

# DSP

## Uni Bank

Viktoriia Sergeeva

20.04.2023

### 1. Úvod

#### 1.1. Účel

Cílem semestrální práce STIN je vytvoření aplikace, která simuluje komunikaci uživatele s bankou a umožňuje provádět platby a zobrazovat historii účtu. Aplikace bude se skládat z backendu a frontendu, které budou vzájemně komunikovat a nacházet se na různých zařízeních. Součástí testování budou vytvořené testovací uživatelé. Oni se budou přihlašovat do aplikace, provádět transakce a ověřovat správnost funkcí systému.

#### 1.2. Rozsah produktu

##### 1.2.1. Funkční požadavky

- Uživatel se může přihlásit do aplikace,
- Uživatel může být ověřen,
- Uživatel může provádět platby,
- Uživatel může zobrazovat historii účtu,
- Uživatel může zobrazit jakoukoliv ze svých měn.

##### 1.2.2. Použití

Aplikace bude použita jako výsledek semestrální práce a jako potvrzení dosazených znalostí z předmětu STIN.

#### 1.3. Definice, zkratky, zkratková slova

- STIN - Softwarové inženýrství,
- DSP - Dokumentace softwarového produktu,
- API - Application Programming Interface,
- REST - REpresentational State Transfer,
- SQL - Structured Query Language,
- SQLite - SQL database engine,
- Node.js - JavaScript runtime environment,
- Svelte - JavaScript framework,
- Fetch API - JavaScript API,
- HTML - HyperText Markup Language,
- CSS - Cascading Style Sheets,
- JavaScript - programovací jazyk,
- Currency - kód měny dle ISO 4217 (EUR, USD, CZK atd.)

#### 1.4. Odkazy

- [Repozitář](#) projektu
- [Zdroj](#) měnového kurzu eura

### 2. Celkový popis

**Název produktu:** Uni Bank

**Jazyk aplikace:** angličtina

**Předpokládané datum dokončení:** 18.5.2023

#### 2.1. Perspektiva produktu

Aplikace bude vytvořena jako webová aplikace, která bude komunikovat s backendem a databází. Backend bude vytvořen pomocí Node.js a bude komunikovat s databází pomocí knihovny SQLite. Databáze bude vytvořena pomocí jazyka SQL a bude komunikovat s backendem pomocí knihovny SQLite. Frontend bude vytvořen pomocí jazyka JavaScript a moderního frameworku Svelte a bude komunikovat s backendem pomocí Fetch API. Backend bude komunikovat s frontendem pomocí REST API.

## 2.2. Funkce produktu

- Přihlášení

Uživatel musí zadat přihlašovací údaje. Pokud v systému bude nalezen uživatel s odpovídajícími údaji: - bude přesměrován na stránku s ověřením, - odešle se mu email s ověřovacím kódem. Pokud v systému nebude nalezen uživatel s odpovídajícími údaji: - uživateli se zobrazí chybová hláška `user was not found`.

- Ověření

Uživatel musí zadat ověřovací kód. Pokud bude ověřovací kód správný: - bude přesměrován na stránku s účty. Pokud bude ověřovací kód nesprávný: - uživateli se zobrazí chybová hláška `wrong code`.

- Platba

**Popis chování:** Po potvrzení platby uživatelem proběhně několik kontrol:

- Má-li uživatel účet v měně platby?
- Má-li uživatel dostatek peněz na účtu?
  - Pokud uživatel nemá účet v měně platby, bude provedena kontrola dostatku peněz na výchozí měně  
**Výchozí měna:** pro všechny uživatele je nastavena na `CZK`.
- Na hlavní stránce s účty je umístěno dva polí a tlačítko `pay`,
- Do prvního pole se uživatel napíše částku,
- V druhém poli si uživatel vybere měnu platby,
- Po zadání všech údajů platby se umožní stisknutí tlačítka `pay`,
- Po kliknutí na tlačítko `pay`:
  - V případě úspěšné transakcí:
    - bude přesměrován na stránku s účty,
    - budou upravena data v databázi.
  - V případě neúspěšné transakce se uživateli zobrazí chybová hláška `transaction was not successful`.

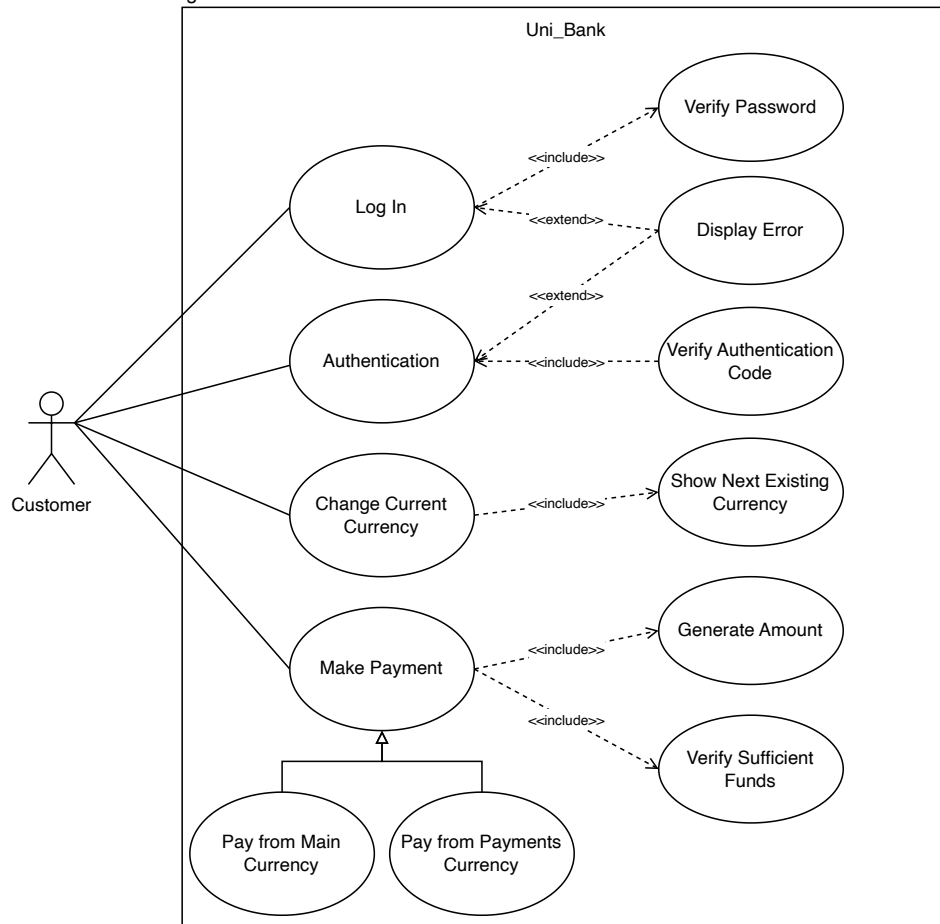
- Historie měnového kurzu

Na hlavní stránce bude zobrazována historie transakcí s aktuálně zvolenou měnou.

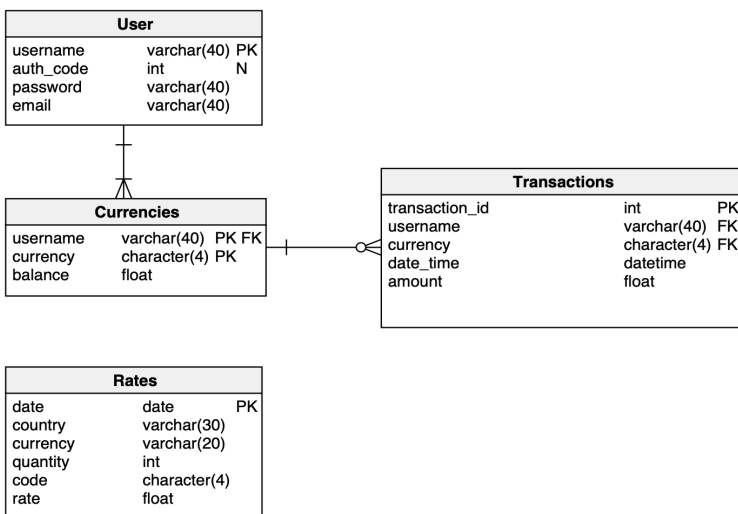
- Zobrazení

- Na hlavní stránce s účty je umístěno tlačítko `change currency`,
- Po kliknutí na tlačítko `change currency` se změní aktuální měna na následující uživatelskou měnu,
- Uživatel může zobrazit jakoukoliv ze svých měn.

### 2.2.1. Use Case diagram



### 2.2.1. SQLite schéma



### 2.3. Charakteristiky uživatele

- Uživatel bude mít přístup do aplikace pouze po přihlášení a ověření,
- Uživatel bude mít přístup pouze k svým účtům,
- Uživatel bude mít přístup pouze k svým transakcím,
- Uživatel bude mít přístup pouze k svým měnám.

### 3. Systémové požadavky

- Klient / Server
  - Každá část aplikace běží samostatně.
  - Frontend klient běží v moderním webovém prohlížeči a je napsán v JavaScriptu s využitím frameworku Svelte.
  - Backend server je napsán v jazyce JavaScript.
- API ČNB
  - Aplikace očekává, že data z API ČNB (<https://www.cnb.cz/>) budou ve výměnném formátu CSV s oddělovačem | a s atributy:
    - První řádek je den, ke kterým všechny hodnoty náleží
    - Druhý řádek jsou atributy
      - země => plný název země měny
      - měna => plný název měny
      - množství => počet kusů měny, ke kterým se vztahuje cena
      - kód => kód měny dle ISO 4217
      - kurz => kurz měny s desetinnou čárkou , a 2 desetinnými místy
  - Pokud nebude možné z API data získat, použijou se poslední dostupná data, tj. data naposledy uložená v databázi
- Měny
  - Data se se stahují jednou za den, automaticky na serveru ve 14:35

### 4. Odhad časové náročnosti

Odpověď ze serveru by měla trvat 8-10 ms.