SRS-DLD

**S/W Detailed Level Design**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name** | Currency Exchange System — Release 2 | | |
| **Block Name** | Compile Commander | | |
| **Author** | **Magnus Jaaska** | **Approver** |  |
| **Team** | Compile Commander | | |

This document represents Detailed Level Design (DLD). It describes the detailed system design and implementation plan in alignment with Agile principles. The DLD is updated incrementally with each release to reflect system evolution.

Contents

[1. Overview 4](#_Toc210821152)

[2. System Overview / Architectural Context 5](#_Toc210821153)

[3. UML Class Diagram (Technical Design) 5](#_Toc210821154)

[4. Class Specifications 5](#_Toc210821155)

[5. Interfaces and Abstractions 7](#_Toc210821156)

[6. Function Responsibilities 7](#_Toc210821157)

[7. Operation Flow 8](#_Toc210821158)

[8. Enumerations & Constants 9](#_Toc210821159)

[9. Validation Rules & Future Work 9](#_Toc210821160)

[10. Traceability Matrix 9](#_Toc210821161)

[11. Code Structure and File Mapping 10](#_Toc210821162)

[12. Revision History 10](#_Toc210821163)

* Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Revised contents | Author | Approver |
| **1.0** | 28.10.2025 | **Initial Release 2 DLD; three-layer refactor; interfaces; method responsibilities; traceability.** | Magnus Jaaska |  |
|  |  |  |  |  |

* Terms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Description** |
| UI | User Interface (ConsoleUI) |
| DLD | Detailed Level Design document |
| SRS | Software Requirements Specification |
| DIP | Dependency Inversion Principle |
| SRP | Single Responsibility Principle |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* References

1. SW Requirements Specification (docs/release-1/SRS.pdf)
2. Lecture 5–7 (Design for Quality, Modularity, Abstraction, Polymorphism)
3. Example DLD Currency Exchange (reference structure)

# 

# Overview

The Currency Exchange System automates currency conversion, verifies reserves before payout, records transactions in memory, and provides a daily summary. This DLD defines the Release 2 design: modular three-layer architecture (UI → Logic → Data), class responsibilities, interfaces, function contracts, constants, operation flow, and SRS traceability. Validation, exception handling, and file-based persistence are planned for Release 3.

# System Overview / Architectural Context

* **UI:** ConsoleUI — console interaction & formatting only.
* **Logic:** ExchangeManager — coordinates conversion, reserve checks, and reporting.
* **Data:** ICurrencyRepository / CurrencyRepository — in-memory balances & exchange rates.  
  Principles: Separation of Concerns, SRP, DIP, encapsulation via header/source split.

# UML Class Diagram (Technical Design)

# 

# Class Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Type** | **Description** | **Attributes** | **Methods** |
| ConsoleUI | UI | Console I/O and formatting; no business logic. |  | run(), parseCurrency(s), showMenu() |
| ExchangeManager | Logic | Coordinates exchange; checks reserve; updates report. | repo\_: ICurrencyRepository&, clock\_: IClock&, report\_: Report& | processExchange(txId, fromC, fromA, toC): Transaction |
| ICurrencyRepository | Interface | Abstracts data access for rates/balances. |  | getRate(from,to), getBalance(c), setBalance(c,amt), applyExchange(fromC,fromA,toC,toA) |
| CurrencyRepository | Data | In-memory balances & rates; implements interface. | balances\_, rates\_, low\_threshold\_ | getRate(), getBalance(), setBalance(), applyExchange(), is\_low(c), seedSampleRates() |
| Transaction | Domain | Immutable exchange record w/ computed to-amount. | tx\_id\_, timestamp\_, from\_currency\_, from\_amount\_, to\_currency\_, to\_amount\_, rate\_ | static make(id,ts,fromC,fromA,toC,rate): Transaction |
| Report | Domain | Aggregates transactions; prints daily summary. | tx\_: vector<Transaction> | add(t), print\_summary(os, repo) |
| Receipt | Utility | Formats receipt and CSV line. |  | print(t, os), csv(t): string |
| IClock | Interface | Time abstraction. |  | now\_iso8601(): string |
| RealClock | Service | System clock implementation. |  | now\_iso8601(): string |

# 5. Interfaces and Abstractions

|  |  |  |  |
| --- | --- | --- | --- |
| Interface | Purpose | Key Methods | Planned For (Release) |
| ICurrencyRepository | Decouple logic from storage; enable File/DB later | getRate(), getBalance(), setBalance(), applyExchange() | |  | | --- | | R2 (implemented) |  |  | | --- | |  | |
| IClock | |  | | --- | | Decouple time for tests |  |  | | --- | |  | | |  | | --- | | now\_iso8601() |  |  | | --- | |  | | |  | | --- | | R2 (implemented) |  |  | | --- | |  | |

# 6. Function Responsibilities

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | Method | Purpose | Input | Output | Notes |
| ExchangeManager | |  | | --- | | processExchange() |  |  | | --- | |  | | |  | | --- | | Computes payout, checks reserve, applies exchange, records transaction |  |  | | --- | |  | | |  | | --- | | txId, fromC, fromA, toC |  |  | | --- | |  | | Transaction | |  | | --- | | Throws on insufficient reserve/bad rate |  |  | | --- | |  | |
| CurrencyRepository | |  | | --- | | applyExchange() |  |  | | --- | |  | | |  | | --- | | Update balances for from/to |  |  | | --- | |  | | |  | | --- | | fromC, fromA, toC, toA |  |  | | --- | |  | |  | |  | | --- | | Encapsulated state change |  |  | | --- | |  | |
| Report | |  | | --- | | add() |  |  | | --- | |  | | |  | | --- | | Append to daily collection |  |  | | --- | |  | | |  | | --- | | Transaction |  |  | | --- | |  | |  | |  | | --- | | Called after success |  |  | | --- | |  | |
| Report | |  | | --- | | print\_summary() |  |  | | --- | |  | | |  | | --- | | Emit daily totals/closing balances |  |  | | --- | |  | | |  | | --- | | ostream, ICurrencyRepository |  |  | | --- | |  | |  | |  | | --- | | Presentation kept minimal |  |  | | --- | |  | |
| Receipt | |  | | --- | | print() |  |  | | --- | |  | | |  | | --- | | Human-readable receipt |  |  | | --- | |  | | |  | | --- | | Transaction, ostream |  |  | | --- | |  | |  | |  | | --- | | UI use |  |  | | --- | |  | |
| Receipt | |  | | --- | | csv() |  |  | | --- | |  | | |  | | --- | | CSV transaction line |  |  | | --- | |  | | Transaction | string | |  | | --- | | For logging later |  |  | | --- | |  | |
| RealClock | |  | | --- | | now\_iso8601() |  |  | | --- | |  | | Timestamp |  | string | |  | | --- | | Local ISO-8601 |  |  | | --- | |  | |
| |  | | --- | | ConsoleUI |  |  | | --- | |  | | |  | | --- | | run() |  |  | | --- | |  | | |  | | --- | | Interactive loop; parse/execute |  |  | | --- | |  | | |  | | --- | | stdin |  |  | | --- | |  | | |  | | --- | | stdout |  |  | | --- | |  | | Uses manager; handles errors |

# 7. Operation Flow

1. ConsoleUI reads input (“EUR 100 USD”).
2. Parses → ExchangeManager::processExchange().
3. Manager gets rate from ICurrencyRepository, computes payout.
4. Validates reserves → applies exchange in repo.
5. Creates Transaction, adds to Report.
6. UI prints Receipt; session end → Report::print\_summary().

# 8. Enumerations & Constants

|  |  |  |
| --- | --- | --- |
| Name | Value / Type | Description |
| |  |  | | --- | --- | | enum class Currency | | |  | | |  |  | | --- | --- | | {EUR, USD, GBP, SEK, DKK} | | |  | | |  | | --- | | Supported currencies |  |  | | --- | |  | |
| |  |  | | --- | --- | | LOW\_THRESHOLD | | |  | | |  |  | | --- | --- | | double = 100.0 | | |  | | |  | | --- | | Alert threshold for low balances |  |  | | --- | |  | |
| |  |  | | --- | --- | | INITIAL\_BALANCES | | |  | | |  |  | | --- | --- | | per-currency doubles | | |  | | Startup config for interactive run |

# 9. Validation Rules & Future Work

|  |  |  |
| --- | --- | --- |
| **Rule / Planned Feature** | **Description** | **Target** |
| Input validation & errors | amount > 0, valid currency codes; clear error messages | R3 |
| Strict reserve enforcement | Block payout > reserve; error codes | R3 |
| Exception policy | try/catch at UI; noexcept in Data | R3 |
| File I/O persistence | Save/load rates, tx, reports | R3 |
| Unit tests | Mock IClock & repo; per-layer tests | R3 |

# 10. Traceability Matrix

|  |  |  |
| --- | --- | --- |
| **Requirement (SRS)** | **Class / Method (DLD)** | **Notes** |
| RQ-01 Convert currency | ExchangeManager::processExchange() | Core conversion via repo rate |
| RQ-02 Show receipt | Receipt::print(), ConsoleUI::run() | UI formatting only |
| RQ-03 Maintain reserves | CurrencyRepository::applyExchange() | Encapsulated data update |
| RQ-04 Daily report | Report::add(), Report::print\_summary() | End-of-day summary |
| RQ-05 Time stamps | RealClock::now\_iso8601() | Abstracted time |

# 11. Code Structure and File Mapping

|  |  |
| --- | --- |
| **Class / Module** | **Files** |
| ConsoleUI | include/console\_ui.h, src/console\_ui.cpp |
| ExchangeManager | include/exchange\_manager.h, src/exchange\_manager.cpp |
| ICurrencyRepository | include/currency\_repository.h (interface) |
| CurrencyRepository | include/currency\_repository.h, src/currency\_repository.cpp |
| Transaction | include/transaction.h, src/transaction.cpp |
| Report | include/report.h, src/report.cpp |
| Receipt | include/receipt.h, src/receipt.cpp |
| IClock / RealClock | include/iclock.h, src/iclock.cpp |
| Entry point | src/main.cpp |
| Build | Makefile |

# 12. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change Summary | Author |
| 28.10.2025 | 1.0 | |  | | --- | | Initial DLD created from Release 2 refactor |  |  | | --- | |  | | Magnus Jaaska |