Hacker Bank Node

Maksym Solonitsyn, Vilma Tomanová

C4a, Informační technologie

(solonitsyn@spsejecna.cz, tomanova@spsejecna.cz)

Date completed: 28.01 2026

Type of project: School

Github: https://github.com/notvivi/HackerBankNode

Střední průmyslová škola elektrotechnická

Praha 2, Ječná 30

2026

Contents

1. Introduction

- This project is P2P (peer-to-peer) node for banks, implemented using TCP/IP with extension for hacking and robbing other banks
- For the purpose of expanding knowledge and learning to work in pairs

---

2. Key features

- Creating and managing bank accounts
- Deposits and withdrawals in USD$
- Persistent storage (Database in SQLite)
- TCP server with configurable log file, port, timeout, ip network and mask
- Robber plan command for hacking/robbing other banks
- CustomTkinter UI for node monitoring
- Logging

---

3. System Overview

1.1 High-Level Architecture

- P2P bank node concept
- Client–Server interaction
- TCP/IP communication model
- Node discovery in IP network

1.2 Technology Stack

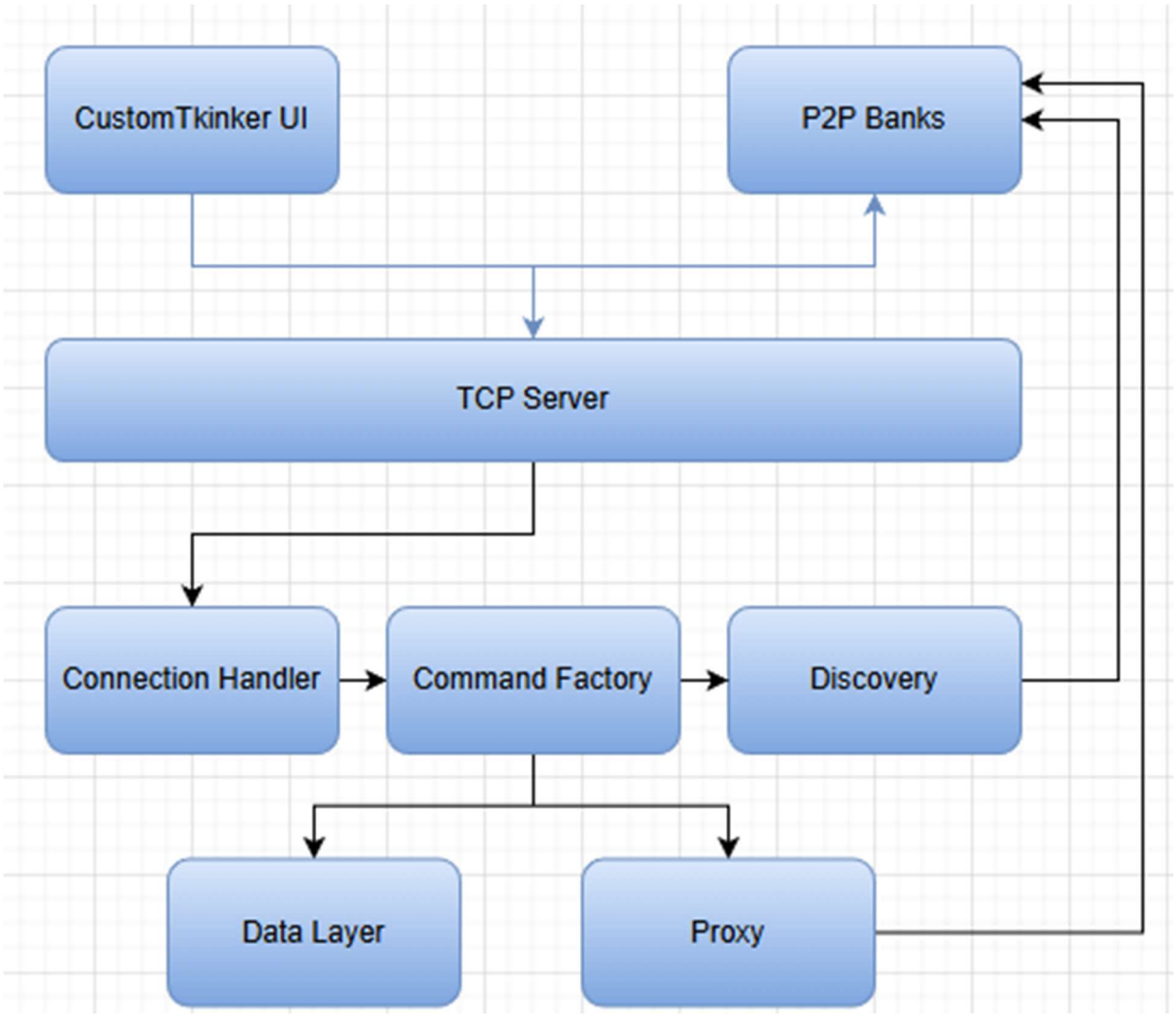
- Python 3.9+
- CustomTkinter
- SQLAlchemy
- SQLite
- TCP sockets

---

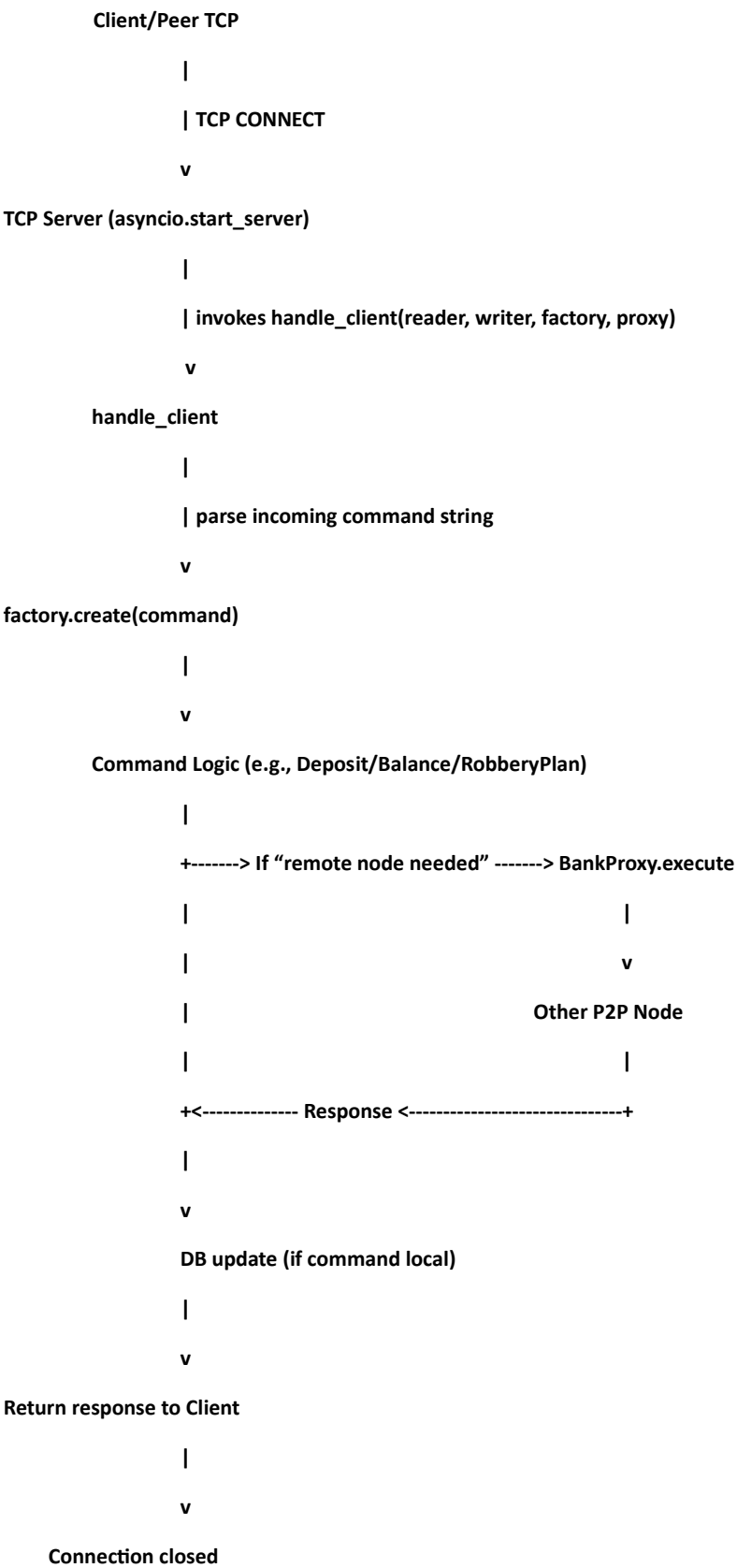
4. Requirements

- Python 3.9+
- Customtkinter 5.2.2
- Sqlalchemy 2.0.25
- Black 26.1.0
- Mypy 1.8.0
- Ruff 0.2.1
- Pytest 8.0.0

5. System diagaram



6. Workflow diagram



7. Configuration

6.1 Configuration File (config.json)

```
{
  "log_file": "file-path.txt",
  "port": port-number,
  "timeout": timeout-number,
  "ip_network": "ip-network",
  "ip_mask": ip-mask,
  "database": {
    "sqlite_path": "db-connection"
  }
}
```

Explanation:

- log_file - path to the log file where all application events are stored (string).
- port - TCP port on which the bank node listens for incoming connections (integer).
- timeout - global timeout in seconds (integer).
- ip_network - base IP address of the local network used for P2P node discovery and communication (string).
- ip_mask - network mask defining the size of the IP network used by the P2P system (integer).
- database.sqlite_path - path to the SQLite database file used for persistent storage (string).

All configuration values are loaded at application startup. Invalid or missing parameters result in application startup failure.

6.2 Runtime Behavior

- Configuration is modified via UI -> then it is saved and used in server

---

8. Networking and Communication

7.1 TCP Server

- The bank node runs a TCP server that listens for incoming client connections and processes banking commands

1. The socket is bound to the configured IP address and port.
2. The server enters a listening state and waits for incoming connections.
3. When a client connects, a new socket is created for the session.
4. The server receives a command, processes it, sends a response, and closes the connection.

Each client request is handled independently, ensuring that failures in one connection do not affect others.

P2P Communication

- The system operates as a peer-to-peer network where each bank node can act both as a server and a client.
- Routing proxy that forwards incoming requests to vendor nodes using the bank address from the request payload
    1. The server extracts the bank IP from the command payload.
    2. If the IP matches the local node, the command is executed locally.
    3. If the IP belongs to a different node:
    4. the request is forwarded to the target bank node
    5. the local node acts as a routing proxy
    6. The response from the remote node is relayed back to the original client.

---

9. Communication Protocol – supported commands

Name	Code	Call	Success Response	Error Response
Bank code	BC	BC	BC <ip>	ER <message>
Account create	AC	AC	AC<account>/<ip>	ER <message>
Account deposit	AD	AD <account>/<ip> <number>	AD	ER <message>
Account withdrawal	AW	AW <account>/<ip> <number>	AW	ER <message>
Account balance	AB	AB <account>/<ip>	AB <number>	ER <message>
Account remove	AR	AR <account>/<ip>	AR	ER <message>
Bank total amount	BA	BA	BA <number>	ER <message>
Bank number of clients	BN	BN	BN <number>	ER <message>

Explanation:

- <ip>, IP address in the format 0.0.0.0 - 255.255.255.255, which is used as the bank code (used as unique identifier for each bank node).
- <account> A positive integer in the range 10000 to 99999, which is used as the bank account number within a bank.
- <number> A non-negative integer in the range 0 to 9223372036854775807

Robbery Plan Command (RP) – command extension

Name	Code	Call	Success Response	Error Response
Robbery plan (local)	**RP**	**RP <number>**	**RP <message>**	**ER <message>**

- RP 1000000
- Response: RP To reach 1000000$, banks 10.1.2.3 and 10.1.2.85 must be robbed, affecting only 21 clients.

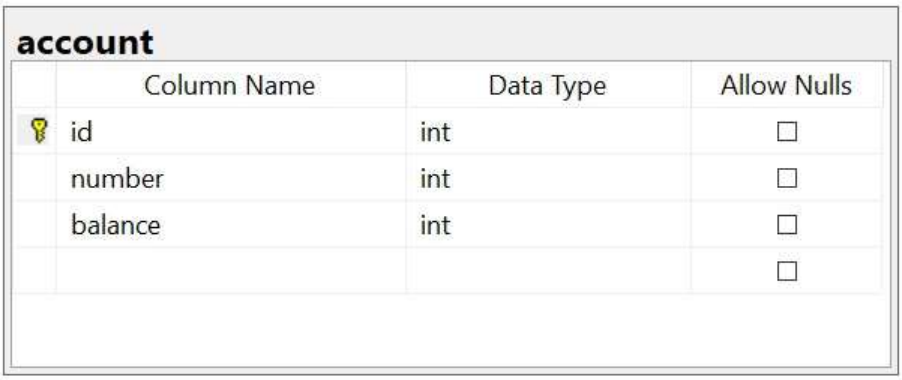
---

10. Database Design

9.1 Database Technology

- SQLite
- SQLAlchemy ORM
- Used for storing bank accounts persistently
- For each connection its own database transaction

9.2 Entity-Relationship Diagram



11. Bussiness Logic

10.1 Account Management

- Creation
- Removal
- Balance handling

10.2 Money Operations

- Deposits
- Withdrawals
- Edge cases (overflow, insufficient funds)

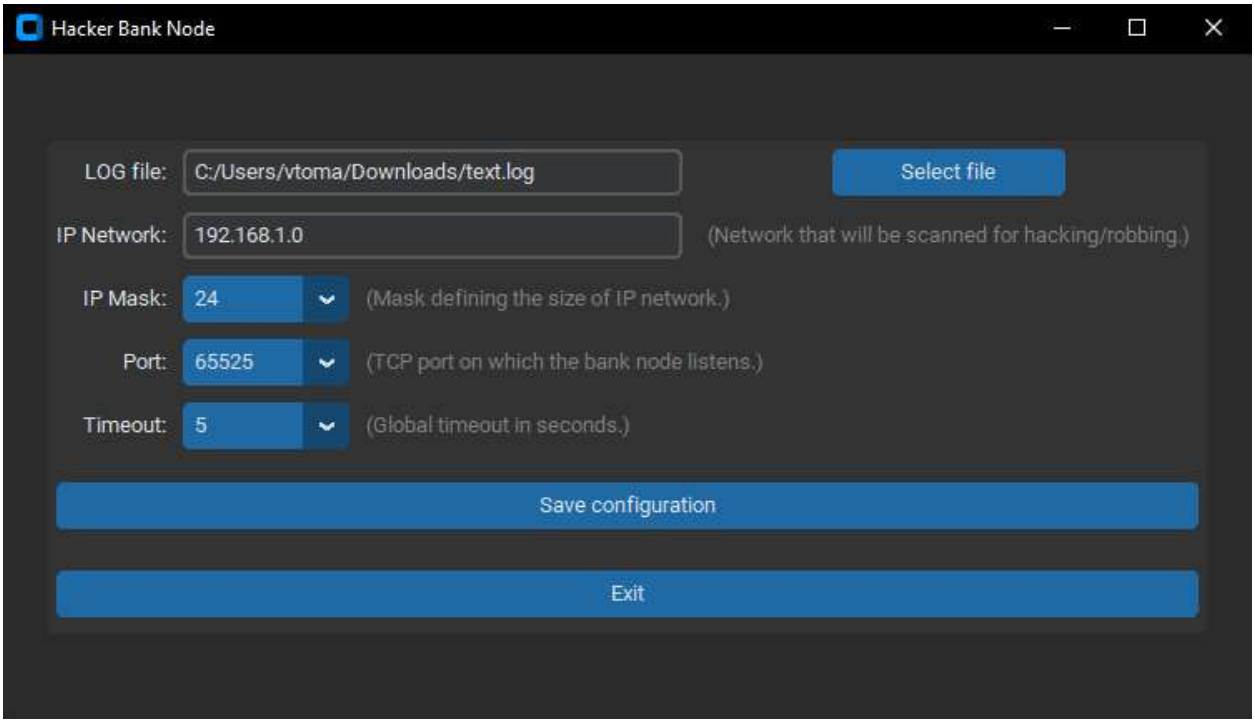
---

12. User Interface (CustomTkinter)

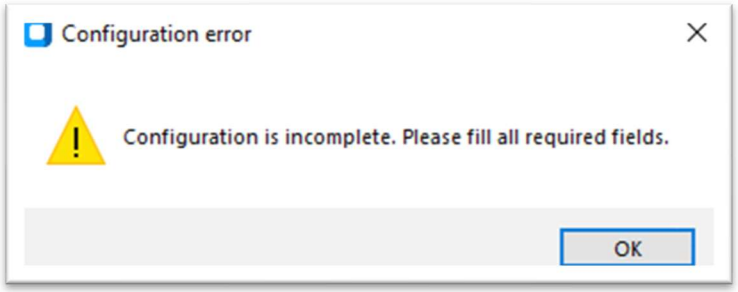
11.1 UI Overview

- The user interface is used for **monitoring the entire bank node**.

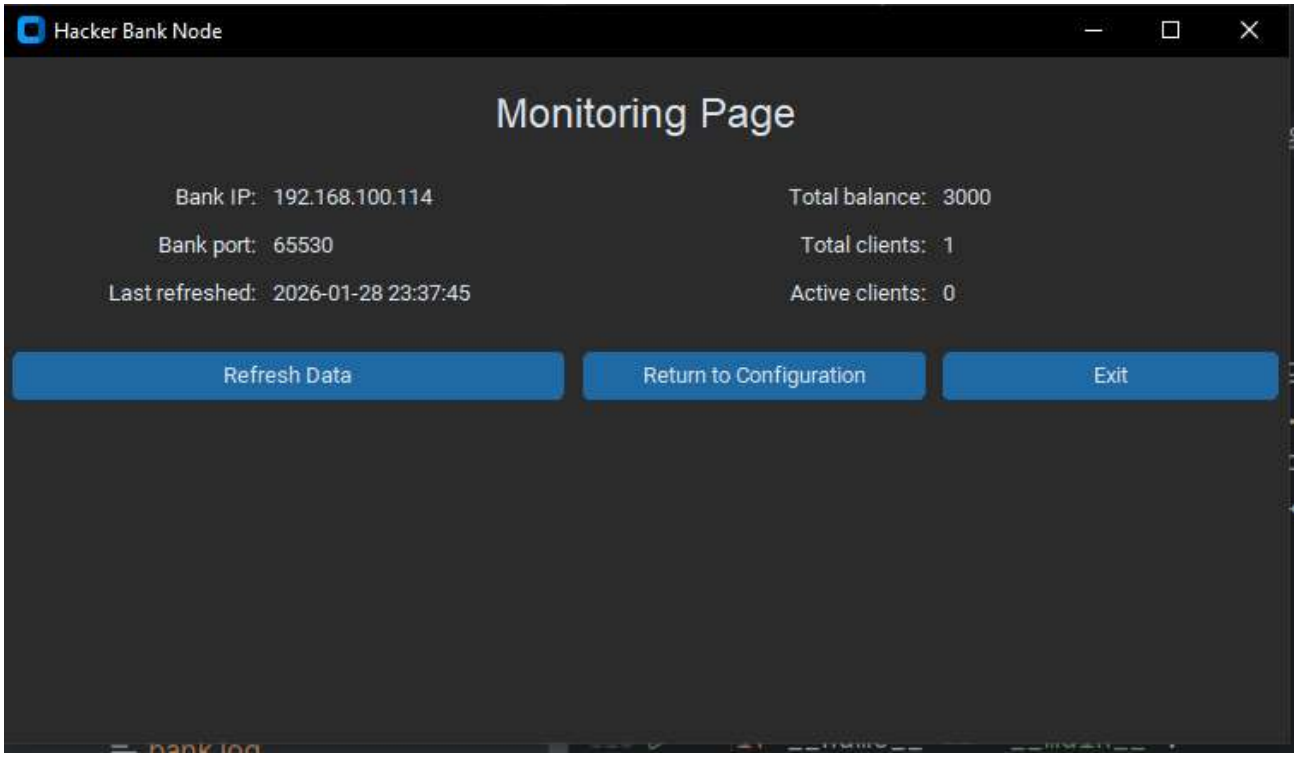
11.2 Photos of UI



1. **First page with configuration**



2. **UI error handling**



3. **Monitoring page with all data**

---

13. Logging

Severity

Level	Description
Debug	Informational events that are most useful to debug an application
Info	Informational messages that highlight the progress of the app
Error	Error events that might still allow the app to continue running

14. Testing

13.1 Testing Tools

- Pytest
- Mypy
- Ruff
- black

13.2 Test Coverage

- 1 Unit test

---

15. Deployment & Usage

- Deployment and usage can be found in readme
  https://github.com/notvivi/HackerBankNode/blob/main/README.md

---

16. Future Work

- The server accepts both **uppercase and lowercase** commands.
- The system is available in **English, Czech, and Ukrainian**.

---

17. Contribution Guide

- You can find guide here:
  https://github.com/notvivi/HackerBankNode/blob/main/CONTRIBUTING.md

---

18. License

- License is specified in the github
- Using MIT license
  https://github.com/notvivi/HackerBankNode/blob/main/LICENSE