

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Monitorování průchodů osob pomocí Raspberry Pi



2016

Vedoucí práce: Mgr. Jiří Zaccal,
Ph.D.

Martin Černý

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Martin Černý
Název práce: Monitorování průchodů osob pomocí Raspberry Pi
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2016
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Jiří Zaccpal, Ph.D.
Počet stran: 34
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Martin Černý
Title: Monitoring the passage of persons using the Raspberry Pi
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2016
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Jiří Zaccpal, Ph.D.
Page count: 34
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

K vývoji nové technologie v oblasti docházkových systémů, bylo nutné přijít s technologií která není v současné době využívána. Využitím bluetooth k identifikaci zaměstnanců a dosažení času 2 s potřebného k identifikaci, plně nahrazuje současné docházkové systémy. Navržený systém je spuštěn na mini počítači Raspberry Pi, využívající webový server pro implementaci uživatelského rozhraní v PHP a Python pro vlastní logiku systému.

Synopsis

For developing new technology in attendance systems, it was necessary to come up with technology that is not currently used. Using bluetooth to identify employees and achieve time 2 s required to identify, fully replaces the current attendance systems. Designed system is being used on minicomputer Raspberry Pi, using a web server to implement user interface in PHP and Python for main logic system.

Klíčová slova: bluetooth; Raspberry Pi; docházkový systém; Linux; Python; PHP

Keywords: bluetooth; Raspberry Pi; attendance system; Linux; Python; PHP

Děkuji Mgr. Jiřímu Zaccalovi, Ph.D. a Davidu Pokornému za cenné rady.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	8
1.1	Požadavky na systém	8
1.2	Návrh systému	8
1.2.1	Identifikační program	9
1.2.2	Webová aplikace	9
2	Použité technologie	10
2.1	PHP	10
2.2	Python	10
2.3	Pojmenované roury	10
2.4	MySQL	11
2.4.1	SQL injection	11
3	Identifikační program	13
3.1	Požadavky na skenery	13
3.2	Signalizace aktivity	14
3.3	Bluetooth skener	16
3.3.1	Půlení intervalu	16
3.3.2	Rychlá identifikace	17
3.3.3	Vracení výsledků	17
3.4	Síťová komunikace	18
3.5	Záloha databáze	18
4	Databáze	19
4.1	struktura	19
4.1.1	Osoby	19
4.1.2	Detail	20
4.1.3	Zařízení	20
4.1.4	Záznamy	20
4.1.5	Temp	20
4.1.6	Zachycen	21
5	Webová aplikace	21
5.1	Navigační menu	21
5.2	Popis funkcionality administrátora	22
5.2.1	Přehled	22
5.2.1.1	Obecné	23
5.2.1.2	Odpracované hodiny	23
5.2.1.3	Poznámky	23
5.2.2	Nastavení	24
5.2.2.1	Přístup	24
5.2.2.2	Záloha	24
5.2.2.3	Skenování	25

5.2.3	Evidence	25
5.2.4	Log	26
5.2.5	Přítomní zaměstnanci	26
5.3	Popis funkcionality běžného uživatele	27
Závěr		29
Conclusions		30
A Postup spuštění docházkového systému		31
A.1	Potřebné balíky	31
A.2	Instalace docházkového systému	31
A.2.1	Identifikační program	31
A.2.2	Webová aplikace	31
A.2.3	Databáze	32
A.2.4	Párování zařízení	32
A.3	Spuštění docházkového systému	32
B Obsah přiloženého CD/DVD		32
Literatura		34

Seznam obrázků

1	Schéma zapojení signalizačních LED diod	15
2	Princip algoritmu půlení intervalu	17
3	Relační model databáze	19
4	Webová aplikace v režimu administrátor s aktivovanou databází .	22
5	Ukázka seznamu přítomných zaměstnanců	27
6	Webová aplikace v režimu běžný uživatel	28

Seznam tabulek

1	Popis signalizace stavů pomocí LED diod	15
---	---	----

Seznam zdrojových kódů

1	Příklad vytvoření a zápis do pojmenované roury v Python	11
2	Příklad čtení z pojmenované roury v Python	11
3	SQL injection rozšíření stávajícího dotazu	12
4	SQL injection vložení vlastního dotazu	12
5	Příklad práce s připravenými příkazy v PHP	13
6	příklad obnovení databáze ze souboru, v Linuxu pomocí shell . . .	18
7	příklad hashování hesla v MySQL	20

1 Úvod

Současné docházkové systémy nabízí způsoby identifikace pouze kontaktní, nebo na velmi krátkou vzdálenost v řádech jednotek, nebo desítek milimetrů. Do této kategorie můžeme zařadit například RfID, čtečka otisku prstů, nebo snímání duhovky oka, tato technologie je nová a zatím není příliš využívána.

Firma OLC Systems s. r. o. pro kterou je vyvíjen docházkový systém, v současné době využívá *Docházka 3000*¹. Hlavní problém tohoto systému je čas potřebný k identifikaci pomocí RfID, který dosahuje 4-5 s. Další z důvodů je zastaralý systém.

Motivací je tedy vytvořit systém u kterého odpadne kontakt se samotným zařízením a vzdálenost se zvýší i na jednotky metrů. Vybraná bezdrátová technologie bluetooth tyto podmínky splňuje a má za výhodu, že je již delší dobu jako standardní výbava mobilních telefonů, který mají lidé neustále u sebe.

1.1 Požadavky na systém

Systém byl především vyvíjen pro firmu OLC Systems s. r. o. Mezi hlavní požadavky bylo využití bluetooth. Čas potřebný k identifikaci se musí pohybovat kolem 2 s, aby identifikace probíhala téměř v reálném čase bez nutnosti delšího zdržení. Identifikace, v případě bluetooth se jedná o MAC adresu², bude odeslána pomocí síťového připojení ke zpracování, v již zavedené databázi firmy. Pro zobrazení aktivity a konfigurace systému je využita webová aplikace.

Systém musí být spustitelný na mini počítači Raspberry Pi, nebo jeho alternativě, kterých je v současné době nepřeberné množství. Výhodou těchto zařízení je jejich velikost, která dosahuje velikosti platební karty. Avšak velikost není jedinou kladnou stránkou, příznivá pořizovací cena a množství přídatných modulů je zapříčiněno vývojem Raspberry Pi pro experimentální a výukové účely. Pro další rozšíření systému může být využit například nabízený NFC³ modul pro identifikaci.

1.2 Návrh systému

Na základě požadavků které je nutné dodržet, je systém pro širší využití obohacen o plnohodnotnou databázi, pro uložení údajů o zaměstnancích včetně záznamů příchodů a odchodů. Databáze se nachází na vlastním zařízení. Databázi bude možno v určitých intervalech zálohovat na lokální úložiště. Pro vyšší bezpečnost budou všechny zálohy v případě připojení na síťové úložiště odeslány pomocí FTP.

Systém je rozdělen na dvě hlavní části, aby byla oddělena reprezentace dat od mechanismu vyhledávání. První část bude mít za úkol získání identifikace

¹Vyvíjený firmou BM-Software, <http://www.dochazka.eu/>

²MAC adresa je jedinečný identifikátor bluetooth zařízení

³NFC je bezdrátová komunikace, v dnešní době čím dál víc využívanější k identifikaci, platbám, přenosům dat atd.

zaměstnance a jeho zpracování. Tato část je pro obsluhu systému „skryta“ a není možný přímý přístup. Pro kontrolu aktivity systému, zobrazení dat a konfiguraci systému slouží webová aplikace, která tvoří druhou část systému.

1.2.1 Identifikační program

První část bude mít za úkol vyhledávání bluetooth zařízení (identifikaci zaměstnanců), v textu tato činnost bude nazývána skenování. Skenování je odvozeno z funkce využívané pro získávání MAC adres. Získané MAC adresy jsou odeslány pomocí síťového připojení ke zpracování, popřípadě jsou MAC adresy zpracovány vlastní databází. Obě metody je možné nezávisle na sobě aktivovat, nebo deaktivovat.

Pro dosažení času 2 s pro identifikaci je nutné využít dva bluetooth adaptéry. Při využití jednoho bluetooth adaptéru a zkrácení času vyhledávání na 2 s je velmi nespolehlivé a zaměstnanec je identifikován až v následujícím cyklu vyhledávání. Algoritmus pro využití současně dvou bluetooth adaptérů popíše v samostatné kapitole.

Systém je modulární a nemusí se jednat pouze o identifikaci pomocí bluetooth. Systém umožňuje souběh více skenerů, které nejsou vázané na žádný konkrétní programovací jazyk. Výhodou souběhu více skenerů je že při výpadku jednoho, je možné využít jiný, nebo si zaměstnanci mohou zvolit která identifikace jim více vyhovuje.

1.2.2 Webová aplikace

Webová aplikace byla zvolena protože nemusí být instalována na zařízení, na kterých bude provozována. A tak jej bude možné spustit na kterémkoliv zařízení. Webová aplikace bude umožňovat dva základní typy uživatelských účtů. Prvním typem je tzv. *Administrátor* a druhým budou běžní uživatelé.

Administrátorovi prostřednictvím webové aplikace je umožněna kompletní konfigurace systému, odpadá nutnost přímého přístupu ke konfiguračnímu souboru, čímž se zabrání dostat systém do nekonzistentního stavu. Další hlavní stránkou je výpis z logovacího souboru, kam systém zapisuje hlavní body aktivity. Při aktivní vlastní databázi, bude administrátorovi umožněno nahlížet do záznamů zaměstnanců, popřípadě je spravovat. Přístup je uložen v konfiguračním souboru, pro funkčnost i při deaktivované vlastní databázi.

Uživatelské jméno a heslo pro přístup běžného uživatele je uložen ve vlastní databázi, pokud je databáze neaktivní, není tomuto uživateli přístup umožněn. Běžný uživatel má přístup pouze ke svým záznamům a to pouze pro zobrazení. Vlastní databázi je možné využít pro účel účetnictví, proto je měnění údajů bez vědomí pověřených osob nežádoucí. Běžný uživatel může měnit pouze své přístupové heslo.

2 Použité technologie

Dva hlavní programovací jazyky využitě pro implementaci systému jsou Python a PHP. Pro ukládání dat byla zvolena databáze MySQL. Dalšími využitými technologiemi je jQuery, CSS a pojmenovaná roura.

2.1 PHP

Jedná se o interpretovaný jazyk, který se především využívá pro tvorbu webových stránek. Pro běh je nutný webový server, navrhovaný systém využívá webový server Apache. V současné době se využívají různé frameworky implementované v jazyce PHP pro vyšší rychlost a více zabezpečení bez nutnosti vlastního ošetření, mezi ně patří například Nette. Pro méně zatížené webové aplikace je dostačující využití čistého PHP.

2.2 Python

Python je interpretovaný jazyk s překladem do bytecode⁴, který urychluje spouštění programu. Python má dynamicky typovou kontrolou. Výhodná rychlost výpočtů je dána implementací jádra v jazyku C. Avšak implementace v jazyku C není jediná, dále je Python implementován například v Jazyce Java (Jython), C# (IronPython) a další^[4]. Python se využívá k výuce programování, jeho syntaxe se považuje za jednoduchou.

V současné době se používají převážně dvě verze Pythonu a to verze 2.7 a 3.3. Identifikační program je implementován ve verzi 2.7.

2.3 Pojmenované roury

Jedná se speciální soubor typu FIFO⁵ využívané v operačním systému Linux. Pojmenované roury slouží pro výměnu informací mezi procesy které jsou na sebe nezávislé. Z toho vyplývá že nezáleží na implementaci daných procesů. Pro vytvoření pojmenované roury slouží příkaz *mkfifo*.

V některých programovacích jazycích je možné využít dva přístupy, bloku-jící a nebloku-jící. V případě bloku-jícího přístupu proces čeká na otevření roury na druhé straně a v případě otevření pouze pro čtení proces čeká dokud není do roury něco zapsáno, nebo není uzavřena. Nebloku-jící nečeká na otevření roury na druhé straně, ani v případě otevření pouze pro čtení nečeká na data.

⁴soubory s příponou *.pyc

⁵první dovnitř první ven z anglického překladu First In First Out

```

1 import os
2
3 #Vytvoření pojmenované roury s názvem test
4 os.mkfifo(test)
5
6 #Otevření pojmenované roury pouze pro zápis
7 pipe_test = os.open(test, os.O_WRONLY)
8
9 #Zápis řetězce "data" do roury
10 os.write(pipe_test, "data")

```

Zdrojový kód 1: Příklad vytvoření a zápis do pojmenované roury v Python

```

1 import os
2
3 #Otevření pojmenované roury pouze pro čtení
4 pipe_test = os.open(test, os.O_RDONLY)
5
6 #Čtení řetězce z roury o délce čtyř znaků
7 word = os.read(pipe_test, 4)

```

Zdrojový kód 2: Příklad čtení z pojmenované roury v Python

2.4 MySQL

Z důvodu předpokládané menší zátěže webové aplikace, je pro nás uspokojivý výkon relační databáze MySQL, která se hodí na méně zatížené služby. Na více vytížené služby je lepší využít konkurenčních databází např. PostgreSQL. MySQL má podporu u všech využitých technologií v navrhovaném systému, čímž je splněn i další z požadavků. Využitím databáze s sebou i nese bezpečnostní rizika, především pokud je uživatelům umožněno vkládat data.

2.4.1 SQL injection

Mezi hlavní úskalí využití databáze je útok tzv. SQL injection. Jedná se o úpravu SQL dotazu v prospěch útočníka. „SQL injection je technika, kde mohou uživatelé vsunout vlastní SQL příkaz do již připraveného SQL dotazu, přes vstupní textové pole webové stránky“[2].

Vsouvání vlastních SQL příkazů může být ve významu rozšíření stávajícího SQL dotazu, aby se útočník dostal ke všem datům tabulky, využívá se vždy pravdivé podmínky viz. příklad.

Vsunutý dotaz může také provádět zcela jinou akci, než ke které byl stávající SQL dotaz určen. Takovým případem může být snaha smazat obsah tabulky, databáze, nebo vložení vlastních dat do databáze. Při této formě útoku je nutné znát strukturu napadené databáze.

```

1 --Správný SQL dotaz, vstup byl zadán: user
2
3 SELECT * from person where username='user';
4
5 --Upravený SQL dotaz, vystup byl zadán: user' or 1=1, 1=1 je zde
   jako vždy platná podmínka, budou vráceny všechny data z tabulky
   person
6
7 SELECT * from person where username='user' or 1=1;

```

Zdrojový kód 3: SQL injection rozšíření stávajícího dotazu

```

1 --Správný SQL dotaz
2
3 SELECT * from person where username='user';
4
5 --Upravený SQL dotaz, vstup byl zadán: user'; DROP TABLE person; --
   po provedení SQL dotazu bude obsah tabulky person smazán,
   cokoliv je za znaky -- je brán jako komentář.
6
7 SELECT * from person where username='user'; DROP TABLE person; --

```

Zdrojový kód 4: SQL injection vložení vlastního dotazu

Neošetřením je možné přijít o všechny data, útočník se dostane k citlivým datům, nebo dalším hrozbám. Řešením toho útoku je hned několik, v navrhovaném systému jsem využil připravených příkazů⁶.

Připravené příkazy se využívají jak jsem již zmínil jako obrana proti SQL injection, ale dále se využívají při opakovaném dotazu pouze s jinými parametry. Jak z principu bude patrné, při vázání hodnot na parametry již nelze SQL dotaz změnit, čímž se zabrání výše uvedenému útoku.

Princip připravených příkazů^[3]:

1. Vytvoření SQL dotazu a nahrazení parametrů za ? a poté je odeslán do databáze
2. Databáze analyzuje , zkompiluje a optimalizuje SQL dotaz a uloží výsledek bez provedení dotazu
3. Na parametry se naváží hodnoty a provede se SQL dotaz

⁶z originálního anglického překladu Prepared Statements

```

1 //Připojení do databáze
2 $mysqli = new mysqli("localhost", "root", "12345", "database");
3 //Připravení dotazu
4 $query = $db->prepare('SELECT jmeno, prijmeni from osoby where id=?'
5     );
6 //Navázání hodnot na parametry. Typy parametrů i - integer, d -
7     double, s - string, b - BLOB
8 $query->bind_param("i", 3);
9 //Vykonání dotazu
10 $query->execute();
11
12 //Získání výsledku a ukončení
13 $data = $query->get_result();
14 $row = $data->fetch_assoc();
15 $query->close();

```

Zdrojový kód 5: Příklad práce s připravenými příkazy v PHP

3 Identifikační program

Identifikační program se skládá ze dvou hlavních částí. První část se stará o získávání identifikací, dále v textu označováno za skenery a druhá část o následné zpracování získaných identifikací, dále v textu označováno za řídicí program.

Komunikace mezi těmito částmi je řešena pomocí pojmenovaných rour s blokujícím přístupem. Důvod tohoto využití je neznámý čas vracení identifikací skenerem a cyklická kontrola zda-li skener poslal identifikaci by buď zbytečně zatěžovala systém, a nebo nezpracovávala výsledky v co nejkratším čase.

Využitím pojmenovaných rour má za výhodu, že při případném rozšiřování navrhovaného systému o další skenery, není nutné skenery implementovat v jazyce Python, avšak je nutné dodržet pravidla nutná k chodu připojovaných skenerů. Nastavení skenerů se ukládá do konfiguračního souboru v sekci *Scanners* ve formě, název modul = cesta k modulu. Cesta může být absolutní, nebo relativní vzhledem k umístění spouštěného identifikačního programu.

3.1 Požadavky na skenery

Skenery není nutné implementovat v jazyce Python, avšak musí existovat program implementovaný v Python přes který se bude spouštět. Toto omezení je dáno způsobem práce s připojovanými moduly.

Postup připojení skeneru k řídicímu programu:

1. Pokus o naimportování modulu v jazyce Python
2. Získání informace o velikosti identifikačního řetězce

3. Získání cesty k pojmenované rouře
4. V případě aktivní lokální databázi získání názvu kategorie do které spadá identifikace (MAC, RFID, NFC, atd...)
5. Spuštění skeneru, zavoláním funkce main s parametrem START

Zvýše popsaného postupu je patrné, že je nutné mít nadefinované následující globální proměnné:

- **BUFFER_SIZE** - velikost identifikačního řetězce
- **PIPE** - cesta včetně názvu pojmenované roury
- **CATEGORY** - kategorie do které spadá identifikace (nutné pouze v případě aktivní lokální databázi)

Naimportovaný modul se nesmí ukončit po dobu života skeneru. Ukončení modulu je bráno za chybu skeneru a je ukončeno spojení s tímto skenerem a nahlášení chyby.

3.2 Signalizace aktivity

Signalizaci bylo nutné navrhnout, aby bylo umožněno kontrolovat běh identifikačního programu, který není uživatelsky kontrolovatelný. Navržená signalizace aktivity je dvouúrovňová. V první řadě jsou využity tři LED diody zelená, oranžová/žlutá, červená, pro signalizaci normálního provozu, provoz s chybou a zastavení systému. Pro připojení se využívá GPIO⁷. Stav skenerů je kontrolován každých deset sekund. Jednotlivé stavy a jejich popis uvádí tabulka 1.

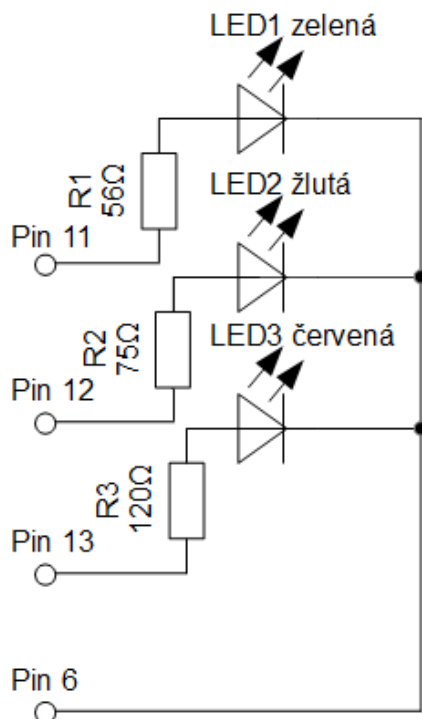
Další úroveň signalizací stavu je zápis zpráv do souboru tzv. logovací soubor. Zde se ukládají všechny hlavní stavy systému. Mezi ně patří například úspěšné, neúspěšné připojení ke skeneru, připojení k databázi a další. Při případné chybě systému se dá jednoduše diagnostikovat co se stalo za chybu. Každý řádek logovacího souboru je ve formátu čas vzniku události, kategorie chyby (informační, výstraha, závažná chyba), popis události. Do kategorie výstraha, spadají takové chyby které nezapříčiní zastavení chodu programu. Jedná se o chyby zálohování databáze a upozornění že ne všechny skenery jsou v provozu.

⁷obsahuje programovatelné vstupy/výstupy s 3.3V logikou, sběrnice, napájení a další

Tabulka 1: Popis signalizace stavů pomocí LED diod

Zelená	Oranžová	Červená	Popis
svítí	nesvítí	nesvítí	Systém je v normálním provozu, bez chyb
svítí	bliká	nesvítí	Systém je v provozu, ale došlo k chybě. Může se jednat o chybu zálohování databáze, nebo některý ze skenerů skončil chybou, ale zbývající skenery jsou v provozu.
nesvítí	nesvítí	svítí	Systém je mimo provoz. Všechny definované skenery skončily chybou, nebo je systém ukončen z jiného závažného důvodu.

Obrázek 1: Schéma zapojení signalizačních LED diod



3.3 Bluetooth skener

Jak již bylo zmíněno, mezi hlavními požadavky bylo dodržení času identifikace 2 s. Při prováděných pokusech pomocí jednoho bluetooth adaptéru, nebyla identifikace spolehlivá. Pro zlepšení spolehlivosti bylo nutné využít dva a více bluetooth adaptéry. Ke správnému chodu bylo nutné navrhnout algoritmus, který se bude starat o synchronizaci skenování na jednotlivých bluetooth adaptérech. Pokud by bylo skenování na bluetooth adaptérech puštěno ve stejný okamžik, bude výsledek srovnatelný s použitím jednoho bluetooth adaptéru. V průběhu návrhu systému a testování byly navrženy dva rozdílné algoritmy, které ve výsledku spolu spolupracují, pro dosažení spolehlivosti a času identifikace během 2 s.

K samotnému získávání MAC adres zařízení, je využíván příkaz *scan* z programu *hcitool*, dostupný na linuxových distribucích v balíčku *bluez*.

Při nasazení navrhovaného systému, nesmí být využíváno více skenerů využívající bluetooth, toto nastavení by vedlo k chybě *Bluetooth adaptér neodpovídá*⁸. Tato chyba je zapříčiněna tím že algoritmy se snaží využít všechny dostupné bluetooth adaptéry a při souběhu více požadavků na jeden bluetooth adaptér, přestane vracet výsledky v požadovaném čase. Tolerance je 60 sekund, aby bylo umožněno krátkodobého využití, například na spárování navrhovaného systému se zařízením zaměstnance. Některé typy telefonů a jejich verze operačních systémů umožňují být viditelné pro spárované zařízení.

3.3.1 Půlení intervalu

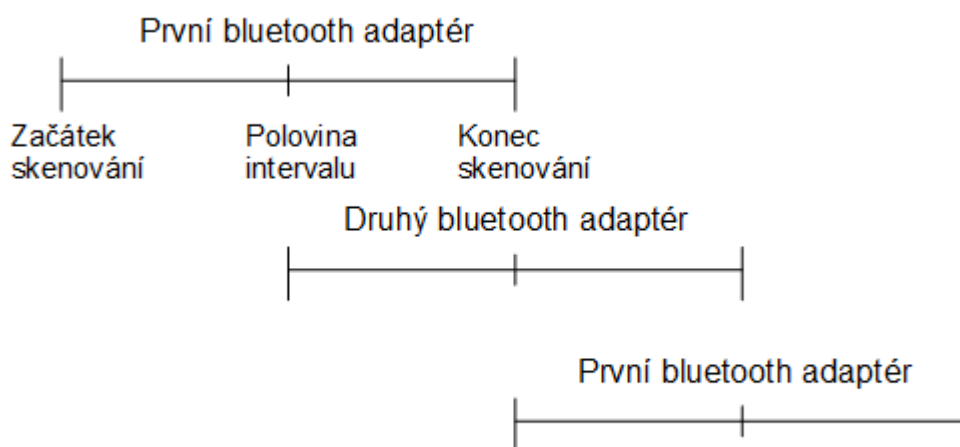
Princip algoritmu si předvedeme na dvou bluetooth adaptérech. Algoritmus využije všechny bluetooth adaptéry v našem případě jsou to dva, celkový skenovací čas (optimální čas je 4s) vydělí počtem aktivních bluetooth adaptéru. Spustí skenování na prvním bluetooth adaptéru, po uplynutí vypočítané doby (v našem případě 2 s) spustí skenování na dalším bluetooth adaptéru, pokud další existuje, v opačném případě se vrací k prvnímu, tento postup je stále opakován. V našem případě se spustí skenování na druhém bluetooth adaptéru, po uplynutí dobu již dojde k vrácení k prvnímu bluetooth adaptéru, který již dokončil skenování a vrátil výsledek. Pro lepší názornost je postup znázorněn na obrázku 2.

Tento algoritmus má vyšší spolehlivost než tomu bylo u použití jednoho bluetooth adaptéru. Avšak spolehlivost stále nedosahuje alespoň 90%. Aby byl zaměstnanec zachycen za 2 s musí se identifikovat v druhé polovině intervalu. Zvyšováním počtu bluetooth adaptéru zvyšuje pravděpodobnost, ale má to negativní vliv na pořizovací cenu a spotřebu.

Vzhledem k uspokojujícím výsledkům je algoritmus i nadále využíván. Využití má na místech kde není potřeba striktně dodržet čas identifikace během 2 s. Modul se nazývá *any_count_bt* a je umístěný ve složce *modules*.

⁸přeloženo z angličtiny Bluetooth adapter not respond

Obrázek 2: Princip algoritmu půlení intervalu



3.3.2 Rychlá identifikace

Výše uvedený algoritmus má za nevýhodu že je nutné skenovat pevně danou dobu. Čímž se nedá zaručit aby identifikace proběhla během 2 s. Myšlenka tedy je, pokud je v dosahu bluetooth zařízení, měl by skener vrátit jeho MAC adresu bez čekání na konec skenovacího intervalu. Tuto myšlenku lze realizovat pomocí nastavení počtu odpovědí. Příkaz *scan* z programu hcitool, umožňuje nastavit potřebný počet odpovědí, po dosažení počtu odpovědí je vrácená MAC adresa zařízení zaměstnance. Pro dosažení identifikace za 2 s je optimální nastavení dvě odpovědi. Při tomto nastavení je identifikace v rozmezí 2-3s a nezahluje se systém. Avšak pokud bluetooth adaptér dostane od zařízení požadované odpovědi, je potom vráceno již v kratším čase, čímž již nedetekuje jiné zařízení. Tento nedostatek je řešen spoluprací výše navrženého algoritmu půlení intervalu. Navržený algoritmus tak splňuje identifikaci zaměstnance během 2 s a zaručuje i zaznamenání více zaměstnanců v jeden čas. Skener je umístěn ve složce modules s názvem *fast_scan_bt*.

3.3.3 Vracení výsledků

Při využívání skeneru rychlá identifikace, dochází k velkému počtu požadavků k ověření identifikace stejného bluetooth zařízení, avšak k tomuto dochází i u ostatních skenerů využívající bluetooth, jelikož skenovací čas je nastaven na 4s. Pokud by měl bluetooth adaptér dosah 10m, člověk tuto vzdálenost ujde⁹ přibližně za 10s. Za 10s bude zaměstnanec minimálně 2x zachycen během průchodu daným prostorem. Aby se omezili požadavky na identifikaci a nedocházelo k zbytečnému zatěžování systému, navrhl jsem třídu která tento problém řeší. Jedná se

⁹Průměrná rychlost chůze člověka je 5km/h

o datovou strukturu slovník[5]¹⁰, kde jako klíč se používá MAC adresa zařízení a hodnotou pak je čas kdy bylo zařízení zaznamenáno. Bluetooth zařízení která jsou zaznamenána ve slovníku jsou ignorovány. Bluetooth zařízení která byla zaznamenána před více jak 10s jsou ze slovníku odstraněna a mohou znovu žádat o identifikaci.

3.4 Síťová komunikace

Získanou identifikaci zaměstnance je možné poslat pomocí protokolu TCP, nebo UDP. IP adresa a port je možné nastavit v konfiguračním souboru v sekci *Network interface*. Pokud bude získáno více identifikací zaměstnanců v jeden čas, budou tyto identifikace odesílány po jedné.

3.5 Záloha databáze

Zálohování databáze je funkční pouze v případě aktivní lokální databázi. Zálohování probíhá především na lokální úložiště. Zálohování databáze se provádí v definovaném časovém intervalu, v konfiguračním souboru je interval uložen v sekundách. K vytváření zálohy databáze je využíván příkaz *mysqldump* dostupný na linuxových distribucích v balíčku *mysql-server*. Takto vytvořená záloha obsahuje všechny potřebné příkazy pro obnovu dat a struktury tabulek. Název vytvořeného souboru je automaticky generován, skládá se z prefixu *backup*, data a času vytvoření. Například: záloha provedená dne 20.5.2016 v 13:55:15 bude mít název souboru *backup_200516135515.sql*. K nastavení zálohování databáze v konfiguračním souboru slouží sekce *Backup database*. V konfiguračním souboru lze nastavit metodu zálohování na síťové úložiště.

Implementovaná metoda zálohování na síťové úložiště je pomocí FTP. Tento modul je možné nastavit pod jménem *ftp_access* umístěné ve složce *modules*. V konfiguračním souboru lze u této metody nastavit IP adresu síťového úložiště, uživatelské jméno a heslo, cestu adresáře na síťovém úložišti. Uživatelské jméno a heslo může zůstat nevyplněné v případě anonymního připojení. Využívaný port je 21.

```
1 mysql -u root -p dochazka < backup_200516135515.sql
2
3 #u - neboli user, uživatel sloužící k přihlášení do databáze
4 #p - neboli password, po potvrzení příkazu je uživatel požádán o
5 zadání hesla
6 #dochazka - databáze do které se mají obnovit data
```

Zdrojový kód 6: příklad obnovení databáze ze souboru, v Linuxu pomocí shell

¹⁰Někdy označované za asociativní pole, anglicky dict. Rozdíl mezi seznamem a slovníkem je způsob definování klíče. U standardně využívaných seznamů jsou klíče definované celými čísly v rozsahu 0 až n, slovníky mohou mít za klíč libovolné řetězce, čísla, atd.

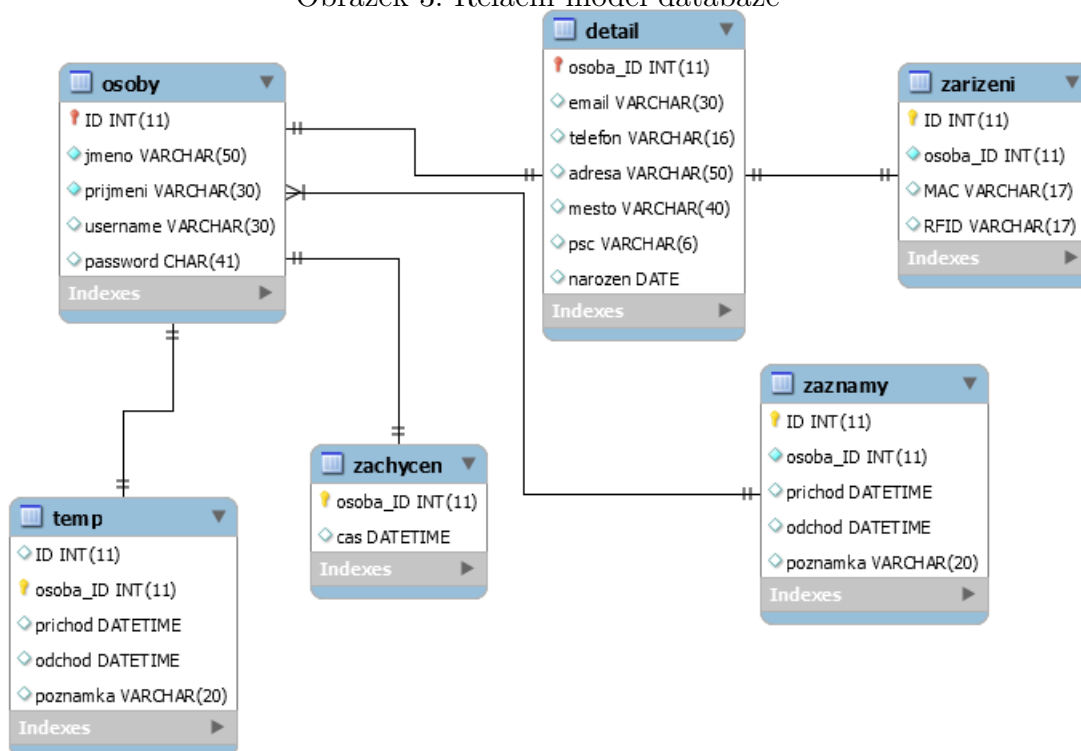
4 Databáze

Databáze navrhovaného systému se skládá ze dvou databází. Hlavní databáze slouží pro ukládání veškerých informací o zaměstnancích a jejich záznamů příchodu a odchodu. Její název je *docházka*. Ztrátu dat nechtěným smazáním, zabraňuje druhá databáze, která slouží jako archiv smazaných zaměstnanců. Tato databáze se jmenuje *archiv*.

4.1 struktura

Níže si popíšeme pouze strukturu hlavní databáze, archiv obsahuje pouze část hlavní databáze a to tabulky osoby, detail a zařízení. Hlavní databáze se skládá z tabulek osoby, detail, zařízení, temp, zachycen a záznamy. Osoby a záznamy jsou ve vztahu 1:N, aby se umožnilo pro jednoho zaměstnance více záznamů příchodů a odchodů. Ostatní tabulky jsou ve vztahu s tabulkou osoby 1:1.

Obrázek 3: Relační model databáze



4.1.1 Osoby

Do této tabulky se ukládají základní informace o zaměstnancích, jako je přiřazené identifikační číslo, jméno, příjmení. Pokud je zaměstnanci umožněn přístup do webové aplikace je zde uloženo jeho uživatelské jméno, minimální délka jsou čtyři znaky, toto omezení je hlídáno ve webové aplikaci. Uživatelské jméno musí

být unikátní, není povoleno stejné uživatelské jméno pro více zaměstnanců. Heslo je uloženo jako 41 bytový hash, pro hashování je využit vnitřní příkaz databázového systému MySQL. Hesla zahashovaná tímto příkazem začínají *.[7]

```
1 SELECT PASSWORD("heslo");  
2  
3 #Výsledkem bude hash - *AADB89769FF364D8537700AE065872FDA6E980F3
```

Zdrojový kód 7: příklad hashování hesla v MySQL

4.1.2 Detail

Do této tabulky se ukládají rozšířené informace o zaměstnancích jako je email, telefon, adresa ve formátu ulice a číslo popisné, město, psč a narozen. Jak jsem již zmínil tato tabulka je ve vztahu 1:1 s tabulkou osoby. To znamená že jednomu zaměstnanci patří jeden řádek této tabulky.

4.1.3 Zařízení

Tabulka zařízení obsahuje identifikační čísla zařízení které vlastní zaměstnanec. Tabulka je oddělena, aby rozšíření systému o další typy identifikačních zařízení, probíhalo bez dalších zásahů do navrženého systému. Toho především využívá webová aplikace, kde se vypisují automaticky všechny typy identifikačních zařízení.

4.1.4 Záznamy

Tabulka záznamy obsahuje čas a datum příchodu a odchodu zaměstnance. Čas a datum jsou datového typu datetime. Další důležitou položkou je poznámka, sloužící jako popis, nebo důvod nepřítomnosti konkrétní den. Tabulka osoby je ve vztahu s tabulkou 1:N, jeden zaměstnanec má více záznamů v této tabulce.

4.1.5 Temp

Příchod a odchod je zapisován postupně, nikoliv v jeden čas, tento úkon je časově náročný a to především na výpočetně slabším zařízení. Časová náročnost závisí především na přístup do tabulky záznamy jejichž počet záznamů dosahuje i několik tisíc. Vyhledání konkrétního zaměstnance s prázdným odchodem a zaznamenání odchodu může trvat i v řádech jednotek sekund. Z tohoto důvodu jsem navrhl tabulku temp, jednomu zaměstnanci patří maximálně jeden záznam v této tabulce. Počet záznamů v tabulce temp je roven, nebo menší počtu zaměstnanců. Hledání požadovaného záznamu je časově méně náročné a po zaplnění příchodu a odchodu daného zaměstnance je záznam přesunut do tabulky záznamů. Dosažený čas tohoto řešení je v řádech desetin sekundy.

4.1.6 Zachycen

Aby nedocházelo k opakovaným záznamům příchodu/odchodu zaměstnance, při průchodu, nebo zdržení ve snímaném prostoru udržuje se tabulka s posledním zaznamenaným časem. Při identifikaci zaměstnance je nahlédnuto do tabulky zachycen a zkontrolován čas. Pokud je rozdíl mezi časy větší jak 10 minut, příchod/odchod je zaznamenan a čas v tabulce zachycen nahrazen. Samostatná tabulka pro tento účel je ze stejného důvodu jako výše uvedený.

5 Webová aplikace

Webová aplikace slouží především jako informační, nikoliv pro účely účetnictví. Tato implementace vyplývá z požadavků OLC Systems s. r. o. Firma již má zavedenou svou databázi zaměstnanců, která splňuje všechny požadavky, včetně požadavků potřebné k účetnictví. Navržená webová aplikace slouží pouze jako základ pro další možné rozšíření a využití i v jiných firmách. Důraz byl tak kladen především na konfiguraci systému, aby umožňoval kompletní nastavení systému.

Webová aplikace je implementovaná v čistém PHP ve verzi 5.5, rychlost a zabezpečení je pro nás dostačující.

Z důvodu využití webové aplikace pouze pro přihlášené uživatele, jako úvodní obrazovka je přihlašovací formulář. Ověřování uživatele probíhá ve dvou fázích, v první fázi je ověřeno uživatelské jméno a heslo s údaji uloženými v konfiguračním souboru, pokud přihlašovací údaje nesouhlasí pokračuje se druhou fází. Ve druhé fázi pro kterou je nutné aktivovaná databáze se ověřují uživatelské jména a hesla zaměstnanců v tabulce osoby.

K automatickému odhlášení dojde při době nečinnosti delší jak 60 minut. Uživatel o tomto odhlášení je informován při aktualizaci stránky, nebo při přístupu na jinou stránku.

5.1 Navigační menu

Navigační menu je umístěné na levém okraji stránky. Toto umístění bylo zvoleno z důvodu orientace obsahu na výšku, na šířku není příliš prostorově náročný. Mezi nejrozšířenější úhlopříčky monitorů využívané u počítačů a notebooků jsou 15" a vyšší při nastaveném vertikálním rozlišení 768 pixelů, nebo vyšší. Obsah je v tomto případě zobrazitelný bez nutnosti scrollování stránky.

Obsah menu je měněn na základě nastavení a na základě typu přihlášení. Aby se zamezilo přístupu k nedostupným stránkám k aktuálnímu nastavení, a nebo ke stránkám ke kterým nemá uživatel oprávnění. Tyto zobrazení jsou tři, v první řadě se jedná o běžného uživatele, tento uživatel má přístup pouze k přehledu a k omezenému nastavení. V druhém případě se jedná o přístup administrátora s deaktivovanou databází, u toho přístupu je zamezen přístup ke stránkám týkající se prací s databází (přehled, evidence). Třetím případem je

přihlášení administrátora s aktivovanou databází, v tomto případě je umožněn přístup ke všem stránkám.

5.2 Popis funkcionality administrátora

Administrátor webové aplikace je pouze jeden. Administrátorovi je jako jedinému uživateli umožněno úplné řízení webové aplikace a editaci záznamů evidence zaměstnanců. Přístup je nutný zabezpečit i při deaktivované databázi, z tohoto důvodu jsou přístupové údaje uloženy v konfiguračním souboru navrhovaného systému. Při deaktivované databázi není nutné instalovat žádné potřebné balíky pro běh MySQL serveru, z toho důvodu není možné využívat databázi pro další využití.

Obrázek 4: Webová aplikace v režimu administrátor s aktivovanou databází

Obecné	Odpracované hodiny	Poznámky
Jméno:	<input type="text" value="Martin"/>	
Příjmení:	<input type="text" value="Černý"/>	
Uživatelské jméno:	<input type="text" value="martin"/>	
Datum narození:	<input type="text" value="16.02.2016"/>	
Adresa:	<input type="text" value="Čechůvky 40"/>	
Město:	<input type="text" value="Prostějov 6"/>	
PSČ:	<input type="text" value="79811"/>	
E-mail:	<input type="text"/>	
Telefon:	<input type="text"/>	
MAC:	<input type="text" value="90:E7:C4:F4:A7:04"/>	
RFID:	<input type="text"/>	

5.2.1 Přehled

Tato stránka je dostupná pouze pokud je aktivovaná lokální databáze. Pro zobrazení informací o zaměstnancích a jejich příchodech/odchodech je navržena stránka přehled. Pro lepší přehlednost je stránka rozdělena na tři podstránky. Využívá se designu záložek, tento design se využívá i na dalších stránkách, kde je potřeba rozdělit hlavní stránku na další podstránky.

Výběr zaměstnanců je implementován pomocí html elementu select. Počet zobrazených elementů, v tomto případě počet zaměstnanců je měněno za pomocí jQuery v závislosti na velikosti zobrazované plochy pro nejvyšší možné využití plochy.

5.2.1.1 Obecné

Podstránka obecné je výchozí podstránkou stránky přehled. Slouží pouze pro zobrazení údajů zaměstnance nikoliv pro úpravu. Data se získávají z tabulek osoby, detail a zařízení. Jak již jsem zmínil, pro usnadnění rozšíření systému o další typy identifikací se na této stránce vypíše všechny atributy tabulky¹¹ týkající se typů identifikace.

5.2.1.2 Odpracované hodiny

Pro zobrazení dat z tabulky záznamy se využívá podstránka odpracované hodiny. Zobrazované data jsou pouze takové které obsahují příchod i odchod, z toho vyplývá že není možné zobrazit čas právě přítomného zaměstnance (obsahuje pouze čas příchodu). Na této podstránce je možné pouze data zobrazit, nikoliv upravovat. Zobrazované data jsou umístěny v tabulce se strukturou: příchod, odchod, čas, poznámka. Čas zde určuje rozdíl mezi odchodem a příchodem, v souvislosti s docházkovým systémem čas znamená počet odpracovaných hodin. Zobrazený čas je ve formátu hh:mm. Pro filtraci zobrazených dat se zde využívá filtrace za dané časové období. Na výběr je několik předdefinovaných hodnot: 1 den, 1 týden, 2 týdny, 1 měsíc, 3 měsíce, 6 měsíců, 1 rok a nebo všechny. Výběr je implementován pomocí html elementu select, se zobrazenou jednou položkou. Pokud ve zvoleném časovém období nebyl nalezen žádný záznam, je o tom uživatel informován hláškou „Ve zvoleném časovém období nebyl nalezen žádný záznam“. Za vybrané časové období se počítá celkový odpracovaný čas, tento údaj se zobrazuje ve spodní části této podstránky.

5.2.1.3 Poznámky

Přidávání nových poznámek a jejich editace je implementovaná na stránce poznámky. Stejně jako u podstránky Odpracované hodiny, i zde je možné filtrovat data za dané časové období. Poznámky je možné vepsat přímo do tabulky k vybranému řádku. Pro uložení změn v tabulce slouží tlačítko *Uložit*. Po zobrazení hlášky o úspěšném, nebo neúspěšném uložení poznámek bude uživatel přesměrován zpět. Přesměrování zabraňuje opakovanému odeslání téhož požadavku. Tlačítko *Storno* slouží pro zrušení neuložených úprav poznámek.

Další implementovanou funkcionalitou je vložení nového řádku. Pokud je potřeba vložit nový řádek s poznámkou z důvodu nepřítomnosti, například návštěva lékaře, školení a podobně. Tato funkce je dostupná pomocí posledního řádku, ve spodní části podstránky. K výběru data slouží html element *input* typu *date*, dalším vstupem je vlastní text poznámky. Tlačítkem *Vložit*, proběhne vložení poznámky do databáze. Takto vytvořená poznámka má rozdílný tvar od záznamu příchodu a odchodu. U ručně vloženého data s poznámkou se zobrazuje pouze jedno datum, bez hodin.

¹¹Atributy tabulky jsou jména sloupců tabulky

5.2.2 Nastavení

Stránka nastavení slouží administrátorovi pro změnu hodnot v konfiguračním souboru navrhovaného systému. Absence displeje a uzavření zařízení na uživatelsky nedostupném místě, neumožňuje konfiguraci přímo na zařízení. Z toho důvodu je nutné řešit konfiguraci vzdáleně, další možností je využít vzdálený přístup mezi které patří například SSH.

Výhodou využití navržené webové aplikace je automatické odeslání informace do identifikačního programu o změně dat v konfiguračním souboru. Ke komunikaci mezi identifikačním programem a webovou aplikací se využívá již zmíněná pojmenovaná roura s neblokujícím režimem, reakce není nutná v okamžiku odeslání zprávy. Soubor pojmenované roury s názvem *commands* je umístěn v kořenu umístění identifikačního programu.

Stránka nastavení je pro lepší přehlednost a orientaci rozdělená na tři tématické podstránky.

5.2.2.1 Přístup

Úvodní podstránkou nastavení je Přístup. Slouží pro změnu přístupových údajů pro administrátora. V konfiguračním souboru se jedná o sekci *Root access*. Administrátorovi je umožněno změnu hesla, ale i přihlašovacího jména. Ověření probíhá na základě hesla původního. Heslo stejně jako uživatelské jméno nesmí být prázdné a musí obsahovat minimálně jeden nebílý znak. Uživatel je informován chybovou hláškou, pokud zadaná hodnota není ve správném tvaru. Po uložení dojde k automatickému přesměrování zpět na stránku.

5.2.2.2 Záloha

Podstránka záloha má dvě části. První část (Obecné) obsahuje nastavení konfiguračního souboru ze sekce *Backup database*. Nastavení umožňuje nastavení intervalu zálohování ve formátu: ss, mm:ss, nebo hh:mm:ss. V případě zapsání intervalu v jiném, nepovoleném formátu je uživatel upozorněn hláškou a je zabráněno uložení, takto se chovají i ostatní položky u kterých je nutno dodržet určitý formát, aby se zabránilo chybám systému. Další položky nastavení souvisí s druhou částí, Metoda. Položka metoda obsahuje název použitého modulu, bez přípony (.py), položka cesta obsahuje cestu k modulu.

Druhou částí (Metoda) je konfigurace použité metody, pokud obsahuje data uložené v konfiguračním souboru. Existuje-li v konfiguračním souboru název sekce, shodující se s názvem metody, jsou automaticky načtena data z této sekce. Pokud sekce nebyla nalezena je ji možné v této části vytvořit včetně potřebných dat. Všechny načtená data je možné upravovat, upravovat je možné jak klíč tak i hodnotu. Přidávat lze libovolný počet řádků, pokud je řádek potřeba vymazat, je potřeba vymazat zadané údaje jak z klíče, tak i z hodnoty. Po uložení, nebo po obnově stránky (pomocí tlačítka Obnovit) proběhne kontrola klíčů, zda-li neobsahují reprezentaci IP adresy, nebo portu. V případě že shoda byla nalezena budou se hodnoty validovat na správnost zadaného formátu. Reprezentace

je kontrolována na základě předpokládaného slovního spojení pro pojmenování položky. U IP adresy jsou to slova: `ipaddress`, `ip`, `ipv4`, `ipadresa`, `ipaddr`. Pokud klíč bude pojmenován jedním z těchto slov, hodnota musí být ve formátu `xxx.xxx.xxx.xxx`, kde `xxx` je číslo 0-255. Pro port jsou definována slova: `port`, `srcport`, `dstport` a poté hodnota musí být celé číslo větší než nula. Po uložení je uživatel informován o úspěšném, nebo neúspěšném uložení a přesměrován zpět na podstránku.

Po uložení je odeslána informace do identifikačního programu o změně nastavení zálohování. Do pojmenované roury `commands` se zapíše „BACKUP“. Po reakci identifikačního programu, jsou signalizační LED diody nastaveny do normálního provozu, v případě že signalizovali chybu, zapříčiněnou zálohováním.

5.2.2.3 Skenování

Poslední podstránkou nastavení je skenování. Tato podstránka je rozdělena na tři samostatné části.

V první části je možné nastavit přístup k lokální databázi. Heslo databáze nesmí být prázdné. Nastavení se ukládá do sekce konfiguračního souboru *Database*.

Druhá část slouží k nastavení síťové komunikaci, která se využívá na odesílání identifikací zaměstnanců. V konfiguračním souboru se jedná o sekci *Network interface*. Nastavení obsahuje volbu protokolu, na výběr je ze dvou možností UDP, nebo TCP. Další položkou je nastavení cílové IP adresy, formát IP adresy je validován. Poslední položkou je nastavení portu, port musí být celé číslo větší než nula.

Poslední částí podstránky je seznam definovaných aktivních skenerů. Pro definování skeneru se využívají dvě vstupní textové pole. První vstupné textové pole se využívá pro zadání názvu modulu, název modulu se shoduje s názvem souboru bez přípony (`.py`). Druhé vstupní textové pole popisuje cestu k modulu.

Po potvrzení nastavení je odeslána informace o změně nastavení skenování do identifikačního programu. Do pojmenované roury se zapíše „SCANNING“. Identifikační program nastaví signalizační LED diody do normálního provozu, pokud byl systém v chybě zapříčiněné skenováním, spustí/ukončí skenery tak aby spuštěné skenery odpovídali nastavenému seznamu skenerů a znovu načte data nastavení vlastní databáze a síťové komunikace.

5.2.3 Evidence

Pokud je aktivní vlastní databáze, je možné využít funkcionalitu stránky evidence. Na stránce je možné přidat nového zaměstnance, odebrat a nebo upravit údaje u existujícího zaměstnance. Pracuje se s tabulkami databáze osoby, detail a zařízení. V levé části stránky je seznam zaměstnanců, výběrem zaměstnance se načtou veškeré dostupné údaje. Povinné položky formuláře jsou jméno a příjmení, pro kontrolu vyplnění se využívá html atributu `required` a vlastní kontrola za pomocí `jQuery`.

Tlačítko *Nový* umístěné pod výběrem slouží k vytvoření nového zaměstnance,

pokud jsou načteny údaje pro vybraného zaměstnance. Dojde ke zrušení výběru zaměstnance a vyprázdnění obsahu formuláře.

Pro odebrání je nutné vybrat zaměstnance a potvrdit tlačítkem Odebrat. Po odebrání jsou záznamy z tabulek osoby, detail a zařízení daného zaměstnance přesunuty do archivu. Pokud není vybrán žádný zaměstnanec je možnost odebrání neaktivní.

Formulář obsahuje všechny definované zařízení pro identifikaci zařízení, stejně jako na stránce s přehledem jsou načítány všechny atributy z tabulky zařízení. Pro zadávání data narození je využit html element *input* typu *date*. Vytvoření přístupu zaměstnance ke svým datům lze provést vyplněním uživatelského jména, po uložení a zobrazení hlášky o úspěšném vytvoření přístupu, bude zobrazeno vygenerované heslo. Heslo je generované z abecedních znaků a čísel o délce 7 až 11 znaků. Z důvodu zobrazování vygenerovaných hesel, stránka evidence jako jediná není po uložení změn automaticky přesměrovaná zpět a je nutné přejít na jinou stránku ručně. Pokud byl vytvořen přístup zaměstnance k docházkovému systému, je umožněno obnovit zapomenuté heslo zaměstnance. Tato nabídka se zobrazí pod údaji zaměstnance pojmenovaná „Obnovit heslo...“. Další možností je spárování bluetooth zařízení, pokud je vyplněná MAC adresa. Nabídka je zobrazena pod údaji zaměstnance. Při využití bluetooth verze 4.0 a vyšší na obou zařízeních není ke spárování nutný zobrazený pin.¹²

Adresa musí být ve formátu: Název ulice číslo popisné. PSČ je možné zadávat ve dvou povolených formátech xxx xx, nebo xxxxx. Telefonní číslo je možné zadávat včetně předvolby a lze ho psát bez oddělení jednotlivých skupin číslic, nebo s oddělením (xxx xxx xxx, xxxxxxxxx). MAC adresa bluetooth zařízení se zdává ve standardizovaném formátu xx:xx:xx:xx:xx:xx, kde x může být číslo, nebo písmeno.

5.2.4 Log

Pro zobrazení obsahu vytvářeného logovacího souboru identifikačním programem složí stránka log. Tabulka s daty je rozdělena do tří sloupců: čas, typ, zpráva. Typ zde popisuje závažnost zprávy. Význam typu již bylo popsáno v kapitole o identifikačním programu, signalizace aktivity. Aby se usnadnilo hledání chyb, jsou data zobrazovány od nejmladších po nejstarší.

5.2.5 Přítomní zaměstnanci

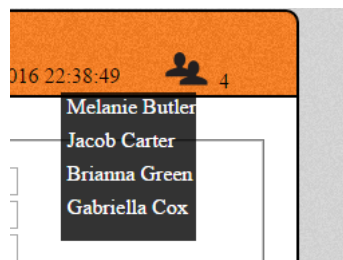
Na podstránce Odpracované hodiny, stránky Přehled se zobrazují pouze kompletní data o příchodu a odchodu, což znamená pokud je zaznamenaný příchod i odchod zaměstnance. Z toho důvodu nejsou informace o právě přítomných zaměstnancích. Implementace zobrazování přítomných zaměstnanců je snadná, protože je přítomna tabulka Temp v databázi, kde jsou uloženy zaměstnanci kteří mají zaznamenaný pouze příchod. Tato funkce je dostupná pouze při aktivní

¹²Párování bluetooth zařízení je funkční pouze při využití programu bluez ve verzi 4.99, nebo starší

vlastní databázi. Přítomní zaměstnanci se zobrazují v hlavičce webové aplikace na pravé straně.

V základním zobrazení se zobrazuje pouze počet přítomných zaměstnanců, po kliknutí na ikonu lze zobrazit seznam jmen zaměstnanců. Po kliknutí na libovolnou část stránky se seznam zavře.

Obrázek 5: Ukázka seznamu přítomných zaměstnanců



5.3 Popis funkcionality běžného uživatele

Běžného uživatele rozumíme přístup zaměstnance pomocí vytvořeného uživatelského jména. Zaměstnanec má přístup pouze ke stránkám přehled a nastavení, obě stránky nabízí pouze vybrané části, oproti administrátorovi.

Stránka přehled nabízí podstránky *Obecné* a *Odpracované hodiny*. Zobrazovaná data jsou pouze pro přihlášeného zaměstnance. Podstránka *Obecné* shrnuje všechny vyplněné údaje o zaměstnanci, údaje nelze měnit. Na podstránce *Odpracované hodiny* může zaměstnanec nahlédnout do svých záznamů o příchodu a odchodu, případně ručně přidaných poznámek. Záznamy lze filtrovat dle definovaných časových období.

Stránka nastavení nabízí pouze změnu vlastního hesla, ověření změny hesla se provádí pomocí hesla původního. Obnovit zapomenuté heslo jde pouze přes stránku administrátora.

Obrázek 6: Webová aplikace v režimu běžný uživatel

Docházkový systém Odhlásit: martin 31. 7. 2016 1:38:57

Přehled Obecné Opracované hodiny

Nastavení

Jméno:

Příjmení:

Uživatelské jméno:

Datum narození:

Adresa:

Město:

PSČ:

E-mail:

Telefon:

MAC:

RFID:

Copyright © 2016 Martin Černý

Závěr

Vyvinutý systém je náhradou za současný docházkový systém ve firmě OLC Systems s. r. o., důraz byl kladen na identifikaci zaměstnance pomocí bluetooth během 2 s. Tohoto bylo dosaženo využitím dvou bluetooth adaptérů, zaručena je i spolehlivost zaznamenání více zaměstnanců ve stejný čas.

Rozšířením navrženého systému je vlastní databáze, pro evidenci zaměstnanců. Databáze je optimalizována pro běh na výkonnostně slabším počítači. K běhu navrženého systému je nutné využít Raspberry Pi, nebo jeho alternativu s kompatibilním rozhraním GPIO využívané pro signalizační LED diody.

Během nasazení docházkového systému do provozu byly zjištěny dva nedostatky. Prvním je dosah bluetooth adaptérů, potíže může způsobovat když sledovaná místnost sousedí s místností s výskytem zaměstnanců. Zde může docházet k nežádoucím záznamům příchodů/odchodů. Řešením může být odstíněním sledované místnosti, nebo úprava antén v bluetooth adaptérech. Druhým nedostatkem je samotná viditelnost zařízení zaměstnance, většina zařízení a jejich operační systémy umožňují viditelnost pouze po manuálním zapnutí, což vyžaduje pro každou identifikaci aktivovat viditelnost. Toto je velmi nepohodlné a téměř ztrácí výhodu využití bezdrátové technologie. Řešením může být dodatečná aplikace do zařízení, která aktivuje viditelnost na základě polohy, nebo dostupnosti firemní wifi sítě.

Navržení systému jako modulárního, je možné využít navržený systém i v jiných firmách, dle jejich požadavků. V dnešní době se již docházkové systémy využívají i ve školách pro záznam příchodů a odchodů žáků, i zde by bylo možné systém využívat.

Conclusions

Newly created system will replace a current attendance system in OLC Systems s. r. o. company. An important insistence has been put on identification of employees by using bluetooth in 2 seconds. This result has been reached by using two bluetooth adapters, with guaranteed reliability register more employees at the same time.

As an extension of suggested system, there is an own database for recording employees . Database is optimised to be use on performance weaker computer. To run suggested system , it is necessary to use Raspberry Pi, or its alternative version with compatible interface GPIO, used for signaling LED diodes.

While using attendance system, there were detected two imperfections. First is the bluetooth adapter coverage. Difficulty can occur if monitored room is next to a room with employees. Then, there is a possibility of an undesirable record of arrival / departure. Solution can be a shielding of monitored room, or an adjustment of the antenna in bluetooth adapters. Second imperfection is visibility of employees devices. Most of the devices and their operating systems allows visibility only after a manual switching on, which requires activation of visibility for each identification. This is very uncomfortable and almost losing advantage of using a wireless technology. A solution can be an additional application into device, which activates visibility based on position, or availability of company wifi network.

Due to designing system as a modular, it is possible to use suggested system in other companies too, according to their requirements. Nowadays, attendance systems are being used also at schools to record arrival and departure of students, so also there would be possible to use the system.

A Postup spuštění docházkového systému

A.1 Potřebné balíky

Níže popsané balíky jsou pro linuxovou distribuci Raspbian, na ostatních operačních systémech se můžou balíky jmenovat odlišně.

- **Webová aplikace** - apache2, php5 (verze 5.5 a vyšší)
- **Vlastní databáze** - php5-mysqldb, mysql-server, python-mysqldb
- **Zálohování databáze pomocí FTP** - ftpd
- **Identifikační program** - python-bluez, bluetooth, bluez

Současné verze operačních systémů pro Raspberry Pi již obsahují některé potřebné balíky pro běh navrženého systému, proto zde nejsou uvedeny. V případě chybějícího balíky bude v logovacím souboru uvedena chyba které balíky schází.

A.2 Instalace docházkového systému

A.2.1 Identifikační program

1. Do umístění `/usr/local/bin/` zkopírovat složku `attendance_system` z archivu umístěného ve složce `bin/` na přiloženém CD.
2. Upravit přístupová práva u konfiguračního souboru (`/usr/local/bin/attendance_system/config.ini`), aby bylo možné provádět úpravy skrze webovou aplikaci.
3. Souboru `/usr/local/bin/attendance_system/main.py` přidat práva pro spuštění.
4. Do umístění `/etc/init.d/` zkopírovat soubor `attendance-system` umístěného ve složce `bin/` na přiloženém CD a přidat práva pro spuštění.
5. V případě potřeby automatického spouštění identifikačního programu po startu systému, spustit program `runatboot.sh` umístěného ve složce `bin/` na přiloženém CD.

A.2.2 Webová aplikace

Do umístění webového serveru zkopírovat složku `dochazka` z archivu umístěného ve složce `bin/` na přiloženém CD.

V případě zobrazení hlášky „Permission denied: access to / 403 Forbidden“ při pokusu o zobrazení webové aplikace, je nutné upravit práva na zkopírované složce včetně obsažených souborů.

A.2.3 Databáze

Při instalaci balíku *mysql-server* je nutné zadat heslo. Identifikační program a webová aplikace neumožňují prázdné heslo.

Ze složky *bin/* umístěné na přiloženém CD je nutné nainportovat do databáze soubory *struktura_archiv.sql* a *struktura_dochazka.sql*. Oba tyto soubory obsahují potřebné příkazy pro vytvoření potřebných databází a struktur tabulek.

Ve výchozím nastavení v konfiguračním souboru je databáze deaktivovaná a nastavené výchozí přístupové údaje, pro správný chod je nutné tyto údaje upravit.

A.2.4 Párování zařízení

Uvedený postup je pouze pro verze balíku *bluez* 4.99, nebo starší. Nové verze již neobsahují používaný program a není možné využívat funkci implementovanou ve webové aplikaci.

1. Zjistit jméno uživatele webového serveru pomocí stránky:
<http://localhost/dochazka/index.php?page=whoami>, localhost lze nahradit za IP adresu Raspberry Pi.
2. Zjistit umístění programu *bluetooth-agent* pomocí příkazu: `which bluetooth-agent`
3. Do souboru *sudoers* umístěného ve složce */etc* přidat řádek:
uživatel ALL= /umisteni/bluetooth-agent

A.3 Spuštění docházkového systému

K ručnímu spuštění identifikačního programu slouží příkaz `sudo /etc/init.d/attendance-system start`, v případě že je již identifikační program spuštěn, je o tom uživatel informován. Pro zastavení identifikačního programu slouží příkaz `sudo /etc/init.d/attendance-system stop`. Webová aplikace je automaticky spouštěna při startu systému.

Výchozími přístupovými údaji do webové aplikace jsou:

- uživatelské jméno: root
- heslo: 1234

B Obsah přiloženého CD/DVD

bin/

Kompletní adresářová struktura webové aplikace (v ZIP archivu) pro zkopírování na webový server a kompletní adresářová struktura identifikačního programu (v ZIP archivu) pro zkopírování do zařízení. Dále obsahuje

soubory struktur databází: *struktura_archiv.sql* a *struktura_dochazka.sql* a soubory sloužící pro spuštění identifikačního programu: *runatboot.sh* a *attendance-system*.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty identifikačního programu a webové aplikace se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory adresářové struktury pro zkopírování na webový server a do zařízení.

readme.txt

Instrukce pro instalaci a spuštění programu, včetně všech požadavků pro jeho bezproblémový provoz. A instrukce pro nasazení webové aplikace na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Navíc CD/DVD obsahuje:

data/

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

Literatura

- [1] *Generátor jmen* 2016 [cit. 2016-07-01]. Dostupné z <http://www.ipsum.me/>
- [2] *SQL Injection* [online]. 1996-2016 [cit. 2016-08-02]. Dostupné z: http://www.w3schools.com/sql/sql_injection.asp
- [3] *PHP Prepared Statements* [online]. 1996-2016 [cit. 2016-07-19]. Dostupné z: http://www.w3schools.com/php/php_mysql_prepared_statements.asp
- [4] *Python* [online]. 2016 [cit. 2016-07-19]. Dostupné z <http://www.itnetwork.cz/python/zaklady>
- [5] *Slovníky* [online]. 2005-2015 [cit. 2016-07-19]. Dostupné z <http://www.sallyx.org/sally/python/python3b.php#hash>
- [6] *PHP* [online]. 2010-2015 [cit. 2016-07-19]. Dostupné z http://phgame.cz/PHGame_serialy/serialy/php/prvni-dil-o-php-historie-a-budoucnost
- [7] *Password Hashing in MySQL* 2016 [cit. 2016-07-20]. Dostupné z <http://dev.mysql.com/doc/refman/5.7/en/password-hashing.html>
- [8] *Nejčastější rozlišení monitoru* 2009-2016 [cit. 2016-07-20]. Dostupné z http://ciselnik.artega.cz/nejcastejsi_rozliseni_obrazovky.php