

---

# CMPE 450

## SOFTWARE ENGINEERING

# OBJECT DESIGN

# DOCUMENT

---

Version 2.0

*Project Name:* Stock Follow Up  
System Project

*Submitted to Instructor :* AYŞE BAŞAR BENER  
*Assistant:* GÜL ÇALIKLI, ONUR GÜNGÖR

*Submitted by:* **Design Group 3**

**Group:**  
ŞAHİN CEM GEYİK  
HÜSEYİN KIRAN  
NESLİHAN KUKUT  
EMRE YILMAZ

*Boğaziçi University, İstanbul*  
*November 14th, 2006*

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	OBJECT DESIGN TRADE-OFFS . . . . .	2
1.2	INTERFACE DOCUMENTATION GUIDELINES . . . . .	3
1.3	DEFINITIONS, ACRONYMS, AND ABBREVIATIONS . . . . .	3
1.4	REFERENCES . . . . .	3
<b>2</b>	<b>PACKAGES</b>	<b>4</b>
2.1	PACKAGE DIAGRAM . . . . .	4
2.2	PACKAGE DEFINITIONS . . . . .	5
2.2.1	User Interface . . . . .	5
2.2.2	Stock Follow Up Package . . . . .	5
2.3	CLASS DIAGRAM . . . . .	6
2.4	CLASS DEFINITIONS . . . . .	7
2.4.1	Stock Creation Class . . . . .	7
2.4.2	Stock Approval Class . . . . .	7
2.4.3	Stock Management Class . . . . .	8
<b>3</b>	<b>Class Interfaces</b>	<b>10</b>

---

## List of Figures

2.1	PACKAGE DIAGRAM of STOCK FOLLOW UP SYSTEM .	4
2.2	CLASS DIAGRAM . . . . .	6
2.3	Stock Creation Class . . . . .	7
2.4	Stock Approval Class . . . . .	7
2.5	Stock Management Class . . . . .	8

# Introduction

The purpose of this document is to summarize the design principles of Stock Follow up System in detail. Stock Follow up System is suppose to be an extension to the system SAS. In this document, the necessary module to integrate the system in SAS is defined with its classes and methods. The design may require subtle changes during the program lifecycle.

## 1.1 OBJECT DESIGN TRADE-OFFS

The purpose of Stock Follow up system is to store the merchandise information in a single database in an organized and easy way. To establish that aim, the previous project SAS and Stock Follow up System have to be integrated and work as a single complete system. While designing the Stock Follow up System we considered this aim and have some trade-offs. Firstly, the integrity is the most important component that the system should have. First trade-off is made on this issue. To make Stock Follow up System compatible with SAS, complexity of the system has been increased. We need to use the fields and variables, which are available on SAS database tables, without any change or modification. Therefore, handling with the data taken from SAS and its database are used in a complex way. Secondly, another trade-off is made between speed and time. We need to use the materials that we have. The previous project SAS make only one user available to display database at a time. It is said that only manager has right to display database information, however, it is important to denote that two different users can access the files by waiting each other to finish their work. But we have to use SAS not to waste time correcting it.

## **1.2 INTERFACE DOCUMENTATION GUIDELINES**

In Stock Follow up System design, there is only one package and it is used by SAS. During the design process, it is tried to implement the same naming style with SAS not to confuse the successor developers. The aim is to be understandable to everyone. Classes have to be named with nouns and their initial letters have to be capitalized. In case there are two or more nouns in the class name, the initial letters of each one of them have to be capitalized. Verbs in methods names are not to be capitalized. e.g. `sendRequestForm()`.

## **1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS**

There are some definitions and abbreviations that you may frequently see in this document, and they are as followings:

SFS: Stock Follow Up System SAS: Satn Alma Sistemi GD: General Depot  
GDC: General Depot Clerk DD: Departmental Depot DDC: Departmental  
Depot Clerk

## **1.4 REFERENCES**

SDD 2.0v of previous SAS Available at <http://satlab.cmpe.boun.edu.tr/cmpe450test>

# PACKAGES

## 2.1 PACKAGE DIAGRAM

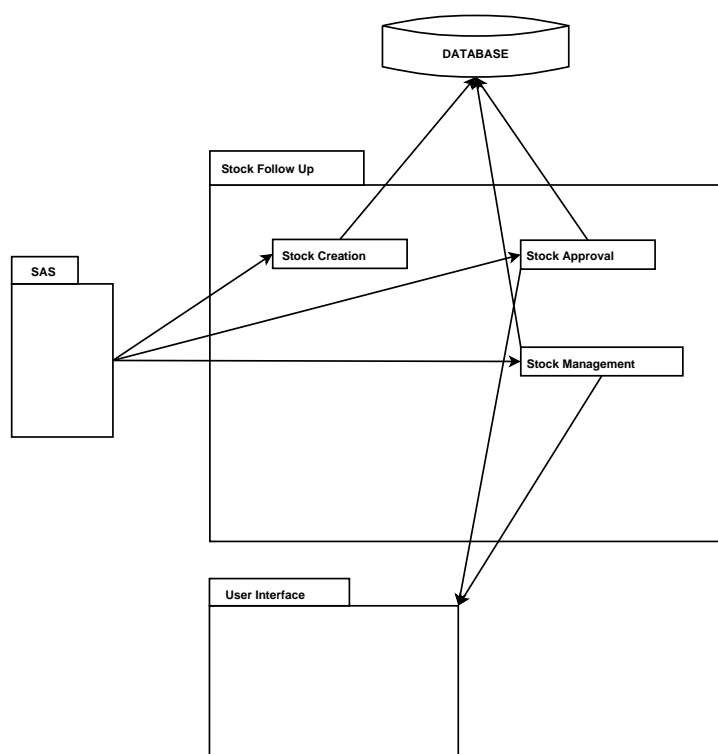


Figure 2.1: PACKAGE DIAGRAM of STOCK FOLLOW UP SYSTEM

## 2.2 PACKAGE DEFINITIONS

There are 2 different packages used in this system. One is named StockFollowUp and is used for the internal structure. This package includes the classes which participate in the internal execution. The second package is UserInterface which is designed for the interface of the StockFollowUp module.

### 2.2.1 User Interface

The package named UserInterface is responsible for handling the operations that are related to the user interface execution of the system. The methods provided by this interface package are explained in detail in section 3.

### 2.2.2 Stock Follow Up Package

This package encapsulates the classes and the methods which are related to the internal execution of the system rather than the interface interactions. The name of these classes are as follows: Stock\_Creation, Stock\_Approval and Stock\_Management. When the user wants to add an item into stock, an object of the class Stock\_Creation is created. Using the function addToStock, the stock is added to the table in the database where the pending items are kept. Approval of an item that is to be added into the stock is done by an instance of Stock\_Approval class. The user is able to view the pending items and approve them by the methods of this class. These methods are view\_pending and approve. Accessing the info in the pending items table of database, the method view\_pending is able to list the items that are not approved, yet. The method approve deletes an item from the pending items table and puts it into the main table where the items that are approved to be put into stock are kept. The class Stock\_Management includes three methods. These are StockInspect, SetLimit and Spending. The method StockInspect generates a query according to the parameters which are criteria that are used while generating the query. After generating the query, it is executed and the result is displayed. The method SetLimit takes StockID and limit as parameters. It assigns a value to the minimum limit field of the corresponding item in the main table. This limit is used to detect the situation that the amount of an item in the stock reaches its minimum limit and must be purchased to increase this amount. Spending method takes two parameters which are StockID and amount. It decrements the amount of the

corresponding item by the parameter amount.

## 2.3 CLASS DIAGRAM

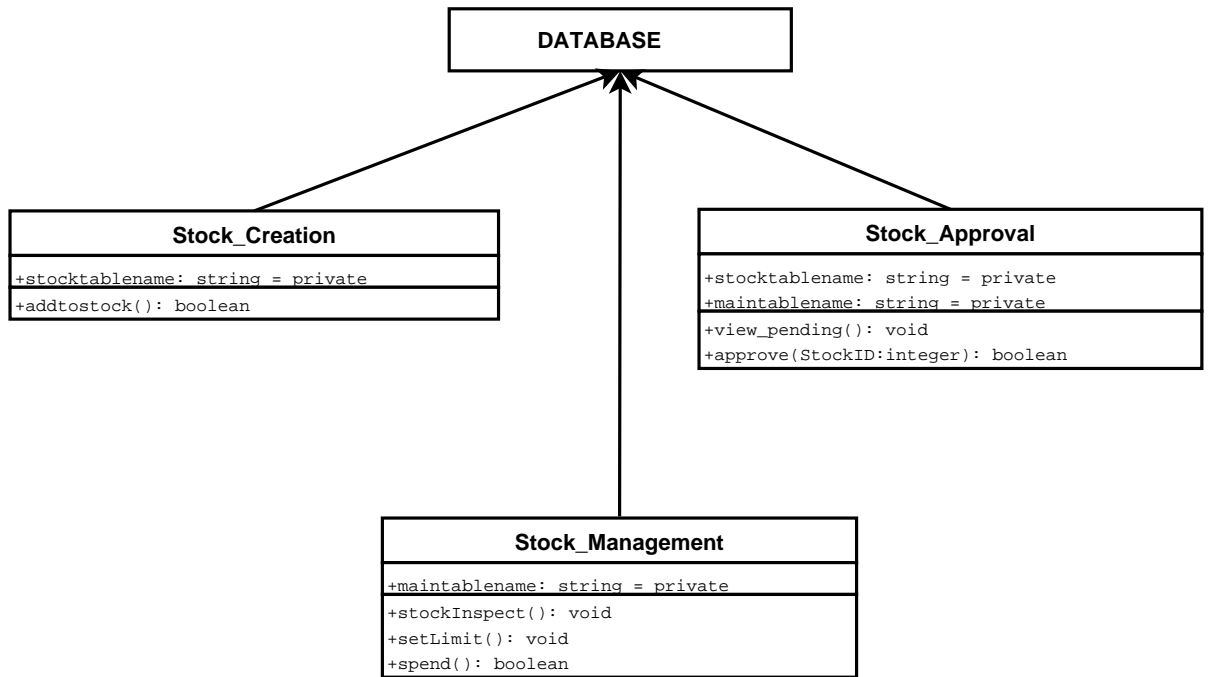


Figure 2.2: CLASS DIAGRAM



## 2.4 CLASS DEFINITIONS

### 2.4.1 Stock Creation Class

<b>Stock_Creation</b>
+stocktablename: string = private
+addtostock(): boolean

Figure 2.3: Stock Creation Class

Stock.Creation Class is used by SAS. Whenever a new stock query is made and approved by SAS, this class is called. The class has a method addstock( )

Boolean addtoStock(): this method gets the data from stock table and display the GDC user to take an action. If GDC approves it, the data will be added to stock database, else the data, which are on pending, will be recorded to table as rejected.

### 2.4.2 Stock Approval Class

<b>Stock_Approval</b>
+stocktablename: string = private
+maintablename: string = private
+view_pending(): void
+approve(StockID:integer): boolean

Figure 2.4: Stock Approval Class

Stock\_Approval class is a part of the package StockFollowUp. There are two methods in this class which are view\_pending and approve. The first method is view\_pending. It displays the list of the items that are not approved, yet. These are the items that are kept in pending items table of the database. This method takes no input parameter. The second method is approve. This method takes StockID as the parameter and removes the corresponding item from the pending items table and adds it to the main table in the

database. The parameter StockID is of type integer. The first attribute StockTableName is of type String. It is the name of table keeping the pending items. The second attribute MainTableName is of type String. It is the name of the table keeping the items that are approved to be put into stock.

### 2.4.3 Stock Management Class

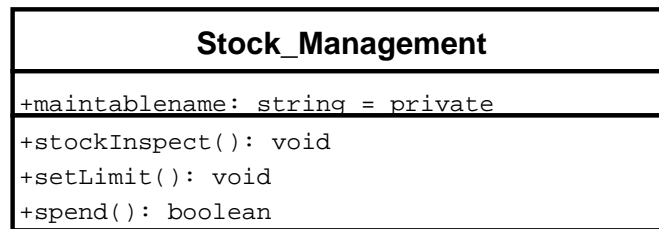


Figure 2.5: Stock Management Class

Stock\_Management Class is for managing the current stock data in our system. The methods belonging to this class are used for viewing the information of the items in stock, removing items from the stock (e.g when we use paper for photocopying, these must be removed from the stock), and setting limits for the minimum amount that an item must be existent in the stock. This class holds the maintablename:String of the main table for the stock system on which it performs actions.

The method stockInspect() gives us the observation capability with many different parameters. This methods merely uses queries to bring out the statistics and information about the current stocks. The method gets the department for which the stock information will be shown, the item names to be viewed, the time domain (for example, a general deposit clerk might want to see the last 3 months of stock entrances for an academic department), the price domain(for the items) and some options for the format the information will be given (monthly, yearly etc.). This method will also enable the users to plan the future about purchasing items etc.

The method setLimit() takes the name of the department and the name of the item and sets a minimum limit for that item in that stock. This method will help the users put the minimum value so that they will get a message when there is "too low amount" of that item in the stock. For keeping the minimum limits, the system might need a seperate table.

The method `spend()` takes the name of the department and the name of the stock id with the amount to spend and removes items from the specified stock. This method also triggers the alert if the spending action causes the amount of goods in stock to be below the minimum limit.

---

## Class Interfaces

The purpose of the Stock Follow Up module to add the functionality of stock management to the previous SAS system. Although the Stock Follow Up system is created as a module, It doesn't work independently and it both needs and provides some interfaces to the SAS system.

The method `addtoStock()` provided to the SAS system gives the system the ability to add the items to stock right after they are purchased. Other than that the system must be reachable from the SAS system and the interfaces must be the same as the SAS system.

---

# Index

Alert, 9

Department, 8

General Deposit Clerk, 8

Inspect, 8

Instructor, 1

Interface, 10

Limit, 8

Price, 8

Remove, 8

SAS, 10

Spend, 9

Statistics, 8

Stock, 8, 10

Stock Approval, 7

Stock Creation, 7

Stock Management, 8

Time, 8

View, 8