Architecture Design Document for PIMS

1. Overview

1.1. System Overview

Personal Investment Management System (PIMS) is intended to help a user keep account of his/her investments in instruments such as shares and bean deposits. PIMS is to provide the user the rate of returns (ROR) he/she is getting, as well as net worth (NW) of the different portfolios and the total net worth. Alerts about any maturity dates can also be provided. For current value of some investment, PIMS is to obtain the information from the web.

1.2. System Context

The system context is defined clearly in the SRS. Basically, the user is the main sink of the information. The user is also a major source of information/data. The other source of data is a website from where data on current prices is obtained.

1.3. Stakeholders of PIMS

The main stakeholders for the system are the individual users who might use the system and the system designer/builder who will build PIMS. The main concerns of the two stakeholders are:

- **For Users:** The usability of the system and providing ROR and NW info. Reasonable response time is also a concern.
- **For designer/builder:** The system is easy to modify, particularly to handle future extensions mentioned in the SRS (i.e. the system may become a multi-user system, which may require use of databases instead of files for keeping data.)

Hence, the key property for which the architecture is to be evaluated is the modifiability or extensibility of the system. Response time performance is another factor for which the system needs to be evaluated.

1.4. Scope of this Document

In this document, we describe two possible architectures for PIMS, compare them for various quality attributes, and then choose the most appropriate one, which is our final proposed architecture for PIMS. By discussing the two alternatives, we also provide the rationale for selecting the final architecture. For architecture, we consider only the component and connector view.

1.5. Definitions and Acronyms

Definitions:

Transaction	A real event that involves flow of personal money. In the context of shares, it is buying/selling a group of shares of the

	same company, and in context of Bank it is deposit/withdrawal of money to/from one's account
Security	A set of all transactions pertaining to a company share or a bank account.
Portfolio	A set of securities.
Net Worth	The sum total of all the money of the investor in form of shares and bank balances.
Rate of Investment	The interest that user gets on a particular investment. In the context of a bank account it is the annual interest and in case of a company share it is computed on a monthly basis, details of which can be found in the appendix A of the SRS.

Acronyms:

PIMS	Personal Investment Management System.
NW	Net Worth.
ROI	Rate of Investment

2. Architecture Design

2.1. Architecture 1: The Repository Model

This architecture consists of a data repository, which contains information about the portfolios, securities, transactions, alerts and current share prices. There are separate modules for performing various tasks such as computation of net-worth/ROI, modification of the data repository etc. Following are the principle components of this architecture.

#	**Component**	**Component Type**	**Description**
1	Data Repository	Database	This module is the database containing information about the Portfolio/Security/Transaction, alerts and current share prices
2	Remote share price database	Database (Website)	This module is a database at some remote site, which is accessed for getting the current share prices.
3	Master Controller	Processing (Interface module)	This is basically the graphical interface module which interacts with the user on one side and makes appropriate calls to the other modules, discussed below to serve the user requests.
4	Create/ Delete/Rename Portfolio/Security /Transaction	Processing (Database modification)	Modifies the information related to the investment in the data repository (Portfolio/Security/Transaction)
5	Compute Net-Worth/ ROR	Processing (Computation)	This module computes the net-worth and ROI using the information in the data repository

6	Get/Set Alerts	Processing (Database access or modification)	This module accesses the data repository to set/get alerts
7	Net Loader	Processing (Download and database modification)	This module downloads the current share prices from a remote database and updates the data repository
8	Get current share price	Processing (Database access)	This module accesses the data repository and returns the price of a share

Following are the connectors of the architecture:

#	Connector	Connector Type	Description
1	R/W connectors	Database access/ modification	These connectors are between the data repository and the modules 4, 5, 6 and 7. These represent access (R) or modification (W) of the data repository
2	Read only connector	Database access	This connector is between the data repository and the modules 5 and 8. It represents reading of relevant data from the data repository
3	Write only connector	Database access / modification	This connector is between the net loader module and the data repository module. This represent the updating of the current share prices
4	URL connector	HTTP	This connector is between the Net Loader module and the Remote share price data module. This represents the
5	Control connector	Module invocation or function calls	This connector is between the master controller module and the modules 4, 5, 6, 7 and 8. This represent the invocation of these modules by the master controller

The diagram below shows this architecture:

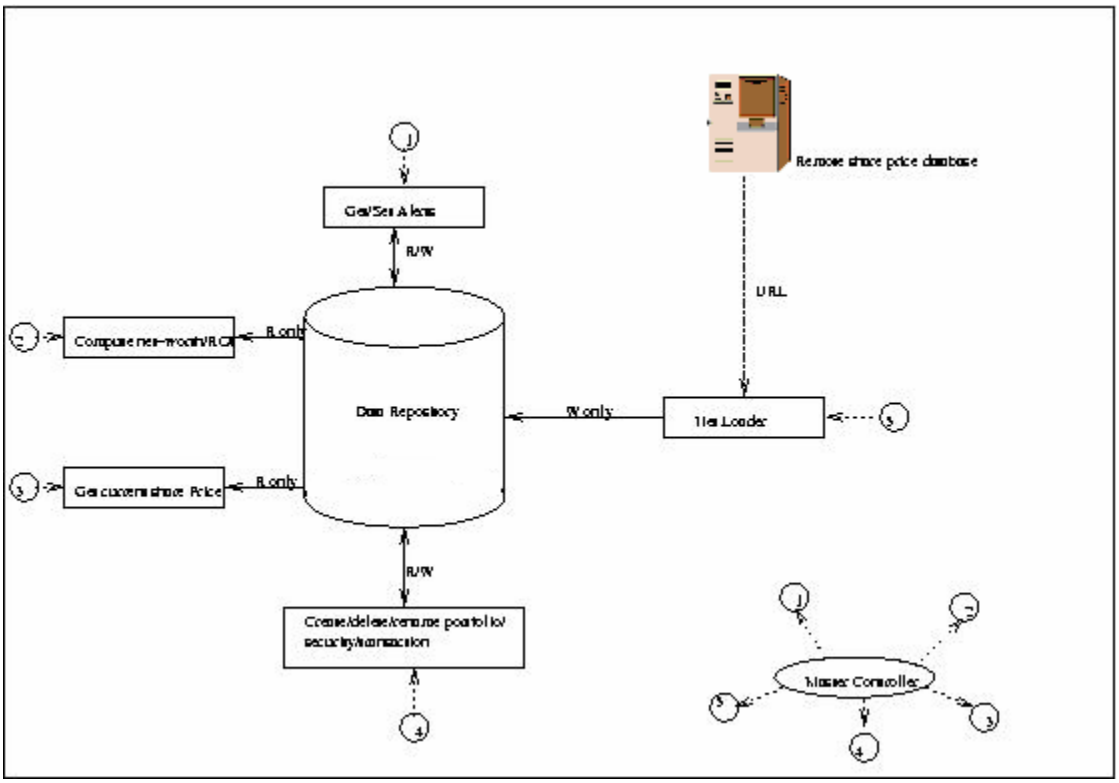
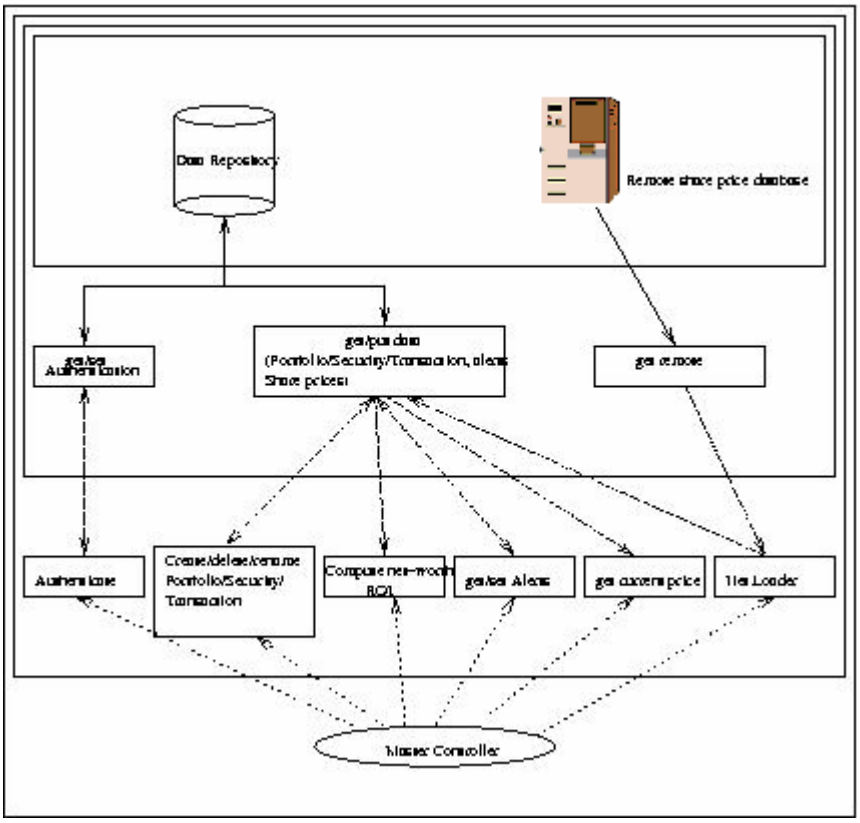


Figure 3.1: The Repository Model

2.2. Architecture 2: The Access Layer Model (4 layered)

This architecture is similar to the data repository architecture. The main difference lies in the fact that here, we have a data access layer separating the business logic and data repository. Data retrieval and modification is done via this data access layer, while all the processing of data or implementation of the business logic done in the business logic layer. The 4th layer is the presentation layer (master control) which is responsible for interacting with the user and invoking modules of the business logic layer. The usefulness of the data access layer comes from the fact that if the type of database is changed then only the access layer needs to be modified while the processing logic remains the same. The components remain the same except that now we would have a new data access component *get/put data* which would be responsible for reading and writing data corresponding to Portfolio/Security/Transaction, alerts and share prices. There is also a component *get remote* which reads the current share prices from the remote database. The diagram below shows this architecture.



1. Data Repository layer

2.Data Access layer

3. Business logic layer

4. Presentation layer

Figure 4.1: The 4 layered architecture

2.3. Comparing the Architectures

Here we compare the architectures with respect to various quality attributes.

Criteria	Architecture 1	Architecture 2
Change In Data repository	Not easy	Easy
Change in the remote database location from which prices are extracted	Easy	Easy
Addition in functionalities	Easy	Easy
Extension to multi-user	Difficult	Easier
Provision of additional securities (passwords, etc)	Difficult	Easier

Catering of new type of securities	Easy	Easy

3. Final Architecture of PIMS

From the above table we see that architecture 2 is better as far as change in data repository is concerned. Moreover, it is also easier to extend it to a multi-user system, which involves additional security issues. However, the performance of Architecture 2 is likely to be poorer than the first one, though it should be only marginally different.

Since, in future data repository may be changed, and the system may be made multi-user we prefer Architecture 2 to Architecture 1. (These have also been stated as the criteria for evaluation.) Thus we choose Architecture 2 for the Personal Investment Management System.