1.1 Purpose

The purpose of this project is to design and implement a basic currency exchange system. The system allows clients to convert money between currencies, while enabling cashiers to manage transactions, track reserves, and generate reports. Management can monitor performance through summaries and apply bonuses based on profit.

1.2 Scope

The Currency Exchange system simulates the operations of a currency exchange office. The system will:

- Convert money between supported currencies.
- Verify reserve availability before approving an exchange.
- Generate receipts for clients.
- Handle partial and denomination-based exchanges.
- Provide daily reports on balances and profit.
- Notify when reserves fall below a threshold.
- Calculate cashier bonuses based on profit.

The system will be implemented as a C++ console-based application and designed to run on standard compilers.

---

2. User Stories

1. As a client, I want to exchange USD to EUR.
2. As a cashier, I want to check if enough EUR is available before confirming the transaction.
3. As a client, I want a printed receipt showing the exchange details.
4. As a cashier, I want to perform partial exchanges (e.g., 50 EUR and the rest in local currency).
5. As management, I want to receive a daily report of currency balances and profit.
6. As a cashier, I want to be notified when a currency reserve is low.
7. As management, I want to award a 5% bonus to the cashier based on daily profit.

---

3. Functional Requirements

- The system must allow currency conversion between supported currencies.
- The system must verify reserve availability before exchange.
- The system must generate a receipt after each transaction.
- The system must support partial and denomination-based exchanges.
- The system must generate a daily report with balances and profit.
- The system must alert when reserves fall below a critical minimum.
- The system must calculate and apply cashier bonuses.

---

4. Non-Functional Requirements

- The system should respond to exchange requests within 1 second.
- The system should store transaction history securely.
- The system should be modular and easy to extend.
- The system should run on standard C++ compilers.

---

5. Constraints

- The system is implemented in C++.
- The application will be console-based.
- Exchange rates are predefined and can be updated only by management.

---

6. Future Enhancements

- Add a graphical user interface (GUI).
- Support online exchange rate updates.
- Enable multiple concurrent cashier sessions.