Cairo University
Faculty of Computers and Artificial Intelligence

CS251

Introduction to Software Engineering

Tharwa

Software Design Specifications

Version 1.5

Name	ID	Email
Fatema El-Zhraa Ahmed Mohamed El-Fiky	20230280	fatmaelfeky922@gmail.com
Nagham Wael Mohamed	20231189	naghamw63@gmail.com
Aly El-Deen Yasser Ali	20231109	ali.el.badry.747@gmail.com

April of 2025

CS251: Optimum
Project: Tharwa

Software Design Specification

Contents

CS251: Optimum
Project: Tharwa

Software Design Specification

Team

ID	Name	Email	Mobile
20230280	Fatema El-Zhraa Ahmed Mohamed El-Fiky	fatmaelfeky922@gmail.com	01221990828
20231189	Nagham Wael Mohamed	naghamw63@gmail.com	01007600773
20231109	Aly El-Deen Yasser Ali	ali.el.badry.747@gmail.com	01286964627

Document Purpose and Audience

Purpose.

This SDS defines the design and structure of the Personal Investment Management Software. The software aims to help users track, analyze, and optimize their investment portfolios, ensuring informed financial decisions. It serves as a reference for consistent development, testing, and future improvements.

Audience.

1. **Development Team**
    - Software Engineers/Developers: Backend and frontend developers who will implement the system architecture and codebase based on the design specifications.
    - Technical Leads: Senior developers who will oversee implementation and ensure alignment with design decisions.
    - QA Engineers: Testers who will use the design documentation to create test cases and verify system behavior.
2. **System Architects**
    - Professionals responsible for reviewing and approving the high-level system design and ensuring it meets all technical requirements.
3. **Project Stakeholders**
    - *Product Owners*: Non-technical stakeholders who need to understand how design decisions fulfill business requirements.

Software Design Specification

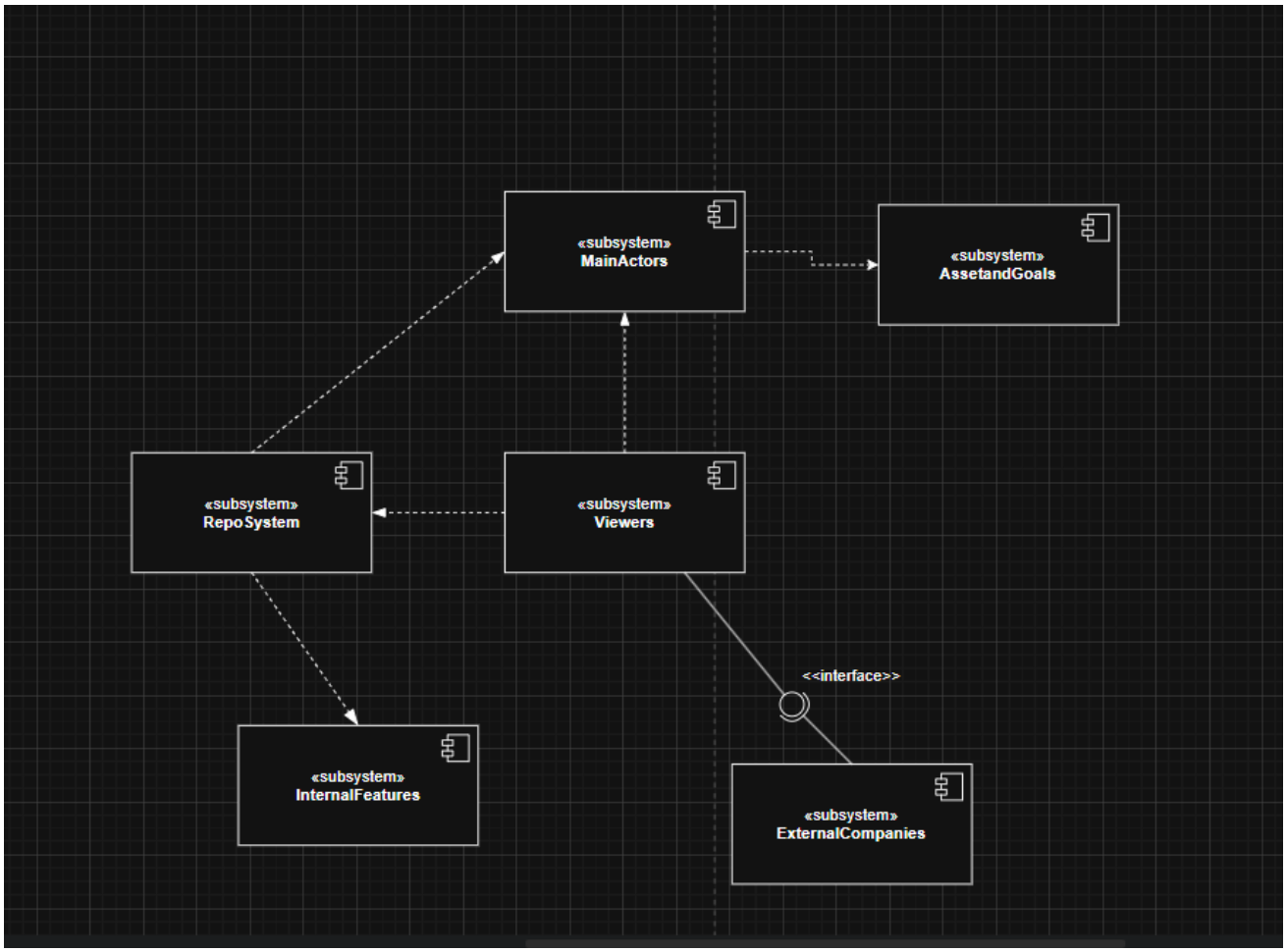
- *Islamic Finance Experts*: Domain specialists who will verify Sharia-compliance aspects of the design.
- *Banking Integration Partners*: Technical representatives from partner institutions (e.g., CIB) who need to understand integration points.

4. **Maintenance Team**
    - Future developers who will maintain, update, or extend the system and need comprehensive design documentation.

Software Design Specification

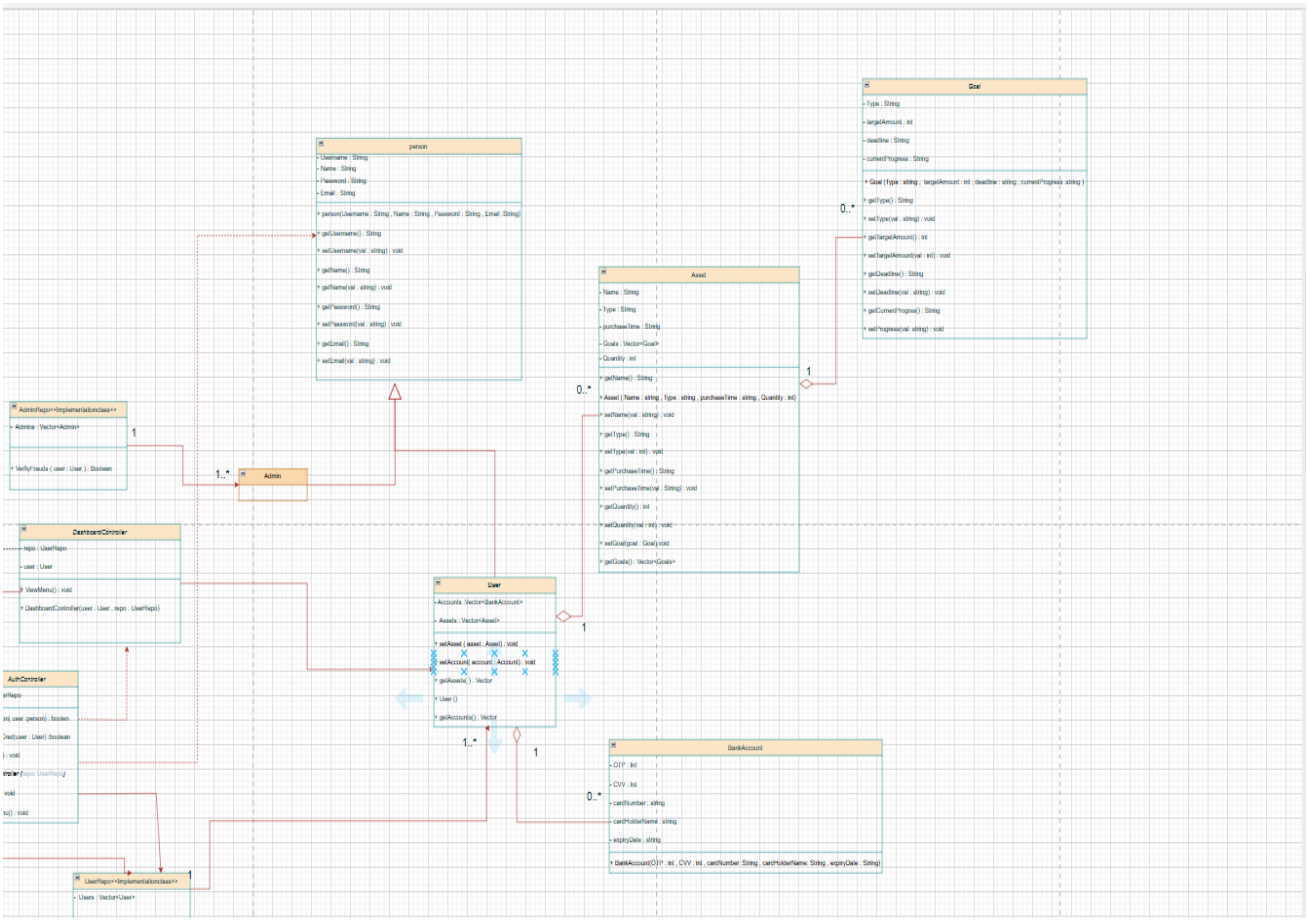
System Models

I. Architecture Diagram



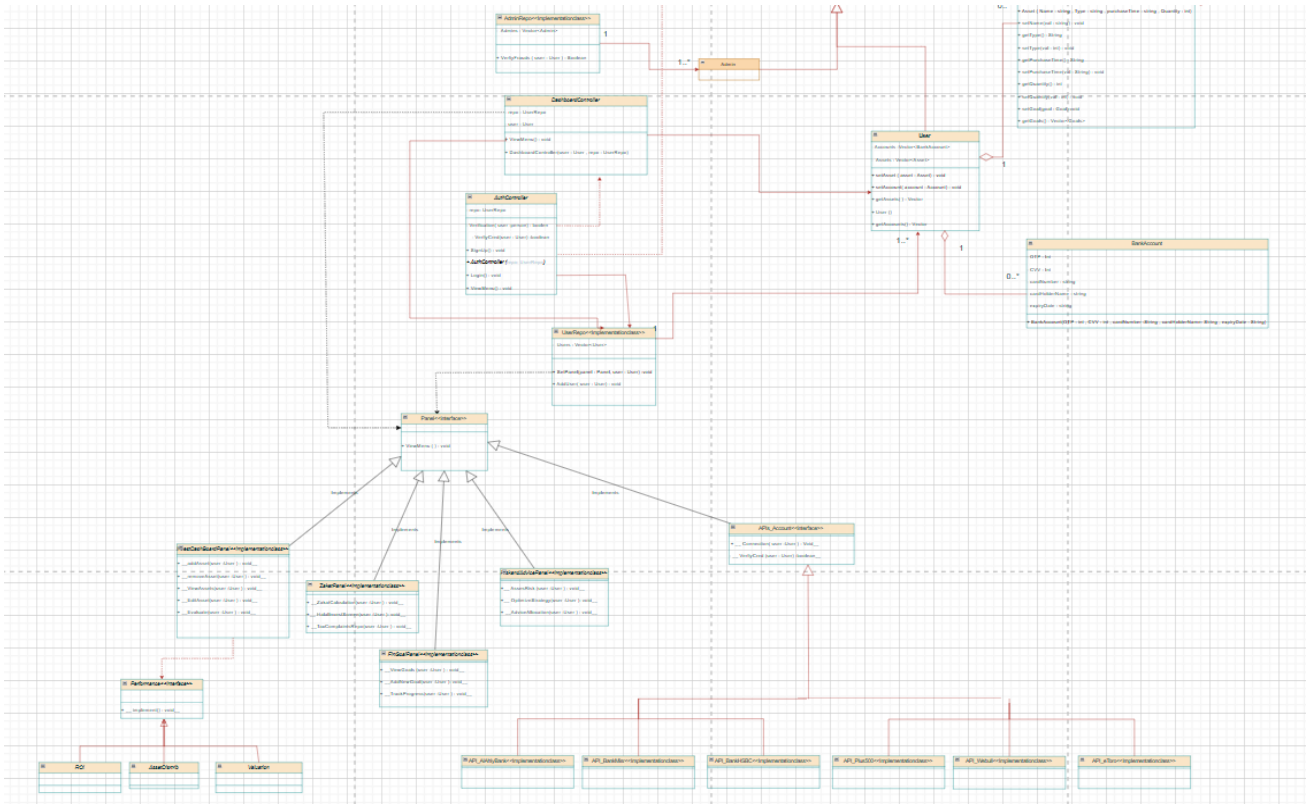
Software Design Specification

II. Class Diagram(s)



CS251: Optimum
Project: Tharwa

Software Design Specification



https://drive.google.com/file/d/1ur9hwcOiEpEGw7NfDLpcxsdzo6gH-VJE/view?usp=sharing

Software Design Specification

III. Class Descriptions

ID	Class Name	Description & Responsibility
1.	**Asset**	**Description**: Represents a financial asset (stocks, real estate, crypto, gold) in a user's portfolio. **Responsibilities**: • Store asset details (Name, Type, Quantity, purchaseTime). • Link to financial goals (Goals: Vector<Goal>).
2.	**Person**	**Description**: Base class storing common attributes for all human actors. **Responsibilities**: • Manage core attributes (Username, Name, Password, Email).
3.	**User** (Extends Person)	**Description**: Represents an investor with linked assets and bank accounts. **Responsibilities**: • Maintain collections of Assets and BankAccounts (Accounts: Vector<BankAccount>). • Provide methods to fetch assets/accounts.
4.	**BankAccount**	**Description**: Stores linked bank/card details for transactions. **Responsibilities**: • Secure sensitive data (OTP, CVV, cardNumber).

Software Design Specification

ID	Class Name	Description & Responsibility
5.	**Goal**	**Description**: Tracks financial objectives (e.g., retirement savings). **Responsibilities**: • Store goal metrics (targetAmount, deadline, currentProgress). • Update progress via setters (setProgress()).
6.	**AuthController**	**Description**: Handles authentication and user sessions. **Responsibilities**: • Verify credentials (Verification(user: Person): boolean). • Manage signup/login flows (SignUp(), Login()). • Interact with UserRepo for persistence.
7.	**ZakatPanel** (Interface)	**Description**: Defines contracts for Sharia-compliant zakat operations. **Responsibilities**: • Declare methods for zakat calculation (ZakatCalculation()). • Enforce halal screening (HalalInvestScreen()).
8.	**RiskAndAdvicePanel** (Interface)	**Description**: Specifies risk assessment and investment advice features. **Responsibilities**: • Define risk analysis methods (AssetRisk()). • Outline strategy optimization (OptimizeStrategy()).

Software Design Specification

ID	Class Name	Description & Responsibility
9.	**FinGoalPanel** (Interface)	**Description**: Template for financial goal tools **Responsibilities**: • Declare goal tracking methods (ViewGoals(), TrackProgress()). • Support adding new goals (AddNewGoal()).
10.	**Admin(Extends Person)**	**Description** Represents an administrator in the system. **Responsibilities:** Potentially manage users, verify fraud activities, and oversee the system's operations.
11.	**AdminRepo<<Implementation >>**	**Description** Represent the interface that the admin will interact with Responsibilities: • Check the Authorization of Admin License • Verify the Frauds of user (VerfiyFrauts (user : User))
12.	**DashboardController**	**Description** Controls the interaction between the user and the system's dashboard, coordinating with user data through a repository. **Responsibilities** • Manage the current User session. • Handle user operations by interacting with the UserRepo. • Display the dashboard menu to the user using ViewMenu()

Software Design Specification

ID	Class Name	Description & Responsibility
13.	**UserRepo<Implementation>**	**Description** Repository class that stores all system users and their associated bank APIs. **Responsibilities:** • Maintain a Vector of User objects and a reference to a Bank API. • Provide access to add new user data through the method addUser(User user). • Determine the panel that user will go to using setPanel()
14.	**InvestDashboardPanel (Interface)**	**Description** Interface defining user investment-related operations on the dashboard. **Responsibilities:** ○ **Add new** assets to a user's profile. ○ Remove assets from a user's profile. ○ View a user's current assets. ○ Edit user assets. ○ Evaluate **the user's investment performance.**
15.	**Performance (Interface)**	**Description:** Interface for measuring different aspects of user performance. **Responsibilities:** • Define a common implement() method that will be customized for various performance evaluations like ROI (Return on Investment), Asset Distribution, and Valuation.

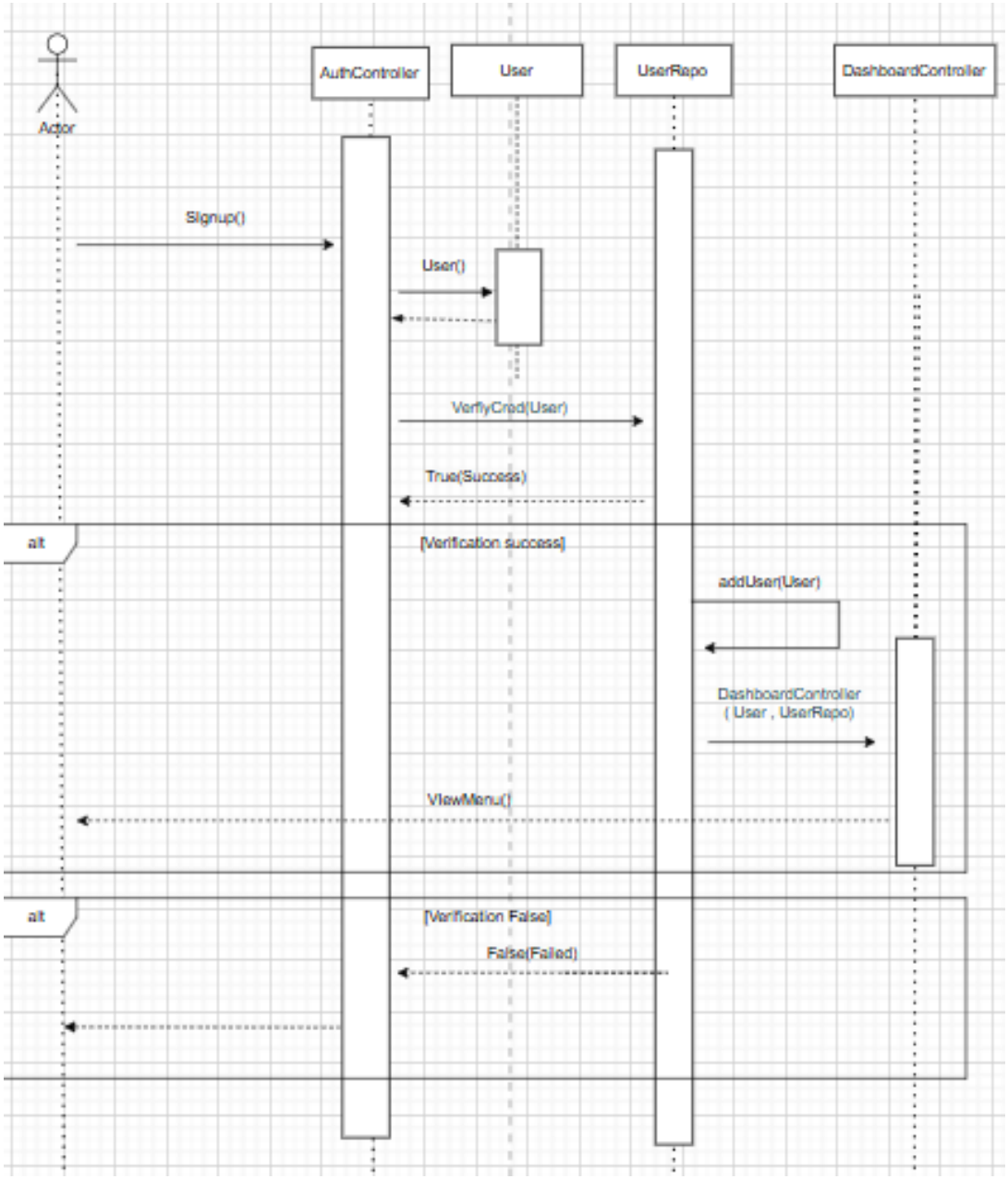
Software Design Specification

ID	Class Name	Description & Responsibility
	Api_Accounts	**Description:** It is an interface that defines the standard operations required for connecting to and verifying user credentials across different financial account systems. It serves as a template to ensure that all financial service implementations follow a consistent structure for integration and authentication. **Responsibility:** • Specify a method to establish a connection between the user and the financial account (Connection(User user)). • Specify a method to verify the user's credentials (VerifyCred(User user)). • Enforce a unified protocol that must be followed by all implementing classes, such as API_AlAhlyBank, API_BankMisr, API_BankHSBC, API_Plus500, API_Webull, and API_eToro. • Allow flexibility for each financial API to provide its own specific connection and verification logic while maintaining a common interface.

Software Design Specification

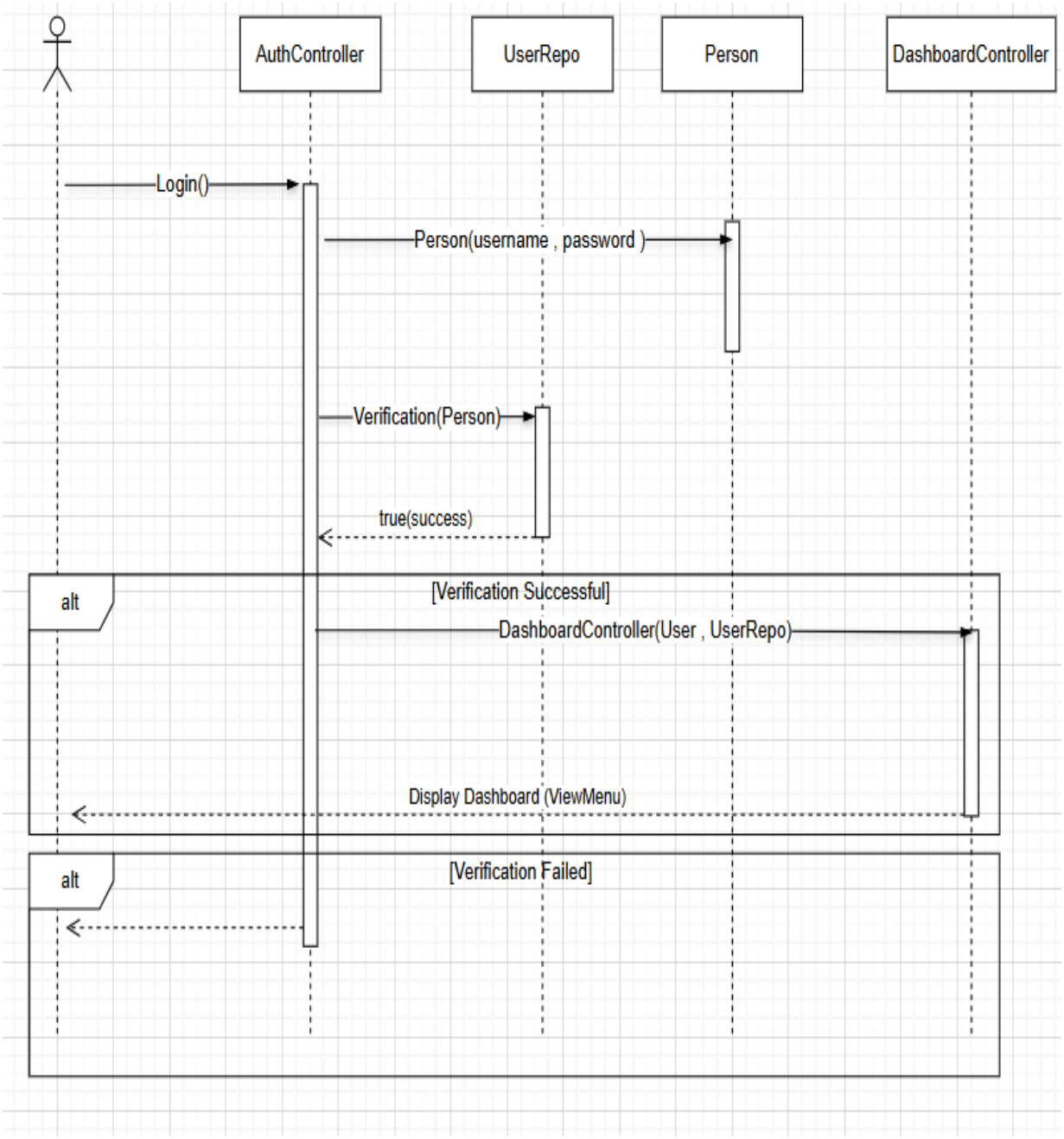
IV. Sequence diagrams

Sign up Use case



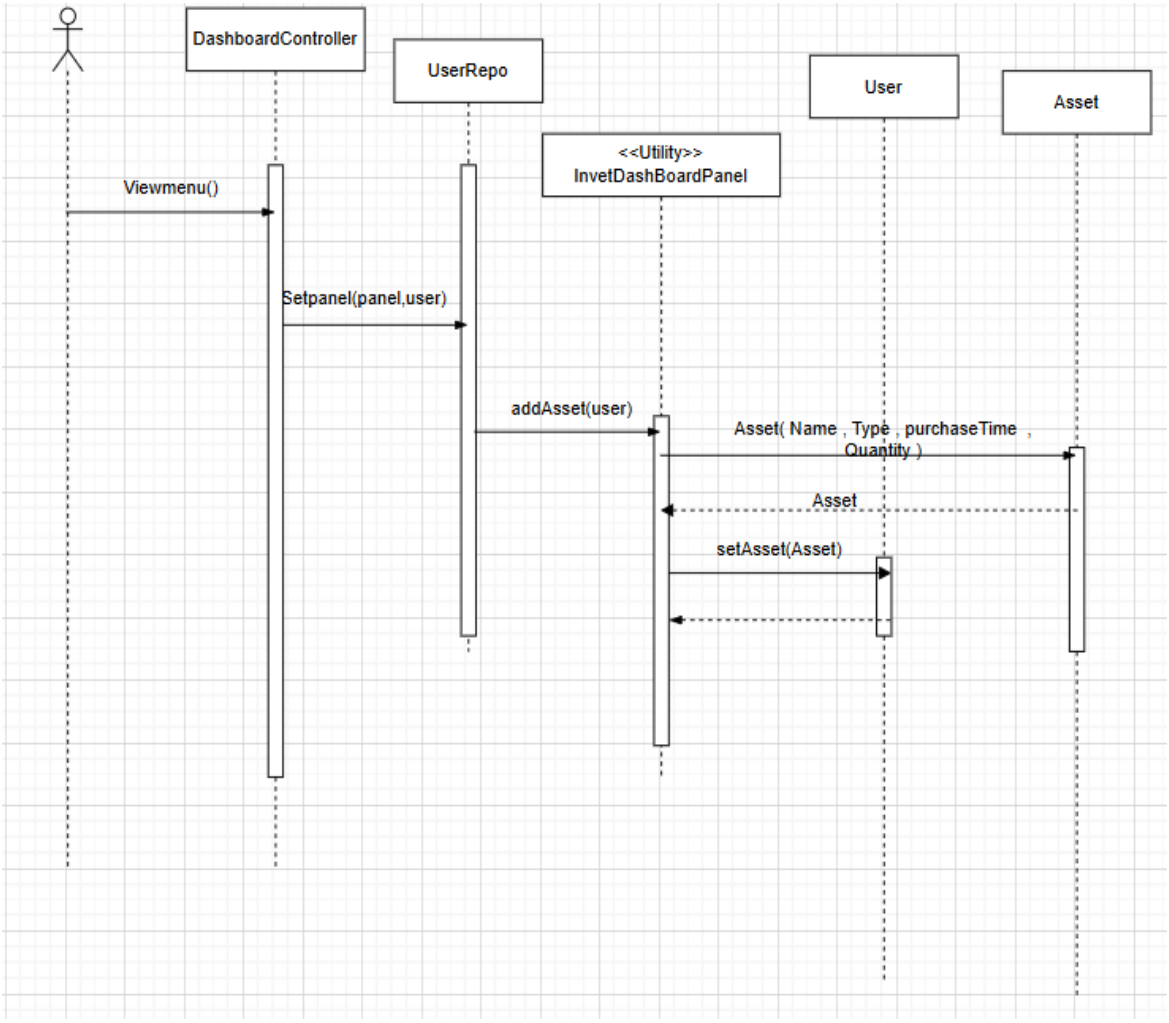
Software Design Specification

Login Use Case



Software Design Specification

Add Assets Use Case.



Software Design Specification

Class - Sequence Usage Table

Sequence Diagram	Classes Used	All Methods Used
1. Sign Up	Authcontroller User UserRepo DashboardController	Signup() User() VeerifyCred(user) addUser(user) DashboardController(user, userRepo) ViewMenu()
2. Login	AuthController UserRepo Person DashboardController	Login() Verification(User : Person)
3. AddAssets	DashBoardController UserRepo <<Utility>>InvestDashBoardPanel User Asset	Viewmenu() Setpanel(panel,user) addAsset(user) Asset(Name,Type,purchaseTime , Quantity) setAsset(Asset)

V. State Diagram

VI. SOLID Principles

VII. Design Patterns

Software Design Specification

Tools

- Draw.io

Ownership Report

Item	Owners
Architecture Diagram, Part of Class Diagram, Add asset Sequence Diagram.	Fatema El-Zhraa Ahmed Mohamed El-Fiky
Part of Class Diagram , Sign up Sequence Diagram, File Organization,Purpose of SDS	Aly El-Deen Yasser Ali
Part of Class Diagram , login Sequence Diagram, Audience of SDS	Nagham Wael