Hobbiton Assetment Task

Version Control

No	Author	Date
1	Samuel Wakumelo	16/02/25

Contents

Introduction

This document outlines the Database of choice and the Design for the implementation of the Wallet Management application.

Database Choice: SQL Server (SSMS)

SQL Server as a relational database management system. Below are some of the key reasons for choosing SQL Server:

1. Strong integration with .NET ecosystem
2. ACID compliance for data integrity
3. Robust support for complex queries and transactions
4. Enterprise-grade security features
5. Excellent tooling support through SSMS

Architecture Design

The architecture follows a clean, layered approach with clear separation of concerns:

Backend (.NET):

1. Controllers - Handle HTTP requests and route to appropriate services
2. Repositories - Implement data access patterns and abstract database operations
3. Utils - Common utility functions and helpers
4. Interfaces - Define contracts for dependency injection and loose coupling
5. Mappers - Transform between domain models and DTOs
6. DTOs - Data transfer objects for API communication
7. DatabaseContext - Entity Framework context for database interactions
8. Models - Domain entities representing database tables

Frontend:

1. Components - Reusable UI elements
2. Pages - Main view containers
3. AuthHook - Authentication logic and user session management
4. Routing - Navigation and URL handling

Design Decisions & Benefits

1. **Dependency Injection Pattern**
   - Writing to abstractions (interfaces) improves testability
   - Loose coupling between components
   - Easier to swap implementations without affecting dependent code

2**. Repository Pattern**
   - Centralizes data access logic
   - Provides abstraction over data persistence

- Makes unit testing easier through interface-based design

3**. DTO Pattern**
- Separates domain models from API contracts
- Provides flexibility in API versioning
- Controls data exposure to clients