CMPE 450
SOFTWARE ENGINEERING
OBJECT DESIGN DOCUMENT

Version 3.0

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
December 14th, 2006

Contents

List of Figures

1 Introduction

The purpose of this document is to summarize the design principles of Stock Follow up System in detail. Stock Follow up System is designed to be an extension to the system SAS. In this document, a package is defined with its classes and methods to integrate into the system SAS.

1.1 OBJECT DESIGN TRADE-OFFS

With the aim of storing merchandise information in a single database in an organized and easy way, the Stock Follow up System is designed. To establish this aim, the Stock Follow up System is embedded into the previous project SAS and they work as a single complete system, now. While designing the Stock Follow up System, there were some trade-offs.

The first trade-off is made to establish integrity as the integrity is the most important component that the system should have. To make Stock Follow up System compatible with SAS, complexity of the system has been increased. We used the tables and variables, which are available on SAS database tables, without any change or modification. To handle with the forms which are available on database, new procedures are written.

Second trade-off is made on time usage. The previous project SAS gives only one user to display database at a time. It does not seem to be a big problem since only the manager has right to display database information, however, it is important to denote that two different users cannot access files simultaneously, they have to wait each other to finish their work. But we have to use SAS in this way not to waste any time correcting it because it is not necessary.

1.2 INTERFACE DOCUMENTATION GUIDELINES

In Stock Follow up System design, there is only one package which is used by SAS. In the designing process, the same naming style with SAS is implemented not to confuse the successor developers. Our purpose is to be clear to everyone. Classes are named with nouns whose initial letters are capitalized. In case there are two or more nouns in the class name, the initial letters are capitalized.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

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

SFS: Stock Follow Up System

SAS: Satin 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

2

PACKAGES

2.1 PACKAGE DIAGRAM

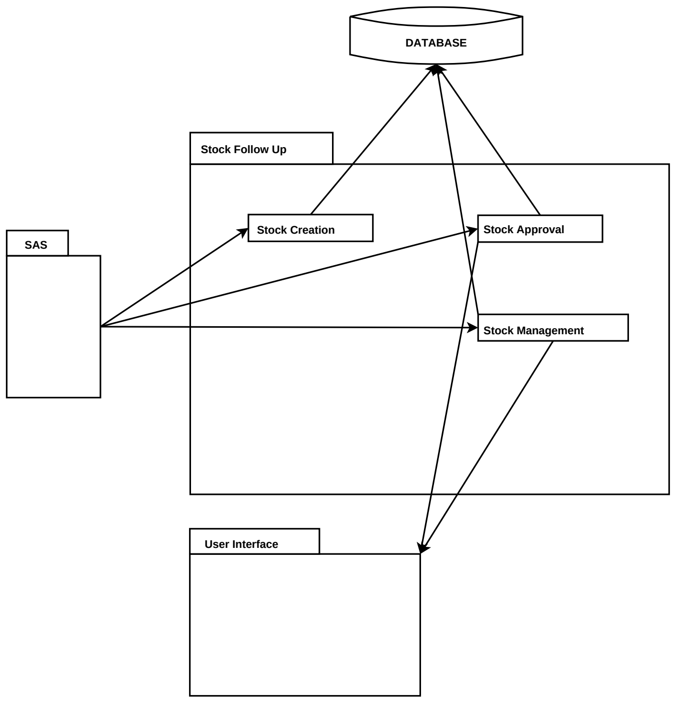


Figure 2.1: PACKAGE DIAGRAM of STOCK FOLLOW UP SYSTEM

2.2 PACKAGE DEFINITIONS

Two different packages are used in this system. One is named StockFollowUp. It is used for the internal structure. This package includes the classes which participate in the internal execution.

2.2.1 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. The definitons for these classes are given in the class definitions section.

2.3 CLASS DIAGRAM

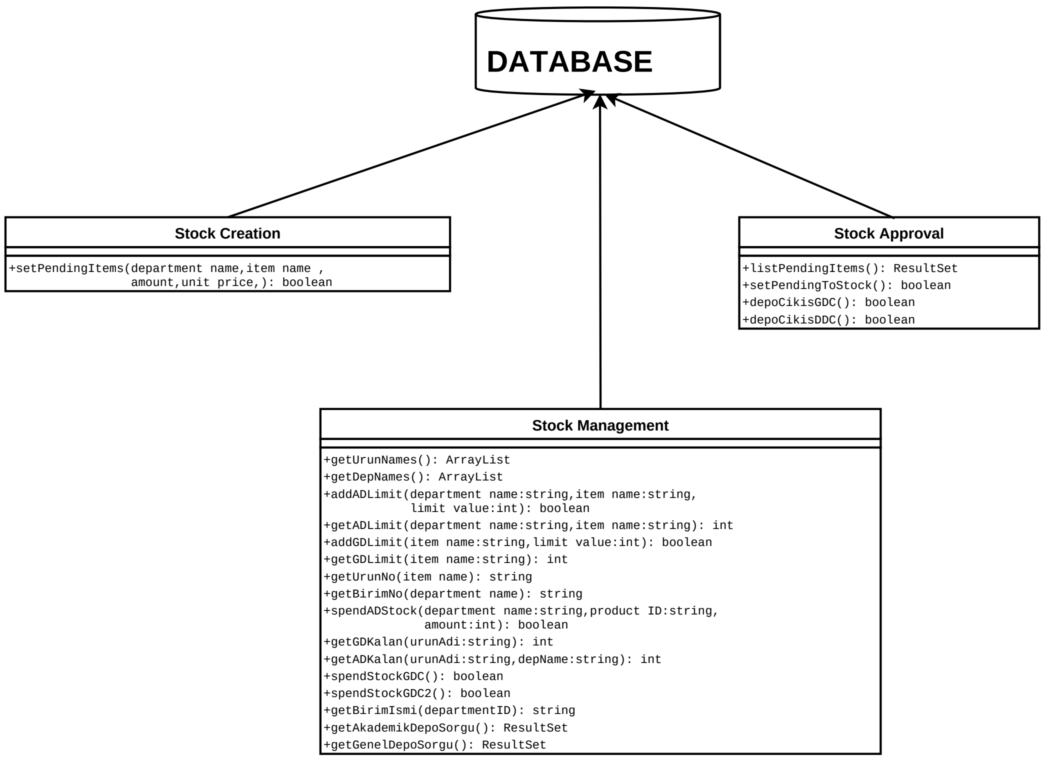

DATABASE
Stock Creation
+setPendingItems(department name,item name , amount,unit price,): boolean
Stock Approval
+listPendingItems(): ResultSet
+setPendingToStock(): boolean
+depoCikisGDC(): boolean
+depoCikisDDC(): boolean
Stock Management
+getUrunNames(): ArrayList
+getDepNames(): ArrayList
+addADLimit(department name:string,item name:string, limit value:int): boolean
+getADLimit(department name:string,item name:string): int
+addGDLimit(item name:string,limit value:int): boolean
+getGDLimit(item name:string): int
+getUrunNo(item name): string
+getBirimNo(department name): string
+spendADStock(department name:string,product ID:string, amount:int): boolean
+getGDKalan(urunAdi:string): int
+getADKalan(urunAdi:string,depName:string): int
+spendStockGDC(): boolean
+spendStockGDC2(): boolean
+getBirimIsmi(departmentID): string
+getAkademikDepoSorgu(): ResultSet
+getGenelDepoSorgu(): ResultSet

Figure 2.2: CLASS DIAGRAM

2.4 CLASS DEFINITIONS

2.4.1 Stock Creation Class

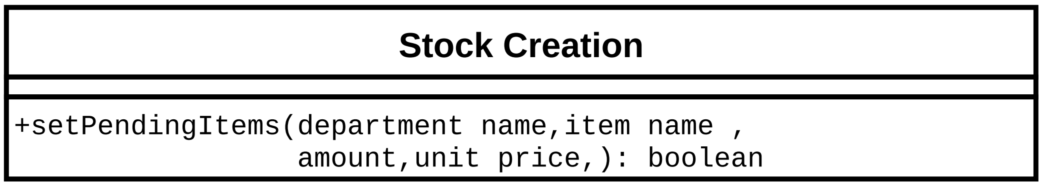

Stock Creation
+setPendingItems(department name,item name ,
amount,unit price,): boolean

Figure 2.3: Stock Creation Class

Stock Creation class is a part of the package StockFollowUp. It is used by SAS system in order to add a newly purchased item to the stock and it is used every time a purchase takes place.

The method boolean setPendingItems() is the one called after the purchase of an item and it puts the new item into the pending items table for approval of the GDC. It gets the department name, name of the item, amount of item and unit price of the item.

2.4.2 Stock Approval Class

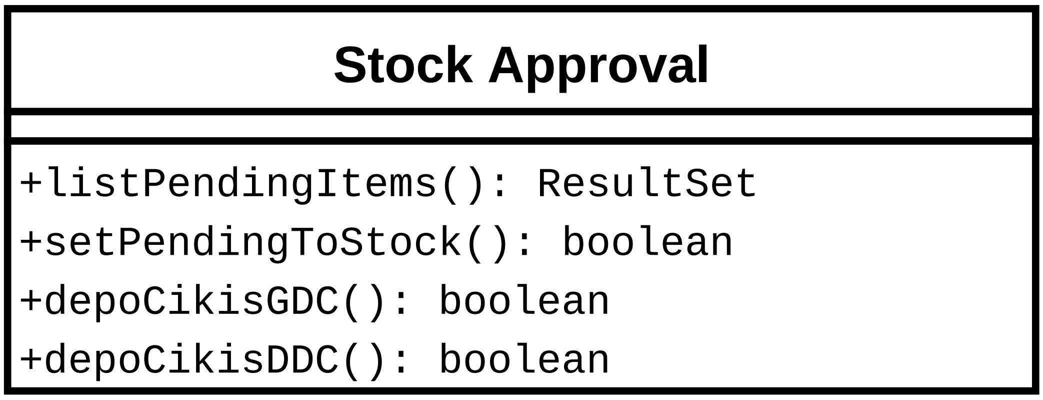


Figure 2.4: Stock Approval Class

Stock Approval Class is a part of the package StockFollowUp. The methods that it contains are listPendingItems(), setPendingToStock(), depoCikisGDC() and depoCikisDDC(). The methods of this class are generally used for approving the entrance and departures of the items to different depots.

The method ResultSet listPendingItems() returns the GDC the items that are waiting for him to approve in order to get into the stock. The method does not get any parameters and it is trivial for GDC to do his job right.

The method boolean setPendingToStock() gets several information of an item and approves these items in order to go into the stock. If the item is for an academic department, then the item is directly taken to the depot of the department,other wise it stays in the general stock or goes to the small virtual administrative departments.

The method boolean depoCikisGDC() takes the information for an item and removes it from the general depot while adding this information to the table that holds the departed items.

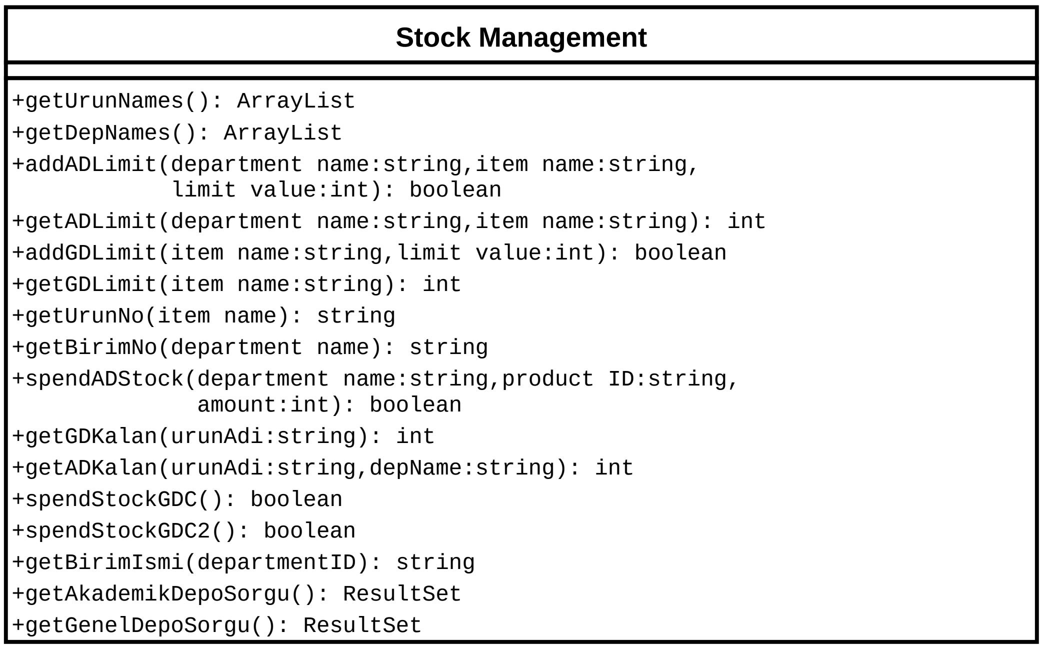
The method boolean depoCikisDDC() takes the information for an item and removes it from the department depot while adding this information to the table that holds the departed items.

2.4.3 Stock Management Class

Stock Management Class is a part of the package StockFollowUp.It serves for managing the stock data already 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. The names of the methods are getUrunNames(), getDepNames(), addADLimit(), getADLimit(), addGDLimit(), getGDLimit(), getUrunNo(), getBirimNo(), spendADStock(), getGDKalan(), getADKalan(), spendStockGDC(), spendStockGDC2(), getBirimIsmi(), getAkademikDepoSorgu() and getGenelDepoSorgu().

The method ArrayList getUrunNames() returns the names of all the items in a string array and it is trivial for the user interface so that the user can see the items when he manages the stock. For example, if user wants to see the number of pens in the stock, we must give him the necessary interface so that he represents his need appropriately.

The method ArrayList getDepNames() is similar to the previous method but it brings out the names of the departments instead of items. This method



```
Stock Management

+getUrunNames(): ArrayList
+getDepNames(): ArrayList
+addADLimit(department name:string,item name:string,
            limit value:int): boolean
+getADLimit(department name:string,item name:string): int
+addGDLimit(item name:string,limit value:int): boolean
+getGDLimit(item name:string): int
+getUrunNo(item name): string
+getBirimNo(department name): string
+spendADStock(department name:string,product ID:string,
              amount:int): boolean
+getGDKalan(urunAdi:string): int
+getADKalan(urunAdi:string,depName:string): int
+spendStockGDC(): boolean
+spendStockGDC2(): boolean
+getBirimIsmi(departmentID): string
+getAkademikDepoSorgu(): ResultSet
+getGenelDepoSorgu(): ResultSet
```

Figure 2.5: Stock Management Class

is also trivial for the user interface.

The method boolean addADLimit() assigns a minimum limit for an item in a department. Input parameters for this method are department name, item name and limit value. Department name and item name are of type String. The type of limit value is integer. If the operation is completed successfully, true is returned otherwise the method returns false.

The method int getADLimit() returns the value of minimum limit for an item in a particular department. It gets the department name and item name as input parameters. Department name and item name are of type String. Minimum limit of the given item for the given department is returned.

The method boolean addGDLimit() assigns a minimum limit for the given item in the general depot. It takes item name, limit value as input parameters. Item name is of type String and the type of limit value is integer. If the operation is completed successfully, true is returned otherwise the method returns false.

The method int getGDLimit() returns minimum limit of a given item in the general depot. It takes item name, which is of type String , as input parameter.

The method String getUrunNo() returns the product ID of a given item. Item name which is of type String is the input parameter.

The method String getBirimNo() returns the BirimNo of the given department. Department name is taken as input parameter and its type is String.

The method boolean spendADStock() decrements the amount of the given item of a particular department by the given amount. Department name , product ID and amount are input parameters of this method. Department name and product ID are of type String. Type of amount is integer. If the operation is completed successfully, true is returned otherwise the method returns false.

The method int getGDKalan() gets the "string urunAdi" as a parameter and returns the existing amount of the item in the general depot in an ArrayList data type.

The method int getADKalan() gets the "string urunAdi" and "string depName " as parameters and returns the existing amount of the item in the departmental depot in an ArrayList data type.

The method boolean spendStockGDC() decreases the amount of items, stored in the general depot, in case the user approved a pending item to send to the related department. Number of items should be decremented basing on UrunNo.

The method boolean spendStockGDC2() decreases the amount of items, stored in the general depot, in case the user approved a pending item to send to the related department. Number of items should be decremented basing on UrunAdi.

The method String getBirimIsmi() is needed to obtain department name. The method takes the departmentID as a parameter.

The method ResultSet getAkademikDepoSorgu() is used to see what is stored in academic depot. The method returns a dataset taken from the table "AkademikDepo" from the database.

The method ResultSet getGenelDepoSorgu() is used to see what is stored in general depot. The method returns a dataset taken from the table "GenelDepo" from the database.

3

Class Interfaces

The purpose of the Stock Follow Up module is 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 setPendingItems() provided to the SAS system gives the system the ability to add the items to stock pending list right after they are purchased. Other than that the stock followup module must be reachable from the SAS system according to user rights and the user interface must be the same as the SAS system.

Index