

# System design document for ASTRO

Table of Contents

Version: 2

Date 2012-04-17

Author:

*Mattias Markehed*

*Daniel Malmqvist*

*Erik Ramqvist*

*Kristian Sällberg*

This version overrides all previous versions.

## 1 Introduction

Stock robot.

### 1.1 Design goals

The design will be divided into many separate parts, making it easy to replace for instance the parser or the Buy/Sell API.

The GUI will be loosely coupled with the under laying model.. The design must also be testable with junit.

### 1.2 Definitions, acronyms and abbreviations

**Algorithm System** - Will load algorithms for a given portfolio. Will also give the option of inserting new algorithms.

**Avanza parser** - A parser to fetch and parse stockdata from the stocktrader Avanza.

**Buy/Sell API** - Our self defined API for buying and selling securities internally. Used by algorithms loaded into the system.

**GUI** - graphical user interface

**Java** - platform independent programming language.

**JPA** - Java Persistence API a way to translate a database into java classes.

**MVC** - Model View Control, a software architectural pattern for isolating the View etc.

**MYSQL** - A free database system.

**Parser** - In this project, the parser is a stand alone program taking in data from a specific source and inserting that data into our Price List Database. Users will define their own parsers for data sources not supported by us.

**Portfolio Database** - The database design that is used in the Portfolio System.

**Portfolio System** - A system for creating a portfolio that will be coupled with one or zero algorithms, will also give the possibility to manually buy/sell stocks.

**Price List Database** - Database that keeps stock prices for each minute (or other time unit) the

parser has been running.

**Stock/Security** - Part ownership of a company.

**SQL** - Structured query language. Used for communication with the database.

## **2 System design**

### **2.1 Overview**

In this section we explain the overall design choices.

### **2.2 Software decomposition**

#### **2.2.1 General**

Package diagram. For each package an UML class diagram in appendix

#### **Application ASTRo Harvester**

Harvester - Fetches stockdata from datasource and saves it to database.

Database - Stores the data recieved from database.

Parser - Parses data recieved from datasource.

Model - Represents the stockdata.

#### **Application ASTRo**

Portfolio

Portfolio - Holds information about economic data and what algorithm to use.

PortfolioHandler - Handles the running algorithms.

PortfolioSettings - Settings for the running portfolio

BuySell - handles buying and selling.

IBuySell - interface for buying and selling

Database

Util - used to setup connection to the database.

Algorithms

Loader - Loads algorithms for use in portfolio

Algorithms - Algorithm that should be run by portfolio

Generics

Pair - Holds a tuple of the type pair.

GUI

Graph - Handles Graph for showing different kind of graph.

Portfolio - Shows the portfolio information.

PortfolioWizard - Wizard for creating a new portfolio.

WorkingStock - Shows stock currently owned.

Sock - Show stockdata.

### 2.2.2 Layering

Se figure

`img.src("error file missing") // 600613`

### 2.2.3 Dependency analysis

The algorithms are depended of parsed data from the web, in order to determine if it should buy or sell.

----- STAY OUT BELOW, VECKA 16!!!

## 2.3 Concurrency issues

X

We will have many different threads for many functions. For example the parser will run in a seperate thread.

The database has build-in concurrency properties.

## 2.4 Persistent data management

All of our data will be stored in a SQL-database, called MySQL.

## 2.5 Access control and security

## 2.6 Boundary conditions

## 3 References

Domain Model: <http://stock-robot.googlecode.com/files/domainmodel.png>

Design Model: <http://stock-robot.googlecode.com/files/designmodel.png>