

ZÁPADOČESKÁ UNIVERZITA  
FAKULTA APLIKOVANÝCH VĚD  
KATEDRA KYBERNETIKY

../pic/fav.jpg

Dokumentace distribuci projektu  
ITE-YELLOW

*Martin Hamar, Radek Kaupe, Samuel Kokoška*

# 1 Úvod

V rámci předmětu KKY/BSOI jsme si vytvořili a nastavili virtuální stroj s operačním systémem Linux. Tento stroj je momentálně používán k provozu aplikace vyvíjené v rámci předmětu KKY/ITE. Zde je na aplikaci a zde je odkaz na Githubový repozitář, kde jsou k dispozici veškeré zdrojové kódy aplikace jak části ITE, tak části BSOI. Tento referát by měl popsat virtuální stroj, jeho nastavení, činnosti a realizaci způsobů přihlášení do již zmíněné aplikace.

## 2 Řešení

### 2.1 Provoz distribuované aplikace

Součástí aplikace je PostgreSQL databáze a dva python skripty, jeden z nich řeší MQTT subscribenta, abychom mohli data přijímat, validovat a ukládat a druhý tvoří backend naší webové aplikace. K založení PostgreSQL databáze, byl také vytvořen python skript, který stačí spustit pouze jedno, aby vytvořil tabulky. Skripty na subscribenta a backend musí běžet neustále, aby aplikace mohla být v provozu. Tyto skripty běží současně pomocí tmuxu. Tmux je ‘Terminal Multiplexer’, který umožňuje rozdělit obrazovku terminálu na více panelů a vytváření ‘sezení’ (sessions), které mohou běžet i na pozadí, poté co se člověk odpojí. Subscriber a backend spolu ‘komunikují’ pouze přes databázi, ke které mají oba skripty přístup.

#### ZAJISTI AUTOMATICKY SPUSTENÍ PO RESTARTU

Momentálně logování výstupu je k vidění pouze v rámci tmux session, kde jsou nastavené výpisy za běhu obou skriptů, aby mohlo dojít ke kontrole kdykoliv některým členům z týmu. Nasazení verzí se může provést pomocí programu WinSCP, kterým se nové soubory nahrají na virtuální stroj.

## 3 Přihlášení a biometrická autentizace

Při implementaci jsem využil učitelů dodaných zdrojů, ve formě skriptů `faceid_server.py`, `extract_embeddings.py` a `train.sh`.

#### Rozšíření backendu o FaceID

V prvním skriptu se nacházejí Tornado Handleři, jeden zajišťující komunikaci s frontendem v rámci posílání a ukládání fotek a druhý v rámci detekci obličeje ve fotce. Předtím oba dva handleři ???fungovali??? v rámci jedné URL adresy. Já je rozdělil. Aplikace funguje následovně:

1. Nepřihlášený uživatel se může buď přihlásit přes uživatelské jméno a heslo (vstupní stránka), přihlásit se přes FaceID nebo se zaregistrovat
2. Po registraci musí uživatel čekat na schválení adminem, jinak se nepřihlásí (ani v případě, že již fotky má uložené v rámci biometrické autentizace)
3. Přihlášený uživatel má přístup k celé aplikaci, včetně možnosti nafotit si své fotky pro FaceID a natrénovat model pro autentizaci obličejů

#### Uživatelské účty a identity

Máme celkem 9 uživatelských účtů pro přihlášení, pro:

- 4 vyučující
- 3 členy týmu
- 1 admin účet, ke kterému mají přístup všichni členové týmu

Identity je 8. Jsou vytvořené pro všechny členy týmu, jeden kategorie ‘unknown’ (aby se ‘nové’ tváře nemohli přihlásit, jinak by je systém identifikoval jako validního uživatele) a 3 identity učitelé, aby mohlo dojít k právě zmíněné identity ‘unknown’. Kategorie ‘unknown’ má celkem TODO fotek. Každý z členů má 10 až 25 fotek, které si každý sám nafotil. Od každého vyučujícího jsme dostali 10 - 25 fotek. Každý s různým počtem, aby šlo testovat úspěšnost identifikace. Rozhodovací práh je nastaven na alespoň TODO: MOŽNA ZMĚN/SNÍŽ 80% pravděpodobnost identifikace uživatele. Je nutné podotknout, že identita ‘unkown’ není vůbec uložena v databázi a nemá možnost se do aplikace přihlásit.

### **Funkčnost přihlašování**

V případě, že se člověk přihlásí, je mu nastaven Json Web Token (JWT). Ten řeší zda je uživatelská session je validní či nikoliv. Pokud jsou uživatelské credentials v pořádku a uživatel je zaregistrován a schválený adminem je mu JWT uložen jako secure cookie. Všechny další žádosti nyní automaticky kontrolují zda je JWT neexpiroval nebo s ním nebylo manipulováno. JWT má totiž nastavenou dobu expirace jedné hodiny a obsahuje ‘podepsán’ tajným klíčem, který je znám pouze serveru.

### **Odhlásování**

Pro odhlásování byl vytvořen Tornado Handler, který vymaže token obsahující cookie a přesměruje uživatele zpět na přihlašovací stránku.

## **4 Závěr**