**Střední průmyslová škola a Vyšší odborná škola, Písek,**

**Karla Čapka 402**

**397 11 Písek**

**Školní rok: 2017/2018**

**Obor vzdělání: 18-20-M/01 Informační technologie**

**Maturitní práce**

**Mobilní aplikace pro ovládání chytré domácnosti**

Téma číslo: 13

Jméno žáka: Tomáš Linhart

Třída:B4.I

Vedoucí práce: Mgr. Jan Turoň

# Čestné prohlášení

Prohlašuji, že jsem maturitní práci na téma „Mobilní aplikace pro ovládání chytré domácnosti“ vypracoval samostatně a s použitím uvedené literatury a pramenů.

V Písku, dne 20.03.2018 ………………….…………….

Tomáš Linhart

# Abstrakt

# Klíčová slova

# Title

# Abstract

# Keywords

# Obsah

Table of Contents

[Úvod 7](#_Toc509384060)

[Databázový systém 7](#_Toc509384061)

[Node.js 7](#_Toc509384062)

[Socket.IO 7](#_Toc509384063)

[Android 7](#_Toc509384064)

[HTML 8](#_Toc509384065)

[Javascript 8](#_Toc509384066)

[jQuery 8](#_Toc509384067)

[NodeMCU 8](#_Toc509384068)

[1. Instalace programů potřebných pro vývoj aplikace 10](#_Toc509384069)

[Vývojové prostředí 10](#_Toc509384070)

[Instalace programu Atom a nadstavby Platformio 10](#_Toc509384071)

[Vytvoření projektu pro vývoj aplikace pro NodeMCU 10](#_Toc509384072)

[Instalace programu Android Studio 10](#_Toc509384073)

[2. Databáze pro ukládání dat 11](#_Toc509384074)

[Omezení vkládaných dat 11](#_Toc509384075)

[3. Webové rozhraní 12](#_Toc509384076)

[Funkčnost stránky a načítání dat 12](#_Toc509384077)

[Registrace uživatele 12](#_Toc509384078)

[Přihlášení uživatele 12](#_Toc509384079)

[Registrace nového zařízení 13](#_Toc509384080)

[Seznam zařízení 13](#_Toc509384081)

[Seznam funkcí 13](#_Toc509384082)

[Správa oprávnění 13](#_Toc509384083)

[Nastavení zařízení 14](#_Toc509384084)

[Ovládací panel 14](#_Toc509384085)

[Historie 14](#_Toc509384086)

# Poděkování

Mé poděkování patří Mgr. Janu Turoňovi za odborné vedení, trpělivost a ochotu,   
kterou mi v průběhu zpracování této maturitní práce věnoval.

# Úvod

## Databázový systém

Pro ukládání dat z chytrých zařízení bylo použito databázového softwaru Microsoft SQL Server 2017. Tento software se vyznačuje kompatibilitou se servery od Microsoftu. Jeho výhodou je část zvaná SQL Agent, ve které lze snadno vytvořit automatizované úlohy pro zálohování dat. Pro použití mimo produkční prostředí ve verzi Developer je tento systém zdarma.

## Node.js

Node.js je serverový systém navržený pro psaní aplikací na serveru. Aplikace pro node.js jsou psané v jazyce Javascript. Vyznačuje se vysokou škálovatelností aplikací a pro maximalizaci výkonu hojně využívá asynchronního programování. Node.js lze využít pro vytvoření webového serveru.

## Socket.IO

Socket.IO je open-source knihovna pro použití v real-time aplikacích psaných pro systém Node.js. Tato aplikace je schopna real-timové komunikace mezi serverem a mnoha dalšími zařízeními. Pro zpracování zpráv obdržených od klientů využívá asynchronních volání. Socket.IO je adaptována pro mnoho programovacích jazyků, jako například C#, Java, C++, PHP a mnoho dalších. Pro zpracování této práce bude využita základní knihovna pro Node.js a dále knihovny pro jazyk Java (pro vývoj Android aplikace) a pro Arduino (pro vývoj chytrých zařízení).

## Android

Android je open source operační systém vyvíjený společností Google pro mobilní zařízení založený na linuxovém jádře. Používá se na mnoha zařízeních, jako jsou chytré telefony, tablety a televize. Má největší zastoupení mezi mobilními operačními systémy. Ve třetím kvartálu roku 2016 byl podíl prodaných zařízení s operačním systém Android 86,7%. Operační systém Android se vyznačuje vysokou univerzálností. Je možné pro něj vytvořit různé nadstavby, které mění funkce systému. Tento způsob rozšiřování způsobuje, že stejná verze operačního systému může mít na zařízeních různých výrobců jiný vzhled a částečně jinou funkcionalitu. Aplikace pro Android se píší v jazyce Java, ve kterém je i část samotného operačního systému napsána. Operační systém nabízí velmi rozsáhlé API, které je možné použít pro vývoj aplikací.

## HTML

Značkovací jazyk HTML tvoří základní kostru veškerých webových stránek. Tento jazyk je charakteristický velkým počtem jasně definovaných tagů, které určují vzhled a funkcionalitu jednotlivých elementů. Tyto elementy poté tvoří jeden celek, který se zobrazuje uživateli jako webová stránka. Webové stránky psané v HTML se interpretují až ve webovém prohlížeči a nelze je využít pro přístup k souborovému systému uživatele. Funkcionalita webových stránek se rozšiřuje pomocí skriptovacího jazyka Javascript a vzhled stránek se určuje styly CSS.

## Javascript

Javascript je objektově orientovaný skriptovací jazyk, který se velmi často používá ve webových aplikacích, ale umožňuje široké spektrum využití. Ve webových aplikacích se používá pro ovládání různých GUI prvků, zpracování dat, asynchronní komunikaci a další. Jazyku Javascript využívá i serverový systém Node.js. Javascript nemá nic společného s jazykem Java, použití slova Java v názvu je pouze z marketingových důvodů. Javascript lze použít i v operačním systému Windows pomocí aplikace Windows Script Host.

## jQuery

jQuery je open source knihovna pro skriptovací jazyk Javascript. Tato knihovna se zaměřuje na interakci s jazykem HTML. Pomocí této knihovny lze mnohem snáze napsat kód, který mění vzhled webové stránky. Základním prvkem jazyka je takzvaná funkce „$“. Tato funkce zajišťuje vybrání správných elementů a je to kořenová funkce pro veškerá volání. Tato knihovna také poskytuje funkce pro asynchronní načítání obsahu ze serveru, dynamické načítání dalších Javascript knihoven a souborů, nebo například animace. Nad knihovnou jQuery jsou postavené další pluginy jako například jQueryUI, Colorbox atd.

## NodeMCU

NodeMCU je programovatelný mikrokontroler založený čipu ESP8266 od Čínského výrobce Espressif Systems. Kontroler má jedenáct digitálních vstupně-výstupních pinů a jeden analogový vstupně-výstupní pin. Na tyto piny můžeme připojit zařízení také přes různé sběrnice, jako například SPI, I2C a další. Na některých pinech také podporuje pulzně-šířkovou modulaci (PWM). Takt čipu mikrokontroleru je ve výchozím nastavení 80 MHz, ale lze jej programově změnit na 160 MHz. Čip odkládá své mezivýpočty na relativně velkou 96kB RAM. Pro uložení zkompilovaného kódu slouží xxkb paměti. Pro uživatele je také k dispozici elektricky nezávislá 4kB flash paměť pro uložení dat. Hlavní výhodou tohoto čipu je možnost komunikace s jinými zařízeními přes WiFi připojení. NodeMCU lze programovat pomocí jazyka Lua, nebo s využitím knihovny pomocí jazyka Arduino.

## Let’s Encrypt

Let’s Encrypt je certifikační autorita, která vydává bezplatné doménově ověřené certifikáty. Tato autorita je mezi většinou moderních webových prohlížečů uznávána jako důvěryhodná. Tato autorita je sponzorována velkými korporacemi zabývající se oblastí internetu jako například Mozilla Foundation, Chrome, Cisco Systems a další. Hlavním cílem bylo rozšíření šifrování webových stránek a tudíž zvýšení zabezpečení webu.

# 1. Instalace programů potřebných pro vývoj aplikace

## Vývojové prostředí

Aplikace SmartHome byla vyvíjena ve dvou vývojových prostředích. V programu Atom byla vyvíjena serverová aplikace, webové rozhraní a program pro chytrá zařízení. Pro kompilování kódu chytrých zařízení byla využita nadstavba Platformio, která zajišťuje kompilaci a nahrání kódu do chytrých zařízení. Aplikace pro chytré telefony s operačním systémem Android byla vyvíjena v programu Android Studio.

## Instalace programu Atom a nadstavby Platformio

Pro nainstalování programu Atom je nutné nejdříve stáhnout jeho instalační balíček z adresy <https://atom.io/>. Tento instalační balíček po spuštění vyzve uživatele pro vyplnění potřebných nastavení, jako je cesta instalace programu. Po potvrzení instalátor nainstaluje aplikaci na uživatelův počítač a spustí aplikaci Atom.

Nadstavba Platformio se nainstaluje pomocí rozšiřitelných balíčků v aplikaci Atom. Manažer balíčků se nachází v nastavení aplikace Atom. Po jeho otevření je nutné kliknout na tlačítko pro instalaci nových balíčků, do vyhledávacího pole zadat text platformio-ide a poté u balíčku platformio-ide kliknout na tlačítko Install. Po jeho stisknutí se spustí automatizovaný proces, který nainstaluje veškeré potřebné závislosti a samotné rozšíření Platformio do aplikace Atom.

## Vytvoření projektu pro vývoj aplikace pro NodeMCU

Pro vytvoření nového projektu je nutné přejít na domovskou obrazovku rozšíření Platformio. Na této obrazovce po kliknutí na tlačítko New Project se objeví nová nabídka, ve které se vyplní jméno projektu, typ mikrokontroleru (v našem případě NodeMCU 0.9) a druh frameworku (v našem případě Arduino). Lze zde ještě nastavit umístění projektu. Po kliknutí na tlačítko Finish se spustí proces, který stáhne požadované knihovny a správně nastaví vývojové prostředí.

## Instalace programu Android Studio

Doplnit

# 2. Databáze pro ukládání dat

Finální databáze se skládá ze sedmi tabulek. (viz. obrázek) Do těchto tabulek jsou ukládána veškerá data, která se v aplikaci SmartHome používají. Tato data zahrnují uživatelská data, informace o zařízeních, oprávnění uživatelů k přístupu k jednotlivým zařízením, historii stavů zařízení a uživatelských příkazů. Databáze odpovídá 2. normální formě.

## Omezení vkládaných dat

Na několika sloupcích, které nejsou klíčové, jsou nastavena omezení, která zaručují, že data vkládaná do databáze jsou unikátní. Toto omezení se vztahuje například na uživatelské jméno uživatele, nebo emailové adresy.

# 3. Webové rozhraní

Webové rozhraní se skládá z několika klíčových prvků, které zajišťují pohodlné a komplexní nastavení zařízení a používá se pro přidělení oprávnění pro další uživatele. Lze zde také naleznout ovládací panel, který je alternativa pro Android aplikaci a poskytuje stejnou funkcionalitu.

## Funkčnost stránky a načítání dat

Webové rozhraní tvoří pouze jediná stránka s horní navigační lištou, menu, pozadím a centrální oblastí pro zobrazení dat, která uživatel vyžaduje. Pokud uživatel otevře klikne na položku v menu, aktuální stránka upraví odkaz v adresním řádku, korespondující s odkazem na stránku, na kterou uživatel klik, ale na tento odkaz se stránka nepřesměruje. Místo toho odešle dotaz na server, aby uživateli odeslal obsah centrálního bloku. Po obdržení obsahu tento obsah vykreslí do zmíněného bloku a načte a spustí javascriptový kód určený pro tuto stránku.

## Registrace uživatele

Nepřihlášení uživatelé mají přístup pouze k hlavní straně, na které lze naleznout základní informace o aplikaci, ale nemají přístup k žádným ovládacím prvkům. Registraci provede stiknutím na tlačítko Přihlásit se a následně na tlačítko Zaregistrujte se zde. Po stisknutí tohoto tlačítka se objeví formulář, do kterého uživatel zadá své uživatelské jméno, heslo, email a jméno a příjmení a stiskne na tlačítko Zaregistrovat. V tomto formuláři je také implementována nejnovější generace služby Google reCAPTCHA, která se vyznačuje tím, že pravost uživatele dokáže ověřit i bez toho, aby musel zaškrtávat políčko Nejsem robot, případně vyplňovat test pravosti uživatele. Tento druh ověření se nazývá Invisible reCAPTCHA. Data z tohoto formuláře se odešlou na server ve formátování JSON, který je zpracuje. Uživatel musí svoji registraci potvrdit kliknutím na odkaz v emailu, který během několika málo minut obdrží do emailové schránky, kterou zadal ve formuláři. (Obrázek) Po ověření pravosti uživatele ověřením emailové adresy se uživatel již může přihlásit do systému.

## Přihlášení uživatele

Pro přihlášení uživatel služby musí kliknout na tlačítko Přihlásit se a zadat do formuláře uživatelské jméno, nebo emailovou adresu a heslo. (Obrázek) Po kliknutí na tlačítko Přihlásit se data odešlou na server, který je zpracuje a obratem uživateli odešle data, zda bylo přihlášení úspěšné a zároveň uživateli odešle obsah menu a obsah aktuální webové stránky. Javascriptový kód po přijetí této odpovědi uživatele informuje, zda bylo přihlášení úspěšné a případně vloží obsah do menu a do centrálního bloku stránky.

## Registrace nového zařízení

Stránku pro registraci zařízení otevře přihlášený uživatel pomocí menu a vybráním možnosti Nové zařízení. Na této stránce je nutné zadat přezdívku zařízení, která se bude zobrazovat u daného zařízení a ID zařízení, kterým se dané zařízení identifikuje při připojení k serveru. Po stisknutí tlačítka Přidat nové zařízení se spustí deseti minutový interval, během kterého je nutné zařízení připojit k serveru. Při úspěšném zaregistrování zařízení se zobrazí hláška o úspěšné registraci a uživatel, který zařízení zaregistroval má nyní práva na veškeré ovládání a přidělování dalších oprávnění pro jiné uživatele na daném zařízení.

## Seznam zařízení

Uživatel může využít funkce Moje zařízení, která mu zobrazí veškerá zařízení, jejichž je majitelem, a zařízení, ke kterým má nějaká oprávnění. Na tuto stránku se dostane přes hlavní menu kliknutím na možnost Moje zařízení nebo otevřením odkazu <https://smarthome.linhy.cz/?devices>. Nalezne zde informace, zda je dané zařízení připojeno k serveru a kdy se naposledy přihlásilo. U každého zařízení se nachází odkaz pro správu oprávnění, a pokud je zařízení online nachází se zde i odkaz na ovládací panel pro dané zařízení.

## Seznam funkcí

Na této stránce uživatel registruje funkce, které bude dané zařízení využívat. Může zde i zobrazit funkce, které vytvořili ostatní uživatelé a které může také využít. U funkce si uživatel musí vybrat, zda daná funkce bude pouze pro čtení, zápis, nebo čtení i zápis. Také si zde vybere z možných grafických prvků, ve kterých se bude hodnota zobrazovat. Hlavně na této stránce uživatel nalezne ID funkce, kterou bude používat pro odesílání dat z chytrého zařízení na server a pro udělování oprávnění jiným uživatelům.

# Správa oprávnění

Pro úpravu, přidání nebo odebrání práv musí uživatel přejít na stránku Oprávnění kliknutím na příslušné tlačítko v menu nebo na adrese <https://smarthome.linhy.cz/?permissions>. Zobrazí se mu zde rozbalovací seznam zařízení, ve kterém se po rozkliknutí daného zařízení otevře další rozbalovací seznam s uživateli, kteří již nějaké oprávnění na daném zařízení mají a tlačítkem pro přidání nového uživatele. Poté, co uživatel rozbalí oprávnění daného uživatele, se vykreslí seznam s funkcemi, ke kterým daný uživatel má oprávnění. Tlačítkem s ikonou popelnice lze oprávnění vybrané oprávnění uživateli odebrat, nebo naopak tlačítkem s ikonou plus lze uživateli přidělit oprávnění k nové funkci. Pokud chce uživatel přidělit oprávnění novému uživateli, klikne na tlačítko Přidat nového uživatele u příslušného zařízení. Otevře se formulář, do kterého se zadá uživatelské jméno uživatele, kterému chce majitel přidat nové oprávnění. Po zadání uživatelského jména webová aplikace na serveru ověří, zda daný uživatel existuje, a pokud ano, tak zobrazí formulář, ve kterém uživatel vybere funkci ze seznamu a tlačítkem s ikonou diskety uloží oprávnění pro daného uživatele a zařízení.

## Nastavení zařízení

Webové rozhraní také umožňuje uživateli vytvořit konfigurační soubor, který může použít zařízení. V tomto konfiguračním souboru může vyplnit různé hodnoty, které si chytré zařízení při spuštění stáhne a uživatel je může dále využít. Toto umožňuje snadnou změnu konfigurace bez nutnosti opět kompilovat kód. Také v tomto konfiguračním souboru vyplní, jaké funkce bude chytré zařízení plnit.

## Ovládací panel

Tato součást webového rozhraní je alternativou pro aplikaci na mobilní telefony. Po otevření se zde v horní části zobrazí posuvný seznam se zařízeními a v dolní části stránky se vykreslí funkce, ke kterým má uživatel oprávnění. Mezí zařízeními se uživatel může přepínat šipkami na posuvném seznamu v horní části, nebo šipkami vlevo a vpravo na klávesnici. Aktualizace hodnot po aktualizaci ze serveru

## Historie

Na této stránce se vykreslí stejný posuvný seznam jako u ovládacího panelu, ale v dolní části stránky se objeví tabulka s veškerými historickými záznamy. Tyto záznamy obsahují čas, kdy byla akce vykonána, název funkce, druh události, ID uživatele a jaká hodnota byla na zařízení odeslána. Druhem události se rozumí, zda na zařízení byl odeslán příkaz, nebo zda zařízení serveru oznámilo změnu hodnoty dané funkce. Další druhy události jsou přidání oprávnění, připojení a odpojení zařízení atd. V případě, kdy se hodnota funkce změnila na zařízení, se nezobrazuje ID uživatele. Pokud druh akce není změna hodnoty, nebo příkaz, nezobrazuje se název funkce.

# 4. Serverová aplikace

Serverová aplikace je hlavní řídící prvek této práce. Přijímá spojení s klienty a zařízeními a stará se o interakci mezi chytrými a klientskými zařízeními. Také zapisuje a čte z databáze a odesílá uživateli data, o která si zažádá.

## Načtení veškerých modulů a certifikátu po spuštění

Po startu aplikace je nutné načíst veškeré knihovny, které budou potřebné správný běh aplikace. Vytvoří se také https server, který bude sloužit knihovně Socket.IO pro komunikaci mezi serverem a uživateli, nebo zařízeními. HTTPS server využívá certifikát vygenerovaný bezplatnou autoritou Let’s Encrypt. Dále spustí Socket.IO poslouchající na tomto webovém serveru a vytvoří připojení k databázi pomocí knihovny tedious-connection-pool. Tato knihovna zajišťuje, že připojení k databázi bude neustále navázáno a při požadavku o dotaz do databáze se z množiny navázaných připojení (pool) vybere jedno připojení, pomocí kterého se dotaz provede. Pokud bude spojení mezi serverovou aplikací a databází ukončeno, naváže nové spojení, které opět vloží do poolu. Velikost poolu se určuje v konfiguraci při spuštění. Dále také zajistí, že nebude spuštěn dotaz, pokud není předchozí dotaz dokončen. Pokud by tato situace nastala, databáze by nevykonala daný dotaz. Na závěr se databáze připojí ke službě Google reCAPTCHA, která se používá při registraci nového uživatele.

## Zpracování uživatelských požadavků

Server přijímá veškeré požadavky pomocí knihovny Socket.IO a její asynchronní funkce on. Poté, co server obdrží požadavek, na zpracování přijatých dat, tak jako první ověří, zda se jedná o autorizovaný přístup. To znamená, že uživatel, nebo chytré zařízení, které se snaží o komunikaci se serverem, je přihlášen. Tato kontrola se netýká požadavků o přihlášení uživatele, registraci nového uživatele a, registraci nového zařízení a ověření uživatelského účtu. Teprve po ověření přístupu začne zpracovávat požadavek a zpravidla odešle odpověď potvrzující úspěšné, či neúspěšné zpracování požadavku.

## Registrace uživatele

Při přijetí požadavku o registraci nového uživatele nejdříve server zjistí, jestli jsou veškerá potřebná pole vyplněna. Poté na serverech služby Google reCAPTCHA pomocí knihovny služby ověří, zda žádost odeslal člověk. Pokud ano, vytvoří hash hesla, které uživatel odeslal serveru, vytvoří 40 znaků dlouhý řetězec znaků, který bude sloužit pro ověření uživatele, a s ostatními daty ho uloží do databáze. Po uložení odešle zprávu register\_response webovému rozhraní, které uživatele informuje o úspěšné, či neúspěšné registraci a odešle ověřovací email na adresu specifikovanou v žádosti o registraci. Emaily se odesílají pomocí knihovny Nodemailer, která využívá emailového účtu vytvořeného na mailovém serveru gmail.com.

## Ověření uživatele

Kliknutím na adresu v emailu se uživateli otevře webová stránka, která na server odešle uživatelské jméno a řetězec znaků vygenerovaný při registraci zprávou validate. Server při přijetí této zprávy vykoná databázový dotaz, ve kterém nastaví hodnotu pole Overeno na 1 u uživatele s uživatelským jménem a tokenem obdrženým v žádosti.

## Přihlášení uživatele

Pokud server přijme od uživatele požadavek na přihlášení, nejdříve ověří správnost uživatelského jména hesla. Z databáze získá veškerá data o daném uživateli. Součástí této informace je hash uživatelova hesla. Pomocí knihovny pbkdf2-password-hash aplikace heslo, které zadal uživatel, zahashuje a tyto hashe porovná. Knhovna poté vrátí, zda hash hesla, odpovídá hashi z databáze. Tímto způsobem se ověří správnost hesla. Pokud uživatel zadal správně heslo, vygeneruje 60 znaků dlouhý náhodný text, který se používá jako identifikátor uživatele. Uživatelské jméno, výše uvedený identifikátor, ID uživatele a samotný objekt socketu se uloží do objektu auth\_users (tento objekt se nachází v souboru authentication.js). Data, která jsme získali dotazem z databáze, identifikátor a obsah menu se odešlou uživateli. Pokud uživatel ve své žádosti o přihlášení požádal o obsah webové stránky (například otevřel adresu /?devices), server data této stránky přiloží k odpovědi a odešle webovému rozhraní, nebo android aplikaci, které ji zpracují.

## Odhlášení uživatele

Při odhlášení uživatele (stiskem tlačítka odhlásit se) se odešle jednoduchá žádost, bez jakýchkoliv dat. Serverová aplikace zavolá funkci unAuthUser(socket), která v objektu auth\_users uživatele vyhledá pomocí objektu socketu, ze kterého provedl žádost. Nalezený záznam z proměnné auth\_users smaže a pro další komunikaci z daného zařízení se uživatel bude muset znovu přihlásit.

## Opětovné přihlášení uživatele

Pokud uživatel znovu načte webovou stránku, nebo stránku zavře (a neodhlásí se) a znovu ji otevře, webová stránka odešle serveru žádost o opětovné přihlášení. V této žádosti se nachází data, která obdržel od serveru při úspěšném přihlášení, nejdůležitější však jsou uživatelské jméno a identifikátor. Server při přijetí této žádosti nalezne v objektu auth\_users uživatele podle jeho uživatelského jména a pokud se identifikátor uložený v této proměnné a identifikátor, který odeslalo webové rozhraní, shodují, server obnoví data v proměnné auth\_users novými daty, která obdržel a provede stejné akce jako při přihlášení (odeslání menu, obsah webové stránky, atd.)

## Registrace zařízení

Pro registraci je nejdříve nutné, aby uživatel odeslal požadavek na odeslání (viz 3 – registrace nového zařízení). Po obdržení požadavku server uloží požadavek do proměnné a vyčkává na připojení chytrého zařízení, které chce uživatel zaregistrovat, k serveru. Po připojení zařízení a obdržení požadavku dev\_login (device login) serverová aplikace porovná, zda se jedná o registraci (zda pole heslo není vyplněno). Pokud se jedná o registraci, vygeneruje 30 znaků dlouhý náhodný text, který uloží s ostatními daty z žádosti do databáze (Název zařízení, status nastaví na true, poslední dobu přihlášení nastaví na aktuální čas). Dále v databázi uloží oprávnění s ID funkce 1 uživateli, který o registraci zažádal, na tomto zařízení. ID funkce 1 znamená práva majitele a uživatel s tímto oprávněním má výhradní práva na daném zařízení.

## Přihlášení zařízení

Po přijetí příkazu dev\_login server nejdříve ověří přijaté ID zařízení a zahashované heslo s údaji v databázi. Pokud se údaje zaslané zařízením shodují, zařízení je autorizováno k dalším příkazům a je o tom informováno zprávou s klíčovým slovem dev\_login\_response. Dále je objekt socketu zařízení a ID zařízení uloženo do proměnné devices v knihovně devices.js. Tato proměnná uchovává veškerá zařízení, která jsou připojena a přihlášena k serveru. Dále se informují všichni uživatelé, kteří jsou přihlášení k serveru a mají jakékoliv právo k zařízení o tom, že zařízení je online.

## Odhlášení zařízení

Zařízení je možné odhlásit pouze jeho odpojením od serveru. To lze provést odpojením od socketu na zařízení, nebo odpojením zařízení od připojení k síti. Pokud server ztratí spojení se zařízením, odstraní ho z proměnné devices a je nutné provést opětovnou autorizaci pomocí ID zařízení a hesla. Podobně jako u přihlášení se uživatelé informují o stavu zařízení.

## Odpovědi na žádosti uživatelského rozhraní

Pro serverovou aplikaci jsem vytvořil množinu klíčových slov, na která server odpovídá (viz tabulka). Po přijetí zprávy server podle klíčového slova zjistí, jaká data uživatel žádá a provede příslušnou operaci (většinou se jedná o dotaz do databáze – získání seznamu zařízení, seznamu oprávnění, atd.). Data poté odešle odpovědí, ve které se nachází klíčové slova, odpověď serveru zpracuje webové rozhraní, nebo Android aplikace.

## Zpracování příkazů

Jelikož serverová aplikace je centrálním bodem této aplikace, chová se jako pomyslný směrovač stavových zpráv a příkazů mezi klientskými a chytrými zařízeními. Pokud server přijme zprávu s klíčovým slovem commandToServer, nejdříve dotazem do databáze ověří, zda má uživatel právo na danou operaci. Dále zjistí ze zprávy ID zařízení a z proměnné devices získá objekt socketu. Pomocí tohoto objektu odešle zprávu ve formátu JSON obsahující ID funkce a hodnotu, kterou uživatel nastavil.

## Zpracování stavových zpráv

Při obdržení stavové zprávy od zařízení server aktualizuje v databázi u příslušného zařízení hodnotu LastValues na aktuální hodnotu pro danou funkci. Poté získá seznam ID uživatelů, kteří mají k dané funkci na daném zařízení oprávnění. Pomocí těchto ID získá veškeré objekty socketů z proměnné auth\_users a pomocí těchto objektů novou hodnotu zašle všem zařízením, které ji zpracují.

## Ukládání a získání dat historie

Při definovaných akcích (registrace, přihlášení a odhlášení zařízení, stavová zpráva a příkaz) se spustí samostatný dotaz do databáze, který do tabulky tab\_historie vloží k příslušnému zařízení záznam o tom, jaká byla provedena akce, kdo akci provedl, kdy byla akce provedena a jaká byla hodnota provedená akcí. Pro získání dat uživatelské zařízení zavolá funkci get\_history, kterou server zpracuje a obratem uživateli odešle veškerá historická data, ke kterým má uživatel přístup.

# Aplikace pro NodeMCU

## Knihovna SmartHome

Knihovna pro NodeMCU na platformě Android slouží pro jednoduchou uživatelskou interakci se serverem. Pro klienta usnadňuje práci s pamětí EEPROM, nastavením WiFi připojení, odesíláním stavových zpráv a přijímáním příkazů. Veškerá data jsou uživateli poskytnuta ve formátu JSON pomocí knihovny ArduinoJson.

## Specifikace výchozích dat, nastavení funkcí zpracovávající příkazy

Knihovna SmartHome poskytuje pouze nastavení údajů pro přihlášení k výchozí WiFi síti. Tyto údaje poskytnete knihovně pomocí funkce setDefaultWiFiConnection, ve které pomocí prvního argumentu specifikujete SSID sítě a pomocí druhého heslo sítě, ke které se chcete připojit. Poté lze použít funkci onReceivedConfiguration(), v jejímž parametru uživatel poskytne ukazatel funkce, která se má spustit, když chytré zařízení přijme konfigurační soubor ze serveru. Tento ukazatel uloží do proměnné a při přijetí konfiguračního souboru obsah tohoto souboru předá funkci ve formátu JSON. Dále může zařízení tento soubor zpracovat a například nastavit výstupní piny. Poslední funkcí, která by se měla zavolat před příkazem begin() je funkce onReceive(). V parametrech této funkce musí uživatel specifikovat ID funkce a ukazatel funkce, která bude zpracovávat příkaz obdržený ze serveru. Tyto ukazatelé se uloží do proměnné typu map, která ke klíči (ID funkce) přiřadí hodnotu (ukazatel funkce). Při přijetí příkazu ze serveru se podle ID funkce v této proměnné nalezne správný ukazatel funkce a dané funkci se předá hodnota příkazu ve formátu JSON.

## Registrace zařízení

Pokud se zařízení poprvé připojí k serveru a ještě neobdržel své vygenerované heslo, odešle serveru v přihlašovací zprávě namísto hesla prázdnou proměnnou. Tímto server pozná, že se jedná o registraci a ne o pokus o přihlášení. Poté, co server žádost zpracuje (viz Serverová aplikace – registrace zařízení), odešle chytrému zařízení heslo zprávou s klíčovým slovem register\_pass. Chytré zařízení přečtě tuto zprávu a heslo v ní obsažené uloží do paměti EEPROM (viz. Ukládání a čtení hesla z paměti EEPROM)

## Ukládání a čtení hesla z paměti EEPROM

Mikrokontroler NodeMCU je vybaven 4kB paměti, do které může uživatel zapisovat. Knihovna SmartHome využívá prvních 40 bytů, do kterých uloží heslo, které mu bylo přiděleno od serveru při registraci. O zápis se stará funkce writeEEPROM v knihovně SmartHome, která předaný řetězec znaků znak po znaku převede na jeho číselnou hodnotu a uloží do paměti na daný byte odpovídající danému znaku. Čtení probíhá podobným způsobem. Aplikaci byte po bytu získá číselné hodnoty znaků uložené v paměti. Tyto znaky poté složí za sebe a uživatel vrátí ucelený řetězec znaků.

## Funkce begin() a loop()

Při zavolání funkce begin se spustí WiFi připojení pomocí knihovny ESP8266WiFiMulti, poté se server připojí pomocí knihovny Socket.IO pro zařízení s čipem ESP8266, odešle přihlašovací zprávu s ID zařízení a heslem, které přečte z paměti EEPROM (Pokud heslo není nalezeno, odešle prázdné pole – jedná se o registraci zařízení), zpracuje odpověď a pokud bylo zařízení úspěšně přihlášeno, požádá o konfigurační soubor. Při jeho přijetí ho předá funkci, kterou uživatel specifikoval. Funkce loop() pouze kontroluje, zda od serveru nepřišla nová zpráva, kterou by mohla knihovna zpracovat. Pro správnou funkčnost aplikace je nutné tuto funkci umístit do smyčky loop().