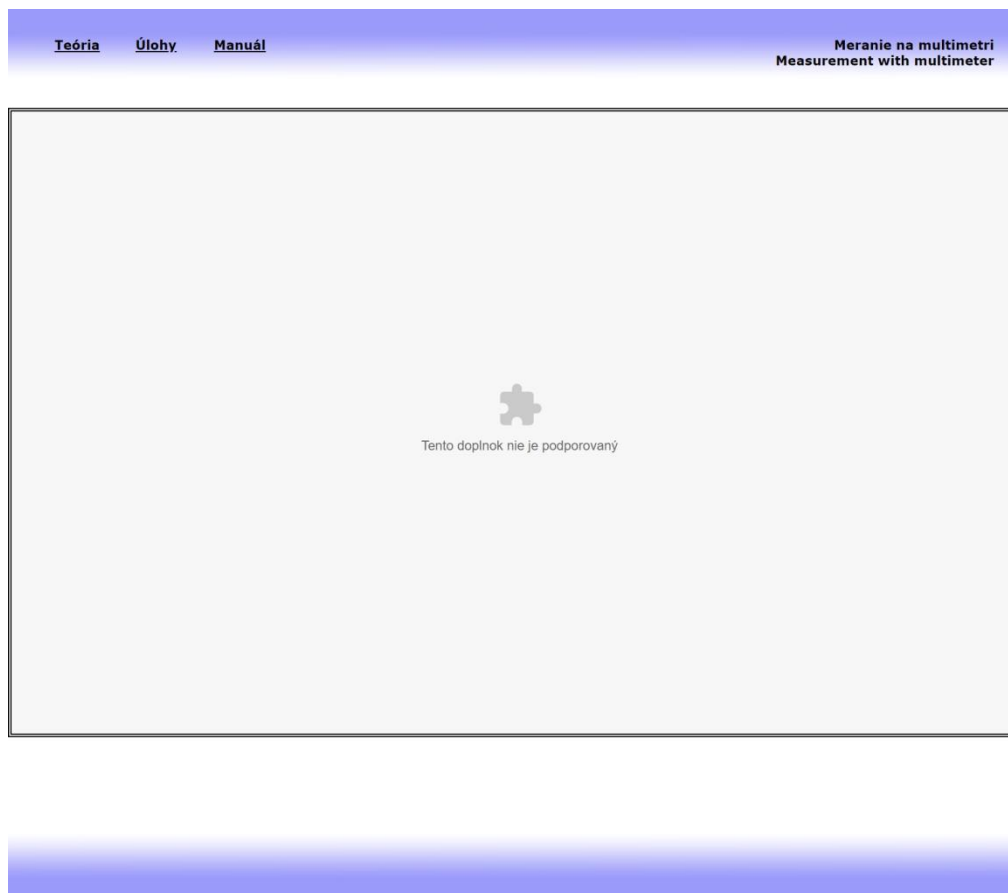
Webová stránka laboratória merania (Obr. 21) je jednoduchá statická stránka s kaskádovými štýlmi vloženými priamo v HTML dokumente. Skladá sa z troch častí:

- Vrchná časť - hlava – tvorí ju názov projektu, školy, fakulty a katedry
- Navigačné menu – tvoria ho odkazy na stránky s iným obsahom
- Časť s hlavným obsahom konkrétnej stránky

Stránka má pevne nastavenú šírku a výšku, čo naznačuje, že nie je prispôbena väčšiemu množstvu obsahu, ktorý môže prekročiť túto hranicu a ovplyvniť celkový vzhľad stránky (Obr. 21).

Stránka je usporiadaná nasledovne:

- Úvod – hlavná stránka, obsahuje opis projektu E-LAB
- Ciele – obsahuje ciele, ktoré si projekt stanovil
- Pracoviská – obsahuje odkazy na jednotlivé pracoviská, ktoré sú zoradené podľa predmetov
- Pomoc – obsahuje informácie o podporovaných prehliadačoch a postup pri inštalácii modulu (plug-in)



Obr. 22 Stránka pracoviska

Stránka pracoviska (Obr. 22) je taktiež jednoduchá a statická. Vo vrchnej časti stránky sa nachádzajú odkazy a popis činnosti, ktoré sa na danom pracovisku vykonávajú. Jednotlivé odkazy odkazujú na podstránky, ktoré dopĺňajú obsah pracoviska, a to konkrétne:

- Teóriu – obsah tvoria teoretické poznatky k danému meraniu
- Úlohy – obsah tvoria úlohy navrhnuté na precvičenie teórie pre konkrétne meranie
- Manuál – obsah tvorí opis pracoviska a možnosti jeho ovládania

V časti s obsahom sa nachádza objekt, ktorý odkazuje na zásuvný modul (plug-in) obsiahnutý v LabVIEW RunTime Engine. Pomocou tohto modulu je možné prísť a ovládať vzdialený panel konkrétneho pracoviska. Tento modul však už nie je v súčasnosti (v nových verziách) webovými prehliadačmi podporovaný (Obr. 22), čo znamená že sa jedná o veľkú nevýhodu. Spodná časť neobsahuje žiaden text. Nachádza sa tam rovnaký obrázok ako vo vrchnej časti, ktorý stránku vizuálne uzatvára.

2.1. Vzdialený panel

Vzdialený panel pracoviska je vytvorený v LabVIEW pomocou vstavaného nástroja Web Publishing Tool. Tento nástroj umožňuje veľmi jednoducho nastaviť vzdialené ovládanie akéhokoľvek programu vytvoreného v LabVIEW. Na to, aby bolo možné prísť na vzdialený panel, musí byť program spustený na počítači. Ak je otvorený v prostredí LabVIEW a beží, je možné ho spustiť. Po spustení je zobrazený len panel bez vrchných líšt.



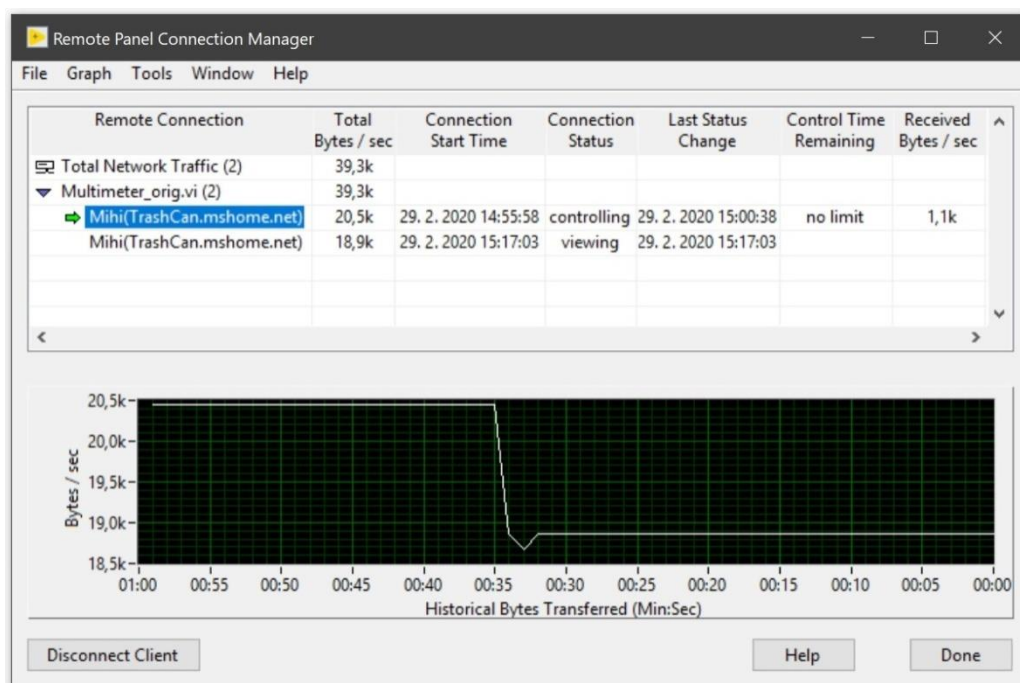
Obr. 23 Front panel na webe

Klient, ktorý chce vzdialený panel ovládať, musí poslať požiadavku o kontrolu (Obr. 23). Vzdialený panel môže ovládať len jeden klient, ostatní klienti môžu panel pozorovať. Na hlavnom počítači (serveri), kde je program spustený, je možné prevziať ovládanie od klienta, ktorý panel ovláda. Kontrolu nad panelom je taktiež možné zablokovať, pozorovanie to ale neovplyvní (Obr. 24).



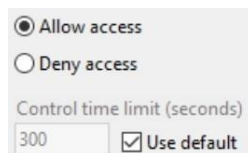
Obr. 24 Možnosti strany servera

V LabVIEW existuje nástroj - Správca pripojení vzdialeného panela (Remote Panel Connection Manager (Obr. 25), ktorý poskytuje prehľad o pripojených klientoch, ako napríklad - meno používateľa počítača a jeho adresu, čas pripojenia, stav, čas poslednej zmeny stavu a ostávajúci čas možnosti kontroly panela. Nachádza sa tu aj prehľad o množstve prenášaných dát a možnosť odpojiť klienta.



Obr. 25 Správca pripojení vzdialeného panela

V nastaveniach je možné zakázať celkový prístup k panelu a taktiež nastaviť maximálny čas kontroly panela (predvolená hodnota je 5 minút) (Obr. 26).



Obr. 26 Možnosť limitovania kontroly

Ak je pripojených viacero klientov na vzdialený panel, každý z nich môže poslať požiadavku na ovládanie panela (Obr. 27). Požiadavky sú podľa času kedy prišli zaradené do poradia a klientovi,

ktorý práve panel ovláda, sa vymedzí maximálny čas, po ktorého uplynutí sa kontrola odovzdá klientovi, ktorý je prvý v poradí (Obr. 28).

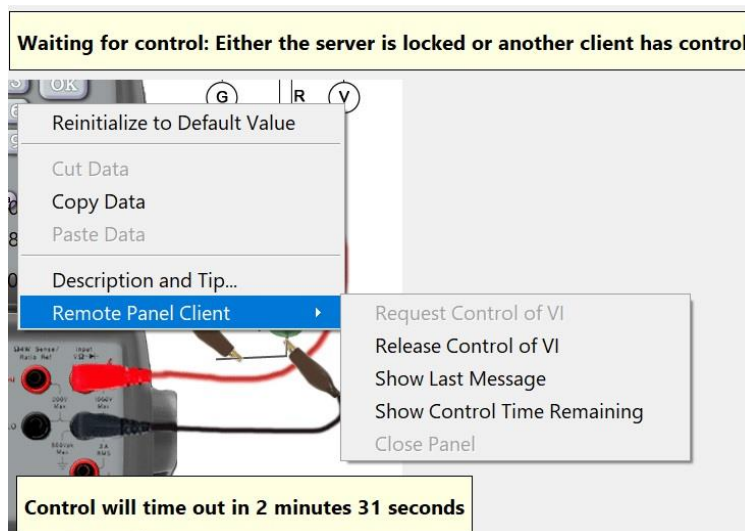
Remote Connection	Total Bytes / sec	Connection Start Time	Connection Status	Last Status Change	Control Rema	Received Bytes / sec
➔ Mihi(TrashCan.mshome.net)	20,5k	29. 2. 2020 14:55:58	controlling	29. 2. 2020 15:00:38	03:41	800,0
⌚ Mihi(TrashCan.mshome.net)	18,9k	29. 2. 2020 15:17:03	waiting for control	29. 2. 2020 15:18:27		

Obr. 27 Pred uplynutím

Remote Connection	Total Bytes / sec	Connection Start Time	Connection Status	Last Status Change	Control T Remaini	Received Bytes / sec
Mihi(TrashCan.mshome.net)	18,8k	29. 2. 2020 14:55:58	viewing	29. 2. 2020 15:23:05		
➔ Mihi(TrashCan.mshome.net)	18,8k	29. 2. 2020 15:17:03	controlling	29. 2. 2020 15:23:05	no limit	0,0

Obr. 28 Po uplynutí

Klient, ktorý vzdialený panel ovláda, dostane informáciu o čase ukončenia ovládania. Ten môže kontrolu nad panelom odovzdať aj skôr (Obr. 29) alebo môže webový preliadač ukončiť.



Obr. 29 Front panel na webe

2.2. Analýza vzdialeného panela

Zo zaujímavosti ohľadom vzdialeného panela som sa pozrel na to, ako to sčasti celé funguje. LabVIEW má zabudovaný webový server, ktorý umožňuje pomocou aplikačného rozhrania komunikovať s LabVIEW programom (aplikáciou pracoviska). V zdrojovom kóde stránky, konkrétne v elemente object sa nachádza odkaz na zásuvný modul LabVIEWControl (Obr. 30), kde

sú nastavené parametre ako napr. názov volaného VI súboru. Ďalej je zrejmé, že vzdialený panel využíva protokol LV_FrontPanelProtocol a špecifikuje verziu rpvi190 (LabVIEW 2019).

```
<object id="LabVIEWControl" classid="CLSID:A40B0AD4-B50E-4E58-8A1D-8544233807BB" width="1028" height="630" codebase="ftp://ftp.ni.com/support/labview/runtime/windows/2019/LVRTE2019min.exe">
  <param name="LVFPPVINAME" value="Multimeter_orig.vi">
  <param name="REQCTRL" value="true">
  <embed src=".LV_FrontPanelProtocol.rpvi190" lvfppviname="Multimeter_orig.vi" reqctrl="false" type="application/x-labviewrpvi190" width="1028" height="630" pluginspage="http://digital.ni.com/express.nsf/bycode/exck2m">
</object>
```

Obr. 30 Zdrojový kód stránky

Dokumentácia o protokole však nie je verejne zverejnená, preto nasleduje môj subjektívny odhad na základe pozorovania komunikácie pomocou analyzátoru sieťových protokolov Wireshark. Po načítaní stránky so vzdialeným panelom a následnou inicializáciou vstavaného modulu, začne komunikácia vstavaného modulu s webovým serverom (Obr. 31).

Time	Source	Destination	Protocol	Length	Time to live	Info
82	2.137583	127.0.0.1	TCP	44	128	65445 → 8000 [ACK] Seq=323 Ack=241 Win=261888 Len=0
85	2.222823	127.0.0.1	TCP	56	128	65447 → 8000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 S
86	2.222875	127.0.0.1	TCP	56	128	8000 → 65447 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=654
87	2.222934	127.0.0.1	TCP	44	128	[65447 → 8000] [ACK] Seq=1 Ack=1 Win=2619648 Len=0
92	2.228272	127.0.0.1	1	106	128	GET /LabVIEW.FrontPanelProtocol HTTP/1.0
93	2.228299	127.0.0.1	TCP	44	128	8000 → 65447 [ACK] Seq=1 Ack=63 Win=2619648 Len=0
96	2.231976	127.0.0.1	2	67	128	8000 → 65447 [PSH, ACK] Seq=1 Ack=63 Win=2619648 Len=23 [TC
97	2.232014	127.0.0.1	TCP	44	128	65447 → 8000 [ACK] Seq=63 Ack=24 Win=2619648 Len=0
98	2.232459	127.0.0.1	3	101	128	Continuation
100	2.232489	127.0.0.1	TCP	44	128	8000 → 65447 [ACK] Seq=24 Ack=120 Win=2619648 Len=0
104	2.233804	127.0.0.1	4	96	128	8000 → 65447 [PSH, ACK] Seq=24 Ack=120 Win=2619648 Len=52 [TC
105	2.233836	127.0.0.1	TCP	44	128	65447 → 8000 [ACK] Seq=120 Ack=76 Win=2619648 Len=0
106	2.236923	127.0.0.1	5	115	128	Continuation
107	2.236947	127.0.0.1	TCP	44	128	8000 → 65447 [ACK] Seq=76 Ack=191 Win=2619648 Len=0
112	2.576800	127.0.0.1	TCP	65539	128	8000 → 65447 [ACK] Seq=76 Ack=191 Win=2619648 Len=65495 [TC
113	2.576824	127.0.0.1	TCP	65539	128	8000 → 65447 [ACK] Seq=65571 Ack=191 Win=2619648 Len=65495
114	2.576838	127.0.0.1	TCP	65539	128	8000 → 65447 [ACK] Seq=131066 Ack=191 Win=2619648 Len=65495
115	2.576852	127.0.0.1	TCP	65539	128	8000 → 65447 [ACK] Seq=196561 Ack=191 Win=2619648 Len=65495
116	2.576867	127.0.0.1	TCP	65539	128	8000 → 65447 [ACK] Seq=262056 Ack=191 Win=2619648 Len=65495

Obr. 31 Zobrazenie komunikácie v programe Wireshark

Klient (modul) pošle HTTP požiadavku s prechodom na protokol vzdialeného panela (Obr. 31 - 1 a Obr. 32) , kde server následne odpovie (Obr. 31 - 2 a Obr. 33) v formáte podobnom HTTP protokolu. TCP spojenie ostáva otvorené a komunikácia od tohto okamihu je výhradne v protokole vzdialeného panela.

```
GET /LabVIEW.FrontPanelProtocol HTTP/1.0\r\n
LVFPREQUEST: TRUE\r\n
\r\n
[HTTP request 1/1]
```

Obr. 32 Zaslanie požiadavky o zmenu protokolu

TCP segment data (65495 bytes)		
0020	64 4d 5e 1b 50 10 27 f9 88 cc 00 00 8d c0 00 10	dM^P.''.
0030	00 00 00 01 00 00 00 11 00 00 00 01 00 00 00
0040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050	00 00 00 00 00 00 00 00 00 00 00 00 00 11 22 af"
0060	00 00 00 00 00 00 00 00 40 80 02 10 00 00 00 01	@.....
0070	12 4d 75 6c 74 69 6d 65 74 65 72 5f 6f 72 69 67	+Multime ter_orig
0080	2e 76 69 00 e0 63 80 cb 43 02 00 00 00 00 00 00	.vi-c- C.....
0090	00 00 00 00 80 63 80 cb 43 02 00 00 00 00 00 00c- C.....
00a0	00 00 00 00 20 63 80 cb 43 02 00 00 00 00 00 00c- C.....

Obr. 37 Prenos VI súboru

TCP segment data (76 bytes)		
02 00 00 00 45 00 00 74 3b ea 40 00 80 06 00 00E..t ;@.....	
7f 00 00 01 7f 00 00 01 1f 40 ff a7 e3 cf 01 78 @.....x	
64 4d 62 5f 50 18 27 f4 fd a5 00 00 8d c0 00 10	dMb_P.''.	
00 00 00 37 00 00 00 14 00 00 00 00 00 00 00	...7.....	
00 00 00 1c 00 00 00 18 00 00 00 00 00 00 00	
00 00 00 00 00 00 00 00 00 00 00 00 00 00 18	
00 00 00 14 56 69 72 74 75 e1 6c 6e 79 20 6d 75Virt u-lyny mu	
6c 74 69 6d 65 74 65 72	ltimeter	

Obr. 38 Prenesená hodnota panela

3. Návrh

3.1. Úvod

Pred samotným návrhom bolo potrebné oboznámiť sa s vývojovým prostredím LabVIEW NXG a jeho doplnkom LabVIEW NXG Web Module. Jednou z úloh práce je navrhnúť nové riešenie webových stránok a web servera laboratória merania, taktiež navrhnúť a implementovať správu klientov pre prístup k jednotlivým pracoviskám pomocou týchto vývojových prostriedkov. Cieľom bolo taktiež umožniť prístup na pracoviská bez použitia zásuvného modulu (pluginu) a nutnosti inštalácie LabVIEW RunTime Engine.

Jednotlivé pracoviská boli vytvorené v LabVIEW a ich vzdialené ovládanie pomocou vstavaného nástroja Web Publishing Tool. LabVIEW však poskytoval aj inú možnosť, a to Web Services (Webové služby), ktoré umožňovali akémukoľvek programu prijímať HTTP požiadavky cez aplikačné rozhranie webového servera (prístupné v palete funkcií). Vo vývojovom prostredí LabVIEW NXG (verzia 4.0) však takéto prostriedky nie sú k dispozícii a komunikácia aplikácie s webovým rozhraním je možná pomocou TCP/UDP protokolov alebo aplikačného rozhrania SystemLink (ako klient cez HTTP). Doplnok LabVIEW NXG Web Module (verzia 4.0) umožňuje ľahšiu tvorbu webových rozhraní aplikácií a jednoduchých stránok. Komunikácia webového rozhrania je možná pomocou protokolov HTTP, WebSocket a webového aplikačného rozhrania SystemLink (pomocou HTTP). SystemLink od National Instruments je platforma, ktorá poskytuje nasadenie softvéru, konfiguráciu systémov, monitorovanie testov a nástroje na správu dát pre testovacie a meracie systémy. Dátové služby SystemLink-u sú súčasťou doplnku LabVIEW NXG Web Module a návrh nového riešenia laboratória merania ich bude využívať, konkrétne službu Tags (Značky, Tagy) ako vzdialené premenné. Služba Tagov poskytuje možnosť vytvárať, zapisovať a čítať dáta identifikované názvom (cestou), ktoré spravuje SystemLink server a pristupovať k nim pomocou aplikačného rozhrania.

Konverzia programov vytvorených v LabVIEW do LabVIEW NXG je zložitá a zdĺhavá (záleží od zložitosti programu). Taktiež v LabVIEW NXG nie sú dostupné všetky doplnky, ktoré sú v LabVIEW k dispozícii. Výhodou SystemLinku je možnosť prepojiť medzi sebou rôzne produkty od NI ako aj programy vytvorené v návrhovom prostredí LabVIEW a LabVIEW NXG spolu s webovým rozhraním vytvoreným pomocou doplnku LabVIEW NXG Web Module.

Po oboznámení sa s možnosťami LabVIEW NXG, jeho doplnku Web Module a dátovými službami SystemLink som zistil, že služba Tagov neposkytuje možnosť nastaviť práva pre zápis a čítanie tagu

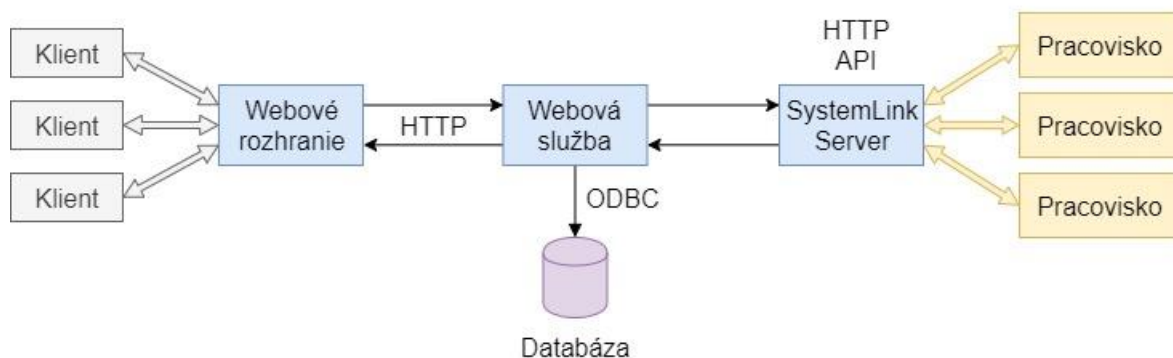
pre jedného klienta, ale len pre skupinu - čítanie a osobitne vytváranie, zápis a mazanie (Obr. 39). Je možné že v budúcnosti bude takáto možnosť k dispozícii.



Obr. 39 Možnosti nastavenia v NI Web Server

3.2. Návrh riešenia

Jedným z cieľov, ktoré som si stanovil, bolo vytvoriť platformu, kde by bolo možné webové stránky pracoviska do nového systému jednoducho prerobiť a spravovať. Ďalej aby bola možnosť vzdialene spustiť program pracoviska, v prípade ak nebeží, kontrolu nad pracoviskom limitovať a zabezpečiť systém rezervácií.

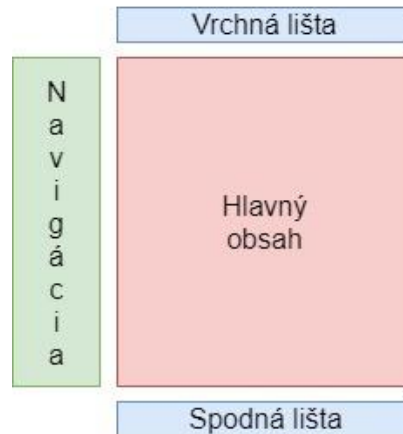


Obr. 40 Navrhnuté riešenie

3.2.1. Webové rozhranie

Webová stránka laboratória merania by mal byť jednoduchá dynamická stránka s kaskádovými štýlmi a potrebnými JavaScriptami v externom súbore. Malo by byť možné ju upraviť a prípadné podstránky vytvoriť jednoducho pomocou doplnku LabVIEW NXG Web Module alebo v prípade potreby priamo jazykom HTML. Stránka(Obr. 41)by sa mala skladať z:

- Vrchná časť (vrchná lišta) - tvorí ju bude informačný text a v prípade potreby hypertextové odkazy
- Navigačné menu (navigácia) – tvorí ho budú odkazy na podstránky stránky s iným obsahom
- Časť s hlavným obsahom konkrétnej stránky
- Spodná časť (spodná lišta, footer, zápätie) – tvorí ju bude informačný text



Obr. 41 Návrh rozloženia stránky

Dynamickosť stránky by bola zabezpečená použitím technológie AJAXu pre zasielanie HTTP požiadaviek na pozadí v prípade žiadosti o obsah a následného vygenerovania a zobrazenia HTML obsahu na strane klienta. Stránka by mala obsahovať rozhranie pre prihlásenie používateľa, úpravu jeho prípadných informácií (napr. profilu), zmeny hesla a spravovania rezervácií.

Stránka pracoviska sa bude skladať z rovnakých častí, dodatkom však bude v časti s obsahom „aplikačná“ lišta s informačným textom o stave kontroly pracoviska, tlačidlami pre prípadné spustenie pracoviska a tlačidlami pre nadobudnutie / odovzdanie kontroly nad pracoviskom. Informáciu o stave kontroly bude treba dynamicky získavať od webovej služby.

Vzdialené panely pracovísk budú vytvorené pomocou LabVIEW NXG Web Module s rovnakým vzhľadom a funkciami. Získavanie a odosielanie dát pre panel bude zabezpečené HTTP protokolom. Aplikačná lišta bude panelu zabezpečovať informácie o možnosti ovládania panela. Webové rozhranie bude sprístupnené webovým serverom NI Web Server, ktorý je súčasťou platformy SystemLink.

3.2.2. Webová služba

Webová služba (web server) bude realizovaná v návrhovom prostredí LabVIEW NXG. Jej hlavnou úlohou bude zabezpečovať prenos dát pomocou HTTP protokolu klientom webového rozhrania a vzdialenému panelu pracoviska zo SystemLink servera cez aplikačné rozhranie (Obr. 40). Ďalšou úlohou bude riadiť prístup k pracoviskám a bude tvorená z týchto služieb:

- Relácie – možnosť vytvorenia a kontrola stavu relácie
- Pracoviská – poskytovanie informácií o stave a možnosti kontroly pracoviska a prenášanie dát vzdialeného panela

- Používatelia – možnosť prihlásenie, úpravy profilu a zmeny hesla
- Rezervácie – poskytovanie informácií používateľom o ich rezerváciách a možnosť tvorby nových či prípadné odstránenie rezervácie

Prístup k jednotlivým službám bude zabezpečený cez aplikačné rozhranie jednotným identifikátorom prostriedku (URI) cez HTTP protokol pomocou požiadaviek POST a GET. Webová služba bude poskytovať nástroje na administráciu a prípadnú konfiguráciu týchto služieb.

3.2.3. Databáza

Pre uloženie informácií o reláciách, používateľoch, pracoviskách a rezerváciách bude potrebná databáza. National Instruments poskytuje pre LabVIEW (aj NXG) súbor nástrojov na pripojenie k databázam - LabVIEWDatabase Connectivity Toolkit. Komunikácia s databázou je možná cez aplikačné rozhranie ODBC (Open Database Connectivity). K tomu je ešte potrebné stiahnuť ovládač (ODBC connector) pre zvolenú databázu.

3.2.4. SystemLink server

Pre zapisovanie a čítanie dát vzdialených panelov pracovísk a sprostredkovateľom medzi webovou službou bude použitý SystemLink server, konkrétne jeho služba Tags pomocou aplikačného rozhrania dostupného pre vývojové prostredie LabVIEW aj LabVIEW NXG.

3.2.5. Pracovisko

Jednotlivé pracoviská bude potrebné konvertovať tak, aby zapisovali dáta zvolených prvkov panela do vopred definovaných tagov a čítali potrebné dáta z tagov - v prípade potreby, v zvolenom periodickom časovom intervale.

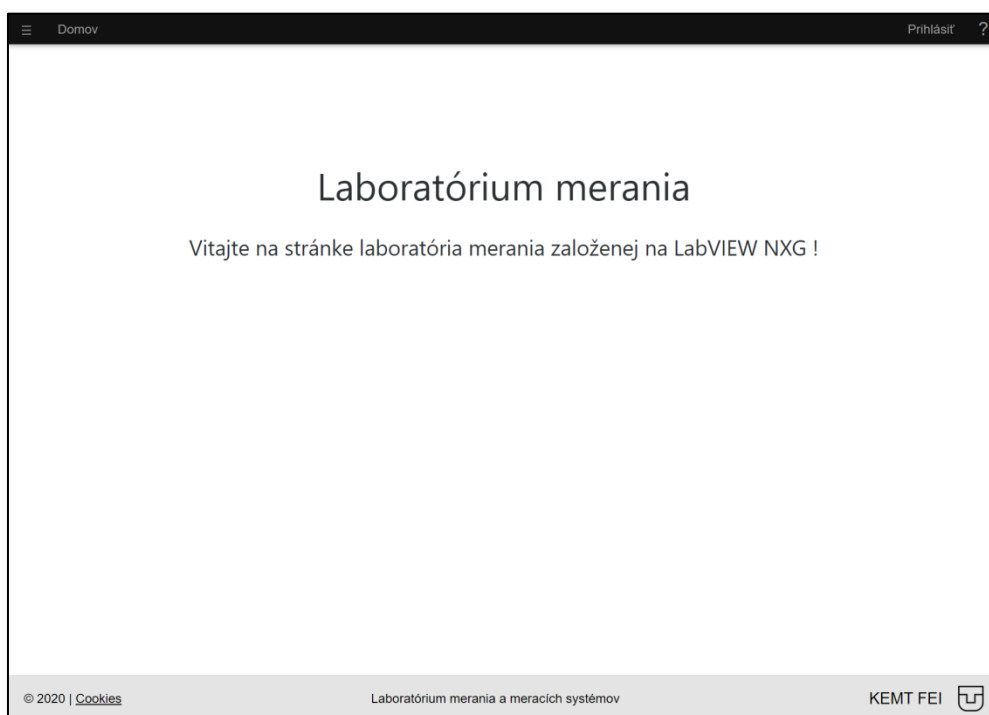
4. Realizácia navrhnutého riešenia

4.1. Úvod

V tejto časti bude prezentované riešenie a popísané vybrané funkcie riešenia.

4.2. Webové Rozhranie

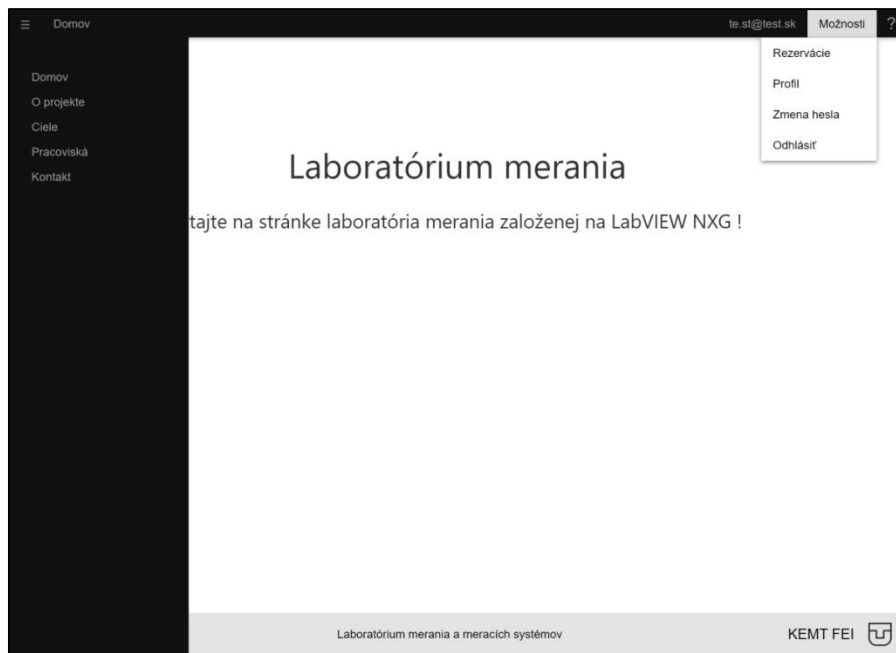
Webová stránka laboratória merania má jednoduchý dizajn. Vo vrchnej časti sa na ľavej strane nachádza odkaz na bočné navigačné menu a názov (titulok) konkrétnej podstránky. Ak používateľ nie je prihlásený, na pravej strane sa nachádza odkaz na prihlasovacie okno a tlačidlo s rozbaľovacou (dropdown) ponukou (Obr. 42).



Obr. 42 Webová stránka laboratória merania

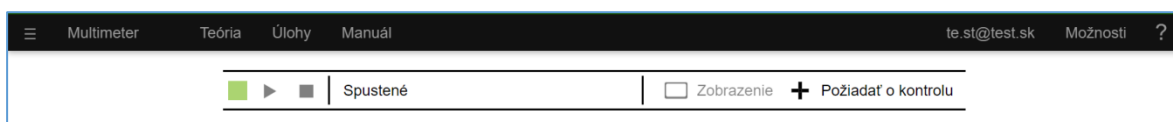
Po prihlásení používateľa, sa na pravej strane nachádza jeho emailová adresa, tlačidlo s rozbaľovacou ponukou (Možnosti), ktorá obsahuje odkazy na okná rezervácií, profilu, zmeny hesla a odkaz na odhlásenie používateľa (Obr. 43). Navigačné menu je vysúvacie a obsahuje odkazy na jednotlivé podstránky (Obr. 43). V spodnej časti sa nachádza informačný text ako napr. skratka katedry a fakulty.

Obsah vrchnej časti, jednotlivých podstránok, navigácie a spodnej časti je možné upraviť podľa potreby jednoducho pomocou doplnku LabVIEW NXG Web Module alebo v prípade potreby priamo jazykom HTML.



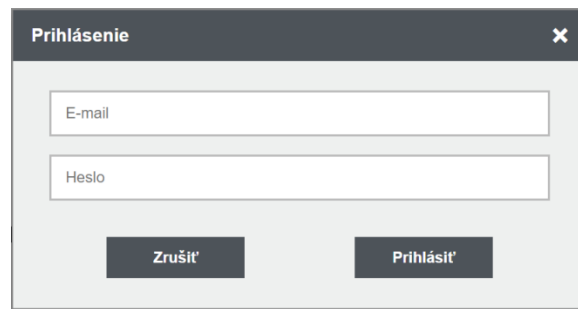
Obr. 43 Navigačná lišta

Vo vrchnej časti stránky pracoviska sa za názvom podstránky nachádzajú odkazy na okná dodatočného obsahu pre konkrétne pracovisko (Teória, Úlohy a Manuál) (Obr. 44). V časti s obsahom je „aplikačná“ lišta, ktorá informuje o stave pracoviska a kontroly. Umožňuje dané pracovisko vzdialene spustiť a zastaviť, požiadať alebo odovzdať kontrolu nad pracoviskom a zakázať zobrazenie dát panela užívateľom, ktorý nie sú v kontrole pracoviska.



Obr. 44 Vrchná časť stránky a aplikačná lišta

Používateľ sa môže prihlásiť pomocou svojej e-mailovej adresy a hesla, ktoré zadá do vyskakovacieho okna prihlásenia (Obr. 45). Ďalej môže zmeniť svoje heslo (Obr. 46), upraviť profil (Obr. 47), spravovať svoje rezervácie (Obr. 48) a vytvoriť novú rezerváciu (Obr. 49).



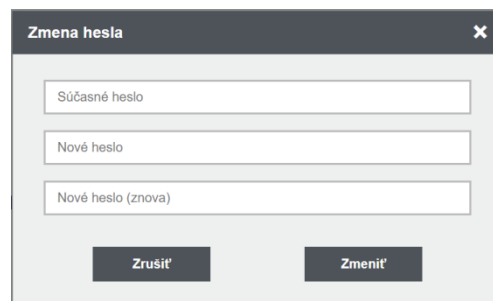
Príhlásenie

E-mail

Heslo

Zrušiť Prihlásiť

Obr. 45 Okno prihlásenia



Zmena hesla

Súčasnú heslo

Nové heslo

Nové heslo (znova)

Zrušiť Zmeniť

Obr. 46 Okno zmeny hesla



Profil

Meno	Ján
Príezvisko	Novák
Ročník	1.
Študijný program	elektronika

Zrušiť Upraviť

Obr. 47 Okno úpravy profilu

Názov pracoviska	Začiatok	Koniec	
Multimeter	14:30 28.3.2020	14:40 28.3.2020	X
Multimeter	13:00 29.3.2020	13:05 29.3.2020	X
MultimeterJSON	15:20 29.3.2020	15:30 29.3.2020	X
MultimeterJSON	18:55 30.3.2020	19:00 30.3.2020	X
MultimeterJSON	10:15 1.4.2020	10:25 1.4.2020	X
MultimeterJSON	12:49 2.4.2020	12:51 2.4.2020	X
Multimeter	14:25 2.4.2020	14:30 2.4.2020	X
MultimeterJSON	16:35 3.4.2020	16:43 3.4.2020	X

Navigation: < >

Odstrániť rezerváciu?

Obr. 48 Okno rezervácií

Rezervácie

multí Defň dd. mm. rrrr

Multimeter

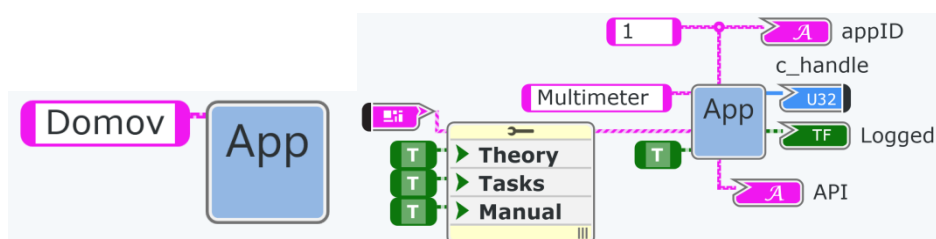
MultimeterJSON

Rezervácie

Obr. 49 Okno pridania rezervácie

4.2.1. Opis funkcií

Web Modul poskytuje rozhranie pre implementáciu vlastných funkcií (programov) napísaných v jazyku JavaScript, ktoré možno použiť v blokovom diagrame stránky. Pre funkčnosť stránky je potrebné mať okrem obsahu aj iné časti ako napr. navigáciu, preto boli jednotlivé časti stránky dodatočne vytvorené v HTML a JS (interakcia s používateľom či webovou službou).



Obr. 50 Podprogram pre podstránku a pracovisko

Podprogram „App.gvi“ (Obr. 50) bol vytvorený na to, aby bolo možné veľmi jednoducho zobraziť navigáciu a ďalšie prvky na každej podstránke. Ako prvé iniciuje na pozadí reláciu s webovou službou (cez HTTP požiadavku), kontrolu či je daný klient prihlásený alebo nie, nastaví titulok stránky (HTML element title) a uloží potrebné informácie do dočasného úložiska prehliadača (sessionStorage pomocou Web Storage API). Následne vygeneruje HTML kód pre vrchnú časť, navigáciu a spodnú časť na základe definovaného obsahu. Potom HTML kód pre vyskakovacie okno a podľa toho či sa jedná o pracovisko tak aj kód aplikačnej lišty. Jednotlivé prvky spolu s ich potrebnými externými JS súbormi sú vložené do stránky pomocou JS funkcie.

Aplikačná lišta (Obr. 43) zabezpečuje komunikáciu s webovou službou ohľadom pracoviska pomocou HTTP požiadaviek. Získava informácie o stave a možnosti kontroly pracoviska v pravidelnom intervale (5 sekúnd) a tie zapisuje do dočasného úložiska prehliadača. Informuje používateľa, umožňuje žiadať alebo odovzdať kontrolu pracoviska či obmedziť zobrazenie dát panela.

Podprogram „AppBarStatus.gvi“ slúži na načítanie informácií o stave a kontrole pracoviska z dočasného úložiska prehliadača, ktoré interpretuje panelu pracoviska či je možné pracovisko ovládať alebo zobrazovať dáta panela.

4.2.2. Štruktúra hlavného priečinka

Štruktúra hlavného priečinka webového rozhrania po kompilácii v LabVIEW NXG s web modulom je nasledovná:

- / - hlavný priečink, obsahuje jednotlivé podstránky (Úvod, Ciele, Pracoviská, atď)
- /css/style.css – súbor obsahuje CSS štýly pre všetky prvky stránky
- /images/ - priečink obsahuje obrázky pre tlačidlá
- /js/appbar.js – obsahuje JS funkcie, ktoré zabezpečujú funkcionality aplikačnej lišty
- /js/functions.js – obsahuje JS funkcie, ktoré sú používané v podprograme „App.gvi“
- /js/modal.js – obsahuje JS funkcie, ktoré zabezpečujú funkcionality vyskakovacieho okna (otvorenie a zatvorenie okna)
- /js/nav.js – obsahuje JS funkcie, ktoré zabezpečujú funkcionality navigačného menu (otváranie, skrývanie vysúvacieho bočného menu a rozbaľovacej ponuky)
- /js/other.js – obsahuje JS funkcie, ktoré zabezpečujú funkcionality okna prihlásenia, zmeny hesla, profilu a rezervácií
- /lab/ - priečink pre podstránky pracovísk
- /modals/ - priečink obsahuje vyskakovacie okná
- /config.js – konfiguračný súbor webového rozhrania

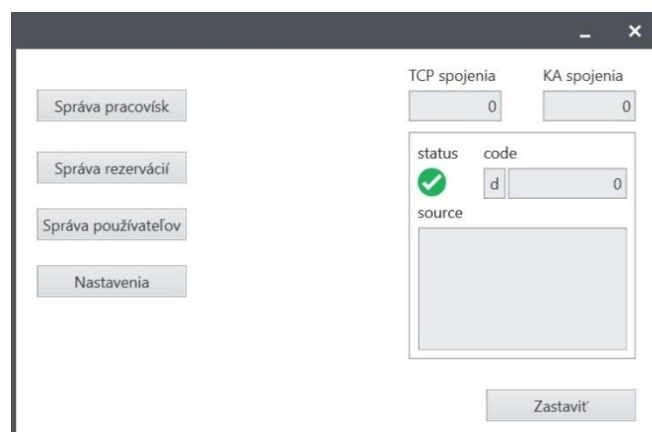
- /ni-webvi-resource-v0/ - priečnik obsahuje súbory vygenerované web modulom potrebné pre chod webového rozhrania

Publikácia webového rozhrania na web je zabezpečená NI Web Serverom, ktorý je súčasťou platformy SystemLink. Adresár webového servera:

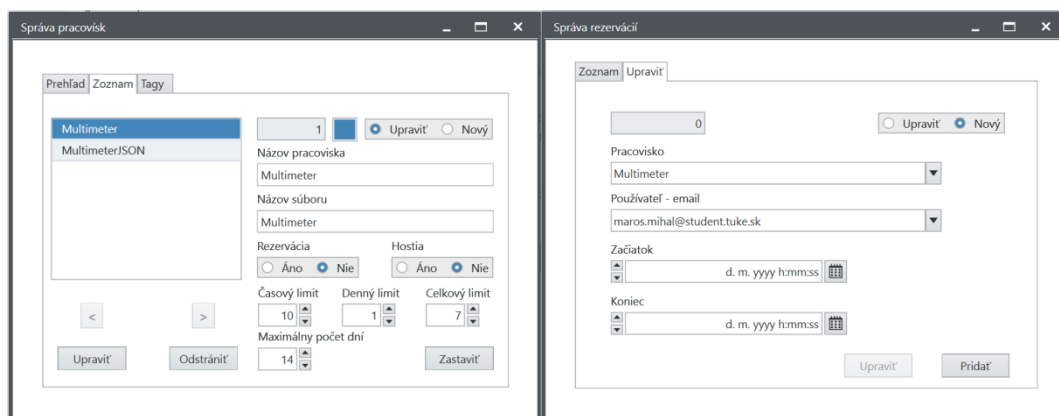
- C:\Program Files\National Instruments\Shared\Web Server\htdocs

4.3. Webová služba

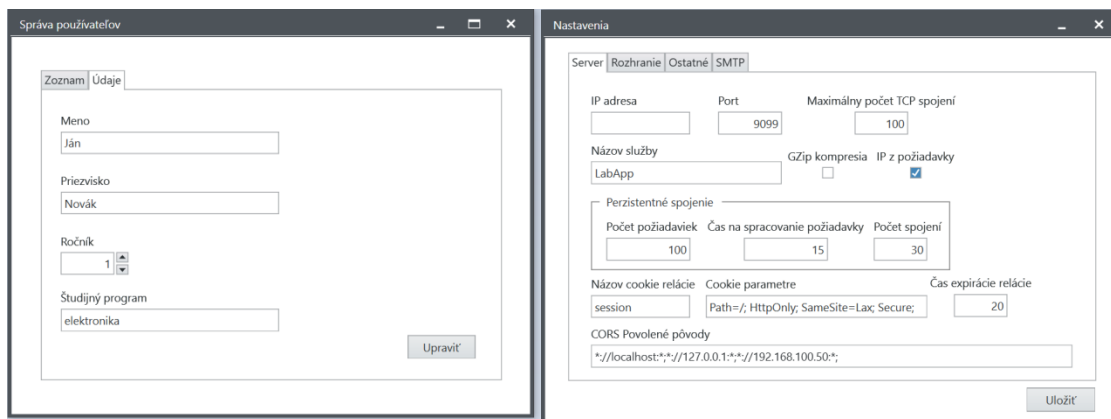
Webová služba (Obr. 51) bola realizovaná v návrhovom prostredí LabVIEW NXG a tvorí ju jednoduchý HTTP server. Jej úlohou je zabezpečenie funkcií webového rozhrania, komunikácia s databázou a pracoviskami cez SystemLink server. Pristupovať na ňu je možné pomocou aplikačného rozhrania určeného jednotným identifikátorom prostriedku (URI). Poskytuje aj nástroje na správu pracovísk, rezervácií, používateľov a úpravu konfiguračných parametrov (Obr. 52 a Obr. 53).



Obr. 51 Hlavné okno aplikácie

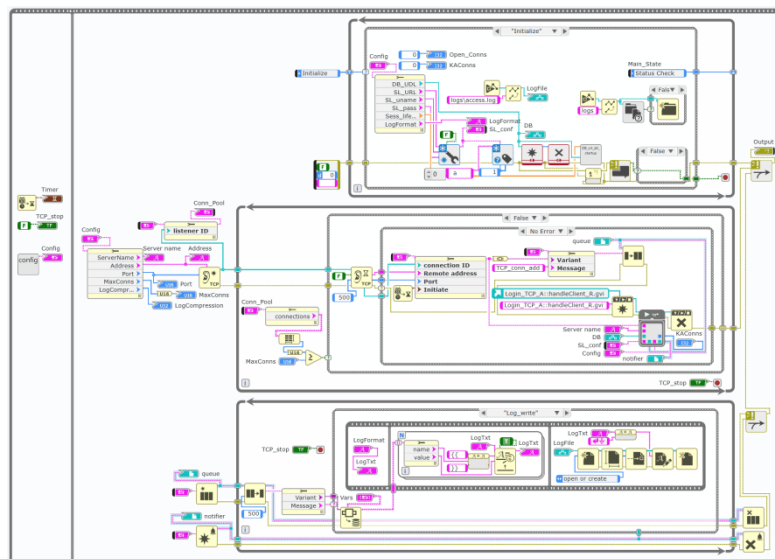


Obr. 52 Okno správy pracovísk a rezervácií



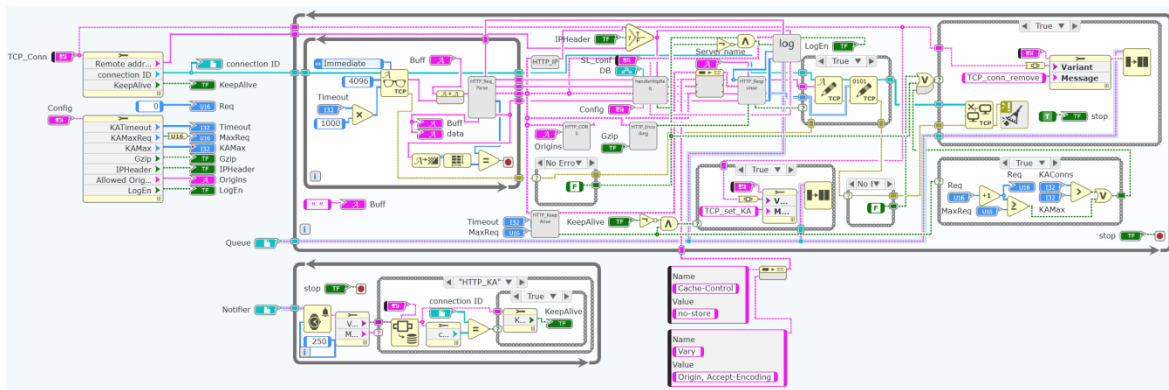
Obr. 53 Okno správy používateľov a nastavení

4.3.1. Opis funkcií



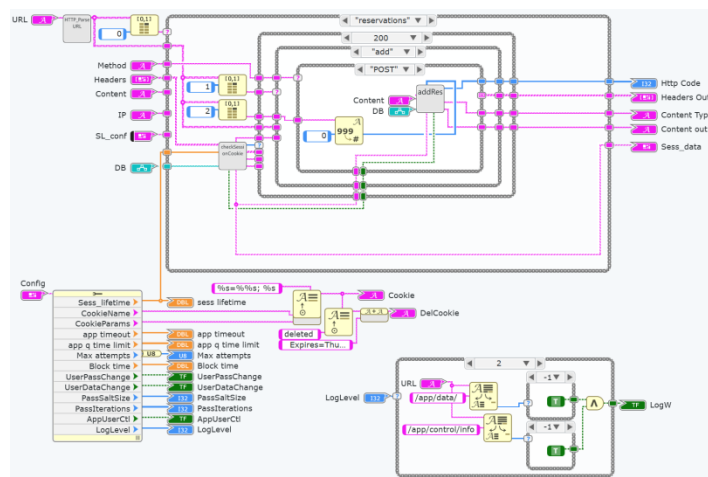
Obr. 54 Náhľad do hlavného programu main.gvi

Hlavným programom je „main.gvi“ (Obr. 54) a pri spustení služby ako prvé dochádza k načítaniu konfiguračného súboru, potom inicializácia a skúška spojenia so SystemLink serverom a serverom databázy. Následne sa otvorí TCP socket (zásuvka) na nastavenej adrese a porte, ktorý čaká na klienta. Ak príde klient, uložia sa jeho informácie (identifikátor spojenia, adresa, port a čas iniciácie spojenia) do zoznamu otvorených TCP spojení a spustí sa podprogram „handleClient.gvi“.



Obr. 55 Náhľad do podprogramu handleClient.gvi

Podprogram „handleClient.gvi“ (Obr. 55) je viacnásobne prístupný (reentrant) s vlastným stavom pre každého klienta, čo znamená, že je možné mať viacerých klientov súčasne. Posielanie informácií hlavnému programu je možné pomocou LabVIEW funkcie Queue. Podprogram slúži na prijatie dát od klienta (požiadavky), potom dochádza k analýze jeho HTTP požiadavky a spusteniu podprogramu „handleHttpRequest.gvi“. Po spracovaní požiadavky sa vytvorí a naformátuje HTTP odpoveď, ktorá sa pošle klientovi a spojenie sa uzavrie. Server podporuje aj perzistentné spojenie, čo znižuje celkovú odozvu. Súčasťou podprogramu je kontrola pôvodu zdroja požiadavky (v rámci CORS, Cross-origin resource sharing) a voliteľná kompresia odpovede.



Obr. 56 Náhľad do podprogramu handleHttpRequest.gvi

Podprogram „handleHttpRequest.gvi“ (Obr. 56) analyzuje požiadavku, skontroluje stav relácie a podľa http metódy a identifikátora (URI) spustí podprogram, ktorý danú požiadavku spracuje a odovzdá stavový kód, prípadné hlavičky (headers), typ obsahu a samotný obsah.

Podprogramy sú rozdelené do kategórií podľa typu požiadavky.

4.3.1.1. Relácie

V tejto kategórii sa nachádzajú podprogramy na vytvorenie, kontrolu a zrušenie relácie. Jednotlivé relácie sú na strane servera uložené do databázy. Identifikátor relácie má dĺžku 32 bajtov (byte-ov), je generovaný kryptograficky bezpečným generátorom pseudonáhodných čísel (CSPRNG) a následne kódovaný do base64. Funkcie generátora a kódovania sú volané pomocou .NET rozhrania (funkcie sú súčasťou .NET core alebo .NET Framework). Na strane klienta je relácia uložená v prehliadači ako HTTP cookie.

4.3.1.2. Pracoviská

V tejto kategórii sa nachádzajú podprogramy na spustenie, zastavenie, informovanie o stave a možnosti kontroly pracoviska a prenášanie dát vzdialeného panela. Vzdialené spustenie a zastavenie skompilovanej aplikácie pracoviska je voliteľné a možné pomocou batch (.bat) skriptov, ktoré je potrebné uložiť do priečinka „apps“ v hlavnom priečinku webovej služby. Ovládanie pracoviska je možné obmedziť len pre prihlásených používateľov.

Informácie o pracovisku zahŕňajú stav (spustené, zastavené), možnosť ovládania bez prihlásenia, možnosť vytvorenia rezervácií a limity na rezerváciu.

Informácie o kontrole zahŕňajú či daný klient môže pracovisko ovládať, či je pracovisko práve rezervované, či je povolené zobrazovanie dát panela, celkový počet požiadaviek o kontrolu od iných klientov, časový limit možnosti kontroly daného pracoviska, čas ukončenia možnosti kontroly, či daný klient poslal požiadavku o kontrolu, v akom poradí sa nachádza (ak je viacero požiadaviek o kontrolu) a či je pracovisko práve ovládané.

Získavanie (čítanie) a zapisovanie dát panela je realizované nasledovne. V databáze je uložený zoznam tagov (premenné) pre konkrétne pracovisko – ich názov (cesta), dátový typ, poradie a či ide o čítanie alebo zapisovanie. Ak sa jedná o čítanie, vyberú sa z databázy tagy pre čítanie, následne sa otvorí spojenie so SystemLink serverom, kde sa požiadava o hodnoty tagov. Tie sa prevedú z ich dátového typu na reťazec, zoradia do poľa v definovanom poradí, prevedú na JSON formát a následne odošlú klientovi, ktorý ich spracuje a zobrazí.

Ak sa jedná o zapisovanie, postupuje sa podobne. Vyberú sa konkrétne tagy pre zápis z databázy, obsah požiadavky sa prevedie z JSON formátu do poľa reťazcov, ktoré sa prevedú na konkrétny dátový typ a v definovanom poradí dôjde k zápisu do SystemLink servera. Prístup klientov k jednotlivým pracoviskám je tak možné týmto procesom obmedziť.

4.3.1.3. Používatelia

V tejto kategórii sa nachádzajú podprogramy na prihlásenie, odhlásenie a úpravu údajov používateľa. Prihlásenie je možné zablokovať na určitý interval (v sekundách) po X nesprávnych prihláseniach používateľa. Po prihlásení sa vygeneruje nový identifikátor relácie. Heslo je pred uložením do databázy zahašované hašovacou funkciou PBKDF2-SHA512 (určité parametre je možné nastaviť), ktorá je volaná pomocou .NET rozhrania. Výstup je kódovaný do base64 a formátovaný do tvaru:

\$pbkdf2-sha512{počet iterácií} \${soľ/salt – base64} \${reťazec hašovacej funkcie - base64}

4.3.1.4. Rezervácie

V tejto kategórii sa nachádzajú podprogramy na vytvorenie, kontrolu a zrušenie rezervácií. Vytvoriť rezerváciu môže prihlásený používateľ. Na zvolenom pracovisku je možné vytváranie rezervácií zastaviť. Taktiež je možné nastaviť určité limity pre používateľov pre dané pracovisko ,a to:

- Maximálny čas jednej rezervácie (v minútach)
- Maximálny počet rezervácií používateľa na deň
- Maximálny celkový počet rezervácií používateľa
- Maximálny hraničný časový interval v ktorom je možné vytvoriť rezerváciu (v dňoch)

Ak má používateľ vytvorenú rezerváciu na danom pracovisku, iný používateľ k tomuto pracovisku pristúpiť nemôže.

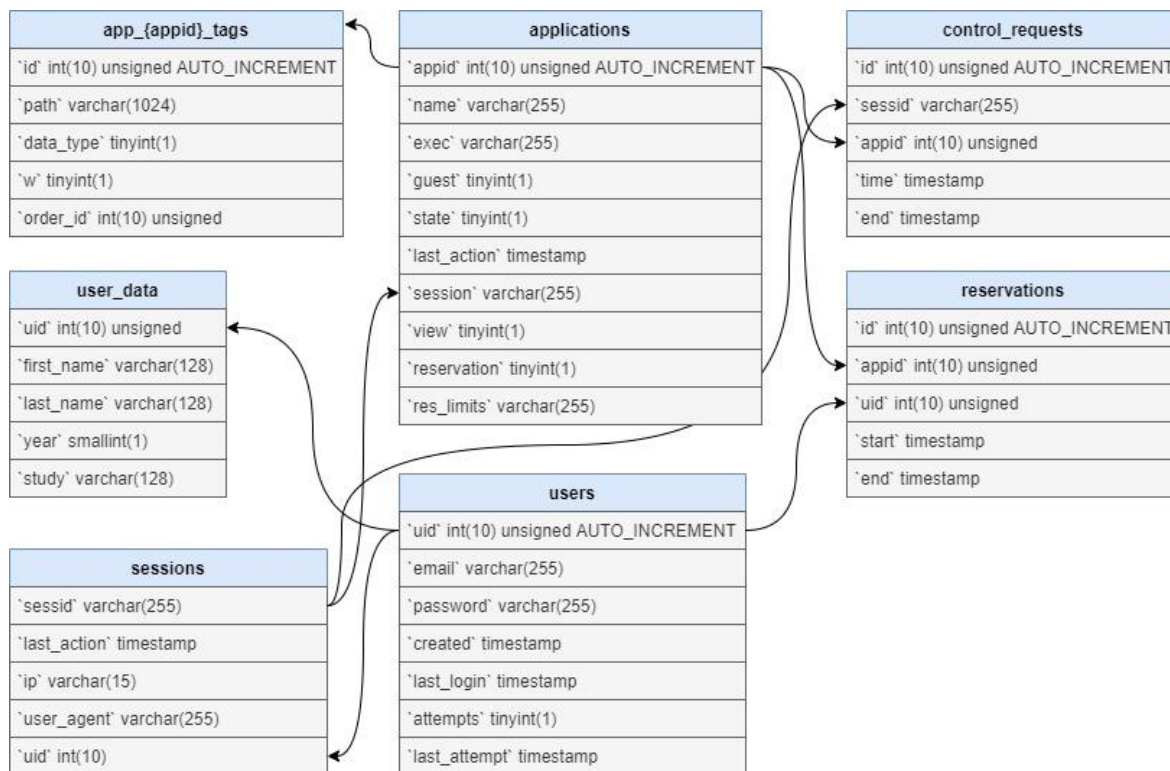
4.3.1.5. Štruktúra hlavného priečinka:

Štruktúra hlavného priečinka po skompilovaní webovej služby je nasledovná:

- / - hlavný priečink, ktorý obsahuje priečinky s „.gll“ knižnicami
- /LabApp.exe/LabApp.exe – skompilovaný súbor webovej služby
- /LabApp.exe/config.ini – súbor obsahuje konfiguračné parametre webovej služby
- /LabApp.exe/DB.udl – tento súbor je potrebné dodatočne vytvoriť, umožňuje spojenie webovej služby s databázou
- /LabApp.exe/mail/ – priečink obsahuje vzory emailov webovej služby
- /LabApp.exe/logs/ – priečink obsahuje záznamové súbory
- /LabApp.exe/apps/ –priečink pre spúšťacie, kontrolné a zatváracie batch skripty pracovísk

4.4. Databáza

Pre uloženie potrebných dát webovej služby bola použitá relačná databáza MySQL (Obr. 57). Časť webovej služby je zabezpečená databázovými udalosťami (events), ako odstránenie starých relácií, rezervácií a resetovanie polí automatického prírastku (auto increment) v tabuľkách „reservations“ a „control_requests“.



Obr. 57 Tabuľky databázy

4.4.1. Tabuľka - sessions

Obsahuje informačné dáta relácií:

- **sessid** – identifikátor relácie
- **last_action** – čas poslednej vykonanej akcie
- **ip** a **user-agent** – ip adresa používateľa a typ prehliadača po prihlásení, tieto dáta nie sú práve využívané
- **uid** – identifikačné číslo používateľa, nastaví sa po prihlásení

4.4.2. Tabuľka - applications

Obsahuje konfiguračné a informačné dáta „aplikácie“ pracoviska:

- **appid** – identifikačné číslo pracoviska

- **name** – názov pracoviska
- **exec** – prefix pre batch skripty, môže byť rovnaký ako názov pracoviska
- **guest**– možnosť kontroly pracoviska bez prihlásenia
- **state** – stav pracoviska, či je spustené alebo nie
- **last_action** – čas poslednej vykonanej akcie na pracovisku
- **session** – identifikátor relácie, ktorý je v kontrole pracoviska
- **view** – povolenie zobrazenia dát panela pre iných používateľov okrem toho kto je kontrole pracoviska
- **reservation** – možnosť vytvorenia rezervácii používateľmi na danom pracovisku
- **res_limits** – jednotlivé obmedzenia rezervácií pracoviska pre používateľov

4.4.3. Tabuľka - app_{appid}_tags

Obsahuje informácie o tagoch konkrétneho pracoviska:

- **id** – identifikačné číslo tagu
- **path** – názov (cesta)
- **data_type** – dátový typ
- **w** – určuje či sa jedná o zápis alebo čítanie
- **order_id** – poradové číslo

4.4.4. Tabuľka - control_requests

Obsahuje podané žiadosti o kontrolu:

- **id**–identifikačné číslo požiadavky
- **appid**– identifikačné číslo pracoviska
- **sessid**– identifikátor relácie
- **time**– čas podania požiadavky
- **end**– čas začatia kontroly (koniec pre toho, kto má práve kontrolu)

4.4.5. Tabuľka - users

Obsahuje informácie o používateľoch:

- **uid** – identifikačné číslo používateľa
- **email** – e-mailová adresa
- **password** – zahašované heslo používateľa
- **created** – čas, kedy bol používateľ vytvorený
- **last_login** – čas posledného prihlásenia
- **attempts** – počet nesprávnych pokusov prihlásenia

- **last_attempt** – čas posledného nesprávneho prihlásenia

4.4.6. Tabuľka - user_data

Obsahuje dodatočné informácie o používateľoch:

- **uid** – identifikačné číslo používateľa
- **first_name** – krstné meno
- **last_name** – priezvisko
- **year** – ročník štúdia
- **study** – študijný program

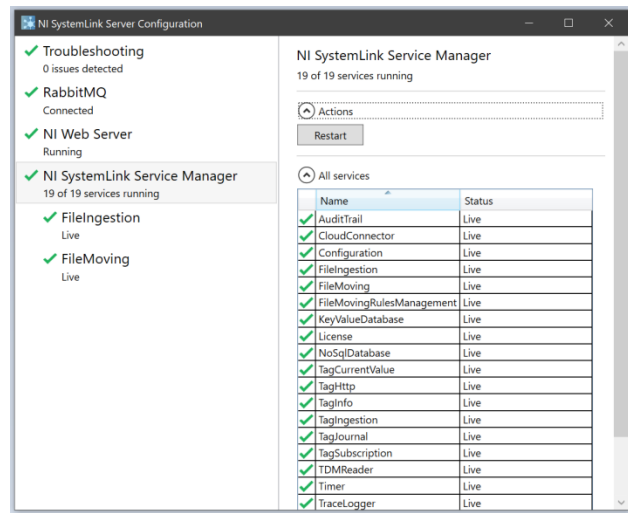
4.4.7. Tabuľka - reservations

Obsahuje rezervácie vytvorené používateľmi:

- **id** – identifikačné číslo rezervácie
- **appid** – identifikačné číslo pracoviska
- **uid** – identifikačné číslo používateľa
- **start** – čas začatia rezervácie
- **end** – čas skončenia platnosti rezervácie

4.5. SystemLink server

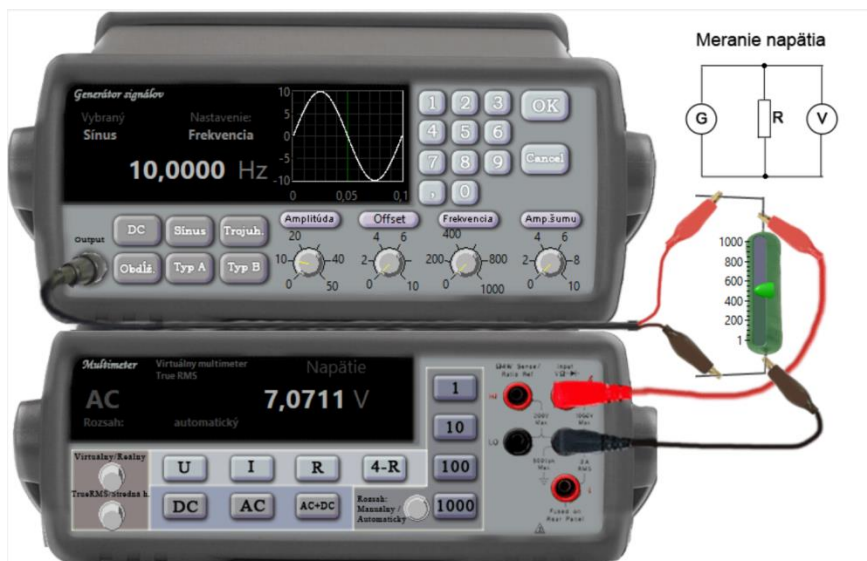
Slúži na prepojenie programu pracoviska a webovej služby. Hlavnou výhodou je jeho podpora pre vývojové prostredia LabVIEW a LabVIEW NXG. SystemLink server neposkytuje žiadne možnosti konfigurácie (Obr. 58). Jeho súčasťou je však NI Web Server a tam bolo potrebné nastaviť heslo pre administrátora. Adresa web servera, jeho port a prihlasovacie údaje slúžia ako konfiguračné parametre pre webovú službu a programy pracovísk, aby bolo možné využívať službu Tagov.



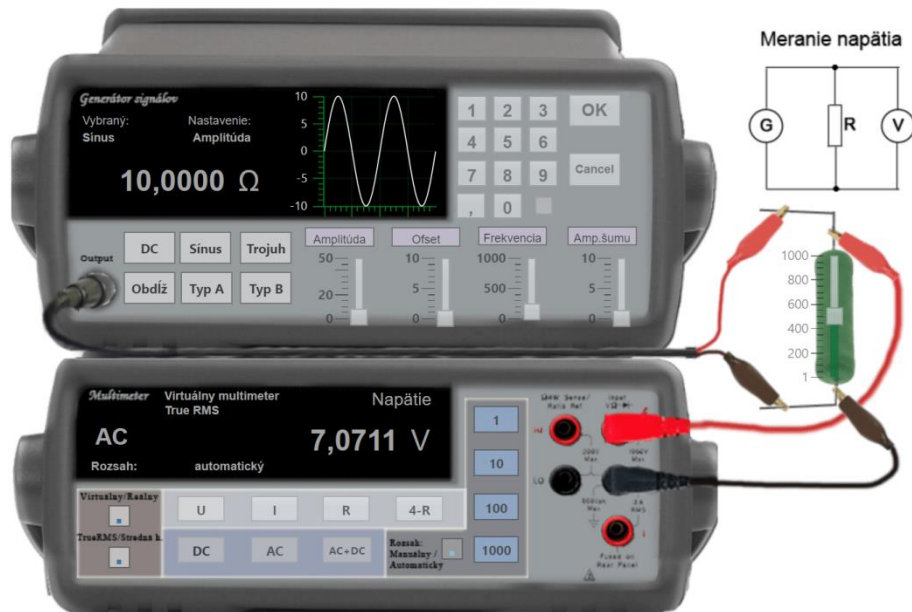
Obr. 58 Konfiguračné okno SystemLink servera

4.6. Pracovisko

Po vybudovaní webovej služby bolo potrebné prerobiť pracovisko a tak vyskúšať funkčnosť celého systému. Vybral som si pracovisko Multimeter (Obr. 59), na základe ktorého som ako prvé vytvoril podobný vzdialený panel pomocou doplnku LabVIEW NXG Web Module (Obr. 60).

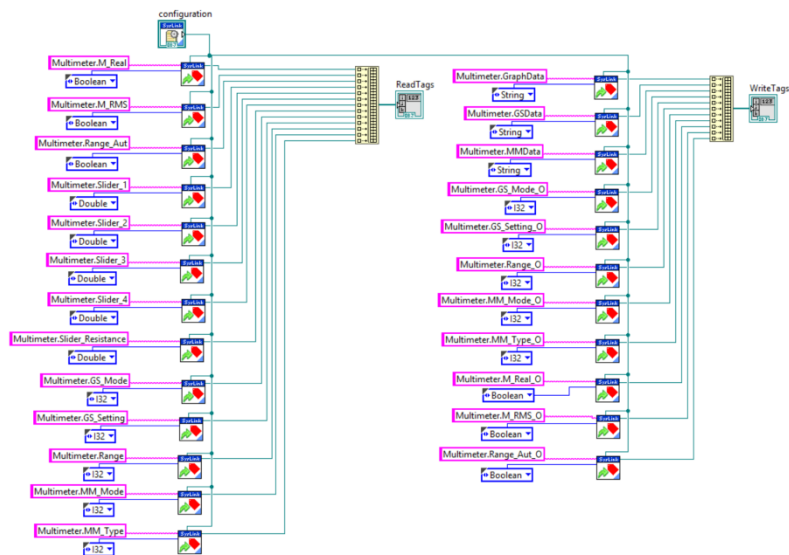


Obr. 59 Predný panel pracoviska Multimeter v LabVIEW



Obr. 60 Vzdialený panel pracoviska Multimeter

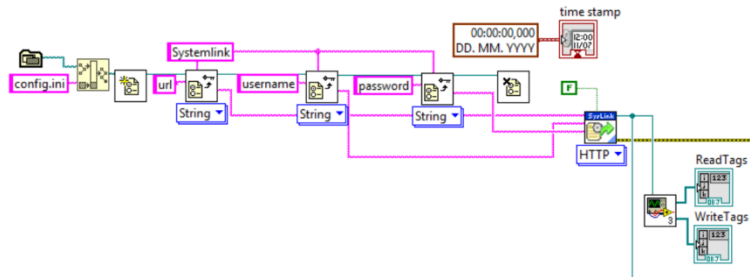
Hodnoty jednotlivých zobrazovacích prvkov pracoviska Multimeter je potrebné na vzdialený panel prenášať a aby mal každý klient po zobrazení vzdialeného panela rovnaké nastavenie je potrebné taktiež prenášať hodnoty ovládacích prvkov. Vzdialený panel (Obr. 60) musí prenášať hodnoty svojich ovládacích prvkov aby bolo možné pracovisko ovládať.



Obr. 61 Otváranie tagov na pracovisku

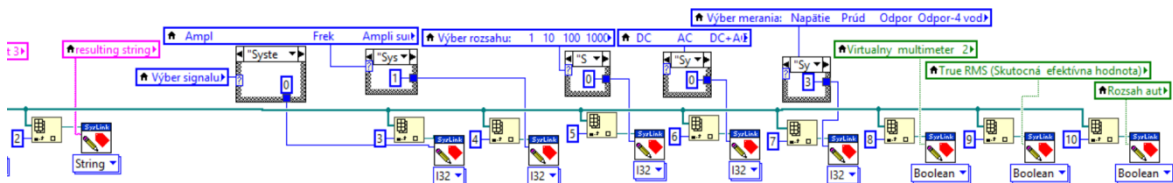
Na základe toho som každému potrebnému prvku priradil tag, ktorý má podobný názov a dátový typ ako vybraný prvok. Tagy sú rozdelené na čítacie a zapisovacie (Obr. 61).

Pred samotným čítaním a zápisom tagov je potrebné otvoriť spojenie so SystemLink serverom (Obr. 62). Ak je program pracoviska skompilovaný, je vhodné mať konfiguračný súbor pre možnosť neskoršej zmeny potrebných parametrov k pripojeniu na server.

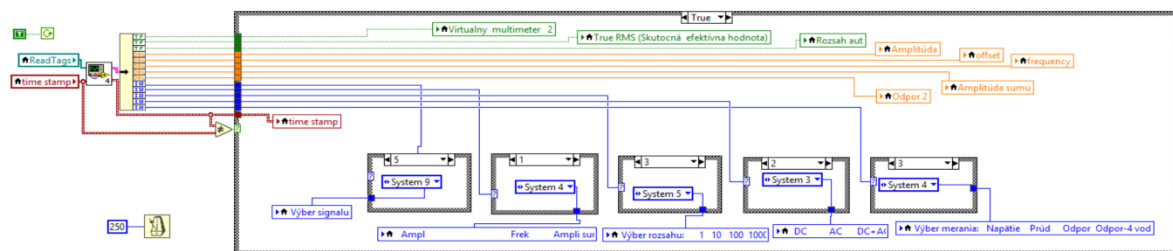


Obr. 62 Otvorenie spojenia

Pracovisko zapisuje hodnoty zobrazovacích a ovládacích prvkov do zapisovacích tagov (WriteTags) (Obr. 63), a číta tagy (ReadTags) ovládacích prvkov vzdialeného panela (Obr. 64), podľa ktorých hodnoty na svojich ovládacích prvkoch prepíše. K prepisu však dochádza až vtedy, ak bola hodnota zapísaná (kontrola poslednej zmeny) a to umožňuje ovládanie pracoviska na strane servera.

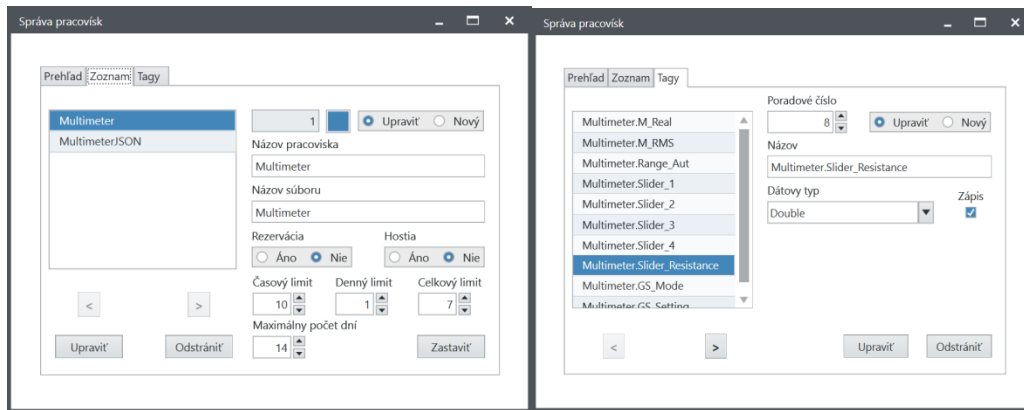


Obr. 63 Zapisovanie hodnôt do tagov

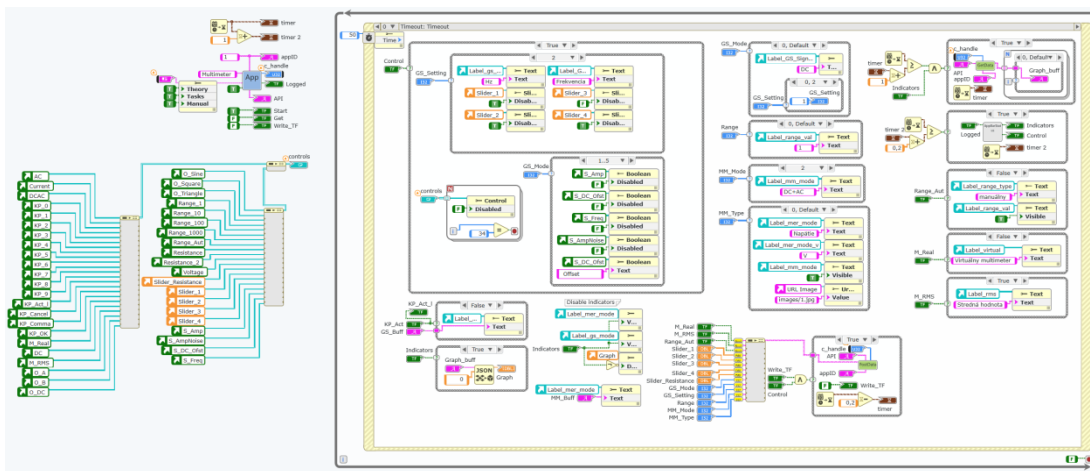


Obr. 64 Čítanie hodnôt tagov vzdialeného panela v pracovisku Multimeter

Na webovej službe som pomocou nástroja na správu pracovísk pridal pracovisko Multimeter a jemu prislúchajúce tagy (Obr. 65).

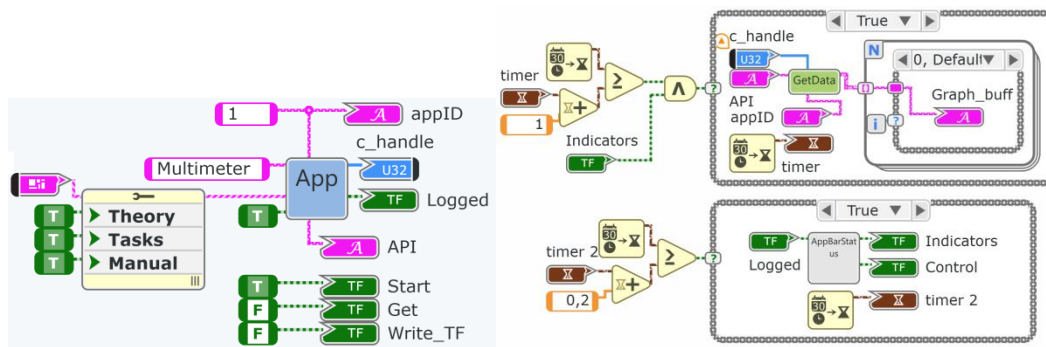


Obr. 65 Správa pracovísk



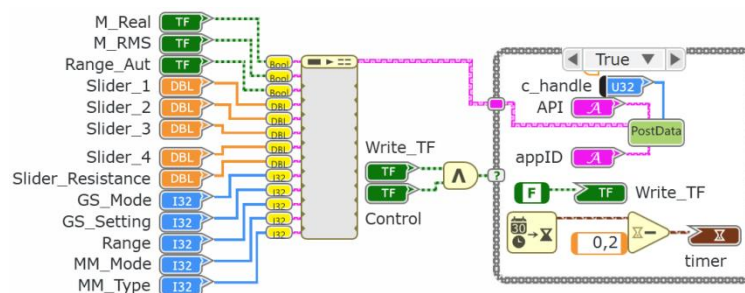
Obr. 66 Blokový diagram vzdialeného panela

Následne som spojzldnil vzdialený panel (Obr. 66). V podprograme „App.gvi“ (Obr. 67 - vľavo) som nastavil titulok stránky, zapol zobrazovanie aplikačnej lišty, uviedol identifikačné číslo pracoviska a nastavil zobrazenie odkazov na stránky s dodatočným obsahom (Teória, Úlohy, Manuál). Vzdialený panel v nastavenom intervale (1 sekunda) posiela požiadavky webovej služby o získanie dát pracoviska a po ich príchode dochádza k ich zobrazeniu (Obr. 67 – vpravo hore). Pomocou podprogramu „AppBarStatus.gvi“ (Obr. 67 – vpravo dole) dochádza k načítaniu informácií o stave a možnosti kontroly z dočasného úložiska prehliadača, ktorý interpretuje panelu pracoviska či je možné pracovisko ovládať alebo zobrazovať dáta panela. Tieto informácie zabezpečuje aplikačná lišta.



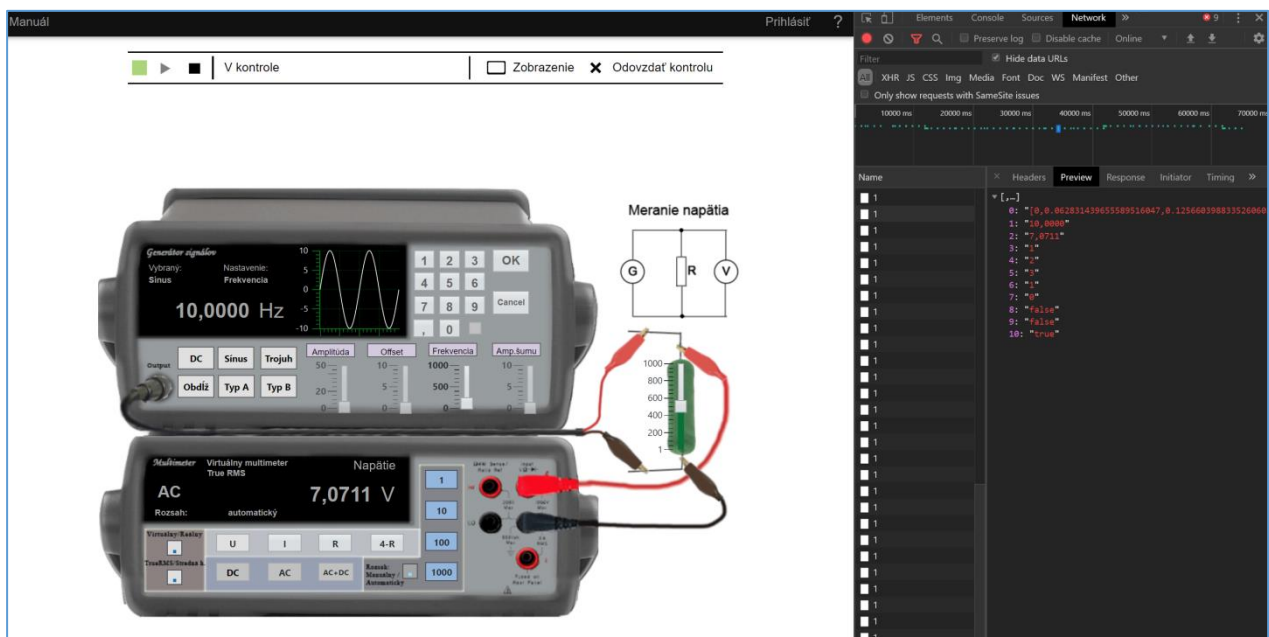
Obr. 67 Časť blokového diagramu

Vzdialený panel posíla (Obr. 68) hodnoty svojich ovládacích prvkov webovej službe, ktorá zabezpečuje zápis do im príslušných tagov. K odoslaniu dochádza až po interakcii používateľom.



Obr. 68 Posielanie hodnôt ovládacích prvkov

Na Obr.69 je možné pozorovať prichádzajúce dáta, ktoré panel načíta a následne zobrazí.



Obr. 69 Pohľad na prenášané hodnoty panela

4.6.1. Štruktúra priečinka vzdialeného pracoviska

Priečinkov vzdialeného panela pracoviska Multimeter sa nachádza v priečinku „lab“ pri webovom rozhraní

- /lab/Multimeter/ – hlavný priečink
- /lab/Multimeter/images/ – priečink obsahuje obrázky
- /lab/Multimeter/Custom.css – súbor obsahuje CSS štýly
- /lab/Multimeter/index.html – vygenerovaný hlavný súbor stránky
- /lab/Multimeter/index.via.txt – vygenerovaný hlavný súbor stránky
- /lab/Multimeter/theory.html – súbor obsahuje podstránku s teóriou
- /lab/Multimeter/manual.html – súbor obsahuje podstránku s manuálom
- /lab/Multimeter/tasks.html – súbor obsahuje podstránku s úlohami

5. Iné možné riešenia

Táto práca sa sústredila na vytvorenie nového riešenia webového rozhrania laboratória merania použitím vývojového prostredia LabVIEW NXG a jeho doplnkom Web Module. Existujú však aj iné možné riešenia.

5.1. Komunikácia

Navrhnuté riešenie využíva na komunikáciu HTTP protokol. Pre zobrazenie dát na vzdialenom paneli musí klient poslať požiadavku webovej službe a tá mu následne vráti odpoveď aj s dátami. Tento proces sa opakuje v určitom intervale a pri viacerých klientoch môže predstavovať záťaž na danú webovú službu ale aj počítač, na ktorom služba beží. Jednou z možností je posielanie dát službou vtedy, ak nastane zmena na pracovisku, bez toho aby klient poslal požiadavku. To si však vyžaduje aby TCP spojenie medzi klientom a službou bolo počas celej doby ovládania alebo pozorovania otvorené.

Možnosťou je zvoliť iný komunikačný protokol ako HTTP. Webové prehliadače a doplnok LabVIEW NXG Web Module podporujú protokol WebSocket. Protokol WebSocket umožňuje nadviazanie dlhodobého TCP spojenia medzi klientom a serverom, čo umožňuje okamžitú obojsmernú komunikáciu, plne duplexných správ. Tento protokol by mohol byť vhodnou alternatívou. Vyžadovalo by si to nový návrh riešenia webovej služby a webového rozhrania.

5.2. Webová služba

Webová služba by mohla byť nahradená klasickým webovým serverom (napr. apache, nginx). Na poskytovanie dynamického obsahu (konkrétneho používateľa) pre webové rozhranie by mohli byť skripty vytvorené v rôznych jazykoch (napr. PHP, ASP, Perl, Python).

Záver

Cieľom tejto práce bolo oboznámiť sa s existujúcim laboratóriom merania a navrhnúť nové riešenie webových stránok a web servera využitím vývojového prostredia LabVIEW NXG a jeho doplnkom LabVIEW NXG Web Module. Prístup na vzdialený panel pracoviska vyžadoval aby klient mal nainštalovaný zásuvný modul. Tento modul však už nie je v moderných prehliadačoch dostupný. Mojim zámerom bolo vytvoriť platformu, kde by bolo možné webové stránky a pracoviská do nového systému jednoducho prerobiť a spravovať. Taktiež aby bolo možné limitovať kontrolu nad pracoviskom a program pracoviska vzdialene spustiť.

Výsledkom tejto práce je nové webové rozhranie laboratória merania a webová služba, ktorá zabezpečuje chod webového rozhrania, spravuje prístup klientov k jednotlivým pracoviskám a obsahuje nástroje na správu pracovísk, rezervácií a klientov. Do tohto systému bolo prerobené jedno pracovisko, ktoré slúži ako príklad. Prístup kvzdialenému panelu pracoviska je možný pomocou webového prehliadača bez nutnosti inštalovania dodatočných doplnkov.

System je možné rozšíriť o ďalšie pracoviská, obmedzením môže byť limitovaný počet typov zobrazovacích prvkov, ktoré web modul ponúka. Ďalším krokom ako systém vylepšiť, by mohlo byť optimalizovanie komunikácie medzi klientom a webovou službou alebo využitie iného komunikačného protokolu.

Pri vypracovaní tejto práce som nadobudol mnoho vedomostí ako napríklad vytvoriť komunikáciu medzi serverom a klientom, HTTP protokole, tvorbe dynamických webových stránok a prácou s ladiacimi nástrojmi.

Zoznam použitej literatúry

- [1]. Getting Started with LabVIEW - National Instruments, [Online]. Dostupné na internete <<http://www.ni.com/pdf/manuals/373427j.pdf>>
- [2]. LABVIEW NXG 4.0 MANUAL, Differences between LabVIEW and LabVIEW NXG [Online]. Dostupné na internete <<https://www.ni.com/documentation/en/labview/4.0/migration/differences-labview-labviewnxg/>>
- [3]. SystemLink Architecture - National Instruments, [Online]. Dostupné na internete <<https://www.ni.com/cs-cz/support/documentation/supplemental/18/systemlink--architecture.html#section-728115818>>
- [4]. Peter Bednár, Aplikácie Webových technológií, [Online]. Dostupné na internete <<http://people.tuke.sk/peter.bednar/awt/prednaska1.pdf>>
- [5]. Representational state transfer, [Online]. Dostupné na internete <https://en.wikipedia.org/wiki/Representational_state_transfer>
- [6]. Uniform Resource Identifier (URI): Generic Syntax, [Online]. Dostupné na internete <<https://tools.ietf.org/html/rfc3986>>
- [7]. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, [Online]. Dostupné na internete <<https://tools.ietf.org/html/rfc7231>>
- [8]. Architecture of the World Wide Web, Volume One, [Online]. Dostupné na internete <<https://www.w3.org/TR/webarch/>>
- [9]. Ing. Michal Halás, PhD., Komunikačné a informačné siete [Online]. Dostupné na internete <<http://www.ut.fei.stuba.sk/~halas/kis/zal%202012/prednaska%2009.pdf>>
- [10]. HTML, Living Standard, [Online]. Dostupné na internete <<https://html.spec.whatwg.org/>>
- [11]. CSS: Cascading Style Sheets, [Online]. Dostupné na internete <<https://developer.mozilla.org/en-US/docs/Web/CSS>>
- [12]. Javascript, [Online]. Dostupné na internete <https://www.tutorialspoint.com/javascript/javascript_tutorial.pdf>
- [13]. Level 1 Document Object Model Specification, [Online]. Dostupné na internete <<https://www.w3.org/TR/WD-DOM/introduction.html>>
- [14]. Ajax (programming), [Online]. Dostupné na internete <[https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))>
- [15]. Introducing JSON, [Online]. Dostupné na internete <<https://www.json.org/json-en.html>>
- [16]. ISBN 9783642080128, Sumathi, S., Esakkirajan, S., Fundamentals of Relational Database Management Systems, 2007
- [17]. Hans-Petter Halvorsen, Introduction to Database Systems, [Online]. Dostupné na internete <[https://www.halvorsen.blog/documents/tutorials/resources/Introduction to Database Systems.pdf](https://www.halvorsen.blog/documents/tutorials/resources/Introduction%20to%20Database%20Systems.pdf)>

- [18]. Hans-Petter Halvorsen, Structured Query Language, [Online]. Dostupné na internete <[https://www.halvorsen.blog/documents/tutorials/resources/Structured Query Language.pdf](https://www.halvorsen.blog/documents/tutorials/resources/Structured_Query_Language.pdf)>
- [19]. Sam Sharp, WebSockets: Bringing LabVIEW to the Web [Online]. Dostupné na internete <https://www.academia.edu/30685376/WebSockets_Bringing_LabVIEW_to_the_Web>

Prílohy

Príloha A: CD médium – diplomová práca v elektronickej podobe, príloha v elektronickej podobe, programy, zdrojové kódy.

Príloha B: Manuál k platforme

Príloha B:

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Manuál k platforme

6. Úvod

Tento manuál je určený pre webovú službu, pre spustenie je potrebné mať minimálne nainštalovaný LabVIEW NXG runtime engine, ideálne LabVIEW NXG spolu s web modulom, MySQL databázu a ODBC ovládač pre MySQL databázu.

Pre úpravu zdrojového kódu je potrebný: LabVIEW NXG a LabVIEW NXG Web Module

Odporúčam do budúcnosti operačný systém Windows 10 64-bitový

LabVIEW NXG will drop support for Windows 7 and Windows Server 2008 R2 starting in 2021. Versions of this product that ship after May 1, 2021, will not install or run on these operating systems.

<http://www.ni.com/pdf/manuals/376808g.html>

7. Inštalácia a spustenie

Manuál nezahŕňa inštaláciu a nastavenie databázy. Odporúčam stiahnuť a nainštalovať najnovšiu MySQL databázu. (<https://dev.mysql.com/downloads/installer/>)

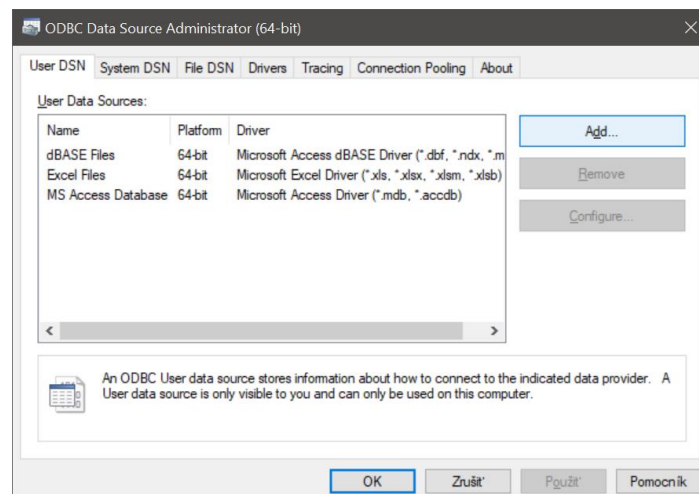
Po nainštalovaní databázy je potrebné vytvoriť databázu, napr. s názvom lab

Následne je potrebné stiahnuť ODBC ovládač pre MySQL (ODBC Connector). (<https://dev.mysql.com/downloads/connector/odbc/>)

Po inštalácii ODBC ovládača, je potrebné spustiť aplikáciu na správu zdrojov ODBC a to nasledovne.

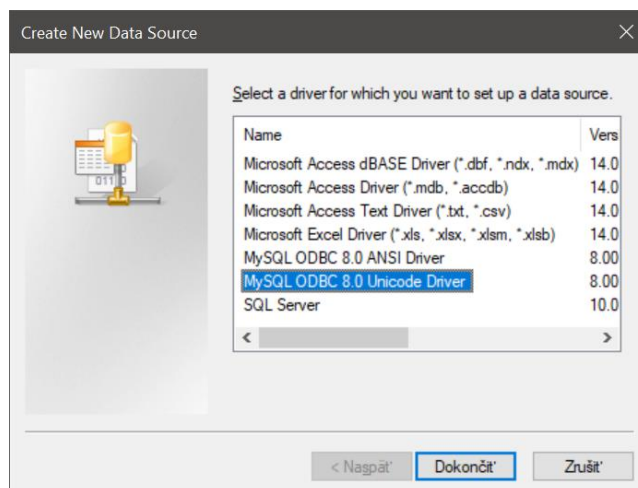
Štart → napísať ODBC → vybrať 64 bitovú

Po spustení okna, kliknúť na **Add**(pridať) v záložke User DSN



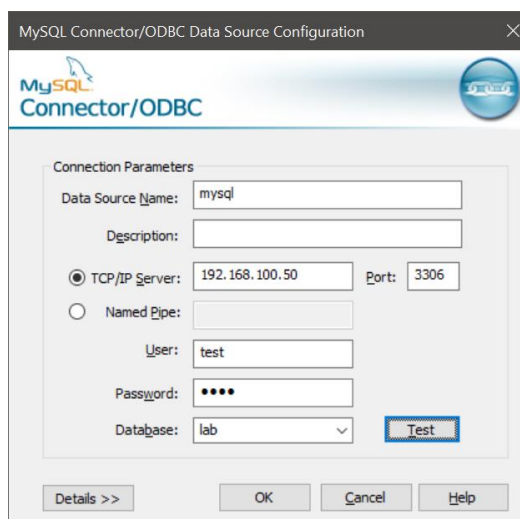
Obr. 70 Okno ODBC

Otvorí sa okno **Create New Data Source**. Následne je potrebné vybrať nainštalovaný ovládač (Unicode)



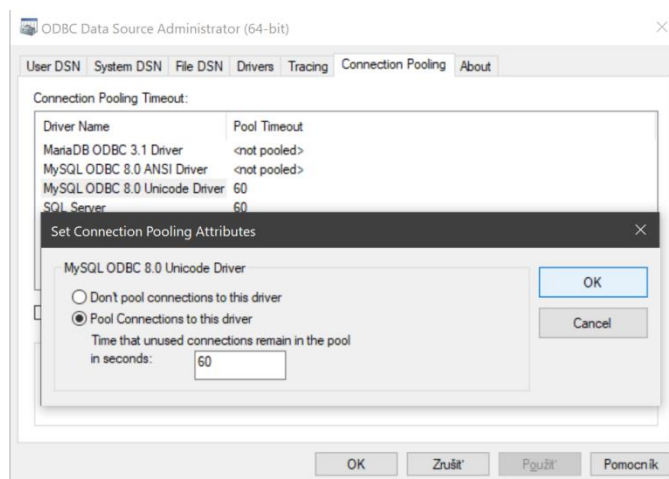
Obr. 71 Pridať nový zdroj

Otvorí sa okno **MySQL Connector/ODBC**. Potrebne je vyplniť názov zdroja (**Data Source Name**) napr. mysql, Následne IP adresa, potom meno, heslo a názov databázy. Pripojenie je možné otestovať pomocou tlačidla **Test**.



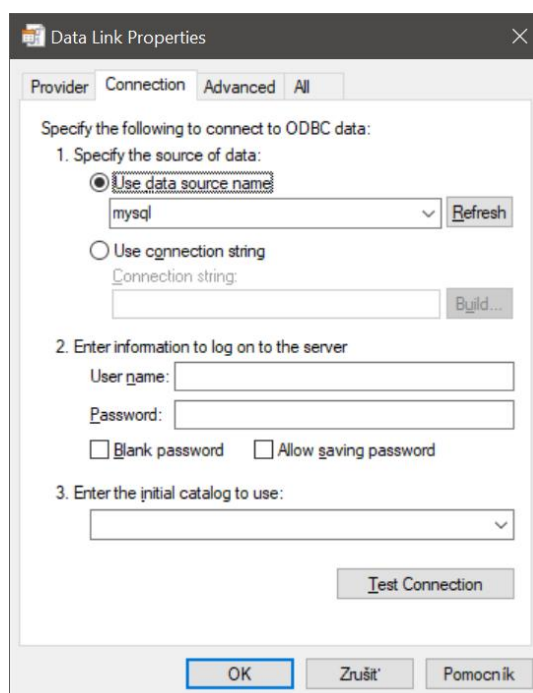
Obr. 72 Nastavenie údajov

Po potvrdení, je potrebné následne ísť do záložky **Connection Pooling**, vybrať ovládač MySQL Unicode Driver a po dvojkliku sa otvorí okno, kde je potrebné zapnúť **Pool Connections** a potvrdiť.



Obr. 73 Nastavenie connection pooling

Potom je potrebné vytvoriť **UDL** (.udl) súbor. Ten vytvoríme tak, že v Prieskumníkovi vytvoríme nový textový súbor. Následne súbor premenujeme na napr. DB.udl. Na to je potrebné mať zapnuté zobrazovanie prípon názvov súborov (Hore v Prieskumník → Zobrazíť → zaškrtnúť Prípomy názvov súborov). Po premenovaní súboru ho dvojklikom otvoríme. Otvorí sa okno **Data Link Properties**. Ďalej je potrebné vybrať názov zdroja (ktorý bol vytvorený v predchádzajúcich krokoch) a potvrdiť. Opäť je možné otestovať spojenie pomocou Test Connection.



Obr. 74 Okno UDL súboru

Súbor následne potrebné premiestniť na vhodné miesto, napr. do adresára k webovej službe.

NI Web Server

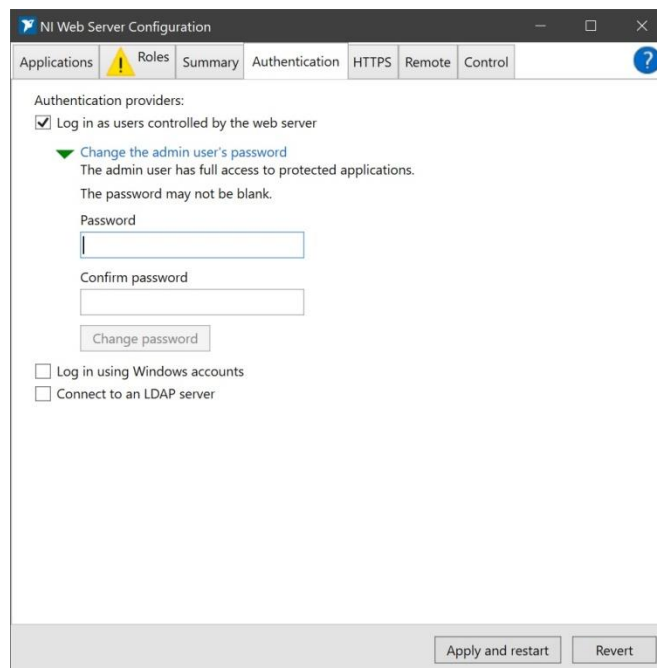
Cesta pre publikovanie stránky:

C:\Program Files\National Instruments\Shared\Web Server\htdocs

Nastavenie NI Web Servera

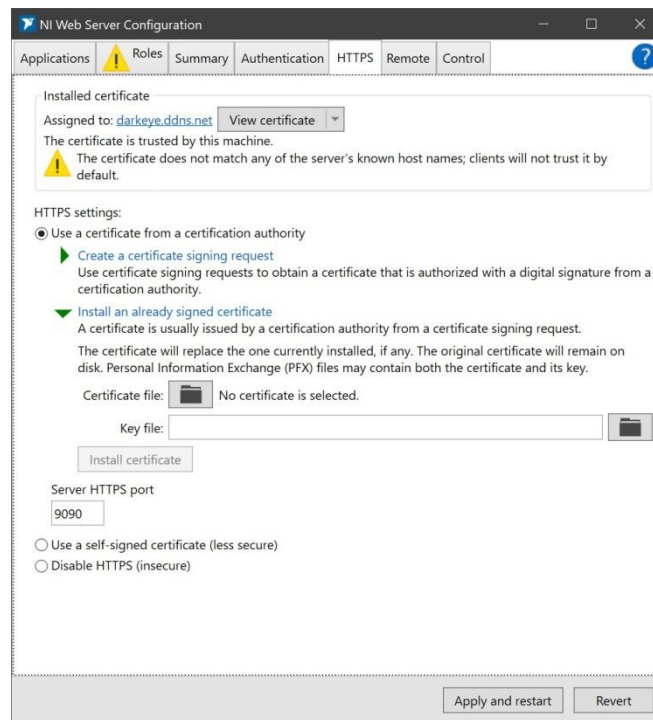
Štart → napísať NI Web Server Configuration → vybrať

Otvorí sa okno **NI Web Server Configuration**. V záložke **Authentication** vhodné nastaviť heslo pre používateľa „admin“ – administrátora



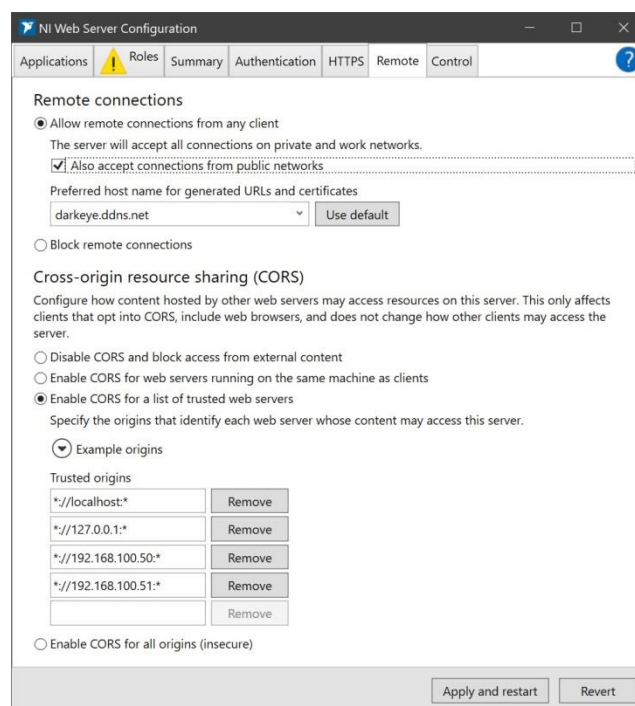
Obr. 75 Nastavenie hesla

V záložke **HTTPS** je možné špecifikovať port na ktorom server bude spustený. Taktiež je možné prepnúť na bezpečné pripojenie a pridať certifikát spolu s kľúčom.



Obr. 76 Záložka HTTPS

Ak je potrebné, v záložke **Remote** možné nastaviť kontrolu pôvodu zdroja požiadavky (**CORS**)



Obr. 77 Záložka Remote

SystemLink server má webové rozhranie na adrese 127.0.0.1:port , po prihlásení je k dispozícii Tag Viewer

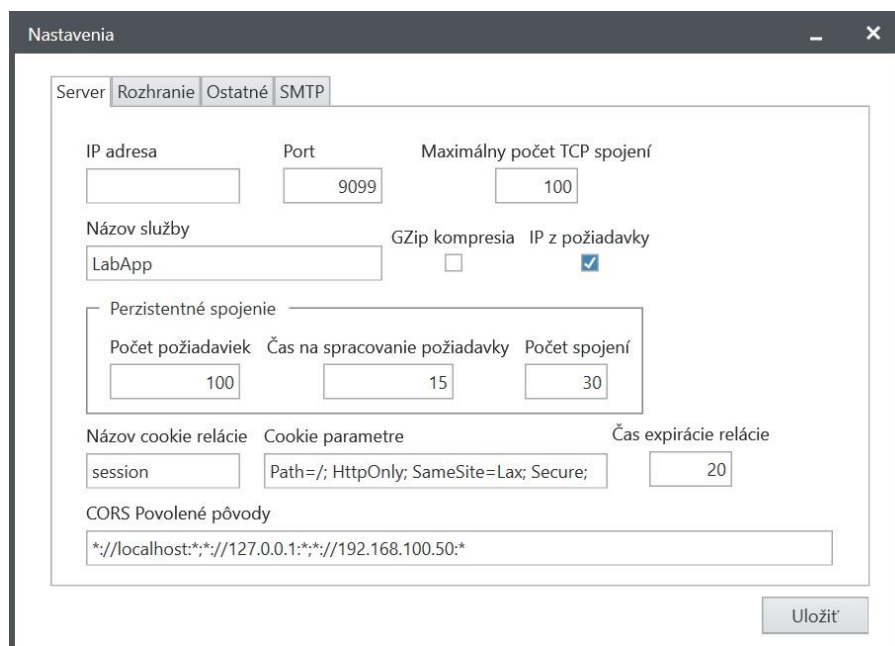
Pred spustením služby „LabApp.exe“ je potrebné v konfiguračnom súbore **config.ini** nastaviť:

1. Cestu k .udl súboru
2. SystemLink URL adresu (http:// ipadresa : port, alebo https://) (<http://127.0.0.1:9090>)
3. Meno a heslo pre SystemLink server (ak nebol vytvorený iný používateľ, štandardne je admin)

8. Konfigurácia webovej služby

V tejto časti sú opísané jednotlivé parametre webovej služby, ktoré je možné nastaviť.

8.1. Záložka – Server



Obr. 78 Okno nastavenia – záložka Server

IP Adresa - IP adresa sieťového rozhrania, na ktorej bude server prijímať požiadavky. Pri prázdnej hodnote prijíma na predvolenom sieťovom rozhraní. To znamená, že pri nastavení konkrétnej IP adresy prijíma len na nastavenej adrese. Vhodné nechať prázdne ak má byť prístupná aj na lokálnej adrese (127.0.0.1, 192.168.xxx.xxx a pod.) aj na verejnej.

Port – Možné špecifikovať konkrétny port, na ktorom bude server prijímať požiadavky.

Maximálny počet TCP spojení (Predvolená hodnota - 100) – maximálny počet otvorených TCP spojení je možné limitovať. Hodnota závisí na počte klientov, ktorý službu využívajú, ale aj od hardvérových parametrov počítača na ktorom server beží. Po prekročení neprijíma ďalších klientov. Zo zmenou hodnoty je vhodné upraviť aj maximálny počet spojení v MySQL databáze. SQL príkaz na kontrolu počtu maximálny spojení a zmenu:

```
show variables like "max_connections";  
set global max_connections = 200;
```

Názov služby (Predvolená hodnota - LabApp) – hodnota zodpovedá hodnote vHTTP hlavičke „Server“ pri odpovedi servera

GZip kompresia (Predvolená hodnota - false) – po zapnutí server každú odpoveď komprimuje. Vhodné nastaviť. Ďalšia možnosť je prenechať kompresiu na proxy server – ak sa používa.

IP z požiadavky (Predvolená hodnota - false) – Použije sa IP adresa nachádzajúca sa v hlavičke požiadavky (X-Real-IP). Vhodné nastaviť ak je použitý proxy server pred webovou službou.

Perzistenté spojenie (Keep-Alive) HTTP

- **Počet požiadaviek** (Predvolená hodnota - 100) – maximálny počet požiadaviek v rámci jedného perzistentného spojenia (Keep-Alive Max Requests), po prekročení hranice sa spojenie uzavrie
- **Čas na spracovanie požiadavky** (Predvolená hodnota – 15 sekúnd) - čas na spracovanie požiadavky v sekundách v rámci perzistentného spojenia (Keep-Alive Timeout). Platí aj všeobecne ako Timeout pre funkciu TCP read.
- **Počet spojení** (Predvolená hodnota - 30) - maximálny počet aktívnych perzistentných spojení (Keep-Alive Max Connections). Podobne ako počet TCP spojení, po prekročení sa však požiadavky prijímajú ale spojenie sa hneď uzavrie a neostáva otvorené.

Názov cookie relácie (Predvolená hodnota - session) – možné špecifikovať názov cookie pre uschovanie identifikátora relácie

Cookie parametre (Predvolená hodnota - Path=/; HttpOnly; SameSite=Lax;)– možné špecifikovať parametre cookie relácie. Pri použití SSL terminácie (https), potrebné pridať ďalší parameter "Secure;".

Čas expirácie relácie (Predvolená hodnota – 20 minút) – možné špecifikovať hodnotu času v minútach, po ktorej sa relácia bude považovať za expirovanú

CORS Povolené pôvody (Predvolená hodnota – *://localhost:*;*://127.0.0.1:*)– kontrola pôvodu zdroja požiadavky na základe hodnoty v HTTP hlavičke „Origins“. Pôvody je možné špecifikovať v nasledujúcom formáte: **schéma://host:port**- jednotlivé pôvody je potrebné oddeliť bodkočiarkou. Ako schému možno vybrať len http/https. Použitie znaku „*“ označuje všetky špecifikované pôvody za platné.

Príklady:

*://doména.sk – požiadavky z http://doména.sk alebo https://doména.sk budú spracované

http://ipadresa:* - požiadavky budú spracované nezávisle od toho aký je port zdroja

:/:* - povolí požiadavky zo všetkých pôvodov http:// a https://

8.1.1. Záložka - Rozhranie

The screenshot shows a window titled "Nastavenia" with four tabs: "Server", "Rozhranie", "Ostatné", and "SMTP". The "Rozhranie" tab is selected. The window contains three main sections:

- Prihlasovanie**: "Maximálny počet nesprávnych prihlásení" (5) and "Čas blokovania" (300).
- Kontrola pracovísk**: "Čas kontroly" (1), "Ukončenie ovládania" (15), and "Ovládanie spúšťania" (checked).
- Používatelia**: "Zmena údajov" (checked) and "Zmena hesla" (checked).

An "Uložiť" button is located at the bottom right of the window.

Obr. 79 Okno nastavenia – záložka Rozhranie

Prihlasovanie

- **Maximálny počet nesprávnych prihlásení** (Predvolená hodnota – 5) - maximálny počet neprávnych prihlásení po ktorom dôjde k zablokovaniu možnosti prihlásenia na určitý časový interval
- **Čas blokovania** (Predvolená hodnota – 300 sekúnd) - časový interval zablokovania možnosti prihlásenia v sekundách. Platí aj ako interval, počas ktorého sa deteguje počet nesprávnych prihlásení.

Kontrola pracovísk

- **Čas kontroly** (Predvolená hodnota – 300 sekúnd) - časový interval v minútach, počas ktorého môže užívateľ pracoviska ovládať pracovisko ak sa nachádza požiadavka o ovládanie v zozname požiadaviek o kontrolu
- **Ukončenie ovládania** (Predvolená hodnota – 15 sekúnd) - časový interval v sekundách, v ktorom sa deteguje či sa aplikácia používa alebo nie (posledná akcia pracoviska - posledná akcia používateľa pracoviska). Vhodné nastaviť čo najnižšiu prípustnú hodnotu.

- **Ovládanie spúšťania** (Predvolená hodnota – true) - povolí spúšťanie a vypínania aplikácie pracoviska používateľom

Používatelia

- Zmena údajov (Predvolená hodnota – true) - povolenie zmeny údajov (profilu) pre všetkých používateľov
- Zmena hesla (Predvolená hodnota – true) - povolenie zmeny hesla pre všetkých používateľov

8.1.2. Záložka - Ostatné

Obr. 80 Okno nastavenia – záložka Ostatné

UDL súbor databázy – cesta k .udl súboru pre možnosť pripojenia k databáze

SystemLink

- **URL** – adresa URL SystemLink servera (schéma://ipadresa:port)
- **Prihlasovacie meno** – prihlasovacie meno na SystemLink server
- **Heslo** – heslo používateľa na SystemLink server

PBKDF2 hašovací funkcia

- **Veľkosť salt** (Predvolená hodnota – 16 byte) – veľkosť náhodného reťazca hašovacej funkcie v byte-och

- **Počet iterácií** (Predvolená hodnota – 10000) –počet iterácii hašovacej funkcie

Záznamy (log files)

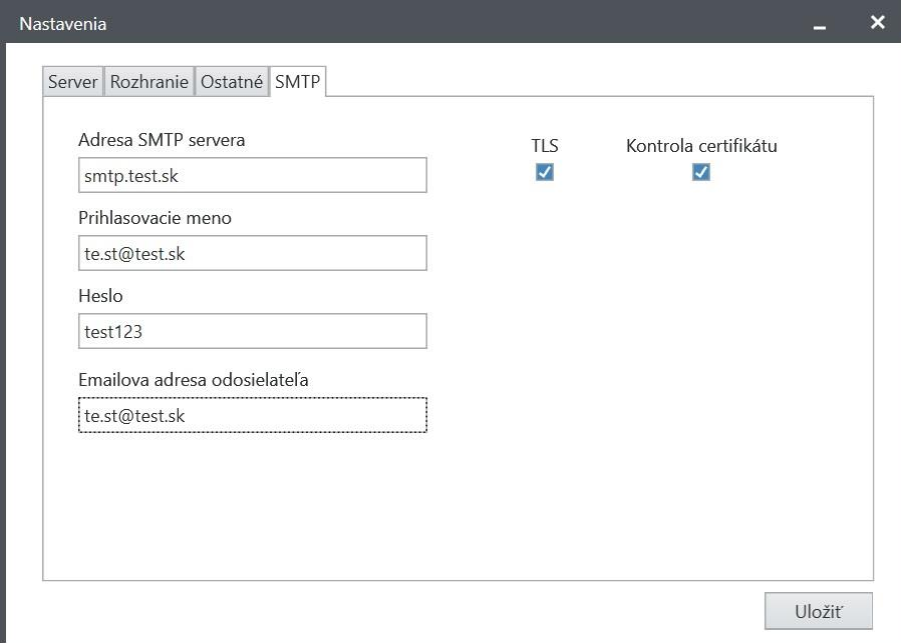
- **Zapnuté** (Predvolená hodnota – true) - zapnutie zápisu požiadaviek do záznamového súboru access.log v priečinku logs
- **Úroveň** (Predvolená hodnota – 2) – určuje, ktoré požiadavky sa vynechajú
 - 1 - žiadne požiadavky
 - 2 - požiadavky /app/data a /app/control/info
- **Veľkosť súboru** (Predvolená hodnota – 5000 KB) - hraničná veľkosť súboru záznamov požiadaviek v KB (kilobyte), po ktorej sa komprimuje a zálohuje. Kontrola prebieha raz za 5 minút.
- **Formát** - formát záznamov. Množnosť určiť, čo bude záznam požiadavky obsahovať.

Dostupné premenné:

- {{ip}} - IP adresa klienta
- {{referer}} - hodnota hlavičky Referer v požiadavke klienta
- {{user-agent}} - hodnota hlavičky User-Agent v požiadavke klienta
- {{time}} - čas a dátum požiadavky klienta
- {{reqline}} - požiadavka klienta
- {{httpcode}} –HTTP stavový kód odpovede na požiadavku klienta
- {{cntlen}} – dĺžka (veľkosť) odpovede na požiadavku klienta
- {{uid}} - Identifikačné číslo používateľa

8.1.3. Záložka – SMTP

V prípade, ak sa využíva funkcia zasielania mailov, je potrebné nastaviť mailový server a ďalšie potrebné údaje. Nastavenia je vhodné pred používaním otestovať.



The screenshot shows a window titled "Nastavenia" with a tabbed interface. The "SMTP" tab is active. The form contains the following fields and controls:

- Adresa SMTP servera:** Input field containing "smtp.test.sk".
- Prihlasovacie meno:** Input field containing "te.st@test.sk".
- Heslo:** Input field containing "test123".
- Emailova adresa odosielateľa:** Input field containing "te.st@test.sk".
- TLS:** Checkmark is checked.
- Kontrola certifikátu:** Checkmark is checked.
- Uložiť:** Button at the bottom right.

Obr. 81 Okno nastavenia – záložka SMTP

Adresa SMTP servera - adresa mailoveho (SMTP) servera (adresa:port). Port niekedy nie je nutnéšpecifikovať.

Prihlasovacie meno – prihlasovacie meno na SMTP server, väčšinou sa jedná o e-mailovú adresu

Heslo - heslo používateľa

Emailova adresa z ktorej sa email odošle – e-mailová adresa

TLS- použitie bezpečnej transportnej vrstvy pri komunikácii so SMTP serverom

Kontrola certifikátu – možnosť zapnutia kontroly certifikátu SMTP servera u certifikačnej autority

9. Aplikačné rozhranie webovej služby

V tejto časti sa nachádzajú informácie o aplikačnom rozhraní webovej služby. Na to aby server vrátil odpoveď, je ako prvé potrebné vytvoriť reláciu zaslaním požiadavky o vytvorenie relácie a následne s každou požiadavkou poslať identifikátor relácie (ako HTTP cookie). Všeobecné pravidlá:

- všetky GET požiadavky vrátia odpoveď v JSON formáte
- všetky zasielané POST požiadavky musia byť v JSON formáte

Možné požiadavky sú rozdelené do týchto kategórií:

9.1. Relácia

- **Vytvorenie a kontrola relácie**
 - Metóda: **POST**
 - URI: **/session**
 - Odpoveď – stavový kód:
 - **200** – relácia existuje a je platná
 - **201** – relácia bola úspešne vytvorená, vráti identifikátor relácie ako cookie

9.2. Pracovisko

- **Informácia o pracovisku** – slúži na zistenie informácií o pracovisku
 - Metóda: **GET**
 - URI: **/app/info/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
 - Odpoveď – stavový kód:
 - **200** – vráti odpoveď s údajmi
 - **404** – neexistuje

Vzor odpovede:

```
{
  "state": "1",           //stav pracoviska, spustené, zastavené
  "guest": "1",          //možnosť kontroly bez prihlásenia
  "reservation": "1",    //možnosť vytvárať rezervácie
  "limits": "10,1,7,14", //limity rezervácie
  "ctl": "1"             //možnosť spúšťania/zastavenia pracoviska
}
```

- **Spustenie pracoviska**
 - Metóda: **POST**

- URI: **/app/start/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
- Odpoveď – stavový kód:
 - **200** – pracovisko bolo úspešne spustené
 - **403** – neoprávnená požiadavka, neprihlásený, rezervované
 - **404** – neexistuje alebo spustená
 - **500** – chyba pri spúšťaní

- **Zastavenie pracoviska**
 - Metóda: **POST**
 - URI: **/app/stop/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
 - Odpoveď – stavový kód:
 - **200** – pracovisko bolo úspešne zastavené
 - **403** – neoprávnená požiadavka, neprihlásený, rezervované
 - **404** – neexistuje alebo zastavená
 - **500** – chyba pri zastavovaní

- **Poslanie dát pracoviska** – slúži na poslanie dát pre panela pracoviska, požiadavka pozostáva z hodnôt, formát - reťazce znakov v poli v definovanom poradí
 - Metóda: **POST**
 - URI: **/app/data/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
 - Vzor požiadavky: **["hodnota","hodnota","hodnota", ...]**
 - Odpoveď – stavový kód:
 - **200** – dáta boli úspešne poslané a zapísané
 - **403** – neprihlásený, rezervované, neoprávnená požiadavka
 - **404** – neexistuje alebo zastavená

- **Získanie dát pracoviska** – slúži na získanie dát panela pracoviska, požiadavka vráti pole hodnôtusporiadaných v definovanom poradí
 - Metóda: **GET**
 - URI: **/app/data/{appid}**

- atribút – appid (identifikačné číslo pracoviska)
- Odpoveď – stavový kód:
 - **200** – dáta boli úspešne načítané
 - **403** – neprihlásený, rezervované, neoprávnená požiadavka, zobrazenie nepovolené
 - **404** – neexistuje alebo zastavená

Vzor platnej odpovede:

```
[
  "1,0000",
  "0,7071",
  "1",
  "1",
  "false",
  "true"
] //pole hodnôt
```

- **Zobrazenie pracoviska**—slúži na prepnutie zobrazenia dát panela, zobrazovať dáta môže len klient, ktorý je v kontrole pracoviska
 - Metóda: **POST**
 - URI: **/app/control/view/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
 - Odpoveď – stavový kód:
 - **200** – zaradený do fronty
 - **403** – neoprávnená požiadavka
 - **404** – neexistuje alebo zastavená

- **Požiadavka o kontrolu pracoviska**
 - Metóda: **POST**
 - URI: **/app/control/request/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
 - Odpoveď – stavový kód:
 - **200** – zaradený do fronty
 - **403** – neprihlásený, rezervované, neoprávnená požiadavka, už v kontrole, už požiadané
 - **404** – neexistuje alebo zastavená

- **Odobzanie kontroly pracoviska**
 - Metóda: **POST**
 - URI: **/app/control/release/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
 - Odpoveď – stavový kód:
 - **200** – úspešné odobzanie kontroly
 - **403** – neoprávnená požiadavka
 - **404** – neexistuje alebo zastavená

- **Informácia o stave kontroly** – slúži na zistenie stavu kontroly klienta, zároveň iniciuje kontrolu požiadaviek, potrebné zasielanie v pravidelnom intervale
 - Metóda: **GET**
 - URI: **/app/control/info/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
 - Odpoveď – stavový kód:
 - **200** – vráti odpoveď s údajmi
 - **403** – neprihlásený
 - **404** – neexistuje alebo zastavená

Vzor odpovede:

```

{
  "master": "0",           //Možnosť ovládania
  "reserved": "0",       //Rezervované
  "view": "1",           //Zobrazovanie
  "queue": "0",         //Počet požiadaviek v fronte
  "end": "0",           //Čas ukončenia kontroly (unix timestamp v sekundách )
  "limit": "1",         //Maximálny čas kontroly (v minútach)
  "requested": "0",     //Zaslaná žiadosť o kontrolu
  "position": "0",     //Poradie v fronte
  "used": "0"           //Používané
}

```

9.3. Používateľ

- **Prihlásenie používateľa**- slúži na prihlásenie používateľa, požiadavka pozostáva z e-mailovej adresy (maximálna dĺžka 255 znakov) používateľa a hesla (maximálna dĺžka 128 znakov), formát - reťazce znakov v poli
 - Metóda: **POST**
 - URI: **/user/login**
 - Vzor požiadavky: **["jan.novak@gmail.com","mrkvicka22"]**

- Odpoveď – stavový kód:
 - **200** – používateľ bol úspešne prihlásený, vráti nový identifikátor relácie ako cookie
 - **403**– nesprávne údaje, zablokované alebo neoprávnená požiadavka
- **Odhlásenie používateľa**– slúžni na odhlásenie prihláseného používateľa
 - Metóda: **POST**
 - URI: **/user/logout**
 - Odpoveď – stavový kód:
 - **200** – používateľ bol úspešne odhlásený, vymaže identifikátor relácie, potrebné znova požiadať
 - **403**– neprihlásený
- **Kontrolapoužívateľa a možnosti** – slúži na kontrolu, či je používateľ prihlásený a vrátenie informácie o možnosti zmeny hesla alebo úpravy profilu
 - Metóda: **GET**
 - URI: **/user/info**
 - Odpoveď – stavový kód:
 - **200** – vráti odpoveď s údajmi
 - **403**– neprihlásený

Vzor odpovede:

```
{
  "uid": 1, //Identifikačné číslo používateľa
  "email": "jan.novak@gmail.com", //E-mail používateľa
  "passchange": true, //Možnosť zmeny hesla
  "datachange": true //Možnosť úpravy profilu
}
```

- **Údajepoužívateľa (profil)**– slúži na získanie údajov o používatelovi (profil), používateľ musí byť prihlásený
 - Metóda: **GET**
 - URI: **/user/data**
 - Odpoveď – stavový kód:
 - **200** – vráti odpoveď s údajmi
 - **403**– neprihlásený

Vzor odpovede:

```
{
  "fname": "Ján",           //Meno
  "lname": "Novák",        //Priezvisko
  "year": "1",             //Ročník
  "study": "elektronika"  //Študijný program
}
```

- **Zmena údajov používateľa (profil)**– slúži na zmenu údajov používateľa (profil), používateľ musí byť prihlásený, požiadavka pozostáva z mena (maximálna dĺžka 128 znakov), priezviska (maximálna dĺžka 128 znakov), ročníka (platné hodnoty od 1 do 5) a študijného programu (maximálna dĺžka 128 znakov), formát – reťazce znakov v poli
 - Metóda: **POST**
 - URI: **/user/data**
 - Vzor požiadavky: **["Ján","Novák","2","elektrotechnika"]**
 - Odpoveď – stavový kód:
 - **200** – údaje boli úspešne zmenené
 - **403**– neprihlásený, nesprávne údaje alebo neoprávnená požiadavka
- **Zmena hesla používateľa** – slúži na zmenu hesla používateľa, používateľ musí byť prihlásený, požiadavka pozostáva z pôvodného hesla (maximálna dĺžka 128 znakov) a nového hesla (minimálna dĺžka 8 znakov, maximálna dĺžka 128 znakov), formát – reťazce znakov v poli
 - Metóda: **POST**
 - URI: **/user/password**
 - Vzor požiadavky: **["stareheslo","noveheslo","noveheslo"]**
 - Odpoveď – stavový kód:
 - **200** – heslo bolo úspešne zmenené
 - **403**– neprihlásený, nesprávne údaje alebo neoprávnená požiadavka

9.4. Rezervácie

- **Vytvorenie rezervácie** – slúži na vytvorenie rezervácie používateľom, požiadavka pozostáva z dátumu a času začiatku a konca rezervácie, formát (YYYY-MM-DD HH-MM)
 - Metóda: **POST**

- URI: **/reservations/add/{appid}**
 - atribút – appid (identifikačné číslo pracoviska)
- Vzor požiadavky: **{"start":"2020-03-19 22:40", "end":"2020-03-19 22:50"}**
- Odpoveď – stavový kód:
 - **200** – rezervácia bola úspešne vytvorená
 - **403** – neprihlásený,
 - **400** – požiadavka v nesprávnom formáte, neexistuje alebo rezervácia vypnutá, chybný dátum alebo čas, prekročený interval dátumu rezervácie, prekročený maximálny časový interval, dosiahnutý denný limit, dosiahnutý celkový limit, kolízia dátumu

- **Odstránenie rezervácie** – slúži na odstránenie rezervácie používateľa, odstrániť je možné rezerváciu, ktorá ešte nezačala
 - Metóda: **POST**
 - URI: **/reservations/delete/{id}**
 - atribút – id (identifikačné číslo rezervácie)
 - Odpoveď – stavový kód:
 - **200** – rezervácia bola úspešne odstránená
 - **403** – neprihlásený, neoprávnená požiadavka

- **Zoznam rezervácií používateľa** – slúži na získanie zoznamu rezervácií používateľa, vráti maximálne 8 rezervácií
 - Metóda: **GET**
 - URI: **/reservations/user/{offset}**
 - atribút – offset (ofset) – realizuje posun výberu (číselná hodnota)
 - Odpoveď – stavový kód:
 - **200** – vráti odpoveď s zoznamom
 - **403** – neprihlásený

Vzor odpovede:

```

{
  "count": "2",           //celkový počet rezervácií používateľa
  "list": [             //zoznam rezervácií
    {
      "id": "68",       //identifikačné číslo rezervácie
    }
  ]
}

```

```

        "appid": "1",
        "appname": "Multimeter",
        "start": "1588840200",
        "end": "1588840800"
    },
    {
        "id": "69",
        "appid": "2",
        "appname": "MultimeterJSON",
        "start": "1589478000",
        "end": "1589478600"
    }
]
}

```

//identifikačné číslo pracoviska
//názov pracoviska
//začiatok (unix timestamp v sekundách)
//koniec(unix timestamp v sekundách)

- **Zoznam rezervácií pracoviska (dátum)**– slúži na získanie zoznamu rezervácií pracoviska na konkrétny dátum
 - Metóda: **GET**
 - URI: **/reservations/app/{appid}/{date}**
 - atribút – appid (identifikačné číslo pracoviska)
 - atribút – date (dátum) – formát YYYY-MM-DD
 - Odpoveď – stavový kód:
 - **200** – vráti odpoveď s zoznamom
 - **404** - neexistuje
 - **403**– neprihlásený
 - **400** - požiadavka v nesprávnom formáte

Vzor odpovede:

```

[
  {
    "start": "1588840200",
    "end": "1588840800"
  }
]

```

//zoznam rezervácií
//začiatok (unix timestamp v sekundách)
//koniec (unix timestamp v sekundách)

- **Zoznam pracovísk**– slúži na získanie zoznamu pracoviska, na ktoré je možné vytvoriť rezerváciu
 - Metóda: **GET**
 - URI: **/reservations/applist/**
 - Odpoveď – stavový kód:
 - **200** – vráti odpoveď s zoznamom
 - **403**– neprihlásený

Vzor platnej odpovede:

```
[ //zoznam pracovísk
  { //identifikačné číslo pracoviska
    "appid": 1, //názov pracoviska
    "appname": "Multimeter"
  },
  {
    "appid": 2,
    "appname": "MultimeterJSON"
  }
]
```

9.5. Zoznam chybových správ

Formát chybovej správy je nasledovný:

```
{
  "code": "100", //kód chybovej správy
  "message": "Neexistuje" //obsah chybovej správy
}
```

Chybové správy sú rozdelené do týchto kategórii:

9.5.1. Všeobecné

- **100** - Neexistuje
- **101** - Neexistuje alebo zastavená
- **102** - Neexistuje alebo rezervácia vypnutá
- **110**– Neprihlásený
- **111** - Neoprávnená požiadavka
- **112** - Neoprávnená požiadavka, zobrazenie nepovolené
- **113** - Požiadavka v nesprávnom formáte
- **114**– Rezervované

9.5.2. Kontrola

- **120** - Už v kontrole
- **121** - Už požiadané
- **122** - Existujúca požiadavka v poradí

9.5.3. Rezervácia

- **130** - Chybný dátum alebo čas
- **131** - Prekročený interval dátumu rezervácie
- **132** - Prekročený maximálny časový interval
- **133** - Dosiahnutý denný limit
- **134** - Dosiahnutý celkový limit

- **135** - Kolízia dátumu

9.5.4. Prihlasovanie

- **140** - Nesprávne údaje
- **141**– Zablokované

9.5.5. Pracovisko

- **150** - Chyba pri zastavovaní
- **151** - Chyba pri spúšťaní

9.5.6. Relácie

- **160** - Cookie relácie neexistuje
- **161** - Relácia neexistuje
- **162** - Relácia neplatná