



P2P bankovní systém

Školní projekt v rámci předmětu Programové vybavení

Autoři:

Martin Chmelík C4c, chmelikmartin123@gmail.com

Sofia Hennelová C4c, sofia.hennelova@gmail.com

Datum: 24.1. 2026

Obsah

1. Úvod.....	3
2. Analýza.....	3
2.1 Popis uživatelů.....	3
2.2 Funkční požadavky.....	3
2.3 Nefunkční požadavky.....	4
3. Návrh (Design).....	4
3.1 Přehled architektury.....	4
3.2 Vrstvy a zodpovědnosti.....	4
3.2.1 Network Layer - TCP Server.....	4
3.2.2 Command / Application Layer.....	4
3.2.3 P2P / Proxy Layer (ESSENTIALS).....	5
3.2.4 Data Layer - databáze a perzistence.....	5
4. Databáze.....	6
4.1 Diagram.....	6
5. Protokol a příkazy.....	6
5.1 Obecná pravidla.....	6
5.2 Přehled příkazů.....	6
BC - Bank code.....	6
AC - Account create.....	7
AD - Account deposit.....	7
AW - Account withdrawal.....	7
AB - Account balance.....	7
AR - Account remove.....	7
BA - Bank (total) amount.....	7
BN - Bank number (of clients).....	7
6. Běh aplikace.....	7
7. Konfigurace programu.....	7
8. Instalace a spuštění.....	8
9. Chybové stavy.....	8
10. Knihovny.....	8
11. Licence.....	8
12. Závěr.....	9

1. Úvod

Cílem této práce bylo vytvořit síťovou aplikaci podle architektury peer-to-peer (P2P), kde každý spuštěný uzel představuje jednu „banku“ v rámci školní sítě. Aplikace umožňuje spravovat bankovní účty a jejich zůstatky. Součástí řešení je také možnost získat základní statistiky banky.

Ovládání aplikace probíhá přes TCP/IP pomocí přesně definovaných textových příkazů v kódování UTF-8, které lze posílat ručně například přes PuTTY nebo telnet. Aby systém fungoval jako mezibankovní prostředí v rámci třídy, uzel při práci s účtem používá jako identifikátor banky IP adresu počítače a u vybraných příkazů dokáže požadavky směřující na jinou banku přeposlat (proxy) na cílový uzel a vrátit jeho odpověď zpět klientovi.

2. Analýza

2.1 Popis uživatelů

- Klient (uživatel přes PuTTY/telnet): Posílá příkazy na bankovní uzel a čte odpovědi. Může zakládat účty, měnit zůstatky, zjišťovat stav a statistiky banky.
- Správce: Spouští aplikaci, nastavuje konfiguraci (port, timeouty, DB), sleduje monitor, řeší provozní problémy a kontroluje logy.

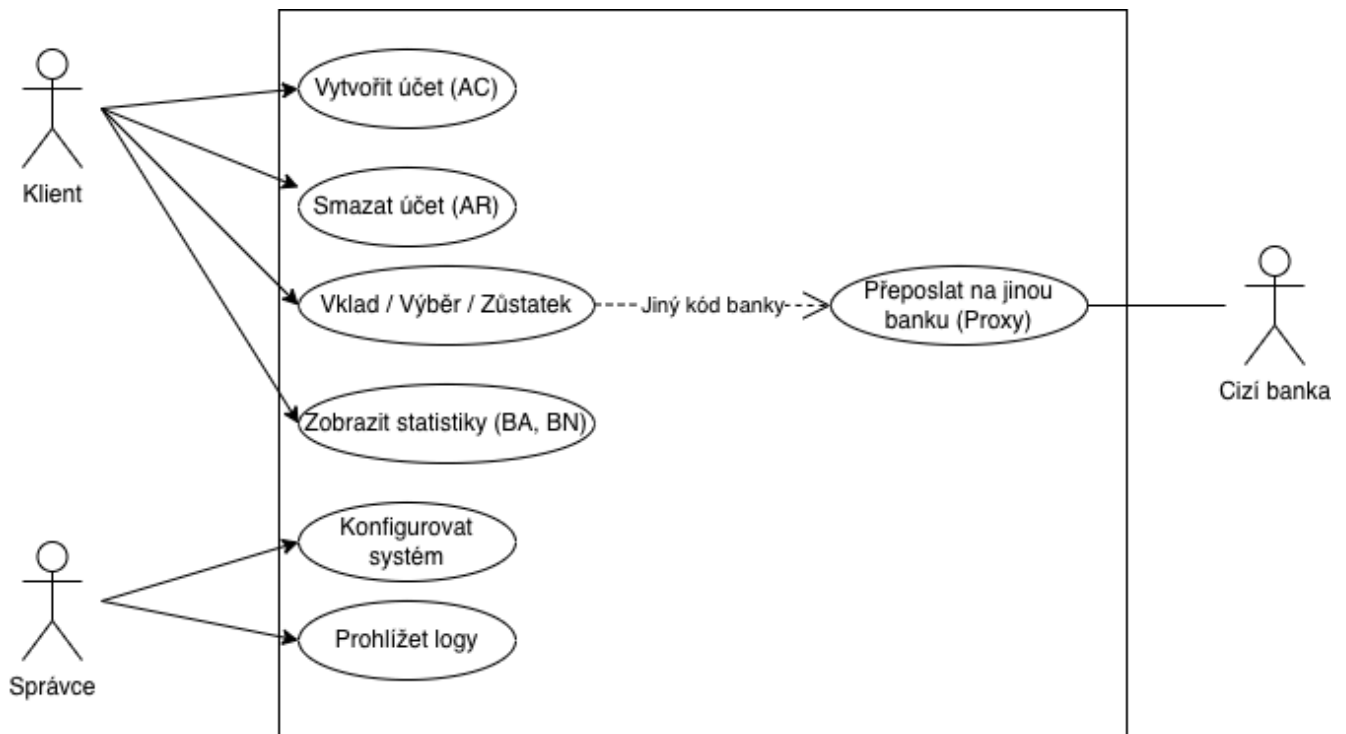
2.2 Funkční požadavky

- FR1 - Bank code
 - Systém musí odpovídat na příkaz BC a vracet svou IP adresu jako kód banky.
- FR2 - Správa účtů (vytvoření / smazání)
 - Systém musí umožnit vytvoření účtu příkazem AC a vrátit AC <account>/<ip>, kde <account> je v rozsahu 10000-99999 a je unikátní v rámci banky.
 - Systém musí umožnit smazání účtu příkazem AR <account>/<ip>, ale pouze pokud je zůstatek 0.
- FR3 - Vklady a výběry
 - Systém musí umožnit vložení prostředků příkazem AD <account>/<ip> <number>.
 - Systém musí umožnit výběr prostředků příkazem AW <account>/<ip> <number>, pouze pokud je na účtu dostatek prostředků.
- FR4 - Zůstatek na účtu
 - Systém musí vracet aktuální zůstatek příkazem AB <account>/<ip> ve formátu AB <number>.
- FR5 - Statistika banky
 - Systém musí vracet celkový součet všech prostředků v bance příkazem BA.
 - Systém musí vracet počet účtů/klientů v bance příkazem BN.
- FR6 - Proxy (ESSENTIALS)
 - Pokud příkazy AD, AW nebo AB obsahují jiný bank code (IP) než má lokální uzel, systém se musí připojit na cílovou banku přes TCP, předat jí původní příkaz a vrátit její odpověď zpět klientovi.
- FR7 - Logování
 - Systém musí logovat příchozí příkazy, odeslané odpovědi a chyby tak, aby bylo možné zpětně dohledat průběh komunikace a operací.
- FR8 - Persistence dat
 - Systém musí ukládat účty a zůstatky do persistentního úložiště (DB), aby se data neztratila po restartu aplikace.

2.3 Nefunkční požadavky

- NFR1 - Konfigurace
 - Aplikace musí být konfigurovatelná externím souborem.
- NFR2 - Validace vstupů
 - Aplikace musí validovat veškeré vstupy
- NFR3 - Síť a protokol
 - Komunikace probíhá přes TCP/IP v textovém formátu UTF-8.
 - Každý příkaz musí vrátit odpověď (úspěch nebo ER).
- NFR4 - Timeouty
 - Aplikace musí mít nastavitelný timeout pro zpracování požadavků (default 5 s) a timeout pro klientské spojení.
- NFR5 - Paralelní obsluha
 - Server musí zvládnout více klientů současně.
- NFR6 - Spustitelnost ve škole
 - Aplikace musí být spustitelná na školním PC bez použití IDE.

Use case diagram



3. Návrh (Design)

3.1 Přehled architektury

Aplikace je navržena jako jeden bankovní uzel v P2P síti. Běží jako TCP server, přijímá textové příkazy a vrací odpovědi podle daného protokolu. Vnitřně je řešení rozdělené do několika částí, aby se oddělila síťová komunikace, zpracování příkazů a práce s databází.

3.2 Vrstvy a zodpovědnosti

3.2.1 Network Layer - TCP Server

- Naslouchá na listen_host:listen_port (port 65525-65535).
- Přijímá více klientů současně (paralelní obsluha).
- Čte data po řádcích (\n), dekóduje UTF-8.
- Na každý příkaz pošle přesně jednu odpověď.
- Řeší client timeout (neaktivní klient se odpojí).
- Předává text příkazu do aplikační vrstvy.

3.2.2 Command / Application Layer

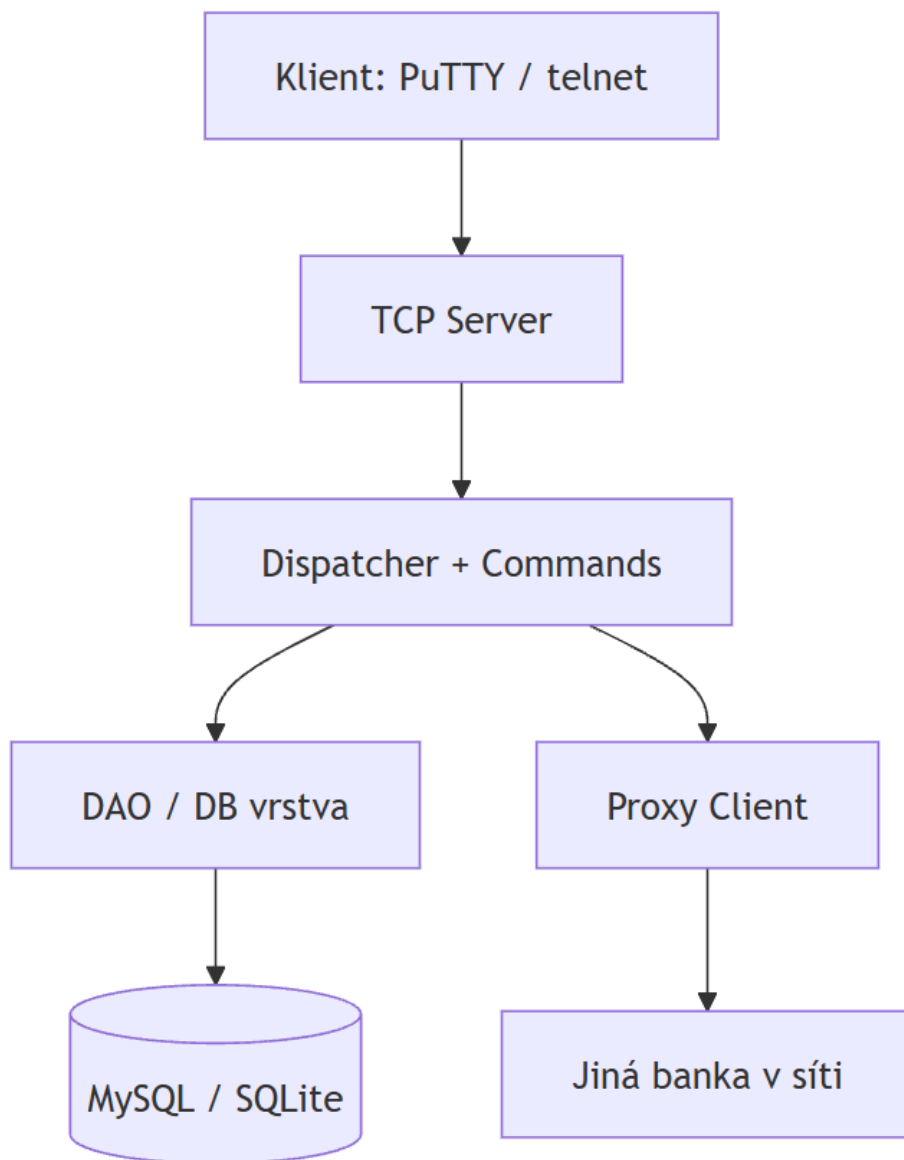
- Pro každý příkaz existuje samostatná implementace (styl „Command pattern“).
- Dispatcher mapuje kód příkazu (BC, AC, ...) na konkrétní handler.
- Společná část je:
 - validace formátu příkazu (počet argumentů, account/ip, amount),
 - jednotné formátování odpovědí (XX ... nebo ER ...),
 - jednotné zachytávání chyb.

3.2.3 P2P / Proxy Layer (ESSENTIALS)

- Pro příkazy AD, AW, AB platí: pokud <ip> v account/ip není lokální IP uzlu:
 - aplikace se připojí na cílovou IP na remote_port,
 - pošle původní příkaz,
 - přečte první řádek odpovědi a vrátí ho klientovi.
- Proxy používá command timeout (default 5 s).

3.2.4 Data Layer - databáze a perzistence

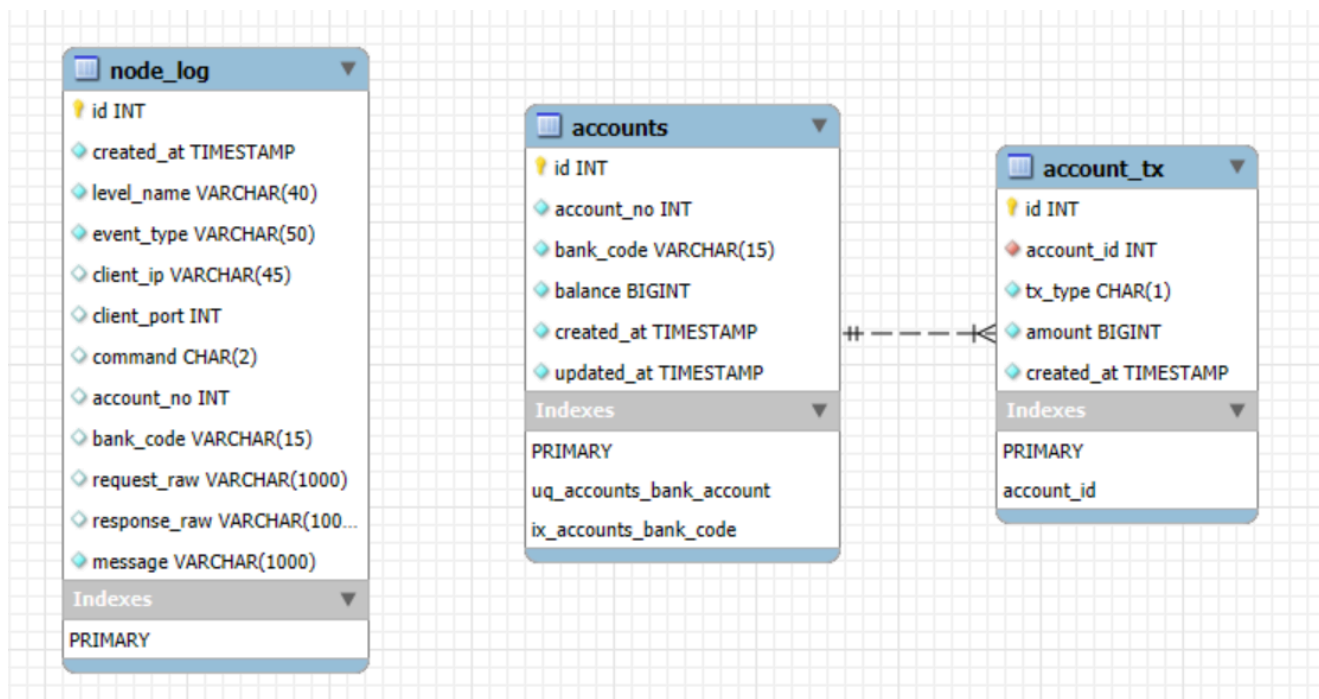
- Účty a zůstatky jsou uloženy v DB, aby přežily restart.
- DB strategie:
 - primárně MySQL (pokud je dostupná a správně nastavená),
 - fallback SQLite.
- Datová vrstva je oddělená přes DAO.



4. Databáze

- accounts - účty
- account_tx - historie vkladů/výběrů
- node_log - log provozu (příkazy, odpovědi, chyby)

4.1 Diagram



5. Protokol a příkazy

5.1 Obecná pravidla

- Komunikace je textová, UTF-8.
- Klient posílá příkaz jako jeden řádek ukončený \n.
- Server vrací právě jednu odpověď také jako jeden řádek ukončený \n.
- Příkazový kód je case-insensitive (server ho převádí na upper()).

Chyby se vrací ve formátu:

- ER <popis_chyby>

5.2 Přehled příkazů

BC - Bank code

- Vrací kód banky (aktuální IP adresa uzlu).
- Požadavek: BC
- Odpověď: BC <ip>

AC - Account create

- Vytvoří nový účet v lokální bance. Účetní číslo je generováno v rozsahu 10000-99999 a musí být unikátní v rámci banky.
- Požadavek: AC
- Odpověď: AC <account>/<ip>

AD - Account deposit

- Vloží částku na účet. Pokud <ip> neodpovídá lokální bance, příkaz je proxován na vzdálený uzel.
- Požadavek: AD <account>/<ip> <amount>
- Odpověď (úspěch): AD

AW - Account withdrawal

- Vybere částku z účtu. Pokud není dostatek prostředků, vrátí chybu. Pokud <ip> neodpovídá lokální bance, příkaz je proxován.
- Požadavek: AW <account>/<ip> <amount>
- Odpověď (úspěch): AW

AB - Account balance

- Vrací aktuální zůstatek účtu. Pokud <ip> neodpovídá lokální bance, příkaz je proxován.
- Požadavek: AB <account>/<ip>
- Odpověď: AB <number>

AR - Account remove

- Smaže účet, pouze pokud je zůstatek 0.
- Požadavek: AR <account>/<ip>
- Odpověď (úspěch): AR

BA - Bank (total) amount

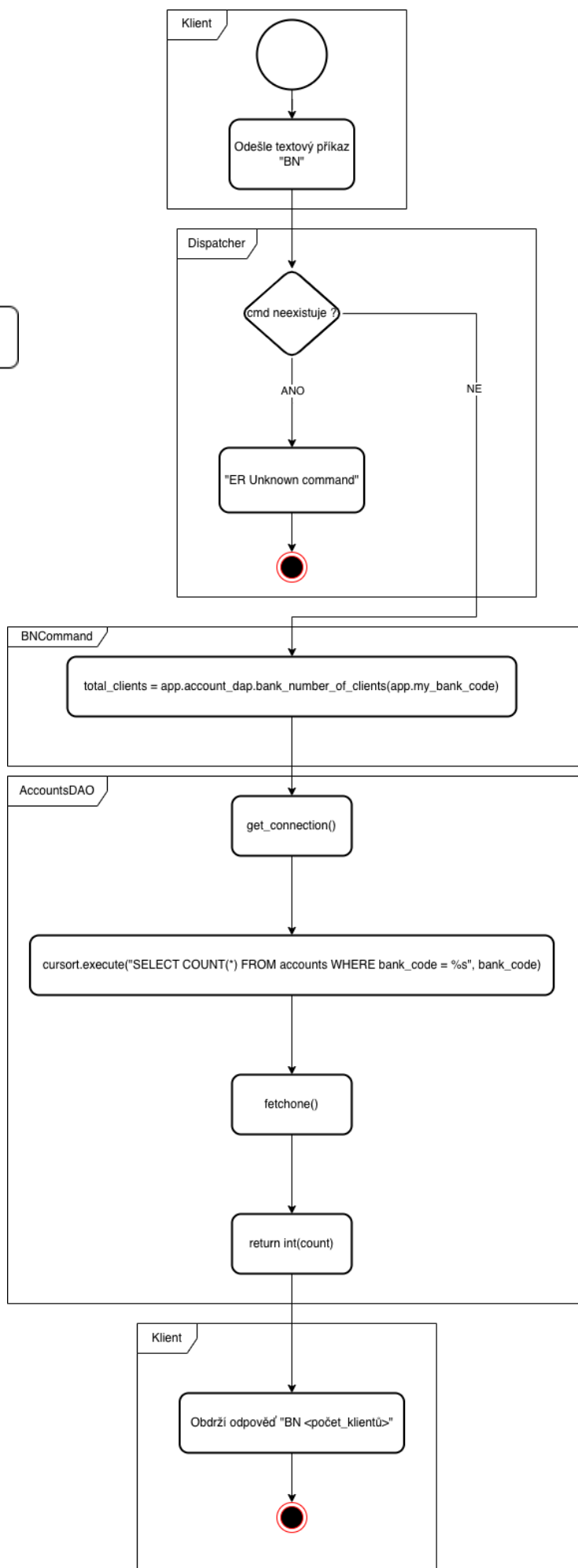
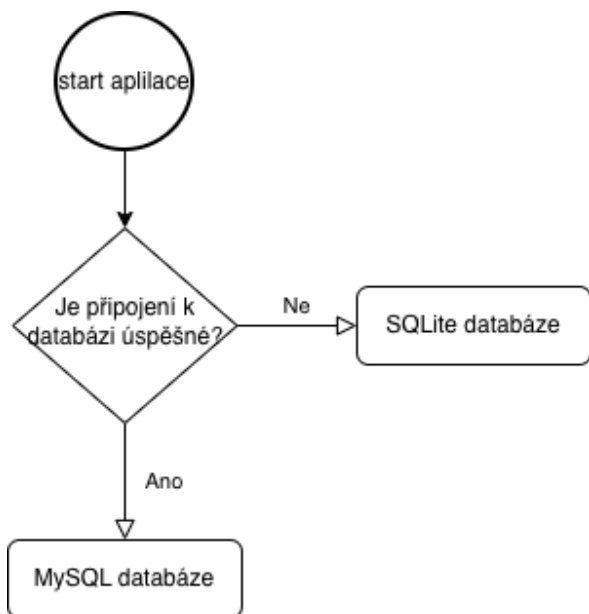
- Vrací součet všech prostředků ve všech účtech lokální banky.
- Požadavek: BA
- Odpověď: BA <number>

BN - Bank number (of clients)

- Vrací počet účtů v lokální bance.
- Požadavek: BN
- Odpověď: BN <number>

6. Běh aplikace

Spuštění aplikace



7. Konfigurace programu

Informace o konfiguraci naleznete v README.md.

8. Instalace a spuštění

Informace o instalaci a spuštění naleznete v README.md.

9. Chybové stavy

- ER Invalid Command format.
- ER Not enough funds.
- ER Account not empty.

10. Knihovny

- MySQL konektor: mysql-connector-python
- SQLite: sqlite3
- Interní knihovna my_lib/my_library:
 - načítání konfigurace
 - detekce IP
 - základ pro command pattern

11. Licence

MIT License

Copyright (c) 2026 Martin Chmelík, Sofia Hennelová

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

12. Závěr

Projekt implementuje jednoduchý bankovní uzel fungující v P2P prostředí školní sítě. Nabízí základní operace nad účty, podporuje přeposílání požadavků na jiné banky (proxy), loguje provoz do databáze a udržuje perzistentní stav účtů. Díky textovému protokolu je aplikace snadno testovatelná ručně přes telnet/PuTTY a je vhodná pro demonstraci síťové komunikace, paralelní obsluhy klientů a práce s perzistentním úložištěm.