UNIVERZITA Hradec Králové  
Fakulta informatiky a managementu

Hello FIM

Seminární práce z předmětu Databázové systémy 2

Členové pracovního týmu:

Ivan Osipchyk,

Denis Feshchenko,

Palina Fralova

V Hradci Králové dne….

## Obsah

[Obsah 2](#_heading=h.3znysh7)

[**Úvod 3**](#_heading=h.2et92p0)

[**Zadání 4**](#_heading=h.tyjcwt)

[**Uživatelská dokumentace 5**](#_heading=h.3dy6vkm)

[Základní popis používané aplikace 6](#_heading=h.1t3h5sf)

[Instalace 6](#_heading=h.4d34og8)

[Přístupová oprávnění 6](#_heading=h.2s8eyo1)

[Použití aplikace 6](#_heading=h.17dp8vu)

[**Programová dokumentace 7**](#_heading=h.3rdcrjn)

[**Datová část 8**](#_heading=h.26in1rg)

[Analýza 8](#_heading=h.lnxbz9)

[Fyzický model dat 8](#_heading=h.35nkun2)

[Číselníky 8](#_heading=h.1ksv4uv)

[Pohledy 8](#_heading=h.44sinio)

[Funkce 8](#_heading=h.2jxsxqh)

[Uložené procedury 8](#_heading=h.z337ya)

[Spouště 8](#_heading=h.3j2qqm3)

[Indexy 8](#_heading=h.1y810tw)

[Sekvence 8](#_heading=h.4i7ojhp)

[**Aplikace 9**](#_heading=h.2xcytpi)

[Použité prostředí 9](#_heading=h.1ci93xb)

[Řízení uživatelských účtů 9](#_heading=h.3whwml4)

[Moduly 9](#_heading=h.2bn6wsx)

[Formuláře 9](#_heading=h.qsh70q)

[Orientace ve zdrojovém kódu 9](#_heading=h.3as4poj)

[**Závěr 10**](#_heading=h.1pxezwc)

[Závěr 11](#_heading=h.49x2ik5)

[Přílohy 12](#_heading=h.2p2csry)

[Backup databáze 12](#_heading=h.147n2zr)

[Zdrojové kódy aplikace, grafika, apod. 12](#_heading=h.3o7alnk)

[Případně ostatní 12](#_heading=h.23ckvvd)

## Úvod

Obsahuje základní seznámení s organizací, procesy uvnitř organizace a způsob současného využívání IT.

Projekt zahrnuje vývoj webové aplikace. Frontend aplikace bude vyvinut pomocí technologie React. Frontend zahrnuje stránky pro registraci uživatelů, registraci administrátorů, výpis produktů a možnosti vyhledávání a filtrování. Denis je zodpovědný za vytvoření uživatelského rozhraní a komponent frontendu. Backend aplikace bude vyvíjen pomocí programovacího jazyka Python a rámce Flask. Lehká struktura Flaska umožní efektivní směrování, zpracování dat a integraci s frontendem. Backend bude komunikovat s frontendem pomocí RESTful API. Ivanova hlavní odpovědnost bude zpracování registrace uživatelů a administrátorů, správa uživatelských sezení a implementace dvoukrokového procesu registrace administrátorů. Paliny je správa databáze a dokumentace. Databáze bude spravována v kontejneru Dockeru pro snadnou nasazení a škálovatelnost. Zvolenou technologií databáze je SQL, která poskytuje strukturované uložiště pro ukládání uživatelských dat, informací o produktech a administrativních údajů. Zdrojový kód projektu bude hostován na GitHubu, což usnadní spolupráci na vývoji a verzování mezi členy týmu.

## Zadání

Obsahuje popis prostředí, v němž bude aplikace nasazena (stručná charakteristika organizace a dosavadní využívání IT), a požadavky na nově tvořený systém (sledované údaje, vstupy a výstupy, přístupová práva, ...)

Cílem našeho projektu je udělat použitelnou databáze a frontend pro obchod tenísek. Náš projekt bude podobou tohoto webu: https://www.footlocker.cz/ . rozdílem bude to, že se náš tým soustředí na implementace databáze, registrace uživatelů, registrace administrátorů, registrace nového sortimentu a vyhledávání zboží. Na našem webu bude dva formuláře pro registrace nového uživatelů a sortimentu. Pro uživatelé budeme sledovat jeho jméno, příjmení, email, telefon, heslo a adresu doručení. Pro heslo použijeme hash funkce, aby data byly spolehlivě ochraněny. Pro nový sortiment budeme sledovat název, cenu, výrobce, barvu a rok vydání. Nový sortiment může přidat pouze jeden z administrátorů. Pro administrátora budeme sledovat jmeno, příjmení, email a heslo. Také při registrace nový administrator bude muset uvést klíč, který dostane od jiného, už zaregistrovaného administrátora. Vyhledávání na webu bude pracovat podle názvu zboží. Také budou existovat možnost filtrování zboží podle výrobce a barvy, a také seřazení podle ceny a roku vydání. Jak už bylo řečeno dříve, na webu bude dvě možnosti přístupu: uživatelský a administrátorský. Pro registrace uživatelů je potřeba uvést pouze osobní údaje. Pro registrace administrátora je potřeba uvést klíč, ke kterému má přístup již registrovány admin. Na začátku fungování systému do databáze bude uveden jeden výchozí administrátor, který bude mít klíč pro registrace dalších administrátorů

# Uživatelská dokumentace

### Základní popis používané aplikace

Popište základní popis aplikace, co může uživatel od aplikace očekávat apod., co řeší a co nikoliv.

Aplikace obchodu se teniskami je webová platforma, která má za cíl poskytnout uživatelům intuitivní rozhraní pro prohlížení a nákup tenisek. Uživatelé mohou vyhledávat produkty podle názvu sáčku. Kromě toho existují možnosti filtrování produktů podle výrobce a barvy a třídění podle ceny a roku vydání. To zlepšuje uživatelský zážitek tím, že jim pomáhá snáze najít sáčky, o které mají zájem. Uživatelé mohou přidávat sáčky do svého nákupního košíku, prohlížet si svá výběry a přejít k nákupu. Mohou také odstraňovat položky z košíku, pokud je to potřeba. Tato funkce umožňuje uživatelům plynulý nákupní zážitek.

### Instalace

Uveďte informace o instalaci aplikace.

Před instalace aplikace musí být nainstalované MySQL, python a next.js

Pro instalace:

1. Otevřít project a terminal
2. Spustit database/schema.sql, database/configurations.sql, database/views.sql, database/functions.sql
3. Spustit backend/server.py
4. V terminálu ‘cd frontend’ a ‘npm run dev’

### Přístupová oprávnění

Popište, jakým způsobem je možné vystupovat v různých rolích v aplikaci, uveďte ukázkové přihlašovací údaje (loginy a hesla) pro všechny úrovně.

Uživatelé se mohou registrovat na platformě tím, že poskytnou své osobní údaje, včetně křestního jména, příjmení, e-mailu, telefonního čísla, hesla a dodací adresy. Hesla jsou zabezpečeně zahešované před uložením do databáze, aby byla zajištěna bezpečnost dat.

Administrátoři se mohou registrovat pomocí jedinečné registrační klíče poskytnutého jiným již registrovaným administrátorem. Musí poskytnout své křestní jméno, příjmení, e-mail a heslo. To zajišťuje, že pouze autorizované osoby mohou získat přístup administrátora.

Pouze administrátoři mají schopnost spravovat inventář. Mohou přidávat nové položky inventáře, včetně názvu sáčku, ceny, výrobce, barvy a roku vydání. To umožňuje efektivní kontrolu nad produkty dostupnými pro uživatele.

Uživatelé i administrátoři musí provést přihlášení pomocí svých příslušných přihlašovacích údajů, aby získali přístup k platformě.

### Použití aplikace

Sepište jednoduchý návod na používání aplikace, rozdělte jej na jednotlivé moduly aplikace. Z této části musí být jasné, jak je možné program využívat.

Aplikace je možné využívat dvěma způsoby: jako uživatel a jako admin.

Uživatel může prohlížet zboží s filtry, dávat je do tašky, a nakupovat.

Admin může přidat nové zboží, přidat zboží do skladu, změnit nebo odstranit zboží z databáze.

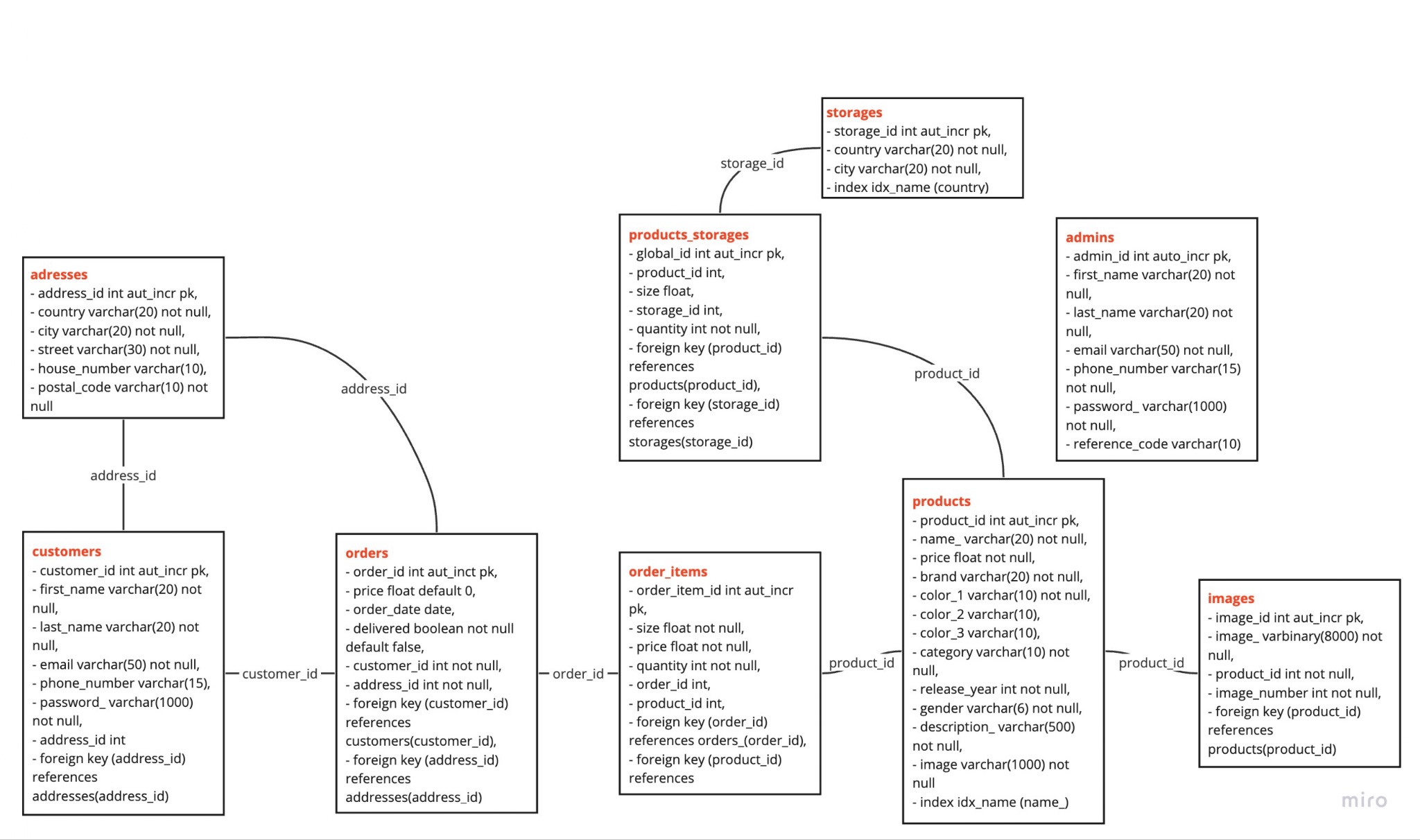
# Programová dokumentace

## Datová část

Zde uveďte řešení zadavatelské dokumentace z pohledu datové části, postupujte dle zadání projektu.

### Analýza

Bude obsahovat entitně-vztahový diagram (ERD) a datový model popisující datové prostředí aplikace a dále stručnou charakteristiku uživatelského rozhraní (typy formulářů, tiskových sestav, .), které bude nad zmíněným datovým prostředím pracovat.



Databáze obsahuje 9 tabulek.

Tabulka customers chrání data o uživatelích, včetně odkazu na jejich adresy.

Tabulka addresses chrání jednotlivé adresy. Když se uživatel registruje, uvádí adresu. Jestli taková adresa již existuje, přidává do tabulky customers její index. Jestli ne, pak ji vytváří a přidává do tabulky customers nový index.

Tabulka orders má informace o objednávkách. Když se nový řádek vytvoří, pak order\_date je null. Když objednáno, mění order\_date na dnešní datum.

Order\_items chrání informace o jednotlivých položkách v objednávce. Má odkazy na objednávky, do které patří, a do produktu, který je v té položce. Také má počet produktů a jeho velikost.

Products chrání informace o produktech. Ale není v tabulce products informace, jaká je velikost a jaký počet zboží je na skladu. Pro to slouží tabulka products\_storages.

Storages má informace o skladech a má index na sloupci země, aby bylo možné tabulky přes ten sloupec propojit.

Images musí mít jednotlivé obrázky. Ale rozhodli jsme se používat obrázky přes odkazy.

Admins má informace o adminech.

### Pohledy

Uveďte jaké pohledy Váš projekt používá včetně SQL všech použitých views.

V našem projektu využíváme následující pohledy spolu s SQL dotazy:

* order\_availability - tento pohled se používá pro kontrolu dostupnosti objednávky. Je využit uvnitř SQL funkce k ověření, zda je požadovaný produkt k dispozici na skladě.
* nike\_products - tento pohled slouží ke zrychlení získávání dat o produktech značky Nike. Jelikož jsou dotazy na tuto značku časté, pohled umožňuje rychlejší získání relevantních informací.
* adidas\_products - podobně jako předchozí pohled, tento pohled je vytvořen pro urychlení získávání dat o produktech značky Adidas.
* jordna\_products - podobně jako předchozí dva pohledy, tento pohled je vytvořen pro urychlení získávání dat o produktech značky Jordan

### Funkce

Uveďte jaké funkce Váš projekt používá včetně SQL všech použitých funkcí.

* find\_or\_insert\_address - se užívá během registrace uživatele.
* customer\_registration - pro registraci uživatele.
* customer\_login - pro přihlášení uživatele.
* update\_address - pro změnu adresy.
* check\_availability - používá se při přidávání do košíku.
* add\_order - používá se při přidávání do košíku.
* update\_price\_and\_quantity - používá se při přidávání do košíku.
* add\_product\_to\_order - přidání do košíku.
* update\_quantity - pro změnu množství zboží v košíku.
* remove\_from\_order - pro odstranění z košíku.
* admin\_registration - pro registraci administrátora.
* admin\_login - pro přihlášení administrátora.
* add\_storage - pro přidání skladu.
* add\_product - pro přidání produktu.
* add\_product\_to\_storage - pro přidání produktu do skladu.
* purchase\_inner - používá se při nákupu.
* purchase - pro nákup zboží.
* change\_product - pro změnu dat produktu.
* delete\_product - pro odstranění produktu z databáze.

### Indexy

Uveďte jaké typy indexů používáte, vypište všechny indexy na neklíčových sloupcích.

* na sloupce country v tabulce storages pro spojení s jinou tabulkou
* na sloupce name v tabulce products

### Sekvence

### Uveďte jaké sekvence a na jakých sloupcích používáte včetně SQL všech použitých sequences.

* na primary key všech tabulek pomocí auto\_increment

## Aplikace

Zde uveďte řešení zadavatelské dokumentace z pohledu výsledné aplikace, postupujte dle zadání projektu. V celé této části použijte printscreeny Vaší aplikace včetně detailního popisu.

### Použité prostředí

Detailně popište použité prostředí, technologie.

React je nástroj na psaní webových stránek a aplikací. Umožňuje nám vytvářet různé části stránky jako malé stavební kameny, které se dají opakovaně použít.

Python je programovací jazyk, který nám umožňuje psát různé druhy programů, od malých skriptů po velké aplikace.

Flask je framework vytvořený pro psaní webových aplikací v jazyce Python. Framework je jakýsi soubor pravidel a nástrojů, které nám pomáhají stavět složité věci jednodušším způsobem.

SQL je způsob, jak mluvit s databázemi, které uchovávají informace. Je to jako mluvení s velkým úložištěm informací, kde můžeme ptát se na data, přidávat do nich a měnit je.

### Řízení uživatelských účtů

Popište jaký způsob využíváte pro řízení uživatelských účtů, jaké úrovně přístupů využíváte např. RBAC, ACL, apod.

Pro řízení uživatelských účtů a zabezpečení přístupu využíváme model RBAC. Tímto způsobem rozlišujeme dvě základní role: "Administrátor" a "Uživatel". Administrátoři mají přístup k jedné určité části aplikace, kde mají širší oprávnění pro správu dat a administrativní úkoly. Uživatelé mají přístup k jiné části aplikace, kde mohou prohlížet produkty, nakupovat a spravovat svůj profil.

### Formuláře

Popište použité formuláře.

* formulář pro registrace adminů a uživatelů
* formulář pro vyhledávání produktů
* formulář pro dodávání produktů do databáze
* formulář pro editování produktu v databáze
* formulář pro odstranění produktu z databáze
* formulář pro dodávání skladů do databáze

### Orientace ve zdrojovém kódu

Detailně popište základní orientaci ve zdrojovém kódu, použité třídy, metody, apod. потом

* CustomerRegistration - třída pro registraci uživatele.
* CustomerLogin - třída pro přihlášení uživatele.
* UpdateAddress - třída pro aktualizaci adresy.
* AddProductToOrder - třída pro přidání produktu do objednávky.
* UpdateQuantity - třída pro aktualizaci množství produktu v košíku.
* RemoveItemFromOrder - třída pro odstranění produktu z košíku.
* AdminRegistration - třída pro registraci administrátora.
* AdminLogin - třída pro přihlášení administrátora.
* AddStorage - třída pro přidání skladu.
* AddProduct - třída pro přidání produktu.
* AddProductToStorage - třída pro přidání produktu na sklad.
* Purchase - třída pro nákup zboží.
* ChangeProduct - třída pro změnu produktu.
* DeleteProduct - třída pro odstranění produktu z databáze.
* FilterProducts - třída pro filtrování produktů.
* GetId - třída pro získání ID uživatele podle e-mailu.
* GetCustomerData - třída pro získání dat o uživateli podle ID.
* GetOrderData - třída pro data o aktuální objednávce podle ID uživatele.

# Závěr

## Závěr

V závěru pochvalně i kriticky zhodnoťte realizovanou databázovou aplikaci a uveďte, jak by se aplikace mohla v budoucnu využívat.

Dokumentace k našemu projektu poskytuje detailní pohled na celý proces vývoje naší webové aplikace. Projdeme zde základní informace o použitých technologiích, architektuře, struktuře databáze a dalších klíčových aspektech. Aplikace se zaměřuje na provoz e-commerce platformy pro prodej tenisek, kde uživatelé mohou procházet a nakupovat různé produkty.

Frontend aplikace je postaven na technologii React, která umožňuje vytvářet interaktivní uživatelská rozhraní. Backend je realizován pomocí Pythonu a frameworku Flask, což umožňuje efektivní směrování dat a komunikaci s frontendem přes RESTful API. Všechna data jsou ukládána do SQL databáze, která obsahuje 9 tabulek pro ukládání informací o uživatelích, produktech, objednávkách a dalších relevantních datech.

Aplikace podporuje dvě úrovně přístupu: uživatele a administrátory. Pro uživatele je možné procházet produkty, provádět nákupy a spravovat svůj profil. Administrátoři mají rozšířená oprávnění pro správu produktů, skladů a uživatelů.

Využíváme model RBAC pro správu uživatelských rolí a oprávnění, což nám umožňuje rozlišovat mezi administrátory a běžnými uživateli. Tímto způsobem zajišťujeme bezpečný a strukturovaný přístup k aplikaci.

Projekt zahrnuje několik pohledů, které usnadňují získávání informací o produktech, a také funkce pro správu uživatelů, produktů, objednávek a skladů. Dokumentace nám pomáhá pochopit strukturu aplikace, její klíčové komponenty a způsob práce s daty.

V budoucnu může být aplikace dále rozšířena o další funkcionality, například podporu plateb, recenze produktů, rozšířené filtrování nebo integraci s externími službami. Základní architektura a způsob práce s daty však zůstane zachován, což umožní efektivní rozšíření a úpravy aplikace podle nových potřeb.

## Přílohy

### 