**Federal State Autonomous Educational**

**Institution National Research University**

**"Higher School of Economics"**

Moscow Institute of Electronics and Mathematics named after A.N. Tikhonov (MIEM HSE)

Department of Computer Engineering

Course: Algorithms and Programming

**TECHNICAL DOCUMENTATION**

for the project

**Bank Transfer Optimizer (MVP Version)**

Developer: Poptsova Dasha BIV243

Documentation Language: English

Moscow 2025

**Оглавление**

[**1. General Information** 3](#_Toc194264010)

[**1.1 Purpose of the Product** 3](#_Toc194264011)

[**1.2 Scope of Application** 3](#_Toc194264012)

[**1.3 Development Justification** 3](#_Toc194264013)

[**2. Developer Information** 3](#_Toc194264014)

[**3. Technical Description** 3](#_Toc194264015)

[**3.1 System Architecture** 3](#_Toc194264016)

[**3.2 Implementation Details** 4](#_Toc194264017)

[**3.3 Technologies Used** 4](#_Toc194264018)

[**4. Algorithm Logic** 4](#_Toc194264019)

[**4.1 Commission Calculation Algorithm** 4](#_Toc194264020)

[**4.2 Example Calculation** 4](#_Toc194264021)

[**5. Safety Considerations** 5](#_Toc194264022)

[**6. Standards Compliance** 5](#_Toc194264023)

[**7. Appendices** 5](#_Toc194264024)

[**8. Development Recommendations** 5](#_Toc194264025)

**1. General Information**

**1.1 Purpose of the Product**

Bank Transfer Optimizer is a console application designed to calculate the most efficient route for interbank transfers considering internal and external commissions, international borders, and correspondent relationships between banks.

**1.2 Scope of Application**

This software product can be used by:

- Financial systems with interbank settlements

- Internal banking tools for minimizing transfer costs

- Financial analysis and simulation platforms

**1.3 Development Justification**

The development was initiated as an MVP prototype of a commission optimization system aimed at validating the economic model and forming the architectural basis of the future product.

**2. Developer Information**

Organization: Investing Company IT Invest

**3. Technical Description**

**3.1 System Architecture**

The system is built using a modular console architecture and includes the following components:

- main.cpp — Main execution module

- TransferProblem — Loads and stores data: banks, commissions, borders

- TransferSolver — Calculates optimal transfer route

- Tier — Commission threshold model

- PiecewiseCommission — Tier-based fee logic

- Bank — Bank description structure

- BorderCommission — Fixed international commission structure

**3.2 Implementation Details**

Language: C++17

Data Format: CSV (UTF-8, without BOM)

External Libraries: None

Interaction: Command-line interface (CLI)

Input/Output: std::cin / std::cout

Dependencies: C++ STL only

**3.3 Technologies Used**

- C++ Standard Library (vector, map, string, fstream, sstream)

- Linux/Windows terminal

- g++ or clang++ compiler

**4. Algorithm Logic**

**4.1 Commission Calculation Algorithm**

1. Load list of banks, commissions, and borders from CSV.

2. User provides:

- Source bank

- Destination bank

- Transfer amount

3. Evaluate routes:

- Direct (with correspondent match)

- Indirect via one intermediary bank

4. Calculate total commission using tiered logic.

5. Attempt split from 1 to 10 equal parts.

6. Select route with minimum total commission.

**4.2 Example Calculation**

Amount: 3000 USD

Route: BankA → BankB

Output commission: 36

Border commission: 20

Input commission: 60

Total commission = 116

**5. Safety Considerations**

- The application does not require network connectivity.

- All data is processed locally from CSV files.

- The tool does not interact with real financial systems.

**6. Standards Compliance**

The project follows documentation and software standards as follows:

- ESKD – documentation structure and detailed description

- ESPD – algorithmic and logic structure

- ESTD / ESTPP – technical architecture and scalability framework

**7. Appendices**

- Banks.csv — Bank, country, correspondents

- Commissions.csv — Commission tiers

- Borders.csv — Fixed international transfer fees

- main.cpp — Source code

**8. Development Recommendations**

- Add support for non-equal split strategies

- Allow multi-node transfer chains

- Implement commission and route visualizations

- Build web interface and REST API

- Add unit testing

- Add language localization