CMPE 450
SOFTWARE ENGINEERING
SYSTEM DESIGN DOCUMENT

Version 3.0

Project Name: Stock Follow Up
System Project

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

Submitted by: **Project Group 3**
Group

Design Group 3
Database Group 3
Requirements Group 3
Coding Group 3

Boğaziçi University, İstanbul
January 8th, 2007

Contents

List of Figures

1

Goals and Trade-Offs

The Stock Follow Up Project will be used by General Depot Clerk, Departmental Depot Clerk, managers of the departments, a general depot manager(GDM) from the strategy office, deans of faculties and the rector. Since these users have no experience in using the system, the system should have some properties that are important for the users of the system. These properties are the following:

1.1 User Friendliness

The system should have some properties that make it user-friendly. The graphical user interface should be easy to use and attractive. The system should provide a detailed and explanatory help section. The mistakes that may be made by the users should be handled. When a user makes a mistake, he should be alerted by an easy to understand message. The message should be helpful to use the system correctly. User-friendliness is one of the most important goals of the system.

1.2 Ease of Use

The Graphical User Interface (GUI) of the system should have meaningful labels that explain the usages of the text boxes, combo boxes and other components of the GUI. The help section may contribute to the ease of use by containing snapshots that explain various usages of the system. The users

should use the system easily so system should provide better functionality than calling, sending e-mails. Furthermore administrators should manage the system easily. Because of these reasons, the system must be easy to use.

1.3 Reliability

Reliability is a very important aspect for the system. The system will require a username and a password to be accessed. So, a non-user will not be able to access the system. Another security rule is that the users' access rights are limited. For example, the manager of a department will not be able to see the stored information belonging to another department. Another aspect of reliability is that the system will handle probable exceptions. The hardware will be sufficient to serve all the users that access the system at the same time.

1.4 High Performance

The system will be used on the campus' intranet. The intranet should be able to support all the connections from the users. The system will use a sufficient primary and secondary storage that will handle all the transactions. An additional server that will keep the database may be considered as a solution to increase the performance of the system. If we consider that 15-20 percent of the users can use the system at the same time, our aim would be having a system which has a high performance.

1.5 Minimum Number of Errors

The system should be error-free. The errors that can be faced during the usage of the system should be considered during the implementation. The necessary code that will handle the exceptional cases should be added. After the implementation of the system, the testing phase should be performed carefully. The system should be tested for different scenarios including the ones in which users make mistakes.

1.6 Security

Security is crucial for the system since important data about the university are stored and processed. It should be protected against unauthorized people. Also, the users should not be able to view or change the data that are related to their departments.

There are different types of users in the system and each user has different rights, so it is important to determine which users can reach which areas in the system. As a result, the security of the system must be provided without regard to cost issues.

1.7 Completeness of Functionality

The system should be able to perform the tasks that are stated in the RAD correctly. So, each module of the system is important and should be designed, implemented and tested carefully.

2

System Decomposition

2.1 System Decomposition

Stock follow up system consists of two subsystems. These subsystems are the stock approval subsystem, which serves the approval purposes of a new stock, and the stock management subsystem which serves the management purposes of existing stocks. Stock approval system is where new stocks enter the database and are ready for future usage. This system can only be used by the general depot clerk. Whenever a new stock, which was approved by the SAS system before, enters the General Depot, this stock is approved by the General Depot Clerk and is entered to the database.

Stock approval system has two main tasks. First of all, stocks that are pending and will arrive at the general depot can be shown using this system. If one of these stocks arrive at the general depot, it is entered to the database by using the approve function of the stock approval system.

Stock management system is where stocks that are already in the inventory are managed. It can be used by all clerks. It is used whenever there is a change in the status of a stock. It can also be used in order to see the current situation of the stock.

Stock management system has three main tasks. First of all, it can be used to inspect the current situation of the stock. When used by a depot clerk, it can show information about the depot of that clerk. When used by the general depot clerk, it can show all stock information. For inspecting purposes, several different queries can be made and the results can be ordered

as the user wishes. Second task of this system is setting a limit to an existing stock, which means setting a lower limit up to which stock can be used. If this limit is reached later on, no one can use that stock. Third task of this system is spending, which is used whenever a stock is spend. This task also gives a warning whenever the lower limit is reached.

2.1.1 Layers and partitions

Stock follow up system actually consists of one layer, which includes the approval system as well as the management system. However, since stock follow up system is integrated to the SAS system, SAS system is an upper layer for the stock follow up system. Users reach the stock follow up system by using the SAS system.

2.1.2 System topology

Stock follow up system can be represented by the following topology:

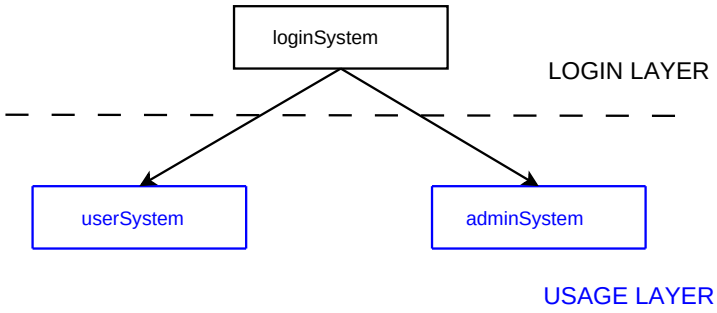

Figure 2.1: TOPOLOGY DIAGRAM

3

Concurrency Identification

3.1 Independency Between Objects in the Object Model

Since decreasing the dependency to the minimum among the subsystems of the StockFollowUpSystem Module is desired, the module is divided into several subsystems which are loginSystem, userSystem and adminSystem. These subsystems enable the system with the higher level of modularity.

Any access can be performed concurrently, since many of the operations performed by the subsystems require read-only access to the database and these operations do not bring any dependency between objects.

3.2 Identifiable Threads of Control

The users are able to access the program from different locations and the operations requested by these users are performed individually and separately. So there is not any usage of threads in the design of StockFollowUpSystem Module.

In addition, since the operations are atomic and do not require any usage of threads, the execution of these processes can be performed without problems.

3.3 User Accesses to the System

The StockFollowUpSystem Module will provide an interface for the users from the internet. Thus, it is inevitable that the program will be inherently multiuser.

First, the users trying to register to the system will be presented a login web page prepared and managed by UserInterface module. During login procedure, the password and the username of the user will be compared to the relevant data on the database. Since this process requires read-only access, there will be no problem encountered while the users from different locations are trying to access the system. Thus, this process can be performed simultaneously. On the other hand, if the operation, that is intended to be performed by the user , requires read-only access to the database, since this operation does not bring any dependency, it can be performed concurrently without causing any inconsistencies.

Secondly, after the registration procedure is completed, according to the type of the user (General Depot Clerk, Departmental Depot Clerk, managers of the departments, a general depot manager(GDM) from the strategy office, deans of faculties and the rector), the appropriate access rights will be given to the user while creating the session. According to the access rights of the user, the operations that the user has the right to perform will be active on the page prepared by the UserInterface module. As the user attempts to perform an operation using the system, while finalizing the process, it will possible for the data to be written on the database which requires a read-write access to the database. In that case the necessary fields of the database will be blocked and simultaneous access from different users will be denied.

On the other hand, if the operation, that is intended to be performed by the user, requires read-only access to the database, since this operation does not bring any dependency, it can be performed concurrently without causing any inconsistencies.

3.4 Availability of Multiple Queries in a Single Query

The user does not have to know a SQL-like language, by entering the criterions on the appropriate fields, the query is generated and executed. If there is entered no criterion on a field, this field is ignored and not included in the query. Therefore, this feature of the system enables the user create single or

multiple queries.

3.5 Parallel Handling of Queries by Different Subsystems

Since the subsystems are mostly independent of each other, queries can be handled in parallel between them. The only exception to this is the situation in which one of the subsystems tries to have a read-write access to one or more of the database tables while another one is trying to access. This situation will be handled by providing a locking mechanism on the table fields during updates, in order to ensure the consistency of the database.

3.6 Concurrency Scheme

The operations that are concurrent between the subsystems can be performed by the StockFollowUpSystem module since these subsystems are independent of each other except in the case of the read-write access to one or more of the tables in the database. In case of updating the tables, the necessary fields of the table will be blocked and concurrent access will be denied. In all other cases, subsystems will operate concurrently without causing any inconsistencies in the database.

4

Hardware / software Allocation

This project will have an Internet based platform for both the implementation and the functional access. As it has been mentioned earlier during the requirements analysis phase, the system will be dealing with three sort of users namely internal employees and managers and the administrator. All of these groups will access the system via Internet pages in different manners. The hardware allocation of the above described system consists of two main aspects, a closely related couple composed of client and server. The project will initially need two servers to handle the web page operations and database storage. It will be extremely important to separate these two tasks into two different servers in the case of a remarkable increase in user numbers. Doing this, the Web Server will handle the Internet connections and the Data Server will deal with the database operations. Besides the server aspect, the client part of the system will only be asked to have a PC with efficient Internet connection via modem, LAN or other protocols and the MS IE software installed on their machine. An oce environment in traditional company idea is not necessary. The design platform will be based on SQL server to handle secure multi-connection and Java programming to deal with excessive Internet based form operations.

4.1 System Performance

The system performance is based on some important measures: information retrieval speed, connection handling performance, data processing capability and memory usage efficiency.

4.1.1 General System Performance

The information retrieval should be as fast as possible for customer satisfaction. To ensure an efficient response time, the customer should be able to download a page in 5 seconds with a 33.6 Kbps modem. The form submission operations are required to be processed in a short time. Since the project is an online system, it will have multiple-users to access it. This multiple-user idea will definitely result in a heavy transaction and request traffic. The server should be able to serve 25 percent of registered customers simultaneously. Separation of connection handling and data processing over two different servers would definitely improve the overall system performance as well as the cost incurred.

4.1.2 Input / Output Performance

The project will basically have web forms and web pages as inputs and outputs. The system will not need special hardware for input/output purposes and the input/output performance will be again dependent on the overall system performance mentioned in the general performance measures part, so the response time, the information retrieval speed and the server abilities will play important roles to determine the input/output performance. Slow data retrieval because of poor programming must absolutely be discarded since the software runs on an online platform.

4.1.3 Processor Allocation

The system is not supposed to handle heavy arithmetic operations or long computations. It will be dealing with simple database queries at most, so it is not crucial for the system to manage a multi-processor environment up until to a certain noticeable database size.

4.1.4 Memory Allocation

The larger the primary memory, the faster the applications would run. As the system performance is basically related to the information retrieval speed and the response time, it would be preferable to have the servers with primary memories in the order of some GB's. Moreover, the size of the secondary storage should be sufficient for data swapping, recovery and backup procedures. An acceptable range for secondary storage should at least cover

some tens of GB's.

4.2 Connectivity and Network Architecture

The project has the feature to be online and global over the Internet and web pages as mentioned before. The only physical component of the project is a couple of connected servers managing the performance of the web page operations and the data storage issues. Admin, managers and staff will all access the servers via TCP-IP.

However, there are some special operations which can not be performed via internet connection. Among those special operations are the administrative operations such as adding new users to the system, changing passwords of users, etc. Those kinds of operations are performed by directly accessing the server computer.

5

Data Management

Our project needs to define database tables according to the requirements of The Stock Follow Up System while integrating it with existing project SAS. In order to achieve that, Stock Follow Up System uses database.

5.1 Necessity of Database

Database will be used to keep information about goods that came to the general depot and it is an extension to the last year's SAS system in this sense. This data will consists of type of the good, quantity of the good, its related department and other things that are specified in the Database Design Document. Each of these will be stored in a table. That will allow all system user access necessary information according to their permission.

5.2 Data Distribution

The data included by the GUI interface will be stored into the separated tables of the grand database.

5.3 Extensibility of the Database

During the improvement of the system, there could be some necessity changes on the database attributes such as creating new tables, adding different fields or changing data types of some fields.

5.4 Average Request Rate

Average request rate is expected to be high because the system will be used by several departments (such as purchasing, budget, etc.) and several users will use this system to retrieve or enter information. In the worst case, there may be requests for the same data set or several users may try to access to the same part of the system (such as the same page). So the system is designed for handling such cases.

5.5 Average Request Size

The size of a typical request is the size of the result set. It is not expected to be very large on the basis that tables do not include a large number of columns and most of the used operations in the database is to insert and to retrieve forms and not to insert all of the forms in the system.

5.6 Frequency of Database Access

Database is accessed very often by most of the departments in the university.

5.7 Query Format and Interface

The query format of the database is SQL, but this is strictly kept away from the end users by the help of the user friendly graphical user interface. Queries are used by filling text boxes, clicking mouse on check boxes and combo boxes. Actually there are 4 interfaces for each type of users according to their authentication. The first type of users General Depot Clerk, whose interface can reach only the table of bought items, table of pending items, table of item in the general depot, table of items that are being asked for delivery. Second type of user is academic depot clerks who can reach the table of item in the depot, table of item that are being asked for delivery. The third user type is department managers. They can reach the table of items that are in their own department depot. Last user type is Rector who can reach all tables in the database.

5.8 Usage of Hidden Location

The STS module keeps private information about items that are bought and used in all departments of the university. Unauthorized access to private data can lead to unwanted consequences. Because of this reason location of the databases should be hidden.

5.9 Usage of Relational Database

The database is selected to be relational since it is based on relational algebra. Data is presented as 2-dimensional tables. Tables have a specific number of columns and arbitrary numbers of rows to dynamically keep records.

Relational database allows uniquely identifying a row in a table with primary keys and reference to another primary key in another table (foreign key). Also SQL can be used since it is the standard language defining and manipulating tables in relational databases.

5.10 Selection of Archival Data

FaturaNo, FaturaTarihi, TalepEdenGrupNo, TalepEdenBirimNo, TalepEdenUstBirimNo, FaturaTarihi, Miktar, Aciklama and bosmu specify the item pending for approval, are archived in BEKLEYEN_URUNLER table.

DepoGirisNo, UrunNo, TalepEdenBirimNo, TalepEdenGrupNo, TalepEdenUstBirimNo, RafNo, Miktar, FaturaNo, FaturaTarihi and Onaylandimi are archived in GENEL_DEPO_GIRIS table that stores the items has been stored in the general depot.

GDepoHarcamaNo, UrunNo, TalepEdenGrupNo, TalepEdenBirimNo, TalepEdenUstBirimNo, RafNo, Miktar, HarcamaTarihi and Aciklama are archived in GENEL_DEPO_HARCAMA table that stores the items delivered from the general depot.

GrupNo, BirimNo, UstBirimNo, DepoGirisNo, UrunNo, Miktar, FaturaNo, FaturaTarihi and Aciklama are achieved in the BIRIM_DEPO_GIRIS table that stores the item that are given to each department depot.

GrupNo, BDepoHarcamaNo, UstBirimNo, BirimNo, UrunNo, Miktar, Har-

camaTarihi and Aciklama are archived in the BIRIM_DEPO_HARCAMA table that stores the items that are delivered from the each department depot.

UrunNo, AltLimit and Mevcut are archived in GENEL_DEPO_URUN_MIKTAR table that stores the limit and current amount of each item in the general depot.

RafNo, Aciklama and bosmu are archived in GENEL_DEPO_RAF table in order to store the each item shelf no in the depot.

GrupNo, BirimNo, UstBirimNo, UrunNo, AltLimit, Mevcut and Aciklama are archived in BIRIM_DEPO_URUN_MIKTAR table that stores the each item with the minimum limit and current amount.

5.11 Password Encryption

One of the extensibility issue we encountered during the development process is the encryption of the user passwords that are stored in the "Kullanici" table of the grant database. Encapsulation of such an important data is necessary because of the security issues. We should not permit this since almost all of the professional softwares do not permit even the administrators to see the passwords of users. While we did not actually add this functionality to the project because of the practical problems it would lead in the development process we find it necessary to add to the design document as an extension to the SAS system.

There were two problems we encountered while applying the encryption to the passwords. First one is the method of encryption and second one is where to put it. We choose MD5 encryption since it is a common method and a one way encryption schema which is almost unbreakable and so suitable in our case. Second choice was to implement encryption in the database which has same purposes as using stock procedures in the system and also because of finding the security of the database more important than the network security. We thought that network intrusions into the communication between the client and the server in the campus intranet are too less probable to consider.

There are two processes to change in the database for this. First one is the user login process and second one is the password change process. For the user login process "KullaniciGiris_sp" is triggered in the database. Plain password is sent to the database and checked with the encrypted password

in the "Kullanici" table. So we have to use "MD5()" function in the SQL syntax inside the "KullaniciGiris_sp" to encrypt the plain password and to check with the encrypted password.

For the second process a new stored procedure have to be written which will encrypt the send string from the jsp form. Since the user will write his/her old password and the requested new password, both have to be encrypted. First the old plain password in the form will be encrypted using the new stored procedure and checked. Then again using that stored procedure the new password will be encrypted and replaced with the old one using "KullaniciSifreGuncelle_sp". Having done these successfully the old password will be replaced by the encrypted new one.

Another issue that will arise while applying the above procedure to the working database will be the plain passwords in the "Kullanici" table which will cause login problem to the system. This can be handled either by adding the users one by one with encrypted passwords by using "KullaniciEkle_sp" or adding "MD5()" function to the passwords of users in the overall SQL script and rerunning it.

6

Global Resource Handling

The project will use limited resources. So, there will be some constraints on the usage of global resources. The usage of global resources includes memory management, database management and authentication and security issues. The following are the details about these resources and their usage.

6.1 Memory Management

The system will provide service to many people from several departments. These people may produce huge amounts of data. The primary and secondary storage used for the system should be large enough to handle the transaction data. The system may be designed and implemented in such a way that the space complexity is decreased. Usage of an additional server may be a solution too.

6.2 Database Management

The database will include tables for the users (General Depot Clerk, Departmental Depot Clerk, managers of the departments, a general depot manager (GDM) from the strategy office, deans of faculties and the rector), goods and allowable input values. Some of the users have authentication to make changes on the database. The employees will store new data about transactions and register to the system. The administrator will approve the registrations and modify the allowable input values.

The users of the system will be able to view the data according to their authority. A back up operation is crucial for the management of the database.

The database should be backed up after the last transaction of a day.

6.3 Authentication and Security Issues

It is very important for the security of the system that a non-user of the system should not login the system, view or modify the stored data. The registered users can only login the system by providing their usernames and passwords. There are also restrictions on the registered users. There are four types of users and each type of users can access different parts of the system and perform different tasks.

7

Software Control Implementation

7.1 External Control Flow

Control flow of the stock follow-up system is defined like web application. For a single user we have the certain structure of branched web pages. Although the login functionality of the SAS is pretty linear, after entrance to the stock follow-up system; we have tree shaped structure running freely according to user commands.

7.2 Concurrent Control

According to the nature of the system, many users can use the stock follow up system through the SAS system, any user can access a single page at a time. This will protect the data integrity of the system as different actions on different pages might effect the database in an unstable manner.

7.3 Internal Control

User request and web page forms implements the process control. This structure makes the designed procedures to be simple and mostly linear.The run of a single user may bring up different pages and different methods might be called by the different classes of the system.

7.4 User Interface

The user interface of the system will be constructed through web pages. This interface enables the user manage the Stock Follow Up System. Using the appropriate fields and buttons of the interface, management is done. The control of the next state of the interface is up to the action taken by the user. According to the type of the operation that is done on the system, status of the interface is changed and the appropriate data are displayed.

8

Boundary Conditions

8.1 Initialization

8.1.1 Dynamic Model of the System Startup

The STS module is an event-driven program working on the web server. For the STS module in order to start working properly, the program should be installed on the Web server and the database should be installed on the Data Server.

8.1.2 Description of Data Accessed at Startup

Since the system has operations related with the time and date, these should be gathered at the startup. This is necessary for the transactions table.

Also "KULLANICI" table must be available at the start up in case of a new login request. This is particularly important for the robustness and accuracy of the program.

8.1.3 User Interface at Startup

User interface is created by web pages which are specialized according to different types of the user. The system design for the user interface imposes different user interfaces for example for an akademik personel or a department

manager or Rector as they have different rights. Therefore; in KULLANICI table there is a"OzelAnahtarKullaniyor" field. This is a special key. Also statu is another key. If statu is 1 user is active else user is inactive. And users privileges are defined in "KULLANICI SAYFA YETKISI MN" table. Every user has different right. According to their right they can access any area or not.

8.1.4 Presentation of System to the User

The system should be user-friendly to increase its functionality and to attract the users. It should also have a simple interface to simplify the usage of the program and to make it more understandable for the users.

In our system at home page there is an help section "Yardim". Any user which want to use our system can learn everything from this section. Our help section is written in Turkish language and have section like :

- Baslangic
- Sisteme Giris
  - Sisteme Giris
  - Hatali Giris
  - D.Imza Appleti Guvenlik Uyarisi
- Onay Bekleyen Istekler
  - Onay Bekleyen Istekler
  - Detay Gor. (Dogrudan Alim)
  - Detay Goruntuleme (Satin Alim)
- Guvenlik
  - Ozel Imza Anahtarini Yukleme
- Dekanliklar
  - Dekan Sayfasi
  - Dekanlik Sekreteri Sayfasi
- Bolumler

  - Bolum Baskani Sayfasi
  - Bolum Sekreteri Sayfasi
- Destek Birimi
  - Destek Birimi Sayfasi
  - Destek Birimi Muduru Sayfasi
- Ihaleler
  - Ihale Rapor Ekrani
  - Ihale Teklif Detay Gor.
    - Ihale Teklif Detay Goruntuleme
    - Ihale Detay
  - Ihale Yonetimi
    - Ihale Yonetimi
    - Ihale Olusturma
    - Acilmis Ihalelerin Yonetimi
    - Teklif Alinacak Ihalelerin Yonetimi
    - Yeni Teklif Hazirlama
    - Teklifi Gorme Guncelleme
    - Sonuclanmis Ihalelerin Yon.
    - Ihale Detayi Goruntuleme
- Butce
  - Butce Anasayfa
  - Detay Goruntuleme
  - Butce Baskani Kullanicisi
    - Butce Baskani Sayfasi
    - Detay Goruntuleme
  - Kurumsal Kod Yonetimi
    - Kurumsal Kod Yonetimi
    - Yeni Kurumsal Kod Atanmasi
  - Butce Kod Yonetimi
    - Butce Kod Yonetimi
    - Yeni Butce Tipi Tanimlama

        * Yeni Butce Dagilimi Girisi
    - Butce Dagilimi Yonetimi
    - Birimlere Butce Atanmasi
- Depo
    - Rektor
    - Dekan
    - Bolum Baskani
    - Genel Depo Sorumlusu
        * Alt Limit Belirleme
        * Alt Limit Uyari
        * Depo Sorgulama
        * Urun Cikisi
        * Urun Girisi
        * Urun Gonderimi
    - Bolum Depo Yoneticisi

8.2 Termination

The Stock Flow up System is an event-driven program working on the web server without explicit startup and termination. When user done with the system simply can logout from the system "Sistemden cikis".

8.3 Failure

8.3.1 Behavior of System in Failures

The server will be located at Purchase Department but the users won't have direct access permission to the server so we do not need a hidden location. The only access way to the database will be our Stock Follow up System and it provides abstraction and restricted access. In the case of network or system failure it will be directly accessed by the administrator.

8.3.2 Availability of Backup Communication Links

The SAS module does not deal with the most vital pieces of information and the operations do not have as much urgency. Therefore, no additional backup communication links will be provided. Backups will be taken on regular basis, in the least crowded time of the process, and the downtime will not effect the whole system considerably.

8.3.3 Recovery From Failure

Recovery from failure will be handled by the Database Management System, which is MySQL. The routines for recovery will be handled by the DBMS and no special handling will be performed by the SAS module.

8.3.4 Differences Between Recovery and Initialization

Recovery is different from initialization, such that after recovery, the consistency of the database has to be checked using the synchronization points in time and logs of database operations, whereas database can immediately start functioning after startup. This control over the status of the database is performed by the DBMS, MySQL.

9

Design Rationale

The Stock Follow up System is an extension of SAS; therefore, the main design issue was to establish integrity of two systems. The development environment and structure of the database is reconsidered according to old project SAS.

The Stock Follow up System had to use the database of SAS and had to be called by SAS. Not to have a congruity problem, we designed a new class to call the Stock Follow up System, instead of writing a new trigger, which would red a function to make a call to the Stock Follow up System whenever a new data is stored on Stock table. If we had changed the structure of the database, adding new parameters or extending the value range of existing parameters, it would cause problems in SAS.

The previous coding group of SAS used Java Server Pages (JSP), Netbeans and Eclipse. Our coding group also uses JSP to establish a congruity with SAS although some of the coders in group are not familiar to JSP, however, it is not a big problem since various sources are available for JSP on internet. Our GUI group uses HTML as coding language since it is easy to learn and clear to everyone. While being simple, the user interface did not turn into something boring. Its design is enriched by using ash applications in the menu. JSP allows software developers to generate HTML documents dynamically in response to a Web client request. JSP also allows embedding certain pre-defined actions into static content. Therefore, JSP language is more appealing to use with HTML. By designing new user interfaces, GUI Group tried to apply the same design principles with SAS group because the Stock Follow up System and SAS would constitute an integrated system,

therefore their user interfaces should have the same designing style. As for the object design, classes and their methods are designed basing on user rights and their abilities. SAS calls the Stock Follow up System according to user's will. When user needs to use the Stock Follow up System, the current user's rights are provided for the Stock Follow up System by SAS. The Stock Follow up System can be discussed in terms of scalability, extensibility and modifiability of the design.

As for the scalability, the system can be easily extended and can be used in different departments. The Stock Follow up System will be used by purchasing department, managers (rector and administer) and others. The definition of "other users" can be extended if a new department would be added to the system. In terms of networking, the system will not allow additional communications links such as wireless communication because there is no need to access the data from outside the department building. The data should not be accessed from outside as it stores the accounting information of the university.

As for the extensibility, the system can be extended by adding new users. This system is going to be used for a long time; therefore, it may need an extra database. A new database can be added easily and new data types as well as a response to changing technology.

As for the modifiability, software does not affect the stability of hardware platform. The system does not require any complex calculations which would exhaust operating system. However, the users may log into the system and require data at the same time which may lock the system due to the extension of number of operations which can be done at the same time. However, our customer wants only twenty-five users to log on at a time at most, which will not cause such a capability problem. In future, the system would pursue to satisfy the customer will as a dramatic increase in number of workers is not expected.

Index