



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

<b>Název:</b>	Automatický docházkový systém s rozpoznáváním tváří
<b>Student:</b>	Filip Bořil
<b>Vedoucí:</b>	Ing. Viktor Černý
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Webové a softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2019/20

### Pokyny pro vypracování

Vytvořte webový portál, který bude zajišťovat automatický docházkový systém s rozpoznáváním tváří.

- Zmapujte dostupné knihovny pro rozpoznávání tváří a jednu z nich vyberte pro vaši aplikaci.
- Vyberte vhodnou technologii pro webové rozhraní vašeho portálu.
- K řešení nevytvářejte vlastní technická zařízení.
- Celé řešení navrhnete s modulární strukturou, která umožní snadnou aktualizaci nebo výměnu jednotlivých částí systému.

Výsledný webový portál bude umět následující:

- podpora uživatelských rolí
- kontrola historie docházek jednotlivých zaměstnanců
- zadávání dovolených samotnými zaměstnanci
- zobrazení statistik a přehledů z dat docházek zaměstnanců
- bezpečné zpřístupnění dat aplikacím třetích stran pomocí REST API

Řešení bude řádně otestováno.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 30. září 2018





**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

Bakalářská práce

# **Automatický docházkový systém s rozpoznáváním tváří**

*Filip Bořil*

Katedra softwarového inženýrství  
Vedoucí práce: Ing. Viktor Černý

10. ledna 2019



---

## Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce za ochotu a zájem, přítelkyni za trpělivost a také svojí rodině za podporu při studiu.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 10. ledna 2019

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2019 Filip Bořil. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Bořil, Filip. *Automatický docházkový systém s rozpoznáváním tváří*. Bachelářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.



---

# Abstrakt

Tato bakalářská práce řeší tvorbu docházkového systému, který pro získávání dat docházky využívá automatické rozpoznávání tváří. Systém je určený pro malé a střední firmy pro usnadnění evidence docházky a správu dovolené. Systém není bezpečnostním prvkem a nelze jej použít např. k odemykání dveří.

Systém je tvořený následujícími moduly. Modul pro detekování pohybu a zachycení obrazových dat je navržený pro mikropočítač Raspberry Pi s připojenou kamerou a senzorem pohybu a je naprogramovaný v jazyce Python. Modul pro rozpoznávání tváří je též vyvinutý v jazyce Python a pro rozpoznávání využívá již existujících knihoven. Nejdůležitějším modulem je webový portál, pomocí kterého se systém spravuje, kontroluje se docházka a žádá se o dovolené. Je naprogramovaný v jazyce PHP a využívá frameworku Nette. Moduly mezi sebou komunikují přes REST API a data ukládají do databáze MySQL.

Vytvořený systém lze spolehlivě provozovat. Tváře rozpoznává spolehlivě díky postupnému učení tváří již v průběhu provozu systému. Provozovatel systému musí pouze zajistit vhodné světelné podmínky a umístění kamery.

Práce obsahuje uživatelskou příručku systému včetně screenshotů webového portálu a v příloze lze nalézt příručku instalační.

**Klíčová slova** docházkový systém, rozpoznávání tváří, webový portál, umělá inteligence, hloubkové učení, Raspberry PI, PHP, Python, Nette

---

# Abstract

This bachelor thesis deals with attendance system creation, which uses automatic face recognition. The system is determined for smaller and medium-sized companies to simplify them attendance evidence and holiday management. The system is not a security element and it cannot be used i.e. for unlocking doors.

This system consists of following modules. Module for motion detection and picture data capturation is designed for microcomputer Raspberry Pi with camera and PIR sensor connected and is programmed in Python. Face recognition module is also developed in Python and uses existing libraries for face recognition. The most important module is web portal, which is used for system management, attendance check and holiday application. It is programmed in PHP and uses Nette framework. Modules communicate via REST API between each other and they save data to MySQL database.

Created system can be used reliably. It recognizes faces reliably thanks to progressive faces learning yet during system operation. System operator just have to ensure suitable lighting conditions and camera placement.

Thesis contains user guide including screenshots of web portal and in the attachment installation guide.

**Keywords** attendance system, face recognition, web portal, artificial intelligence, deep learning, Raspberry PI, PHP, Python, Nette

---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza a návrh</b>	<b>5</b>
2.1 Existující řešení . . . . .	5
2.2 Požadavky . . . . .	6
2.3 Případy užití . . . . .	6
2.4 Doménový model . . . . .	8
2.5 Architektura systému . . . . .	12
2.6 Technologie . . . . .	13
2.7 Rozpoznávání tváří . . . . .	18
<b>3 Realizace</b>	<b>21</b>
3.1 Modul kamera . . . . .	21
3.2 Rozpoznávací modul . . . . .	22
3.3 Webový portál . . . . .	24
3.4 Propojení modulů . . . . .	30
3.5 První kroky . . . . .	32
<b>4 Testování</b>	<b>35</b>
4.1 Propojení modulů . . . . .	35
4.2 Rozpoznávání tváří . . . . .	36
4.3 Webový portál . . . . .	36
4.4 Vyhodnocení . . . . .	36
<b>Závěr</b>	<b>39</b>
<b>Bibliografie</b>	<b>41</b>

<b>A</b>	<b>Seznam použitých zkratk</b>	<b>47</b>
<b>B</b>	<b>Obsah přiloženého CD</b>	<b>49</b>
<b>C</b>	<b>Instalační příručka</b>	<b>51</b>
C.1	Kamera . . . . .	51
C.2	Rozpoznávací modul . . . . .	52
C.3	Webový portál . . . . .	53
C.4	Databáze . . . . .	54

---

## Seznam obrázků

2.1	Diagram případů užití . . . . .	9
2.2	ER-diagram entit v databázi . . . . .	11
2.3	Diagram komponent (modulů) . . . . .	12
2.4	Schéma zapojení PIR senzoru k Raspberry PI (zdroj: <a href="https://projects.raspberrypi.org">https:// projects.raspberrypi.org</a> [51]) . . . . .	17
2.5	Rozpoznávání – Nalezení tváří (zdroj: <a href="https://medium.com">https://medium.com</a> [60]) .	19
2.6	Rozpoznávání – Transformace tváře (zdroj: <a href="https://medium.com">https://medium.com</a> [60]) . . . . .	19
2.7	Rozpoznávání – Zakódování znaků (zdroj: <a href="https://medium.com">https://medium.com</a> [60])	20
2.8	Rozpoznávání – Nalezení jména osoby (zdroj: <a href="https://medium.com">https://medium.com</a> [60]) . . . . .	20
3.1	Webový portál – stránka přehled . . . . .	25
3.2	Webový portál – stránka moje docházka . . . . .	25
3.3	Webový portál – stránka moje dovolená . . . . .	26
3.4	Webový portál – detail záznamu (snímku) . . . . .	27
3.5	Webový portál – detail uživatele (pohled manažéra) . . . . .	28
3.6	Webový portál – přidávání kamery do systému . . . . .	29
3.7	Webový portál – nastavení parametrů systému a zpráva o úspěšném uložení . . . . .	30
3.8	Webový portál – údržba (manuální spouštění úkonů) . . . . .	30



---

## Seznam výpisů kódů

C.1	Instalace balíčků a knihoven pro python nezbytných pro provoz modulu kamera . . . . .	51
C.2	Instalace balíčků a knihoven pro python nezbytných pro provoz rozpoznávacího modulu . . . . .	52
C.3	Instalace virtuálních kontejnerů a jejich spuštění . . . . .	53
C.4	Instalace softwaru potřebného pro provoz webového serveru . .	53
C.5	Instalace knihoven potřebných pro provoz webu a nalinkování adresáře www . . . . .	53
C.6	Povolení symbolických adres v Apache . . . . .	54
C.7	Povolení modulu rewrite v Apache . . . . .	54
C.8	Restartování Apache . . . . .	54
C.9	Nastavení práv v databázi pro docházkový systém . . . . .	55





---

# Úvod

V posledním desetiletí se stále více rozvíjí oblast umělé inteligence. Jednou z oblastí, která je založena na umělé inteligenci, je rozpoznávání tváří. Na oblast rozpoznávání tváří dnes existují uspokojivá řešení, která se dají integrovat do celé řady informačních systémů.

Rozhodl jsem se proto vytvořit docházkový systém, který zautomatizuje zaznamenávání docházky v ideálním případě natolik, že nebude nutný žádný ruční zásah, ale zároveň umožní žádání a schvalování dovolené a další doprovodné činnosti. Systém ale nebude bezpečnostním prvkem a nebude možné jej použít např. k odemykání dveří.

Tento systém je určený k nasazení do malých či středních firem, kterým pomůže automaticky evidovat a kontrolovat docházku s nízkými pořizovacími i provozními náklady. Systém bude totiž tvořený jednotlivými moduly, které bude možné jednoduše vyměnit či aktualizovat. Každý modul může být provozován na zvláštním zařízení (například modul pořizující obrazový záznam může být provozován na levném mikropočítači Raspberry pi s kamerou a zbylé moduly, tedy výpočetní modul, databáze a webový server, mohou být provozovány na již existujícím firemním serveru).

V práci se zabývám analýzou, návrhem a implementací všech zmíněných modulů systému a komunikací mezi nimi. Systém přizpůsobím tak, aby nebyl v rozporu s GDPR, ani jinými normami. Naopak se nebudu zabývat samotným procesem rozpoznávání tváří, použiji totiž již existující řešení.

Tato práce dále pokračuje v následující struktuře: Nejprve si vytyčím jednotlivé cíle práce, dále analyzuji a navrhnu výběr technologií a struktury systému, poté popíši realizaci a zprovoznění celého systému, popíši testování a vyhodnotím jej a na závěr shrnu splnění cílů.



---

## Cíl práce

Cílem této práce je navrhnout a implementovat webový portál, který bude automaticky evidovat docházku zaměstnanců a který využije již funkční řešení rozpoznávání tváří. Součástí portálu bude administrace a doprovodná funkcionalita jako například žádání a schvalování dovolené.

Rešeršní část práce si klade za cíl zmapovat dostupné knihovny pro rozpoznávání tváří a jednu z nich vybrat pro aplikaci. S tím se rozumí také pochopit proces rozpoznávání, ale jen v obecné rovině.

Dalším cílem rešeršní části je podrobněji prozkoumat webové technologie, aby bylo snadné systém používat, aby byl rychlý, spolehlivý a dobře udržovatelný. Systém bude ze své podstaty evidovat osobní údaje, takže je nutné se seznámit s právní legislativou jako je např. GDPR.

Cílem praktické části je naprogramovat webový portál pro zaměstnance i manažery a sestavit a naprogramovat zařízení, které bude detekovat pohyb, snímat tváře a odesílat pořízené snímky k rozpoznávání tváří. Ve webovém portále si pak uživatelé budou moci zobrazit docházku vypočtenou z nasbíraných dat.

Výsledný webový portál bude podporovat uživatelské role, umožňovat kontrolu historie docházek jednotlivých zaměstnanců, zadávání dovolených samotnými zaměstnanci, zobrazení statistik a přehledů z dat docházek zaměstnanců a bezpečně zpřístupní data aplikacím třetích stran.

Cílem praktické části je také systém otestovat.



---

## Analýza a návrh

V této kapitole uvedu již existující řešení, která řeší stejný problém jako můj systém a porovnáám je s ním. Dále uvedu seznam funkčních a nefunkčních požadavků, případy užití a architekturu systému. Zdůvodním volbu technologií a srovnám je s ostatními. Krátce uvedu také teorii o rozpoznávání tváří.

### 2.1 Existující řešení

Při hledání podobných systému jsem našel hned několik různých řešení včetně těch od českých tvůrců. Z těch českých firem mohu jmenovat například SAFESCAN [1], FINGERA [2], SYSDO [3] nebo GARDON INT [4], ze zahraničních například MATRIX Comsec Pvt. [5]. Všechny je spojuje to, že se jedná o systémy komerční, vázané na terminály (zařízení obsahující displej, čtečku otisků, kameru. atd.), které firmy samy také prodávají. Některé mají webové rozhraní, některé desktopovou aplikaci, některé mobilní aplikaci. Některé se prodávají za jednorázový poplatek, častější je ale varianta ročního předplatného (8 let provozu vyjde v průměru zhruba na 45 000 Kč bez DPH).

Čím se můj systém od těchto řešení liší, je veliká přenositelnost – modulární struktura, kdy každý modul může běžet na různých zařízeních, z toho modul kamera je naprogramovaný pro populární mikropočítač Raspberry PI [6] s kamerou [7], který je levný (o ceně a dalších parametrech píší v sekci 2.6.6).

Mojí motivací vytvořit tento systém bylo především zapálenost do moderních technologií, jako je umělá inteligence (kterou rozpoznávání tváří využívá), ale také do webových technologií. Byla to zkrátka dobrá příležitost vyzkoušet si naprogramovat větší projekt využívající zmíněné technologie.

## 2.2 Analýza požadavků

### 2.2.1 Funkční požadavky

- Vedení seznamu uživatelů a jejich rolí (zaměstnanec, manažér, admin.)
- Nahrávání fotek tváří a ukládání tváří pořízených kamerou
- Detekce pohybu a pořízení fotografie, na které se spustí rozpoznávání
- Z pořízených dat výpočet docházky
- Manuální editace docházky manažérem a reklamace zaměstnancem
- Žádosti o dovolenou zaměstnanci a schvalování dovolené manažérem
- Zobrazení přehledů a statistik
- Administrace webu včetně přidávání/odebírání kamer přes web
- Zpřístupnění dat aplikacím třetích stran

Funkční požadavky jsou rozepsány detailněji jako případy užití (sekce 2.3).

### 2.2.2 Nefunkční požadavky

- Algoritmy rozpoznávání a porovnávání tváří použít z existující knihovny
- Webový portál naprogramovat ve frameworku
- Řešení navrhnout modulárně – moduly:
  - Detekující pohyb a zachycující obraz
  - Rozpoznávající
  - Webový portál
- Prostředky ke komunikaci mezi moduly:
  - REST API
  - Databáze

## 2.3 Případy užití

### Účastníci

Účastníci jsou rozdělení do tří typů:

- Zaměstnanci
- Manažéři
- Administrátoři (někdy uvedení jako správci)

Mezi účastníky funguje dědičnost práv, tedy všechny případy užití zaměstnance náleží také manažérovi a všechny případy užití manažéra náleží také administrátorovi. Jinými slovy práva zaměstnance má i manažér a administrátor a práva manažéra má i administrátor. Práva jsou definována v následující sekci. Dědičnost ilustruje také diagram na obrázku 2.1. Například manažér si může zažádat o dovolenou a sám si ji pak schválit.

## Akce

Účastníci budou vykonávat následující akce:

- UC1 Zobrazení a reklamace docházky** – Zaměstnanci budou mít možnost zkontrolovat svoji docházku a v případě nesrovnalostí reklamovat daný den. Po měsících budou moci procházet celou historií a ke každému měsíci uvidí souhrn odpracovaných hodin, případně vypočtenou mzdu.
- UC2 Zobrazení a žádání o dovolenou** – Zaměstnanci si budou zobrazit stav jejich dovolené za aktuální rok – počet vyčerpaných dní a celkový nárok na dovolenou. Budou moci požádat o dovolenou a uvidí též historii dovolené a její stav (zamítnuto, čeká na schválení, schváleno).
- UC3 Zobrazení svého profilu** – Zaměstnanci si budou moci zobrazit, které údaje o nich zaměstnavatel eviduje. Na stejné podstránce si bude moci změnit heslo pro přihlášení do systému.
- UC4 Správa vlastních tváří** – Zaměstnanci si budou moci prohlédnout, jaké fotografie s tvářemi, které se systém naučil, jsou evidovány. Tyto fotografie bude moci také mazat nebo nahrávat své vlastní. Jednu z fotek si může nastavit jako profilovou.
- UC5 Vyřizování reklamací** – V případě, že zaměstnanec reklamuje svoji docházku, manažérům se tato informace zobrazí na hlavní stránce a také na podstránce s výpisem všech dosud nevyřízených reklamací, kde si je budou moci prohlédnout a reagovat na ně zamítnutím či schválením s možností případné dodatečné úpravy.
- UC6 Zobrazení a editace historie docházky** – Manažéři si budou moci prohlédnout docházku kohokoliv stejným způsobem, jak to vidí zaměstnanec sám. Navíc bude možné každý jednotlivý den editovat.
- UC7 Zobrazení statistik docházky** – Manažéři budou mít možnost zobrazit si souhrny docházky všech zaměstnanců dohromady ve formě tabulky nebo grafů (například graf celkově odpracovaných hodin všech zaměstnanců).
- UC8 Schvalování dovolené** – Manažéři si budou moci zobrazit výpis žádostí o dovolené od všech zaměstnanců a na stejné podstránce vyplnit formulář o zamítnutí či schválení s případnou dodatečnou editací. Počet nevyřízených žádostí o dovolenou bude též zobrazen na hlavní stránce.
- UC9 Zobrazení historie dovolené** – Manažéři budou moci procházet historií dovolených všech stavů (zamítnutá, čeká na schválení, schválená) a dodatečně je editovat.
- UC10 Zobrazení záznamů zachycených kamerou** – Manažéři si mohou prohlédnout záznamy, které pořídila kamera. Mezi záznamy budou moci filtrovat podle data (od – do) a podle toho, zda a s jakou jistotou byly na snímku rozpoznány nějaké tváře. Každý snímek budou moci zobrazit v detailu, kde budou vypsány také všechny nalezené tváře se jmény, komu patří, a tlačítka na potvrzení/vyvrácení správnosti či reklamaci. V případě potvrzení tváře, u které si systém nebyl jistý, se tato tvář uloží

mezi známé a naučí. Tím se do příště rozpoznávání zpřesní. V případě vyvrácení/reklamace tváře se zobrazí seznam nejpodobnějších osob pro rychlý výběr té správné.

**UC11 Zobrazení seznamu uživatelů** – Manažéři si budou moci zobrazit seznam všech uživatelů s možností filtrování (vyhledávání) a rozkliknout si jejich detail, případně zaznamenanou docházku.

**UC12 Správa údajů a tváří uživatelů** – Manažéři budou moci editovat údaje vedené o uživatelích, včetně fotek s jejich tvářemi, mazat a přidávat je a jednu z nich vybrat jako profilovou. Uživatelům budou moci vygenerovat nové heslo, změnit jim roli nebo je celé rovnou smazat. Dohlédne se na to, aby byl v systému vždy alespoň jeden administrátor.

**UC13 Správa zařízení (kamer a rozpoznávače)** – Administrátor bude moci přidávat, odebírat či měnit kamery, spouštět a zastavovat snímání a spolu s rozpoznávacím modulem je také restartovat či zkontrolovat jejich stav (spuštěno, zastaveno, odpojeno, chyba).

**UC14 Editace parametrů systému** – Administrátoři budou moci měnit parametry systému, kterými se budou nastavovat konfigurace modulů (například zobrazovaná měna v systému).

**UC15 Údržba systému** – Administrátor bude moci spouštět úlohy jako je například dopočet docházky. Některé z těchto úloh (jako například právě výpočet docházky) se budou spouštět pravidelně v předem definovaný čas.

Pro lepší přehlednost případy užití zachytává diagram na obrázku 2.1.

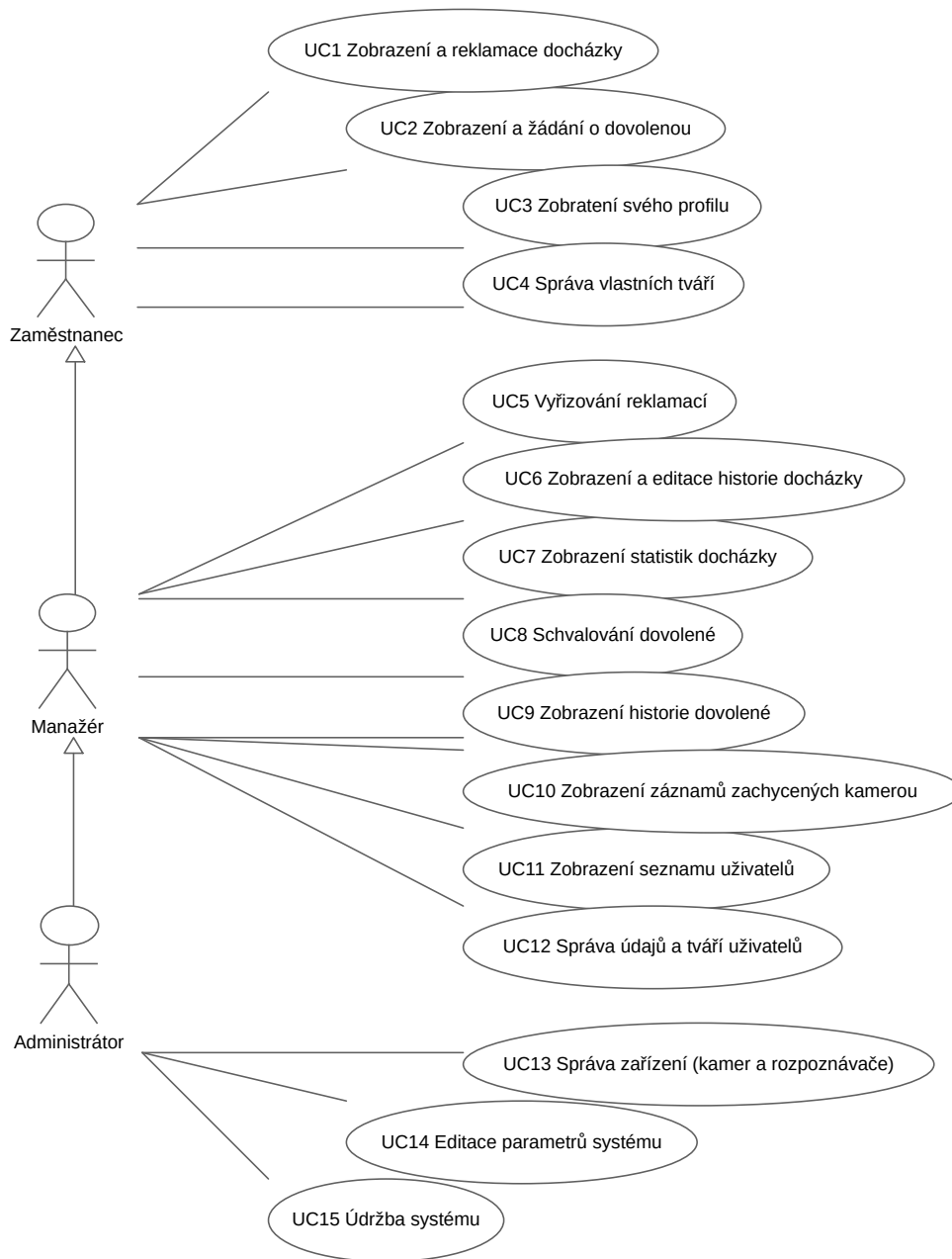
## 2.4 Doménový model

Místo doménového modelu uvedu rovnou databázový model (ER-diagram – obrázek 2.2), protože není o moc složitější než doménový a entity ilustruje stejně dobře.

Stěžejními entitami jsou *user* a *face*. *User* (uživatel) uchovává přihlašovací údaje (*login*, *password\_hash*), jméno, příjmení a další vedené údaje. Každý uživatel vlastní 0 až N tváří (entita *face*) – buď nahrané jako soubor nebo pořízené ze snímání kamerou. *Face* v sobě uchovává *encoding*, tedy kódování, které využívá rozpoznávací modul pro porovnávání s ostatními tvářemi k určení největší schody a tím pravděpodobného vlastníka tváře. Každá tvář má v databázi uloženo 1..N obrázků (*face\_image*) různých rozlišení (*image\_resolution*).

Každý uživatel má právě jednu roli (*role*), která určuje jeho práva v aplikaci. Každý uživatel může mít 0..N docházky (*attendance*) vypočtené algoritmem nebo editovanou manažérem či administrátorem (*updated\_by\_user\_id*). Docházka může být reklamována (entita *reclamation*), která je/není manažérem či správcem schválena (*approved*) s volitelným odůvodněním (*justification*). Každý uživatel má 0..N dovolené (*holiday*), která je nebo není schválena (*approved*) manažérem nebo administrátorem (*confirmed\_by\_user\_id*).





Obrázek 2.1: Diagram případů užití

Informace o pořízených snímcích jsou obsaženy v entitě *log\_camera*, samotné obrázky různých rozlišení (*image\_resolution*) v entitě *log\_camera\_image*. Pokud rozpoznávací modul zpracoval fotografii (*log\_camera.processed*) a našel v ní tváře, uloží je do *log\_face* (samozřejmě s vazbou 0..N k *log\_camera*). Ke

## 2. ANALÝZA A NÁVRH

---

každé tváři pak uloží 0..N vzdáleností (*distance*) s každou známou tváří do entity *log\_face\_distance*.

Zbylé moduly popíší velmi stručně. *contract\_type* – typ smlouvy pro výpočet čisté mzdy, *device* – informace (IP adresa, port,...) o zařízení (*ID* = 1 je rozpoznávací modul, ostatní jsou kamery), *parameter* – definice parametru konfigurace systému a jeho hodnota (*value*), *enum* – definice výčtu parametru, je-li typu (*value\_type*) enum, *public\_holiday* – definice státních svátků (bude-li systém provozován v jiném státě, lze definici změnit).



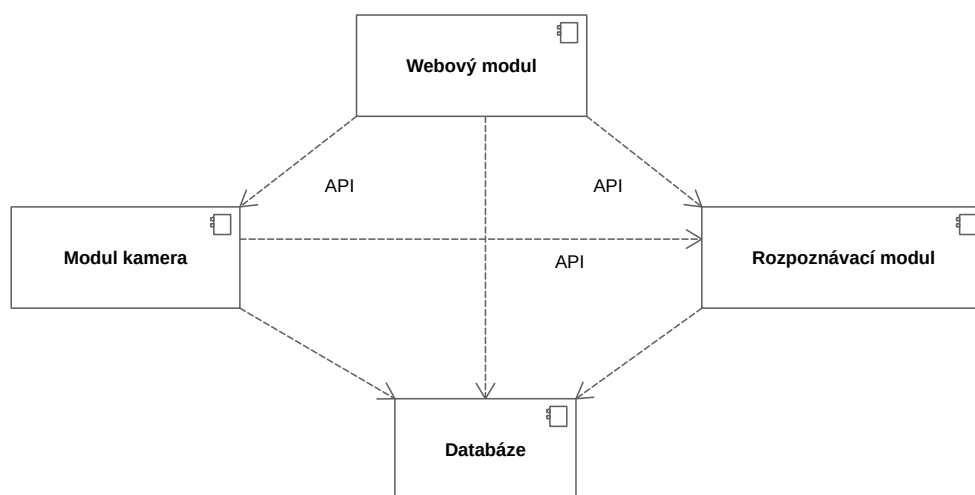
## 2.5 Architektura systému

Jedním ze základních požadavků na systém bylo to, aby byl systém modulární – pro snadnou výměnu nebo upgrade jednotlivých částí systému a také pro široké možnosti nasazení (různé moduly mohou běžet na různých zařízeních i na stejném zařízení a zároveň kamer může být více).

Původně jsem zamýšlel 3 moduly – rozpoznávací s kamerou, webový modul a databáze. Pak jsem ale došel k názoru, že bude lepší rozpoznávání od kamery oddělit a to zejména proto, aby rozpoznávací modul byl jen jeden a mohl běžet na výkonnějším zařízení, než je Raspberry PI (proč Raspberry PI je vysvětleno v sekci 2.6.6).

Na obrázku 2.3 jsou znázorněny moduly a komunikace mezi nimi (webový modul ovládá kamery a rozpoznávač přes REST API, kamera posílá přes REST API příkaz rozpoznávacímu modulu k rozpoznání právě vyfoceného snímku a všechny moduly přistupují k databázi).

V pozdějších fázích jsem došel k tomu, že komunikace přes REST API a zároveň ještě přes databázi je zbytečné zdvojení a později bych chtěl komunikaci omezit pouze na REST API a aby každý modul používal svoji vlastní databázi.



Obrázek 2.3: Diagram komponent (modulů)

Jako příklad komunikace uvedu cestu snímku od pořízení po zobrazení. Modul kamera detekuje pohyb a vyfotí snímek, který uloží do databáze. Poté pošle přes REST API zprávu rozpoznávacímu modulu, že je nutné spustit rozpoznávání na pořízeném snímku definovaném jeho ID v databázi. Rozpoznávací modul načte snímek, spustí rozpoznávání a výsledek uloží do databáze. Modul kamera pošle odpověď, že to proběhlo v pořádku. Manažér se pak při-

hlásí do webového modulu, který si všechny informace načte z databáze. Další příklady jsou vyloženy v kapitole Realizace, sekce 3.4.1.

Typické nasazení je kamera jako jediný modul na Raspberry PI a zbylé moduly na výkonnějším (jediném) PC/serveru.

## 2.6 Volba technologií

V této sekci vyberu nejvhodnější technologie pro dané úlohy ve svém projektu.

### 2.6.1 Webový modul

Jako webový server jsem použil Apache HTTP Server verze 2.4. Vybral jsem ho, neboť je stále vyvíjen, je zdarma a umí pracovat s PHP. [8] Zároveň je to nejpoužívanější webový server. [9]

Pro dynamické generování HTML stránek jsem použil skriptovací jazyk PHP ve verzi 7.1. V této oblasti je nejpoužívanější jazyk na světě a já s ním umím dobře pracovat. [10] Proto byl pro mě jednoznačnou volbou.

A neméně významnou technologií pro tuto bakalářskou práci byl javascript, tedy objektově orientovaný interpretovaný jazyk. V mém případě použitý jako skriptovací jazyk interpretovaný prohlížečem. [11]

Pro usnadnění práce s javascriptem jsem použil knihovnu jquery. Obsahuje mnoho funkcí pro manipulaci se stránkou, jako jsou vizuální efekty nebo načtení části dat bez nutnosti načítat celý HTML dokument (AJAX). [12] Je nejpoužívanější javascriptovou knihovnou. [9]

### Framework

Dnes již pouze málokdo píše rozsáhlejší webové aplikace v čistém PHP, neboť existuje mnoho frameworků, které psaní aplikací usnadňují. *„Už vás nebaví řešit opakující se úkoly, tisíce drobností, které odvádějí pozornost od práce a dělají z programování nudnou činnost? Jste na správném místě! Framework vám ulehčí práci, budete méně psát, mít přehlednější kód a radost z práce.“* [13]

Většina dnešních PHP frameworků je založena na architektuře MVC. *„Architektura MVC dělí aplikaci na 3 logické části tak, aby je šlo upravovat samostatně a dopad změn byl na ostatní části co nejmenší. Tyto tři části jsou Model, View a Controller. Model reprezentuje data a business logiku aplikace, View zobrazuje uživatelské rozhraní a Controller má na starosti tok událostí v aplikaci a obecně aplikační logiku.“* [14]

Model MVC využívá také Nette framework, který je velmi oblíbený mezi českými programátory už jen proto, že je také Čechy vyvíjen a je kolem něj nejaktivnější komunita českých PHP vývojářů. [15] Já si ho navíc zvolil proto,

že v něm umím nejlépe programovat a také proto, že pokud někdo bude pokračovat ve vývoji, nejspíše (bude-li to Čech) v něm bude umět také.

Nezůstal jsem ale pouze u samotného Nette, nýbrž použil jsem ještě několik dalších komponent instalovaných buď přes Composer [16] nebo nalinkovaných přímo do HTML souborů.

### Použité knihovny a doplňky

**guzzle** – HTTP klient pro PHP [17],  
**bootstrap** – Sada nástrojů pro vytváření responsivního designu [18],  
**bootstrap-4-forms** – Bootstrap 4 formuláře v Nette [19],  
**datagrid** – Přizpůsobitelný datagrid [20],  
**doctrine** – Doplněk pro ORM [21],  
**translation** – Překladačový systém pro Nette [22],  
**console** – Spouštění scriptů v Nette přes konzoli [23],  
**fontawesome** – Vektorové ikony [24],  
**pace** – Ukazatel průběhu načítání stránky [25],  
**masonry** – Javascriptové rozmístování prvků do mřížky [26],  
**Chart.js** – Kreslení grafů [27],  
**autocomplete** – Našeptávač [28]

### 2.6.2 Rozpoznávací modul

Od startu hloubkového učení (disciplína strojového učení) v roce 2012 se rozpoznávání tváří stalo velice zajímavým trhem pro mnoho společností nabízející rozmanitou řadu produktů. Integrovaní schopnosti rozpoznávání tváří do projektů se stalo velice snadné. [29]

### SDK vs API

Existují primárně dva typy řešení rozpoznávání tváří. Jsou jimi SDK a API. API jsou obvykle provozovány v cloudu a jsou dostupné na bázi předplatného. Toto řešení je velice jednoduché na použití, protože běží na infrastruktuře poskytovatele, který se o všechno stará. Lze je použít na jakékoliv platformě a lze je jednoduše vyměnit za jiné. Nevýhodami je to, že je nelze použít v real-time aplikacích, vyžadují spolehlivé internetové připojení a se vrůstajícím počtem požadavků může výrazně narůstat cena za službu. Příklady uvedu záhy.

SDK jsou dostupné jako předpřipravené knihovny nebo poskytované spolu se zdrojovým kódem pro integraci s aplikacemi. Pracují kompletně offline, jelikož algoritmus rozpoznávání tváří je v aplikaci obsažen. Nicméně infrastruktura musí být spravována vývojářem a v případě komerčních řešení jsou SDK v porovnání s API dražší. [29]

### Komerční řešení

Mezi nejpoužívanějšími řešeními rozpoznávání tváří v roce 2018 podle RapidAPI [30] se řadí Face++[31], Microsoft Face API [32] nebo Face Recognition and Face Detection[33].

Komerční řešení nicméně nebudu dále zkoumat, neboť do bakalářské práce potřebuji řešení, které je zdarma a více průhledné, než tato API.

### Otevřená řešení

Nejnámějším otevřeným řešením rozpoznávání tváří je knihovna OpenFace [34]. OpenFace je Pythonová knihovna implementující Torch neuronové síť k rozpoznávání tváří. Provozovat knihovnu je možné buď pod programem Docker [35] nebo nainstalovat přímo na systém – to jsem zkoušel a při instalaci se vyskytovalo mnoho problémů a byť se mi nakonec podařilo nainstalovat na Debian [36] (je otázkou, zda by to fungovalo na Raspberry PI [6]), použití mi přišlo také vcelku komplikované a instalaci jsem již nechtěl opakovat.

Proto jsem se rozhodl hledat ještě jiné řešení. Mnoho takových se nachází na GitHubu [37] (repozitářů s tímto tématem je tam 939). Mnoho z nich funguje podobně (využívá stejné či podobné komponenty) jako OpenFace. Nejpoužívanější z nich je face\_recognition [38]. Autor knihovny ji nazývá jako „Světově nejjednodušší API pro rozpoznávání tváří pro Python a příkazovou řádku“. Je postavená na nejmodernějším rozpoznávání tváří dlib [39] založené na hloubkovém učení. Model má přesnost 99.38% v benchmarku Labeled Faces in the Wild. [40]

### Můj výběr

Pro svoji, dle mého názoru, jednoduchost použití a variabilní možnosti použití jsem si vybral knihovnu face\_recognition [38], která je napsaná v jazyce Python [41]. Z toho důvodu jsem modul, který knihovnu využívá, napsal v témže jazyce. Více se o modulu dozvíte v sekci 3.2.

### Použité knihovny a doplňky

**configparser** – modul zpracovávající konfigurační soubory [42],  
**cv2** – balíček pro práci s knihovnou OpenCV [43] [44],  
**face\_recognition** – knihovna na rozpoznávání tváří [38],  
**flask** – webový server pro programy napsané v Pythonu [45],  
**numpy** – balíček pro vědecké výpočty v Pythonu [46],  
**pymysql** – doplněk pro komunikaci s databází MySQL/MariaDB [47],  
**time** – modul s funkcemi pro práci s časem [48],  
**sys** – modul s parametry a funkcemi specifické pro systém [49]

### 2.6.3 Modul kamera

Systém byl navrhovaný tak, aby alespoň modul kamera mohl být provozován na mikropočítači Raspberry PI [6] s kamerou [50] a pohybovým senzorem [51] (proč právě Raspberry PI je popsáno v sekci 2.6.6).

Rozhraní pro ovládání kamery a pohybového senzoru na Raspberry PI je v jazyce Python. Z toho důvodu je v Pythonu naprogramován celý modul kamera. Více o modulu v sekci 3.1.

#### Použité knihovny a doplňky

**configparser** – modul zpracovávající konfigurační soubory [42],  
**cv2** – balíček pro práci s knihovnou OpenCV [43] [44],  
**flask** – webový server pro programy napsané v Pythonu [45],  
**gpiozero** – rozhraní pro práci se zařízeními připojenými přes GPIO piny [52],  
**json** – modul pro kódování/dekódování do/z formátu JSON [53],  
**numpy** – balíček pro vědecké výpočty v Pythonu [46],  
**picamera** – balíček poskytující rozhraní k modulu Raspberry PI camera [7],  
**playsound** – modul s funkcí přehrávání zvuků [54],  
**pymysql** – doplněk pro komunikaci s databází MySql/MariaDB [47],  
**threading** – modul pro práci s vlákny [55],  
**time** – modul s funkcemi pro práci s časem [48],  
**sys** – modul s parametry a funkcemi specifické pro systém [49]

### 2.6.4 Databáze

Relačních databází splňující potřeby projektu a spolupracujících s PHP i Pythonem je opravdu mnoho. Systém nemá žádné zvláštní požadavky. Pouze v případě, že by firma měla opravdu velký počet zaměstnanců, by výběr databáze vyžadoval větší pozornost na výkon.

Pro systém jsem vybral databázi MariaDB [56], což je opensourcová odnož MySQL [57], kterou koupil Oracle. Tyto dvě databáze jsou vzájemně široce kompatibilní [58]. Databázi jsem zvolil proto, že s ní umím v rámci opensourcových databází nejlépe pracovat.

### 2.6.5 Komunikace mezi moduly

Všechny tři moduly (web, kamera i rozpoznávač) jsou spuštěny jako webové servery (web pomocí Apache [8] a zbylé dva pomocí Flask [45]). Moduly mezi sebou komunikují přes REST API a odpovědi posílají ve formátu JSON.

Například modul web odešle požadavek na adresu `http://camera:5001/run` modulu kamera, ten spustí službu focení a webovému modulu odešle odpověď ve tvaru `{"result": "OK"}`.

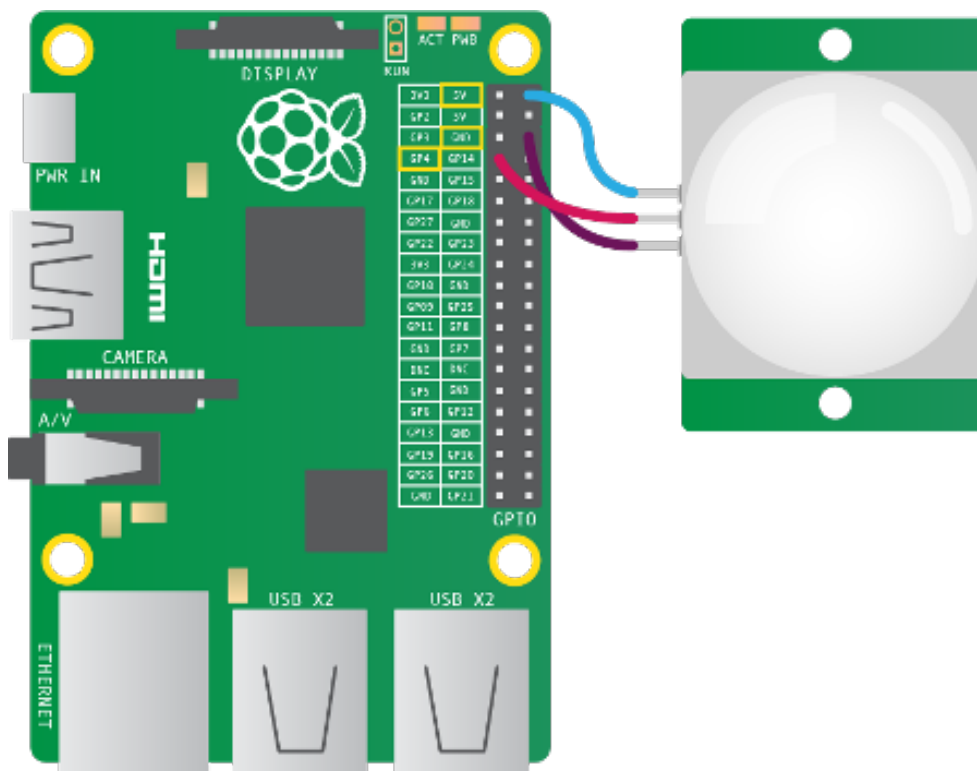


### 2.6.6 Hardware

Webový modul, rozpoznávací modul i databáze mohou být spuštěny na kterémkoli počítači nebo serveru ve firmě. Nicméně ke vstupu do firmy je potřeba umístit malý kus hardwaru s co nejmenším počtem kabelů, který by detekoval pohyb, fotil a komunikoval přes internetovou síť. Ve spoustě případů je nutné mít takových kamer více (aby zaměstnanci měli viditelnou tvář nejen při vstupu do firmy, ale i při výstupu).

Dalším kritériem, zejména v případě více kamer, jsou pořizovací náklady, také široká možnost úprav a dobrá komunita kolem hardwaru i softwaru.

Těmto požadavkům nejvíce vyhoví mikropočítač Raspberry PI. Je o velikosti kreditní karty [6], spolu s ním se prodává také kamera přímo pro Raspberry PI a PIR pohybový senzor (schéma zapojení naleznete na obrázku 2.4). K provozu je nutné také dokoupit napájecí kabel (micro USB), drátové propojky k PIR senzoru, micro SD kartu a šikovná je také krabička na Raspberry PI. Toto všechno se dá pořídit s cenou kolem 2 700 Kč [59].



Obrázek 2.4: Schéma zapojení PIR senzoru k Raspberry PI (zdroj: <https://projects.raspberrypi.org> [51])

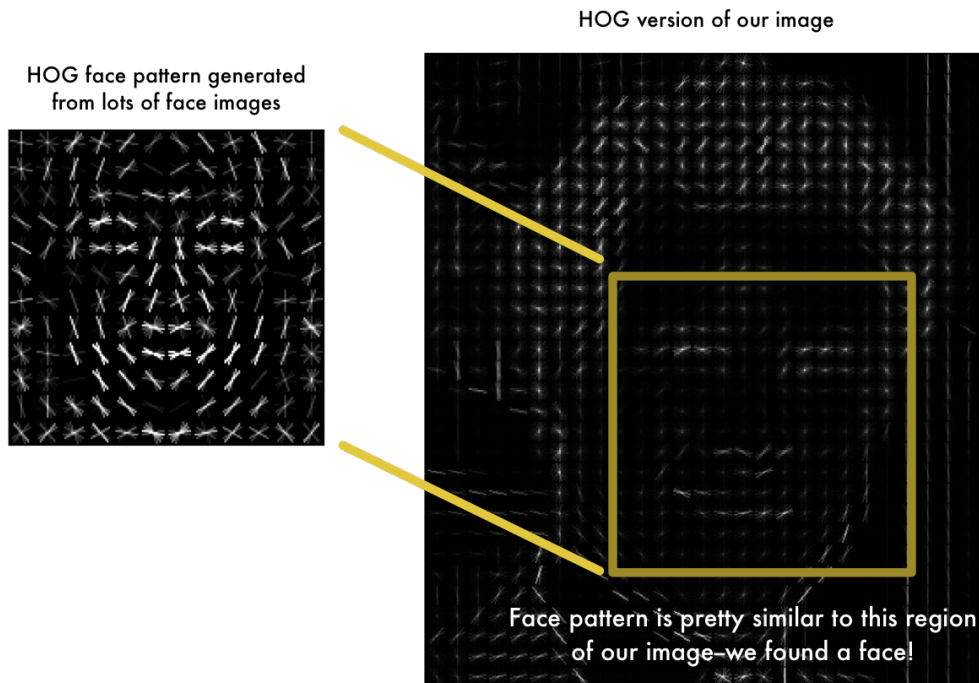
Verze 3 má integrovanou Wifi, takže odpadá nutnost připojovat Raspberry

PI k internetu pomocí kabelu a jediným kabelem bude tedy ten napájecí (v průběhu testování se však ukázalo, že wifi způsobuje mnoho falešných detekcí pohybu – každou minutu – takže doporučuji připojení kabelem), má dostatečný výkon pro provoz modulu kamera mého systému a existuje k němu díky široké komunitě spoustu návodů, tutoriálů a softwaru (například pro Raspberry PI speciálně upravený linuxový OS Debian nazvaný Raspbian, na kterém systém provozuji). [6]

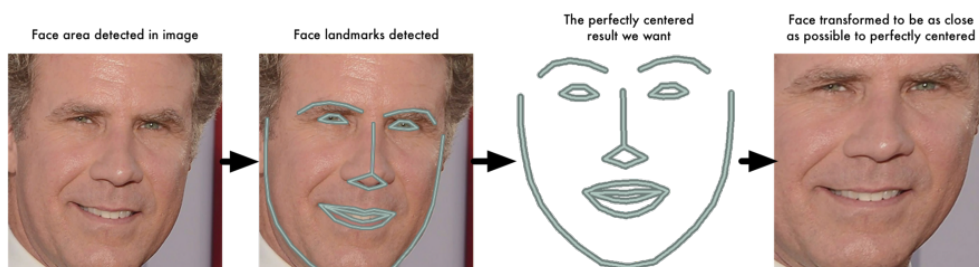
### 2.7 Teorie o rozpoznávání tváří

Rozpoznávání tváří probíhá ve 4 základních krocích. Tuto činnost vykonává knihovna `face_recognition` [38]. Popíši jen velmi stručně, pro ilustraci přidám obrázky.

1. Nalezení všech tváří na obrázku (obr. 2.5)
2. Nalezení orientačních bodů tváře, vycentrování a transformace tváře (důležité zejména v případě, že je tvář zachycena ze strany nebo je nahnutá) (obr. 2.6)
3. Zakódování znaků tváře (například tvar očí) do vektoru čísel, kterým rozumí jen algoritmus sám – jaké znaky má zkoumat bylo strojově naučeno (obr. 2.7)
4. Nalezení jména osoby – porovnávání se známými tvářemi pomocí klasifikačního algoritmu (obr. 2.8) [60]

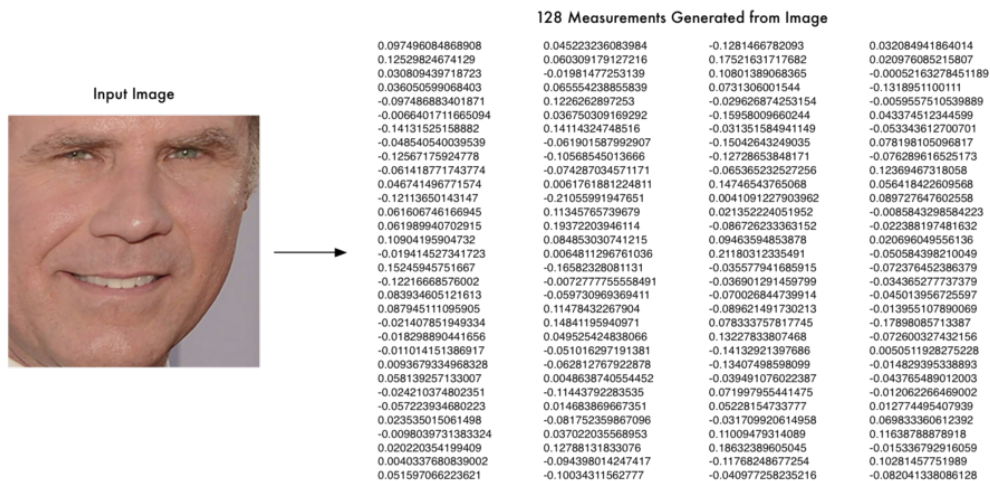


Obrázek 2.5: Rozpoznávání – Nalezení tváří (zdroj: <https://medium.com> [60])

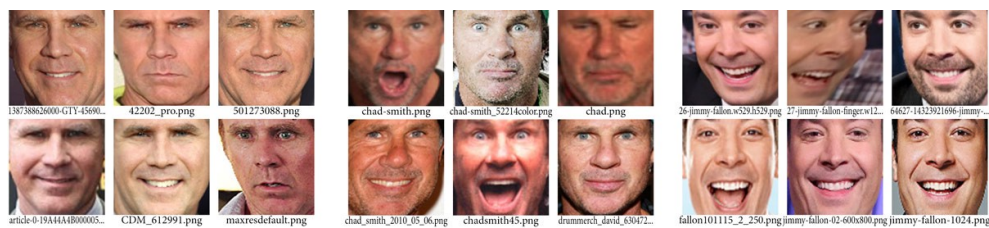


Obrázek 2.6: Rozpoznávání – Transformace tváře (zdroj: <https://medium.com> [60])

## 2. ANALÝZA A NÁVRH



Obrázek 2.7: Rozpoznávání – Zakódování znaků (zdroj: <https://medium.com> [60])



Obrázek 2.8: Rozpoznávání – Nalezení jména osoby (zdroj: <https://medium.com> [60])

---

## Realizace

V této kapitole uvedu to, jak jsem celý systém implementoval – jaké se při tom naskytly problémy, jak jsem jednotlivé moduly propojil, jak si moduly mezi sebou vyměňují data, co mi zařídila knihovna na rozpoznávání a co jsem musel implementovat já. Také uvedu to, jak systém funguje a co musí uživatel zpočátku dělat, aby systém dobře fungoval (postupné učení tváří).

Dále vysvětlím také to, jakým způsobem je systém možné zavést a jaké parametry je vhodné použít. V přílohách pak naleznete instalační příručku (příloha C).

### 3.1 Modul kamera

Původně jsem tento modul vyvíjel také s funkcí rozpoznávání tváří, později se však ukázalo, že kvůli nedostatku výpočetního výkonu bude vhodnější tuto část oddělit a umožnit tak provozovat rozpoznávání tváří na výkonnějším stroji a dosáhnout tak rychlejšího běhu aplikace.

Doporučené rozložení je modul kamera provozovat na Raspberry PI a zbylé moduly a databázi na výkonném počítači/serveru, nicméně systém je navržen modulárně a vše bude fungovat také v případě, že rozpoznávací modul (nebo také všechny moduly) poběží na Raspberry PI, pouze za cenu pomalejší odezvy (v případě načítání obrázků – případ, kdy je na Raspberry PI provozován webový server a databáze – znatelně pomalejší odezvy). Funkčnost však zůstává zachována.

Zpět ale k modulu Kamera. Již v kapitole 2.6.3 jsem zmínil, že celý modul je psaný v Pythonu a které knihovny využívá (připomenou, že knihovny `picamera` [7] a `gpiozero` [52] jsou určeny výhradně pro práci s Raspberry PI a modul tak nelze provozovat na jiném zařízení).

Stejně jako rozpoznávací modul, i modul kamera je provozován jako server využívající knihovnu `Flask` [45]. Je tak možné ho z pozice webového serveru (nebo i manuálně přes prohlížeč) ovládat. Přesnější popis API naleznete po

spuštění modulu na adrese zařízení, na kterém je provozováno, a nastaveném portu (defaultně 5001).

Služba detekující pohyb a snímání fotek je naprogramovaná jako vlákno, je provozovaná tak dlouho, dokud se nepošle modulu přes API příkaz k zastavení. Přes API se také spouští. Hlavním důvodem použití tohoto řešení je to, že zbytek modulu může stále odpovídat na požadavky, ale hlavně je schopen ihned odpovědět, že služba byla spuštěna nebo zastavena.

Konfigurace, kterou není nutné znát ještě před spuštěním modulu, se načítá z databáze a příkazem restart je možné načíst konfiguraci novou. Tato konfigurace se nastavuje přes webový portál → Administrace → Parametry systému. Nastavit se dá například otočení kamery, zpoždění focení po detekci pohybu a další.

Ostatní konfigurace se nastavuje přes strukturovaný textový soubor `config.ini` umístění ve stejné složce, jako je hlavní spouštěcí soubor – `camera.py` (spuštění popsáno v sekci 3.5.1). Takovým nastavením jsou například parametry připojení k databázi nebo port, který bude modul jakožto server poslouchat a odpovídat.

Před spuštěním modulu musí být správně vloženo jméno modulu a IP adresa v databázi (přes webový portál → Administrace → Správa zařízení) a stejné jméno nastavené v konfiguračním souboru, aby modul věděl, které má přiřazené ID.

#### Struktura souborů modulu

**camera.py** – hlavní spouštěcí soubor a server odpovídající na požadavky

**camera\_service.py** – spouští a zastavuje fotografující vlákno

**camera\_thread.py** – vlákno, které rozpoznává pohyb, fotí, ukládá fotografie do databáze a volá přes API rozpoznávací modul, aby provedl rozpoznávání

**config.ini** – konfigurační soubor (vzorová konfigurace = `config-example.ini`)

**config.py** – načítá a drží konfiguraci, ovládá komunikaci s databází

## 3.2 Rozpoznávací modul

Jak jsem zmínil v kapitole 2.6.2, kde jsou také vypsány všechny použité knihovny, celý rozpoznávací modul je napsaný v jazyce Python. Je provozovaný jako webový server pomocí knihovny Flask [45]. Jeho prací je na základě API požadavků rozpoznávat tváře z pořizovaných snímků načtených z databáze.

Požadavky se posílají primárně přes webový portál buď automaticky při určitých akcích nebo manuálně (Administrace → Správa zařízení) a automaticky po pořizení snímku modulem kamera. Aby to fungovalo, musí být správně nastavená IP adresa rozpoznávacího modulu v databázi (přes webový portál → Administrace → Správa zařízení) a tato adresa musí být dostupná webovému modulu i všem kamerám).

Možné API příkazy jsou vypsané na webové stránce dostupné přes IP adresu (a port – defaultně 5002), na které modul poslouchá. Je doporučeno, aby byl tento modul provozován na výkonnějším stroji, neboť proces rozpoznávání tváří je výpočetně náročný. Konfigurace modulu probíhá stejně jako u modulu kamera (viz. 3.1). Liší se jen nastavitelné parametry.

Modul si v paměti drží vypočtené znaky známých tváří a ty porovnává s právě rozpoznávaným snímkem. Ten se porovná s každou známou tváří, vybere se ta nejpodobnější a systém rozhodně, že rozpoznanou osobou je ta, které patří zmíněná nejpodobnější tvář. Proto čím více známých tváří, tím větší pravděpodobnost rozpoznání.

Důležitý je práh jistoty rozpoznávání nastavitelný přes webový portál → Administrace → Parametry systému, který určuje, jak podobné si tváře musí být, aby nedošlo k nesprávnému rozpoznání. Hodnota parametru je v rozmezí 0 (vůbec žádná podobnost) až 1 (identické znaky) a jako ideální se zdá být 0,54, kdy už jsem nezaznamenal žádné falešné rozpoznání. V případě velkého množství známých tváří a podobných lidí ale může být potřebná větší hodnota. Proto je tento parametr nastavitelný.

Knihovna rozpoznává tváře až do určitého úhlu otočení tváře ve smyslu rotace kolem páteře vůči kameře, který bych definoval tak, že je vidět ještě alespoň kus druhého oka, a nesmí být moc natočená ve smyslu rotace kolem špičky nosu (úhel velmi zhruba do 45°). Dalším problémem může být rychlý pohyb osob a následné rozmazání snímku nebo příliš tmavý snímek. Proto je důležité správné umístění kamery, aby tváře osob směřovali čelem ke kameře, a zajištění dobrých světelných podmínek.

Naopak není žádnou překážkou střídavé nošení brýlí a kontaktních čoček či změna účesu (vlasů si algoritmus vůbec nevšímá).

Knihovna `face_recognition` [38], která v tomto modulu zajišťuje rozpoznávání včetně porovnávání tváří, nabízí pro určení správné osoby také algoritmus KNN [61] – K-nejbližších sousedů – který by zpřesnil výsledky rozpoznávání. Bylo by velice vhodné tento algoritmus do systému implementovat.

Načítání tváří z databáze do paměti, API příkazy, konfigurace modulu, ukládání výsledků do databáze a ošetření chyb je v tomto modulu mojí prací; detekce tváří, výpočet znaků a porovnávání tváří je prací knihovny `face_recognition`.

## Struktura souborů modulu

**config.ini** – konfigurační soubor (vzorová konfigurace = `config-example.ini`)

**config.py** – načítá a drží konfiguraci, ovládá komunikaci s databází

**image.py** – rozpoznává tváře v načtené fotce

**recognize\_service.py** – vykonává samotné rozpoznávání

**recognizer.py** – hlavní spouštěcí soubor a server odpovídající na požadavky

### 3.3 Webový portál

Z kapitoly 2.6.1 již víme, že webový portál píše v PHP [10] ve frameworku Nette verze 2.4 [15]. Tvorbu uživatelského rozhraní jsem si usnadnil použitím bootstrapu verze 4. [18] Všechny použité doplňky a knihovny jsou vypsány v sekci 2.6.1.

Každá podstránka webového portálu (kromě přihlašování) má v horní části menu s položkami přehled, docházka, dovolená, záznamy, uživatelé, administrace a tlačítka na přepnutí jazyka (kromě češtiny je k dispozici ještě zatím částečně přeložený jazyk angličtina), zobrazení vlastního uživatelského profilu a tlačítko na odhlášení. Tyto sekce jsou v následujících sekcích popsány a jsou k nim také ukázky (screenshoty).

#### 3.3.1 Přihlašování a přehled

Pro přihlašování uživatelů používám v Nette výchozí autentizaci a autorizaci. Uživatele rozdělují do 3 rolí: zaměstnanci, manažery a administrátoři. Práva jednotlivých rolí zachycuje obrázek 2.1. K vymezení rolí a práv používám Nette ACL.

Po přihlášení je uživatel přesměrován na domovskou stránku nazvanou přehled (obrázek 3.1). Zde uživatel najde ty nejdůležitější informace, jako je aktuální čas, výpis mojí dovolené tento měsíc, souhrn a poslední 3 záznamy o docházce, pro manažery a administrátory ještě tlačítko s počtem dovolené ke schválení, tlačítko s počtem nerozpoznaných tváří ke kontrole, stavy zařízení a další, a ke každé je tlačítko zobrazit více, kde uživatel najde kompletní informaci.

#### 3.3.2 Docházka

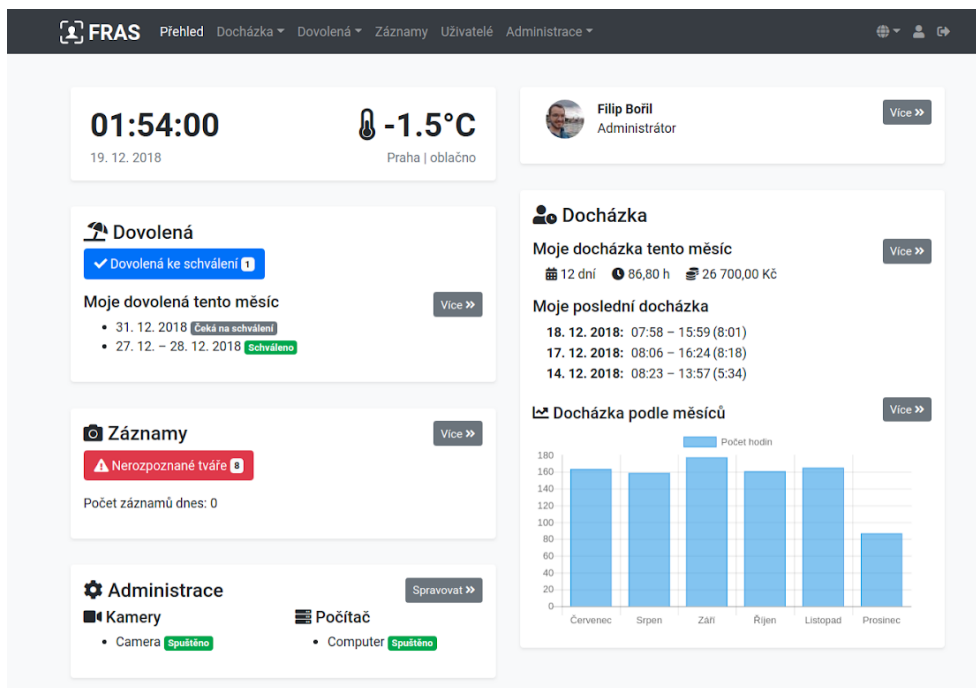
Sekce docházka se dělí do nabídky moje docházka (kam mají přístup všichni) a spravovat (kam má přístup manažer a administrátor).

V moje docházka (obrázek 3.2) si každý může prohlédnout vlastní záznamovanou docházku a v případě nesrovnalostí je zde nabídka reklamovat. Reklamace se pak objeví manažerům a správčům, aby ji prověřili. Uživatel si zde může prohlédnout svoji docházku kdykoliv v historii a u každého měsíce je zobrazený ještě přehled odpracovaných dní, hodin a dovolené.

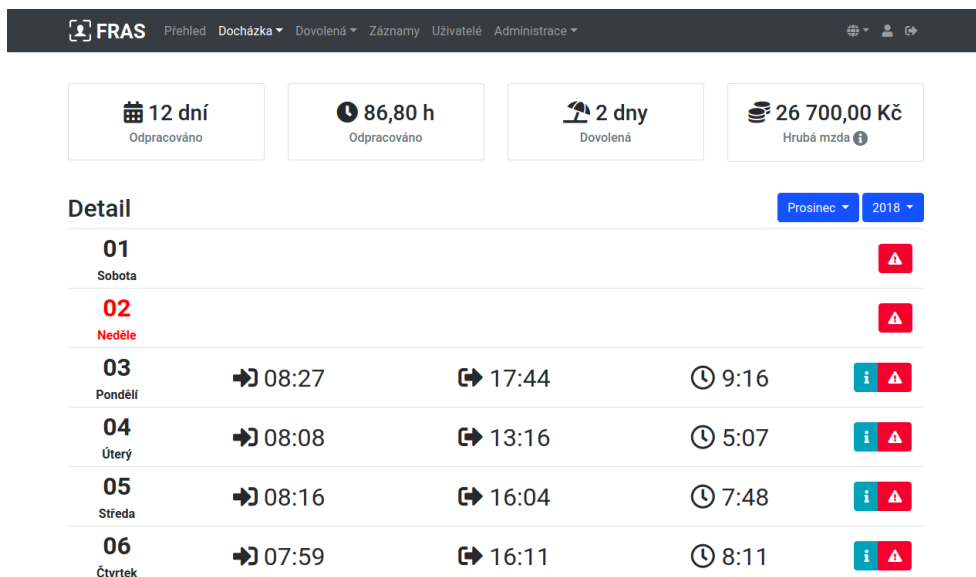
Do nabídky správa (docházky) se již dostane pouze manažer nebo administrátor. Správa se dále dělí do záložek reklamace, editace, historie a statistiky. V záložce reklamace je možno schvalovat či zamítnout nevyřízené reklamace, v editaci je možné po zvolení uživatele zobrazit přehled a editovat docházku v libovolný den, v historii je možné filtrovat veškeré dostupné záznamy o docházce a v záložce statistika lze zobrazit různé grafy, například docházku podle měsíců (sloupcový), podle času (spojnicový) nebo podle zaměstnanců (koláčový).



### 3.3. Webový portál



Obrázek 3.1: Webový portál – stránka přehled



Obrázek 3.2: Webový portál – stránka moje docházka

#### 3.3.3 Dovolena

Stejně jako docházka, i sekce dovolená je rozdělena na vlastní přehled a žádání (dostupné pro všechny uživatele) a na správu dovolených (určenou výhradně pro manažery a administrátory).

V sekci moje dovolená (obrázek 3.3) si každý může prohlédnout přehled veškeré žádané, schválené i zamítnuté dovolené, nachází se zde také formulář pro žádání a je zde k mání i informace o souhrnu dovolené – využití dny a celkový nárok – doplněná progress barem pro grafické znázornění.

The screenshot shows the 'moje dovolená' page in the FRAS system. The top navigation bar includes 'FRAS', 'Přehled', 'Docházka', 'Dovolena', 'Záznamy', 'Uživatelé', and 'Administrace'. The main content is divided into two sections. On the left, there is a list of leave requests, each with a status icon, dates, a status label, and a duration. On the right, there is a 'Žádost o dovolenou' form with a progress bar, date pickers, a comment field, and a 'Požádat' button.

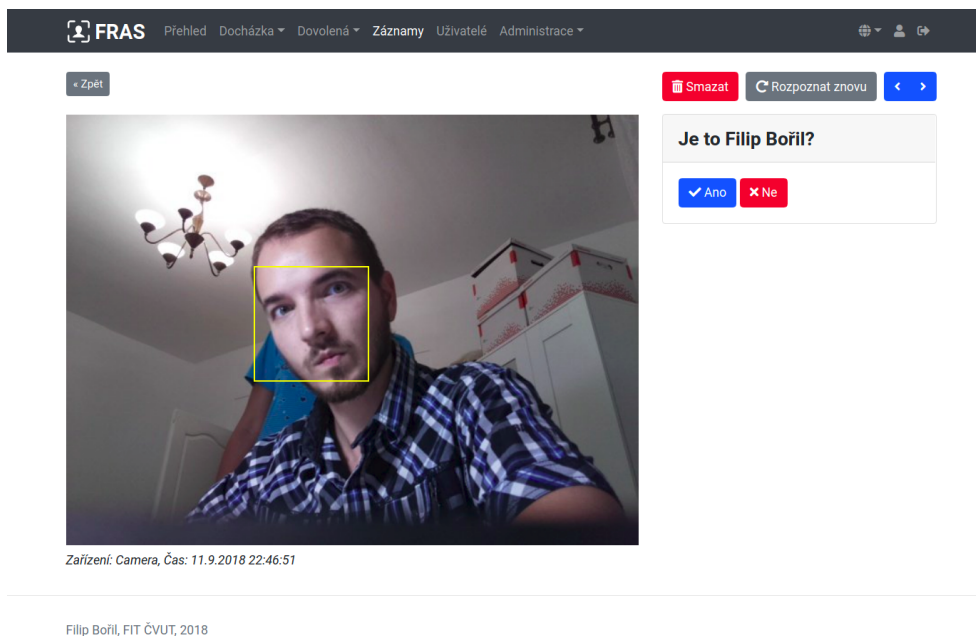
Obrázek 3.3: Webový portál – stránka moje dovolená

Do správy dovolené, kam mají přístup jen manažeri a administrátoři, patří záložky schvalování a historie. Pod záložkou schvalování je v levé části stránky seznam žádostí a na pravé straně zobrazeny detaily zvolené žádosti a formulář na schválení či zamítnutí. V záložce historie lze filtrovat dovolenou všech uživatelů a upravovat (i již schválené či zamítnuté) dovolené.

#### 3.3.4 Záznamy

Sekci záznamy je přístupná pouze manažerům a administrátorům. Najdou zde přehled pořízených záznamů (snímků) a mají možnost mezi nimi filtrovat podle data a času, typu (všechny, s tvářemi, bez tváří, známé, jisté, nejisté, pravděpodobné, nepravděpodobné) a řadit (nejnovější nahoře/dole).

Po rozkliknutí záznamu (obrázek 3.4) se v levé části snímek otevře ve větší velikosti, pod ním je napsané datum, čas a zařízení, které ho pořídilo, v pravé části je nahoře toolbar s možnostmi smazat, rozpoznat znovu (změnila-li se databáze uživatelů známých tváří nebo došlo-li při prvním rozpoznávání k chybě) a tlačítka pro procházení snímky dle zvoleného filtru. Pod toolbarem jsou vypsané rozpoznané tváře – v záhlaví jméno uživatele, kterému dle vyhodnocení patří, a pod ním podle jistoty rozpoznání tlačítka s akcemi. Při najetí kurzorem nad rámeček se jménem a tlačítky akcí se daná tvář na obrázku vlevo žlutě zarámuje (aby bylo jasné, o kterou tvář jde – tváří zde může být více).



Obrázek 3.4: Webový portál – detail záznamu (snímku)

V případě, že je podobnost vyšší, než prah jistoty rozpoznání, tvář se počítá do docházky jako správně rozpoznaná a zde se objeví pouze tlačítko reklamace. V případě kliknutí na tlačítko se objeví seznam dalších podobných tváří a je tak možno přiřazení opravit.

V případě, že je podobnost mezi prahy pravděpodobnosti a jistoty rozpoznání, tvář se také počítá do docházky, ale zde se objeví otázka, zda jde skutečně o rozpoznanou osobu a tlačítka ano, ne. V případě kliknutí na ano, tvář se přiřadí mezi známé tváře osoby a okamžitě se ji systém naučí (systém osobu příště rozpozná s větší jistotou). V případě kliknutí na ne, objeví se seznam dalších podobných tváří a je tak možno přiřazení opravit.

V ostatních případech se jedná o nejisté rozpoznání a tvář se tak do docházky nepočítá. Je zde rovnou zobrazen výpis nejpodobnějších tváří a po výběru správné osoby se tvář přiřadí této osobě a ihned se ji systém naučí.

Při výběru správné osoby je na posledním místě políčko, kam je možné jméno napsat. Políčko obsahuje našeptávač a pokud se sem napíše osoba, kterou systém zatím neneviduje, je otevřena stránka se zakládáním nového uživatele, kterému může být následně tvář přiřazena.

## 3. REALIZACE

### 3.3.5 Uživatelé

Sekce uživatelé v sobě obsahuje několik podstránek. Výpis uživatelů, který je buďto tabulkový (kde lze filtrovat) nebo obrázkový, mohou prohlížet pouze manažéři a administrátoři.

Detail uživatele (obrázek 3.5), který je dostupný z výpisu, mohou editovat manažéři a administrátoři. V detailu uživatele lze editovat vedené údaje, roli (s výjimkou, že manažer nemůže změnit roli administrátorovi a zároveň v situaci, kdy by v systému neměl zůstat žádný administrátor), přidávat, mazat tváře a nastavovat profilovou fotografii. Uživatelům je možné vygenerovat heslo nebo je smazat (s výjimkou situace, kdy by v systému neměl zůstat žádný administrátor).

Kliknutím na ikonu uživatele v menu se otevře vlastní profil, který je přístupný každému uživateli pro čtení, může si zde změnit heslo a udělit souhlas se zpracováním osobních údajů.

Detail tváře (prohlížení fotografie přiřazené uživateli s informací zda je naučena) je dostupná kliknutím na fotografii v detailu uživatele.

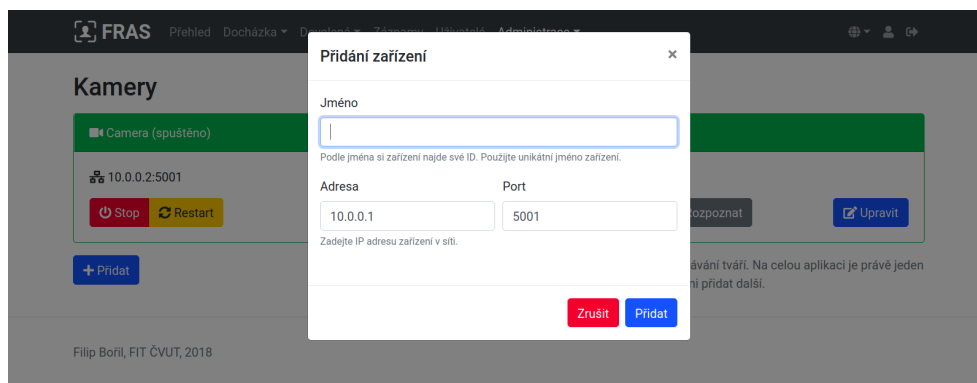
The screenshot shows the user detail page for Filip Bořil in the FRAS system. The page has a dark navigation bar at the top with the FRAS logo and menu items: Přehled, Docházka, Dovolená, Záznamy, Uživatelé, and Administrace. A user profile card for Filip Bořil is displayed, featuring a circular profile picture and a 'Zpět' button. The profile information is organized into a grid of form fields: 'Titul před' (empty), 'Křestní jméno' (Filip), 'Příjmení/další jména' (Bořil), and 'Titul za' (empty). Below this are 'Uživatelské jméno' (filip) and 'Role' (Zaměstnanec). Further down are 'Typ smlouvy' (Jiné/neurčené), 'Hodinový plat' (empty), and 'Týdenní úvazek' (empty). There are also three checkboxes: 'Prohlášení o dani podepsáno', 'Student', and 'Státní pojištěnec'. A red warning message states: 'Filip Bořil nedal souhlas se zpracováním osobních údajů!'. At the bottom of the profile card are three buttons: 'Změnit údaje', 'Generovat heslo', and 'Smazat'. Below the profile card is a 'Tváře' section with three thumbnail images of the user. At the bottom is a 'Nová tvář' section with a 'Browse' button and a 'Nahrát' button.

Obrázek 3.5: Webový portál – detail uživatele (pohled manažéra)

### 3.3.6 Administrace systému

V sekci administrace systému je možné spravovat zařízení (kamery a rozpoznávací modul), nastavovat parametry (jako například zpoždění focení po detekci pohybu v milisekundách nebo měnu) a spouštět úkony, jako je například výpočet docházky nebo mazání starých záznamů.

Ve správě zařízení je možné přidávat (obr. 3.6) a odebírat kamery, nastavovat jméno, adresu zařízení v síti a port, na který poslouchají. Podle jména si zařízení najde svoje ID, aby mohlo do databáze ukládat fotografie, adresa a port zařízení se nastavuje, aby šlo přes webové rozhraní ovládat API a posílat tak zařízení příkazy jako je zapnutí/vypnutí monitorování, restart, naučení nových tváří a další.



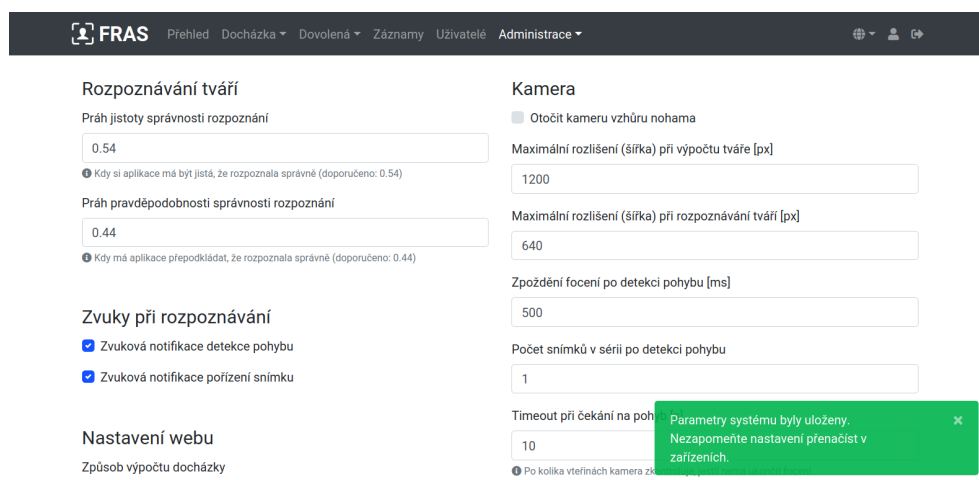
Obrázek 3.6: Webový portál – přidávání kamery do systému

Adresa a port zařízení jsou důležité také pro komunikaci mezi kamerou a rozpoznávacím modulem (po vyfocení záběru kamera ihned pošle příkaz na výpočet tváře). Obrázek samotný se neposílá, posílá se jen ID, pod kterým je obrázek uložen do databáze.

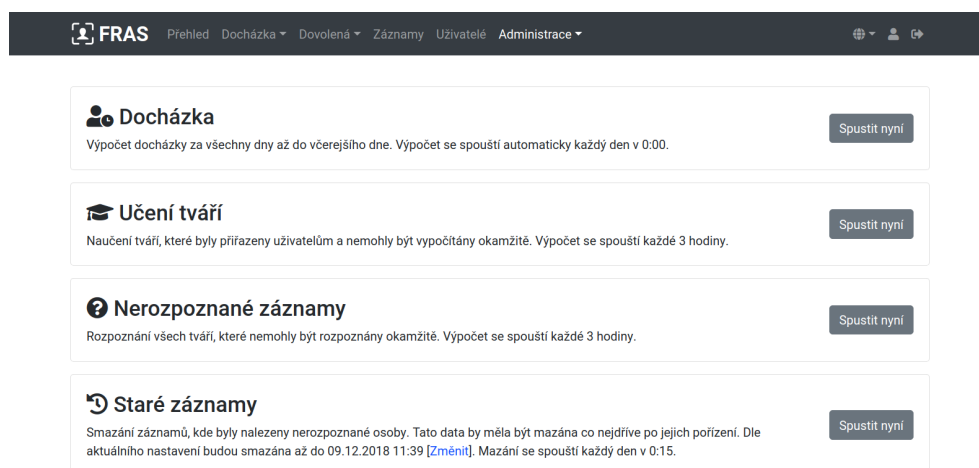
V sekci parametry systému lze nastavovat různé parametry, od těch techničtějších, jako je rozlišení focení, zpoždění focení po detekci pohybu, či práh jistoty správnosti rozpoznávání (k čemu to přesně slouží je popsáno v sekci 3.5.2, kde najdete popsáno více dostupných parametrů) až po ty praktičtější, jako je zobrazení státních svátků v docházce nebo například měna, ve které se v systému zobrazuje mzda. Obrázek 3.7

A nakonec, spouštění úkonů – sekce pojmenovaná jako údržba. Některé z těchto úkonů se spouštějí automaticky v předem nastavenou dobu, některé lze spouštět pouze manuálně přes tuto stránku. Všechny úkony včetně času jejich spouštění jsou zde popsány a je u nich tlačítko „Spustit nyní“ (obr. 3.8).

### 3. REALIZACE



Obrázek 3.7: Webový portál – nastavení parametrů systému a zpráva o úspěšném uložení



Obrázek 3.8: Webový portál – údržba (manuální spuštění úkonů)

## 3.4 Propojení modulů a výměna dat

Schéma modulů jsem již popsal v sekci 2.5 – architektura systému. Zde uvedu podrobněji na to, jakým způsobem spolu moduly komunikují a co všechno ukládají do databáze.

### 3.4.1 Komunikace

Protože je systém složený z modulů, musel jsem vyřešit, jak spolu budou moduly spolupracovat. Z důvodu bezpečnosti a přenositelnosti jsem se rozhodl pro komunikaci pomocí REST API, kdy každý modul (kromě databáze) běží jako webový server a odpovídá na požadavky, a data ukládá do stejné databáze.

Webový portál má pozici řídicího modulu, protože vydává příkazy spustit/zastavit kameru, naučit nové tváře, či rozpoznat dosud nerozpoznané záznamy. Příklad takového příkazu je uveden v sekci 2.6.5. Přes portál se také nastavují údaje pro spojení mezi moduly, tedy jméno, adresa a port, pod kterým běží. Tyto údaje jsou uloženy v databázi. Při spouštění modulů tak je potřeba znát pouze spojení k databázi a zbytek si zjistí z databáze. Také změna stačí provést pouze na tomto jednom místě. Detailněji je to popsáno v sekci 3.3.6.

Modul kamera potřebuje posílat příkaz k rozpoznání fotografie, kterou pořídil a uložil do databáze. Adresu rozpoznávacího modulu zná z databáze, stačí mu tedy poslat příkaz k rozpoznání fotografie s ID, které mu pošle. Rozpoznávací modul rozpozná a vrátí výsledek (úspěch/neúspěch). Webový portál má tak výsledek již připravený v databázi. Sám modul kamera umí odpovídat na dotaz, zda je focení spuštěno (ve webovém portále je tak vidět stav běží/zastaven/odpojen), provést příkaz spustí focení, zastav focení a přenačti nastavení z databáze. Na každý příkaz vrátí odpověď result (OK/Fail). V případě chyby navíc připojí popis chyby.

Rozpoznávací modul sám nikomu požadavky neposílá, ale odpovídá webovému portálu, kterému odpovídá na dotaz na stav (vždy spuštěno – možné stavy jsou tak jen spuštěno/odpojeno (v případě, že odpověď žádná nepřijde)) a na příkazy naučení všech nových tváří, rozpoznání všech nerozpoznaných a přenačtení nastavení včetně naučených tváří z databáze. Modulům kamera odpovídá na příkazy rozpoznání záznamu s určitým ID v databázi. Na příkazy vrací odpověď result (OK/Fail) a připojí dodatečnou informaci – například počet rozpoznávaných záznamů.

Schéma komunikace je zachyceno na obrázku 2.3.

### 3.4.2 Databáze

Všechny moduly používají jednu databázi, kam ukládají (a načítají) informace. Jelikož webový portál je založený na Nette a ten má vlastní vrstvu ke komunikaci s databází, mělo by být velmi snadné databázi MySQL/MariaDB [57][56] vyměnit za jinou. V případě modulů kamera a rozpoznávacího modulu bude vyžadován minimální zásah do jediného souboru config.py, ve kterém se provádějí SQL volání pro konkrétní databázi, kde se místo knihovny pymysql [47] použije jiná pro konkrétní databázi. Řešení bude nutné ještě otestovat (na-

příklad s knihovnou `mysql-connector-python` [62] (také pro MySQL/MariaDB) byly na některých systémech problémy, proto došlo k výměně).

Již v počátcích vývoje systému jsem se rozhodl, ačkoliv to má také nevýhody, ukládat do databáze také obrázky s tvářemi. Hlavní výhodou je to, že k nim mají přístup všechny moduly. Modul kamera je pořídí a spolu s dalšími daty uloží, rozpoznávací modul je načte, rozpozná a uloží informace o rozpoznávaných osobách (podobnosti s tvářemi) a webový modul všechna tato data načte pro uživatele k prohlížení. Další výhodou je přenositelnost dat – data lze zálohovat či přenést pouhým exportem dat z databáze. Žádné další soubory se nepřenáší.

Nevýhodou takového řešení je především vyšší nárok na výkon, což je problém například pokud databáze běží na Raspberry PI a načítá se stránka s velkým množstvím obrázků (všechny z databáze). Každý dotaz trvá zhruba 1 – 2 sekundy. Nicméně v případě použití doporučeného nasazení, tedy na Raspberry PI kamera a na PC/serveru zbylé moduly, žádný problém to nepůsobí.

## 3.5 První kroky

V této sekci rozeberu možnosti zavedení systému, vhodné parametry pro provoz a popíši, jak by měl administrátor systému na začátku jeho používání postupovat (postupné učení tváří). Přesné kroky pro úspěšnou instalaci jsou popsány v příloze C.

### 3.5.1 Zavedení systému

Je více možností, jak jednotlivé moduly provozovat. Nejjednodušší variantou instalace je pomocí Dockeru. [35] Docker umožňuje běh virtuálních počítačů. To se hodí zejména v provozu webového portálu, rozpoznávacího modulu a databáze na PC/serveru. Jediné, co je potřeba, je na PC/server nainstalovat Docker a spustit „image“, který jsem připravil. Ten již má nainstalované všechny závislosti a po zapnutí vše automaticky běží. Přesné kroky pro instalaci jsou uvedeny v příloze C.

Docker ale nelze provozovat na mikropočítači Raspberry PI (nepodporuje virtualizaci), pro který je určený modul kamera. Zbývá provozovat modul napřímo. To má ale nevýhodu v tom, že musí zůstat otevřený terminál a v případě, že modul spadne, automaticky se nerestartuje. Toto řeší program Supervisor. [63] Stejným způsobem je možné provozovat i rozpoznávací modul. Webový server (Apache [8]) a databáze (MySQL/MariaDB [57][56]) tyto problémy již řeší a není potřeba je provozovat přes Supervisor. Provozují se napřímo. Přesné kroky pro instalaci jak přes Supervisor, tak samostatně, jsou uvedeny v příloze C.



### 3.5.2 Parametry systému

Parametry systému jsou definované a nastavené v databázi. Nastavují se přes webový portál (Administrace → Parametry systému), kde je též blíže popsáno, co dělají. Některé parametry používá sám webový portál, některé modul kamera a některé rozpoznávací modul (změny se aplikují po restartu přes webový portál). Následně parametry popíši.

#### Parametry

- Prah jistoty správnosti rozpoznání** – Hodnota podobnosti (1 – identita, 0 – žádné podobné znaky), od které již systém prohlásí s jistotou za správně rozpoznáný; nejvhodnější hodnota 0,54
- Práh pravděpodobnosti správnosti rozpoznání** – stejné jako předchozí, ale pravděpodobně správně rozpoznáný (jak se v tomto případě chová webový portál je popsáno v sekci 3.3.4); nejvhodnější hodnota 0,44
- Zvuková notifikace detekce pohybu** – Pokud je k Raspberry PI připojen reproduktor, ozve se při detekci pohybu zvuk
- Zvuková notifikace pořízení snímku** – Pokud je k Raspberry PI připojen reproduktor, ozve se při pořízení snímku
- Způsob výpočtu docházky** – Zatím je jediná možnost – Od prvního záznamu do posledního – to znamená, že každý den se jako začátek docházky považuje první zachycený snímek a jako konec docházky poslední
- Mazání starých nerozpoznaných snímků** – Doba, po které se automaticky smažou snímky s nerozpoznanými tvářemi (kvůli ochraně osobních údajů)
- Mazat také pravděpodobné** – Mazat se budou také osoby, které nebyly rozpoznány s jistotou (přísnější režim)
- Zobrazovat státní svátky** – V kalendáři se budou zobrazovat státní svátky a nebude tak možné nastavovat v tyto dny docházku
- Proplácet státní svátky** – Dny se budou započítávat do výplaty
- Nárok na dovolenou ročně** – Kolik dní si uživ. může vzít ročně dovolené
- Měna** – Měna, jaká se bude u peněz ukazovat v systému
- Město (pro počasí)** – Nastavuje, pro jaké město se má na domovské stránce zobrazovat počasí
- Otočit kameru vzhůru nohama** – V případě, že se nainstaluje kamera vzhůru nohama, lze snímky automaticky otočit zpět
- Maximální rozlišení (šířka) při výpočtu tváří** – na jaké rozlišení se každá fotografie před rozpoznáváním zmenší (pro rychlejší (ale méně přesné) rozpoznávání); jako naprosto uspokojivou hodnotu vidím 1280 px
- Maximální rozlišení (šířka) při rozpoznávání tváří** – V jakém rozlišení se snímky ukládají do databáze; pokud není kamera příliš daleko od dveří, stačit bude rozlišení 640 px, jinak zvolte vyšší (zabere více místa v databázi)

**Zpoždění focení po detekci pohybu** – Zpoždění mezi detekcí pohybu a pořízením snímku (hodí se, pokud je kamera daleko od dveří)

**Počet snímků v sérii po detekci pohybu** – Kolik snímků pořídí kamera po detekci pohybu

**Timeout při čekání na pohyb** – Jak dlouhá bude maximálně blokáce běhu programu při čekání na pohyb (uvolní prostředky pro ostatní procesy, ale prodlouží se doba, po které proces focení zkontroluje, jestli se má ukončit (byl-li na to zadán příkaz)

**Zobrazit uživatelům čistou mzdu a její výpočet** – umožní zobrazit částku, kterou systém vypočte jako čistou mzdu; je k tomu potřeba nastavit další účetní parametry následující po tomto parametru (nejsou zde popsány)

#### 3.5.3 Začátek používání

Po prvním spuštění se přihlaste na účet s uživatelským jménem „admin“ a heslem „admin“. Přejděte do sekce Administrace → správa zařízení, zadejte adresu rozpoznávacího modulu v síti a přidejte modul kamera. Upravte v podsekcí Parametry požadované parametry systému, jako například práva jistoty správnosti rozpoznávání (viz. sekce 3.5.2), je-li to potřeba. Systém je připraven k provozu.

Nyní můžete spustit rozpoznávání a z menu záznamy můžete začít přidávat první tváře (usmějte se před kamerou). U každého dosud nezaloženého člověka můžete z výběru známých lidí vybrat poslední položku a napsat jméno. Není-li systému známé, přesměruje vás na vytváření nového člověka. Poté se můžete vrátit do menu záznamy a přiřadit tvář vytvořenému uživateli. Nového uživatele je možné vytvářet také přes Uživatelé → Nový uživatel a fotografie je možné také nahrávat (a použít vlastní, nikoliv jen ty pořízené modulem kamera).

Vytvořte také svůj profil, jakožto profil administrátora, zvolte Vygenerovat heslo a pomocí tohoto hesla se do systému přihlaste. Následně si heslo hned změňte (v menu Profil – ikona panáčka v menu) a odstraňte účet admina. Pozor – nezapomeňte heslo administrátora (nebo alespoň ne manažera). Jen tito mohou generovat hesla dalším uživatelům, což je jediná možnost, jak vyřešit zapomenuté heslo.

V prvních dnech bude nutné zvýšit pozornost na nerozpoznané tváře (nejrychleji se k nim lze dostat přes tlačítko Nerozpoznané tváře na domovské stránce, které se objeví, pokud nějaké takové jsou) a přiřazovat je správným uživatelům. Systém se je tak naučí a čím více tváří bude znát, tím přesnější bude následné rozpoznávání. Je dobré také potvrzovat nejisté rozpoznání v sekci záznamy (u uživatele se objeví otazník).

---

# Testování

V této kapitole popíši, jak jsem postupoval při testování, co je potřeba ještě otestovat a zhodnotím, jak se systém povedl.

## 4.1 Propojení modulů

Jako velice problematická se ukázala oblast propojení systémů a zejména propojení modulu kamera a rozpoznávajícího modulu s databází. Často docházelo k pádům modulů kamera i rozpoznávajícího modulu, ze kterého se modul již nezotavil a musel jsem je manuálně restartovat.

Prvním řešením, ke kterému jsem přistoupil, bylo zachytávání všech možných výjimek, které mohou v aplikaci nastat, a přidat je do odpovědi, aby mohl administrátor adekvátně zareagovat, ale hlavně, aby modul běžel dál a mohl být dále používán.

Dále jsem přistoupil k výměně knihovny komunikující s databází, tedy `mysql-connector-python` [62] za `pymysql` [47]. Pak ale začalo docházet často ke ztrátě spojení s databází. To jsem vyřešil tak, že pokud selže vykonávání SQL příkazu, modul se pokusí připojit znovu a znovu provést příkaz. Pokud nepřijde úspěch ani po druhém kole, přeruší se vykonávání aktuální úlohy a v té další se modul pokusí připojit znovu. V případě výpadku databáze tak po opravě není nutné modul restartovat.

Dalším krokem je spouštění modulů přes Supervisor [63] nebo Docker [35] (pouze v případě rozpoznávacího modulu). Ten zajistí automatický start po zapnutí systému a restart v případě, že by i tak modul spadl.

V neposlední řadě jsem také implementoval automatický dopočet (spouštěný periodicky) záznamů, které nebyly rozpoznány například v důsledku výpadku spojení s rozpoznávacím modulem.

Když tedy srovnám funkčnost na začátku, kdy moduly běžel maximálně několik hodin, pak spadly a bylo nutné je manuálně restartovat, a nyní, kdy

moduly běží a lze je ovládat pohodlně přes webový portál, je vidět veliký pokrok a moduly mohu prohlásit za stabilní.

Bezchybnost komunikace mezi moduly záleží také na routeru, na tom, jestli je připojeno kabelem nebo přes wifi a na mnoho dalších elementech. Při testování doma došlo ke zlepšením výměnou wifi routeru.

### 4.2 Rozpoznávání tváří

Knihovna na rozpoznávání tváří funguje výborně (až překvapivě), díky postupnému učení, které jsem v systému programoval, se za velmi krátkou dobu stane rozpoznání s jistotou bezchybné. S parametry, jaké jsem nastavil (práh jistoty správnosti rozpoznávání), nedocházelo v průběhu testování k žádné nesprávné detekci tváře, a také u těch nejistých (ale pravděpodobných) tváří úspěšnost překračovala 95%.

Později jsem narazil na to, že knihovna má implementovaný také klasifikační algoritmus KNN, který by rozpoznávání ještě zpřesnil (více je popsáno v sekci 3.2) a který plánuji v budoucnu implementovat.

Krátce před odevzdáním této práce se mi také podařilo eliminovat falešné detekce pohybu jednak ladění trimrů sensitivity a času a jednak připojením přes kabel a vypnutí wifi (další informace v sekci 2.6.6). Dříve přes noc běhu, kdy v místnosti nikdo nechodil, došlo ke stovkám detekcí a pořízených zbytečných snímků. Nyní přes noc nedojde k žádné falešné detekci.

### 4.3 Webový portál

V průběhu programování jsem mnohokrát měnil přístupy. Největší změnou bylo použití ORM doctrine [21], které mi zajistí namapování dat z databáze na entity v PHP. Dospěl jsem k tomu, že je to čistější řešení než mnoho funkcí s SQL, omezí to duplikace kódu a usnadní psaní.

Docházelo také k častým chybám aplikace, kdy implementování nebo oprava jedné funkce portálu rozbila jinou. S tím má ale zkušenost, řekl bych, každý programátor. Webový portál jsem otestoval proklikáním a simulací všech různých možných stavů a ve fázi odevzdání práce již k žádným chybám nedochází (až na drobnosti, které ale nepřekáží v používání).

Systém se také nasadil na testování v malé firmě, kde ale zatím neběžel dostatečně dlouho, aby se odhalily všechny chyby a nasbíralo větší množství dat a s tím odhalila místa, kde bude docházet k výkonnostním nedostatkům (dlouhé odezvy portálu).

### 4.4 Vyhodnocení

Ačkoliv jsem nestihl naprogramovat všechny funkce (zbývají ale jen drobnosti), odevzdávám systém, za který se nestydím podepsat. Rozpoznávání fun-

guje perfektně, moduly spolu komunikují správně a v případě chyby se zotaví, webový portál má všechny požadované funkce a „vypadá k světu“. Řekl bych, že jsem předčil svá očekávání. Také jsem tím strávil spoustu času.

Snažil jsem se totiž vytvořit systém, který bude možné skutečně využívat (ačkoliv motivací bylo také vyzkoušet si technologii založené na umělé inteligenci). Systém je nasazený v malé firmě k testování a velmi by mě potěšilo, kdyby našel reálné využití. Myslím, že potenciál k tomu má. To ukáže čas.



---

## Závěr

Cílem této práce bylo navrhnout a implementovat webový portál, který automaticky eviduje docházku zaměstnanců a který využívá již funkční řešení rozpoznávání tváří. Součástí portálu měla být administrace a doprovodná funkcionalita jako například žádání a schvalování dovolené.

Vytvořený systém podporuje uživatelské role, umožňuje kontrolovat historii docházek jednotlivých zaměstnanců, zadávat dovolené samotnými zaměstnanci, zobrazit statistiky a přehledy z dat docházek zaměstnanců a bezpečně zpřístupňuje data aplikacím třetích stran.

Navíc se systém v průběhu používání učí tváře, u kterých nebylo jisté, zda patří určitému zaměstnanci, a tím se rozpoznávání zdokonaluje.

Systém byl testován především v domácích podmínkách, kde fungoval takřka bezchybně. Především se mnoho chyb zvládlo opravit. Systém se též nasadil v menší firmě, nicméně ještě nebyl dostatek času na vyhodnocení. Systému z mého pohledu nic nechybí, nicméně každá firma má své potřeby a bude jistě chtít systém rozšířit o nějaké specifické funkce.

Mně samotnému rozšířila obzory v oblasti umělé inteligence, získal jsem zkušenosti s vývojem komplexního systému složeného s více modulů a zdokonalil jsem dovednosti s vývojem webových aplikací.

Systém jsem celou dobu programoval tak, aby bylo možné jej reálně využívat. A myslím, že potenciál k tomu má.





---

## Bibliografie

1. SAFESCAN. *Timemoto TM-838 se RFID a rozpoznávání tváře* [online] [cit. 2019-01-08]. Dostupné z: <https://www.safescan.com/cs-cz/store/dochazkove-pristroje/timemoto-tm-838-dochazkovy-system>.
2. INNOVATRICES S.R.O. *Docházkový systém Fingera* [online] [cit. 2019-01-08]. Dostupné z: <https://www.fingera.com/cs/dochazkovy-system-fingera/>.
3. EUROSAT CS, SPOL. S R.O. *Docházkový systém SYSDO* [online] [cit. 2019-01-08]. Dostupné z: <https://www.sysdo.cz/dochazkovy-system/>.
4. GARDON INT S.R.O. *Docházkový systém - Face Recognition, Mifare/Desfire, PoE, WiFi* [online] [cit. 2019-01-08]. Dostupné z: <http://www.gardon.cz/dochazkovy-system-face-recognition-mifare-desfire-poe-wifi-z1627.html>.
5. MATRIX COMSEC PVT. LTD. *Facial recognition attendance system* [online] [cit. 2019-01-08]. Dostupné z: <https://www.matrixaccesscontrol.com/facerecognition.html>.
6. RASPBERRY PI FOUNDATION. *Raspberry Pi* [online] [cit. 2018-12-27]. Dostupné z: <https://www.raspberrypi.org/>.
7. JONES, Dave. *picamera* [online]. Verze 1.13 [cit. 2018-12-30]. Dostupné z: <https://picamera.readthedocs.io/en/release-1.13/>.
8. THE APACHE SOFTWARE FOUNDATION. *Apache HTTP Server* [online] [cit. 2018-12-13]. Dostupné z: <https://httpd.apache.org>.
9. BUILTWITH® PTY LTD. *built with* [online] [cit. 2018-12-13]. Dostupné z: <https://builtwith.com>.
10. THE PHP GROUP. *PHP* [online] [cit. 2018-12-13]. Dostupné z: <http://php.net>.

11. MOZILLA. *MDN web docs* [online]. 2016 [cit. 2018-12-13]. Dostupné z: <https://developer.mozilla.org/cs/docs/Web/JavaScript>.
12. THE JQUERY FOUNDATION. *jQuery* [online] [cit. 2018-12-13]. Dostupné z: <https://jquery.com>.
13. NETTE FOUNDATION. *Nette dokumentace* [online] [cit. 2018-12-16]. Dostupné z: <https://doc.nette.org/cs/2.4/getting-started>.
14. BERNARD, Borek. *Úvod do architektury MVC* [online]. 2009 [cit. 2018-12-16]. Dostupné z: <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>.
15. NETTE FOUNDATION. *O Nette* [online] [cit. 2018-12-16]. Dostupné z: <https://nette.org/cs/about>.
16. ADERMANN, Nils; BOGGIANO, Jordi. *Composer* [online]. Verze 1.8.0 [cit. 2018-12-16]. Dostupné z: <https://getcomposer.org/>.
17. DOWLING, Michael. *Guzzle, PHP HTTP client* [online] [cit. 2018-12-16]. Dostupné z: <http://docs.guzzlephp.org/en/stable/index.html>.
18. TWITTER. *Bootstrap* [online]. Verze 4.1.3 [cit. 2018-12-16]. Dostupné z: <https://getbootstrap.com/>.
19. ČECH, Petr. *Bootstrap 4 forms for Nette* [online] [cit. 2018-12-16]. Dostupné z: <https://componette.com/czubehead/bootstrap-4-forms/>.
20. NEXTRAS COMMUNITY. *Nextras datagrid* [online]. Verze 3.0 [cit. 2018-12-17]. Dostupné z: <https://nextras.org/datagrid/docs/3.0/>.
21. *Kdyby/Doctrine* [online] [cit. 2018-12-17]. Dostupné z: <https://github.com/kdyby/doctrine>.
22. *Kdyby/Translation* [online] [cit. 2018-12-17]. Dostupné z: <https://github.com/Kdyby/Translation>.
23. *Kdyby/Console* [online] [cit. 2018-12-17]. Dostupné z: <https://github.com/Kdyby/Console>.
24. FONTICONS, INC. *Font Awesome* [online]. Verze 5.6.1 [cit. 2018-12-17]. Dostupné z: <https://fontawesome.com/>.
25. HUBSPOT. *PACE* [online]. Verze 1.0.2 [cit. 2018-12-17]. Dostupné z: <https://github.hubspot.com/pace/>.
26. DESANDRO, David. *Masonry* [online] [cit. 2018-12-17]. Dostupné z: <https://masonry.desandro.com/>.
27. *Chart.js* [online]. Verze 2.0 [cit. 2018-12-17]. Dostupné z: <https://www.chartjs.org/>.
28. *Autocomplete: jqueryUI + Nette* [online] [cit. 2018-12-17]. Dostupné z: <https://gist.github.com/tynovsky/1207351>.

29. MOOSA, Azmath. *Best face recognition SDKs as of 2018* [online] [cit. 2019-01-09]. Dostupné z: <https://www.baseapp.com/computer-vision/best-face-recognition-sdks/>.
30. WALLING, Alex. *Top 10 Facial Recognition APIs of 2018* [online] [cit. 2019-01-09]. Dostupné z: <https://blog.rapidapi.com/top-facial-recognition-apis/>.
31. MEGVII. *Facial Recognition SDK* [online] [cit. 2019-01-09]. Dostupné z: <https://www.faceplusplus.com/>.
32. MICROSOFT. *Tvář* [online] [cit. 2019-01-09]. Dostupné z: <https://azure.microsoft.com/cs-cz/services/cognitive-services/face/>.
33. LAMBDA LABS. *Face Recognition API* [online] [cit. 2019-01-09]. Dostupné z: <https://lambdalabs.com/face-recognition-api>.
34. AMOS, Brandon; LUDWICZUK, Bartosz; SATYANARAYANAN, Mahadev. *OpenFace: A general-purpose face recognition library with mobile applications* [online]. 2016 [cit. 2019-01-09]. Dostupné z: <https://cmusatyalab.github.io/openface/>.
35. DOCKER INC. *Docker* [online] [cit. 2019-01-07]. Dostupné z: <https://www.docker.com/>.
36. SOFTWARE IN THE PUBLIC INTEREST, INC. *Debian* [online] [cit. 2019-01-09]. Dostupné z: <https://www.debian.org/>.
37. GITHUB, INC. *TOPIC face-recognition* [online] [cit. 2019-01-09]. Dostupné z: <https://github.com/topics/face-recognition>.
38. GEITGEY, Adam. *face recognition* [online]. 2018. Verze 1.2.3 [cit. 2018-12-27]. Dostupné z: [https://github.com/ageitgey/face\\_recognition](https://github.com/ageitgey/face_recognition).
39. KING, Davis; DALŠÍ. *Dlib C++ library* [online] [cit. 2019-01-09]. Dostupné z: <http://dlib.net/>.
40. HUANG, Gary. *Labeled Faces in the Wild* [online] [cit. 2019-01-09]. Dostupné z: <http://vis-www.cs.umass.edu/lfw/>.
41. PYTHON SOFTWARE FOUNDATION. *python* [online]. Verze 3.5 [cit. 2018-12-27]. Dostupné z: <https://www.python.org/>.
42. PYTHON SOFTWARE FOUNDATION. *configparser — Configuration file parser* [online] [cit. 2018-12-30]. Dostupné z: <https://docs.python.org/3/library/configparser.html>.
43. HEINISUO, Olli-Pekka. *opencv-python* [online]. 2018. Verze 3.4.5.20 [cit. 2018-12-30]. Dostupné z: <https://pypi.org/project/opencv-python/>.
44. OPENCV TEAM. *OpenCV* [online] [cit. 2018-12-30]. Dostupné z: <https://opencv.org/>.
45. RONACHER, Armin. *Flask* [online] [cit. 2018-12-27]. Dostupné z: <http://flask.pocoo.org/>.

46. NUMPY DEVELOPERS. *NumPy* [online] [cit. 2018-12-30]. Dostupné z: <http://www.numpy.org/>.
47. MATSUBARA, Yutaka. *PyMySQL* [online]. Verze 0.9.3 [cit. 2018-12-30]. Dostupné z: <https://pymysql.readthedocs.io/en/latest/>.
48. PYTHON SOFTWARE FOUNDATION. *time* — *Time access and conversions* [online] [cit. 2018-12-30]. Dostupné z: <https://docs.python.org/3/library/time.html>.
49. PYTHON SOFTWARE FOUNDATION. *sys* — *System-specific parameters and functions* [online] [cit. 2018-12-30]. Dostupné z: <https://docs.python.org/3/library/sys.html>.
50. RASPBERRY PI FOUNDATION. *Getting started with picamera* [online] [cit. 2018-12-27]. Dostupné z: <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera>.
51. RASPBERRY PI FOUNDATION. *Physical Computing with Python: Using a PIR sensor* [online] [cit. 2018-12-27]. Dostupné z: <https://projects.raspberrypi.org/en/projects/physical-computing/13>.
52. NUTTALL, Ben; JONES, Dave. *gpiozero* [online] [cit. 2018-12-30]. Dostupné z: <https://gpiozero.readthedocs.io/en/stable/>.
53. PYTHON SOFTWARE FOUNDATION. *json* — *JSON encoder and decoder* [online] [cit. 2018-12-30]. Dostupné z: <https://docs.python.org/3/library/json.html>.
54. MARKS, Taylor. *playsound* [online]. 2017. Verze 1.2.2 [cit. 2018-12-30]. Dostupné z: <https://pypi.org/project/playsound/>.
55. PYTHON SOFTWARE FOUNDATION. *threading* — *Thread-based parallelism* [online] [cit. 2018-12-30]. Dostupné z: <https://docs.python.org/3/library/threading.html>.
56. *MariaDB* [online] [cit. 2018-12-27]. Dostupné z: <https://mariadb.com/>.
57. ORACLE CORPORATION AND/OR ITS AFFILIATES. *MySQL* [online] [cit. 2018-12-27]. Dostupné z: <https://www.mysql.com>.
58. *MariaDB versus MySQL - Compatibility* [online] [cit. 2018-12-27]. Dostupné z: <https://mariadb.com/kb/en/library/mariadb-vs-mysql-compatibility/>.
59. PRENNER, Michal. *RPishop.cz* [online] [cit. 2018-12-28]. Dostupné z: <https://rpishop.cz/>.
60. GEITGEY, Adam. *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning* [online]. 2016 [cit. 2018-12-28]. Dostupné z: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>.

61. BRONSHTEIN, Adi. *A Quick Introduction to K-Nearest Neighbors Algorithm* [online]. 2017 [cit. 2019-01-02]. Dostupné z: <https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7>.
62. BELL, Chuck; MARIZ, Nuno; TEKRIWAL, Prashant; MYSQL RE TEAM. *mysql-connector-python* [online] [cit. 2019-01-07]. Dostupné z: <https://pypi.org/project/mysql-connector-python/>.
63. AGENDALESS CONSULTING AND CONTRIBUTORS. *Supervisor: A Process Control System* [online] [cit. 2019-01-07]. Dostupné z: <http://supervisord.org/>.



---

## Seznam použitých zkratk

- AJAX** Asynchronous JavaScript and XML  
**API** Application Programming Interface (Aplikační rozhraní)  
**ER** Entity-relationship  
**GDPR** General Data Protection Regulation (Obecné nařízení o ochraně osobních údajů)  
**GPIO** General-purpose input/output  
**GUI** Graphical user interface (Grafické uživatelské rozhraní)  
**HTML** Hypertext Markup Language (Hypertextový značkovací jazyk)  
**JSON** JavaScript Object Notation (JavaScriptový objektový zápis)  
**KNN** K-Nearest neighbors (algoritmus)  
**ORM** Object-relational mapping (Objektově relační mapování)  
**OS** Operační systém  
**PHP** Hypertext Preprocessor (Hypertextový preprocesor)  
**PIR** Passive Infrared Sensor (Pasivní infračervené čidlo)  
**REST** Representational State Transfer  
**SDK** Software development kit (Sada vývojových nástrojů)  
**XML** Extensible markup language





---

## Obsah příloženého CD

src.....	zdrojové kódy práce
├ cvut-logo-bw.pdf.....	logo ČVUT vložené do práce
├ FITthesis.cls.....	šablona práce
├ image.....	obrázky vložené do práce
├ ref.bib.....	informace o citovaných dílech pro biber
├ thesis.tex.....	zdrojová forma práce ve formátu $\text{\LaTeX}$
text.....	text práce
├ thesis.pdf.....	text práce ve formátu PDF



---

# Instalační příručka

Instalace systému se bude lišit modul od modulu podle toho, kde a jakým způsobem budou provozovány. Protože jsou moduly vzájemně nezávislé, rozdělím instalaci podle modulů. Z praktického hlediska je databáze také modul, v instalační příručce k ní budu také tak přistupovat.

## C.1 Kamera

Modul kamera je naprogramovaný přímo pro Raspberry PI. Narozdíl od ostatních modulů popíše pouze provoz pod programem Supervisor. Raspberry PI nemá dostatečný výkon pro provoz virtuálních kontejnerů, proto ani neuvažuji provoz pod Dockerem.

### C.1.1 Předpoklady

Na Raspberry PI je spuštěný OS Raspbian, speciálně upravený Debian pro Raspberry PI. Přes balíčkovací systém se nainstaluje z oficiálních repozitářů následující balíčky (a jejich závislosti – ty se nainstalují automaticky) – kód C.1.

```
sudo apt install -y cmake libatlas-base-dev libjasper-dev \  
libqtgui4 libqt4-test mysql-client python-gst-1.0 supervisor  
pip3 install Flask opencv-python playsound pymysql
```

Výpis kódu C.1: Instalace balíčků a knihoven pro python nezbytných pro provoz modulu kamera

Více o knihovnách je možno se dozvědět v části 2.6.3.

Dalším krokem je nakopírování souborů systému – složky camera a sounds z kořenového adresáře zdrojových souborů do adresáře ~/FRAS.

### C.1.2 Konfigurace

Před spuštěním modulu je nutné vytvořit soubor `config.ini` s konfigurací. Šablonou je soubor `config-example.ini`. Každé nastavení je okomentováno v souboru a to nejdůležitější také v sekci 3.1. Bez správného nastavení připojení k databázi nebude možné modul vůbec spustit.

### C.1.3 Spuštění pod Supervisorem

Celou aplikaci je nyní možné spustit příkazem `./camera.py`. Má to ale nevýhody – aplikace bude běžet v terminálu a nelze ho tak ukončit, pokud v aplikaci dojde k chybě způsobující pád, aplikace se sama nerestartuje, a po restartu systému se sama nespustí. Proto doporučuji aplikaci spouštět přes supervisor. Spouštění se nastaví spuštěním scriptu `install.sh` ve složce `camera` (pokud aplikace nemá být provozována pod uživatelem `pi` (výchozí uživatel v Raspbianu), je nutné tento script pozměnit).

## C.2 Rozpoznávací modul

Rozpoznávací modul může být provozován buď pod programem Supervisor nebo jako virtuální kontejner pod Dockerem. V případě provozu modulu na Raspberry PI je možný pouze provoz pod Supervisorem.

### C.2.1 Provoz pod Supervisorem

V tomto případě je instalace stejná jako u modulu kamera (viz. C.1). Jediným rozdílem je instalace balíčků a knihoven potřebných pro provoz modulu (kód C.2).

```
sudo apt install -y cmake libatlas-base-dev libjasper-dev \
    libqtgui4 libqt4-test mysql-client python-gst-1.0 supervisor
pip3 install face_recognition Flask opencv-python pymysql
```

Výpis kódu C.2: Instalace balíčků a knihoven pro python nezbytných pro provoz rozpoznávacího modulu

Tento postup instalace platí pro linuxové distribuce založené na Debianu nebo Ubuntu.

### C.2.2 Provoz pod Dockerem

Instalace pro provoz pod Dockerem je jednoduchá. Nainstaluje se multiplatformní program Docker, vytvoří se virtuální kontejnery (postup vytváření je již předepsaný) a spustí se. Tímto se vytvoří kontejnery s rozpoznávacím modulem, webovým portálem a databází. Instalace trvá až desítky minut,

spuštění je během okamžiku. Nelze provozovat na Raspberry PI (zde použijte instalaci pod Supervisorem). Vytvoření kontejnerů a spuštění ilustruje kód C.3, který předpokládá zdrojové kódy umístěné ve složce FRAS v domovském adresáři.

```
cd ~/FRAS
docker-compose build
docker-compose up -d
```

Výpis kódu C.3: Instalace virtuálních kontejnerů a jejich spuštění

## C.3 Webový portál

Webový portál lze provozovat napřímo (nainstalováním serveru Apache a PHP) nebo pod Dockerem. V případě provozu na Raspberry PI lze provozovat pouze napřímo.

### C.3.1 Provoz napřímo

Následně je uveden kód pro instalaci potřebného softwaru na linuxové distribuci Debian nebo Ubuntu (kód C.4). Obdobně lze postupovat i na ostatních operačních systémech.

```
sudo apt install -y apache2 composer mysql-client php7.0 \
    php7.0-gd php7.0-mysql php7.0-sqlite3
```

Výpis kódu C.4: Instalace softwaru potřebného pro provoz webového serveru

Dále je třeba nakopírovat soubory webového modulu (například do složky FRAS v domovském adresáři) a nalinkovat podadresář www na /var/www/html. Možností je více, tato je ta nejjednodušší. Poté je třeba nainstalovat používané knihovny pomocí Composeru a nastavit práva k zápisu pro všechny (kvůli Nette) na podadresáře temp a log. Kód C.5.

```
cd ~/FRAS/web
composer update
chmod -R 777 temp log
sudo mv /var/www/html /var/www/html.old
sudo ln -s www /var/www/html
```

Výpis kódu C.5: Instalace knihoven potřebných pro provoz webu a nalinkování adresáře www

Poté je třeba přidat řádky v kódu C.6 do konfiguračního souboru Apache `/etc/apache2/sites-enabled/000-default.conf`, aby správně fungovaly URL adresy a povolit modul rewrite (kód C.7).

```
<Directory /var/www/html>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Výpis kódu C.6: Povolení symbolických adres v Apache

```
sudo a2enmod rewrite
```

Výpis kódu C.7: Povolení modulu rewrite v Apache

A nakonec se prodlouží platnost session (aby přihlášení bylo platné místo 20 minut až 3 dny) – edituje se řádek `session.gc_maxlifetime = 259200` v souboru `/etc/php/7.0/apache2/php.ini` a přenačte se konfigurace (kód C.8).

```
sudo systemctl restart apache2
```

Výpis kódu C.8: Restartování Apache

### C.3.2 Provoz pod Dockerem

V případě provozu modulu pod Dockerem je práce velice usnadněná. Veškerý software se sám nainstaluje a nakonfiguruje ve virtuálním kontejneru. Webový modul se nainstaluje společně s rozpoznávacím modulem a databází – viz. C.2.2.

## C.4 Databáze

Stejně jako webový modul, databáze se dá provozovat buď napřímo nebo pod Dockerem. V případě provozu na Raspberry PI lze provozovat pouze napřímo.

### C.4.1 Provoz napřímo

Pokud v OS není nainstalována databáze MySQL nebo MariaDB, jednu z nich je potřeba nainstalovat (doporučená je MariaDB, ale funguje také MySQL). Poté už jen stačí vytvořit přístup pro nového uživatele `fras` (pro naše moduly – kód C.9). Pokud je databáze právě nainstalovaná a je používán OS Linux, stačí zavolat `sudo mysql`.

```
USE mysql;  
CREATE USER fras@localhost IDENTIFIED BY 'fras1234';  
GRANT ALL PRIVILEGES ON * . * TO 'fras'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

Výpis kódu C.9: Nastavení práv v databázi pro docházkový systém

Druhým krokem je import tabulek a dat. To je možné provést například přes adminer, který je dostupný z webového modulu (<http://localhost/adminer.php>). Data k importu jsou v podadresáři database, soubor fras.sql.

### C.4.2 Provoz pod Dockerem

V případě provozu modulu pod Dockerem je práce velice usnadněná. Veškerý software se sám nainstaluje a nakonfiguruje ve virtuálním kontejneru. Databáze se nainstaluje společně s rozpoznávacím a webovým modulem – viz. C.2.2.