



UNIVERSITI
TEKNOLOGI
PETRONAS

FINAL YEAR PROJECT (PART 2)

STB 5034

JANUARY 2007 SEMESTER

Name : KHAIRIL ISMADY BIN ISMAIL
Student ID : 4432
Course : Business Information Systems
Supervisor : Mr. A. Izzudin Zainal Abidin
Title : User Access Control Management System

Table of Content

Chapter 1	Introduction	
1.1	Abstract	1
1.2	Problem Statement	1
1.3	Objective of Project	2
1.4	Scope of Study	3
Chapter 2	Literature Review	
2.1	Implementation of Hierarchical Authorization For A Web Based Digital Library	4
2.2	Why Internet-Based Software Is The Future Of Corporate Computing	5
2.3	Rapid Application Development	6
2.4	User Interface Design	7
2.5	Ten Usability Heuristics	8
2.6	User Interface Design Tips and Technique	10
2.7	Color Test	12
Chapter 3	Methodology	
3.1	Procedure Identification	13
3.1.1	Research	13
3.1.2	Development Methodology	13
3.1.2.1	Planning	14
3.1.2.2	Analysis	14
3.1.2.3	Design	15
3.1.2.4	Development and Implementation	15
3.2	Tools Required	
3.2.1	Software	15
3.2.2	Hardware	15
3.3	System Architecture Design	16

3.4	UML Diagram	18
3.5	Database Design	26
3.6	Interface Design	27
Chapter 4	Result And Discussion	
4.1	Results	35
4.2	Discussion	38
4.3	Difficulties Faced	40
4.4	Project Scenario	46
Chapter 5	Conclusion	
5.1	Conclusion	47
5.2	Reference	48

Chapter 1: Introduction

The User Access Control Management System is created to assist the intranet's administrator in limiting or expanding the user access to other system. This system has all the necessary data that are maintained by a higher access of user. It even maintained data up to the specified level such as the page data and function data on a page. By administrating all of these data, the administrator will have a better control over each system. In each system, there are assign several group with different level of access to a system. With different level of access, each group can access to different sets of pages and functions in a system. Furthermore, user can view other user access rights, which means that it can display user access and group in the other system.

1.1 Abstract

Access control mechanisms are needed in almost every system nowadays to control what kind of access each user has to which resources and when. On the one hand access control systems need to be flexible to allow the definition of the access rules that are actually needed. But they must also be easy to administrate to prevent rules from being in place without the administrator realizing it. I present the implementation and architecture of a system that allows definition of access rights down to the single function and user level. I use hierarchies on users and groups, hierarchies on access rights and hierarchies on system page and function. These hierarchies allow a maximum of flexibility and still keep the system easy enough to administrate. User Access Control Management System supports positive as well as negative permissions.

1.2 Problem Statement

As an administrator or an IT administrator, I have analyzed if there are any potential problem when user is using applications or system inside the intranet. As I have experience the industrial internship in a factory (Komag(M) Sdn.) in Penang, there are a lot of applications and system that are inside the intranet. Basically, there are some problems that occurred regarding the user access in the intranet. As I can see, the applications and system inside the intranet does not have a central control. Basically each

application have its own control, furthermore if it is a later version, it does not has any control at all. In my understanding, in a factory, if any user can tampered with any information in the intranet, someday it will create a big threat to the company.

As the users are using the intranet, users have accesses to most of the system. There are some systems with an authentication control such as user need to enter user id and password to control user from accessing it. However this control may be included only on the recently develop system. In other situation, there is some system that should not be access by the user from different department. There are some system that user shouldn't have any access to so that administrator can control its private information. This is because, most of the system in the intranet are all a live system. These systems will actually effect the production in Komag. In each system, there is no limitation to users when they can access it. Each user can go to any page that they want to view and can use any function on the page. With this unlimited access, each user can actually effected productivity intentionally or unintentionally. As an example, in university intranet situation, if there a lot of applications and systems where users (students, lecturers, finance staffs, etc) have access to enter, then the administrator will have a problem to secure finance information from a lecture or other users.

1.2 Objectives of Project

The IT department has found some solution to the problem that they are analyzing. In their opinion, they need to create a user friendly interface where the administrator can actually control the access of each user to every system. By developing this access control they can actually control every user that want to access to a system.

Furthermore, this access control can actually control up to the page and the function level. By having this access control, each user will be group into several groups. This group will have different set of access to every system. By grouping the user, the administrator can manage and control easily and precisely.

The User Access Control Management System will be helping the administrator of a system to grant and limit an access to the system. With this access control, the administrator can have a better control over the users that are using the system.

In this system, each system will have few groups of users. Each group will have different set of access to a system. With these accesses, each group can only access some

parts of the system. By doing so, the other parts cannot be access due to the user may not need to use it.

This User Access Control Management System needs all the information about a system. This also includes the information about a System Page and also information about Functions on each page. By having all of this information, the system can maintain this information and manage each group more precisely.

In this system, the higher access user will have a better access. With this access, they can manage the other entire group in the system. They also can edit the access granted to the other group in a system. User will have a user friendly environment, where the layout of the page will be consistent, user can go to the entire system with few clicks.

1.4 Scope of Study

As for the part 1, I have done the proposal of the User Access Control Management System project. Inside this proposal I have explained a little bit about the system and the suggested software and hardware. Then I proceed with the preliminary report which I have been explaining about the project in more detail such as the problem statement, project objective, and the methodology. As in the progress report, I add few other parts such as project abstract, scope of study of the project and literature review which helps this project. Lastly in the Interim Report, I have done the introduction part where it contains problem statement, objective and few others, literature review part where I included all the research that I have done, methodology part where I explain the method I am using, the result and discussion part where I will show a suggested system architecture, the UML design of the system, the database design and the interface design of the system and finally the conclusion part where I conclude my interim report.

As for the part 2, in the implementing stage, I will create the whole system base on the design that I have created and will do some modification according to the current situation. After that, I will proceed to the testing part where I will test my system to make sure that it will function according to project needs. If there are any problem on the coding side, I can do some research on the internet to find solution to my problem. My second option is to ask the lecturer which has the expertise on the problem or I might ask my Supervisor, Mr. A. Izzudin Zainal Abidin for some advice.

Chapter 2: Literature Review and Theory

2.1 Implementation of Hierarchical Authorization For A Web Based Digital Library

In the [5] the authors have presented an access control system that is used for a hybrid library and other web-based applications. The major aim of the proposed system is high flexibility in the definition of permissions which is accomplished by introducing hierarchical relationships on the subjects, the privileges and the objects. Subjects are users or user groups and objects can be different types of resources identified by a combination of resource type and resource identifier. Subjects and objects are organized in a hierarchy of supergroups and subgroups by partial order relationships, whereas the partial order on the privileges expresses implication of a privilege by another. Authors have introduced the propagation rules for positive and negative permissions using a conflict resolution policy that always lets negative permissions prevail. Some motivating examples of the application of the hierarchies show how the flexibility can be exploited in practical situations. Authors have described the architecture and detailed implementation of the function for checking access. The system presented here is implemented on a central server.

As I finished analyzing this journal I find some similarities of my intended system with [5]. The similarities of this journal project and my project are that we are using the hierarchical authorization meaning a system containing few groups and in each group having several users. This also imply to the relation of System, Pages and Functions. With this journal, I find that I can manage a flexible access control over other system using my system. Furthermore inside the architecture, the author is emphasizing that the hierarchical Authorization was supposed to be implemented in the central server and I find that is very useful for my system.

2.2 Why Internet-Based Software Is The Future Of Corporate Computing

In [6] stated that, traditional desktop based software running or using your resources of your computer where it will used up all the space on your hard disk, the memory, the processor power etc. Your computer is a world unto itself. Not only that, the used of individual computer needs to do maintenance, upgrade, installations or even troubleshooting. We are not only talking about individual computers, but imagine a company that has many staff and many computers for each and everyone. Department of MIS then will have a headache when it comes to maintenance of the individual computers. In this case, organization has to purchase one copy of the software for every computer and include the upgrade of the operating system, of a new version of the software installed.

When we used the internet-based application, the program is access as a website. Normally, such application has the ability to connect with the database such as Microsoft Access, Oracle, Ms SQL server etc. The platform is the web browser itself. Each data is stored into the database and it is receive using the web browser, according to the user needs. The different between the traditional forms is the data centralized using the power of the technology. The database is stored at the remote location, the server instead of your individual computer, the local hard drive. You only require the web browser to view.

Internet-based application lowers the hard cost, long term cost, platform independence, location independence and up-to-date information and the information on the server can be protected.

2.3 Rapid Application Development

As explained in [7], businesses now operate in a global rapidly changing environment. They have to respond to new opportunities and markets, changing economic condition and the emerges of competing products and services. Software, application and system is part of almost all business operations so it is essential that new software is developed quickly and take advantage of new opportunities and to respond to competitive pressure. Rapid development and delivery is therefore now often the most critical requirement for software, application and system. In fact, many businesses nowadays are willing to trade of quality and compromise on requirements against rapid software, application and system delivery.

Rapid application development processes are designed to produce useful software quickly. Generally, there are iterative processes where specification, design, development and testing are interleaved. The application is not developed and deployed in its entirety but in a series of increments, with each increment including new system functionality. Although there are many approaches to rapid software development, they share some fundamental characteristics:

1. The processes of specification, design and implementation are concurrent. No detailed system specification. Design documentation is minimized.
2. System is developed in series of increment. End-users are involved in specifying and evaluating each increment. They may include changes or new requirements need to be implemented.

Advantages of rapid application development are:

1. Accelerated delivery of customer services. Early increments of the system can deliver high-priority functionality so that customers can get value from the system early in its development. Customer can evaluate and specify any requirements to be incorporated.
2. User engagement with the system. User of the system has to be involved in the incremental development process to evaluate and give feedback. This will shows that the system is meeting their requirements and the end-user also wants the system to work

2.4 User Interface Design

As in the [8], the author is emphasizing the importance of user interface design. Many technological innovations rely upon User Interface Design to elevate their technical complexity to a usable product. Technology alone may not win user acceptance and subsequent marketability. The User Experience, or how the user experiences the end product, is the key to acceptance. And that is where User Interface Design enters the design process. While product engineers focus on the technology, usability specialists focus on the user interface. For greatest efficiency and cost effectiveness, this working relationship should be maintained from the start of a project to its rollout.

When applied to computer software, User Interface Design is also known as Human-Computer Interaction or HCI. While people often think of Interface Design in terms of computers, it also refers to many products where the user interacts with controls or displays. Military aircraft, vehicles, airports, audio equipment, and computer peripherals, are a few products that extensively apply User Interface Design.

Optimized User Interface Design requires a systematic approach to the design process. But, to ensure optimum performance, Usability Testing is required. This empirical testing permits naïve users to provide data about what does work as anticipated and what does not work. Only after the resulting repairs are made can a product be deemed to have a user optimized interface.

The importance of good User Interface Design can be the difference between product acceptance and rejection in the marketplace. If end-users feel it is not easy to learn, not easy to use, or too cumbersome, an otherwise excellent product could fail. Good User Interface Design can make a product easy to understand and use, which results in greater user acceptance.

2.5 Ten Usability Heuristics

Below are the ten general principles for user interface design suggest by the author Jakob Nielsen in [9]. They are called "heuristics" because they are more in the nature of rules of thumb than specific usability guidelines.

1. Visibility of system status

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. User control and freedom

Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognition rather than recall

Minimize the user's memory load by making objects, actions, and options visible.

The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency of use

Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed.

Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

2.6 User Interface Design Tips and Technique.

There are a lot of tips and technique that can be consider in [10] because all of the tips and technique are generally needed to develop a good interface. However I only took some which I find very useful for me. There are:

1. **Consistency, consistency, consistency.** The most important thing that can possibly do is ensure the user interface works consistently. If double-clicking on items in one list and something will happen, then double-clicking on other item in other list should provide with the same function. Buttons should be place in consistent places on all windows, use the same wording in labels and messages, and use a consistent color scheme. Consistency in user interface enables users to build an accurate mental model of the way it works, and accurate mental models lead to lower training and support costs.
2. **Set standards and stick to them.** The only way to ensure consistency within application is to set user interface design standards, and then stick to them.
3. **Navigation between major user interface items is important.** If it is difficult to get from one screen to another, then users will quickly become frustrated and give up. When the flow between screens matches the flow of the work the user is trying to accomplish, then the application will make sense to users. Because different users work in different ways, the system needs to be flexible enough to support their various approaches.
4. **Word your messages and labels effectively.** The text display on screens is a primary source of information for users. If the text is worded poorly, then the interface will be perceived poorly by users. Using full words and sentences, as opposed to abbreviations and codes, makes text easier to understand. Messages should be worded positively, imply that the user is in control, and provide insight into how to use the application properly. For example, which message is more

appealing “You have input the wrong information” or “An account number should be eight digits in length.” Furthermore, messages should be worded consistently and displayed in a consistent place on the screen.

5. **Follow the contrast rule.** If color is used in the application, then all the screen should still be readable. The best way to do this is to follow the contrast rule: Use dark text on light backgrounds and light text on dark backgrounds. Reading blue text on a white background is easy, but reading blue text on a red background is difficult. The problem is not enough contrast exists between blue and red to make it easy to read, whereas there is a lot of contrast between blue and white.
6. **Align fields effectively.** When a screen has more than one editing field, the fields should be organized in a way that is both visually appealing and efficient. It is always found the best way to do so is to left-justify edit fields: in other words, make the left-hand side of each edit field line up in a straight line, one over the other. The corresponding labels should be right-justified and placed immediately beside the field. This is a clean and efficient way to organize the fields on a screen.
7. **Take measures to minimize errors.** Inside a system, there is a place where users need to input data such as the textbox or any other input box. Make sure that there are instructions so that user will easily know what kind of data should be input. Example Date is 2/23/2007 or 23rd February 2007.

2.7 Color Test

The author of [11] from Stephen F. Austin State University in Nacogdoches, Texas has conducted a test where it was assisted by the internet users. It was designed to measure the readability of various text/background color combinations. Below is a summary of the results of this survey and a discussion of the research project which was designed to study this topic further.

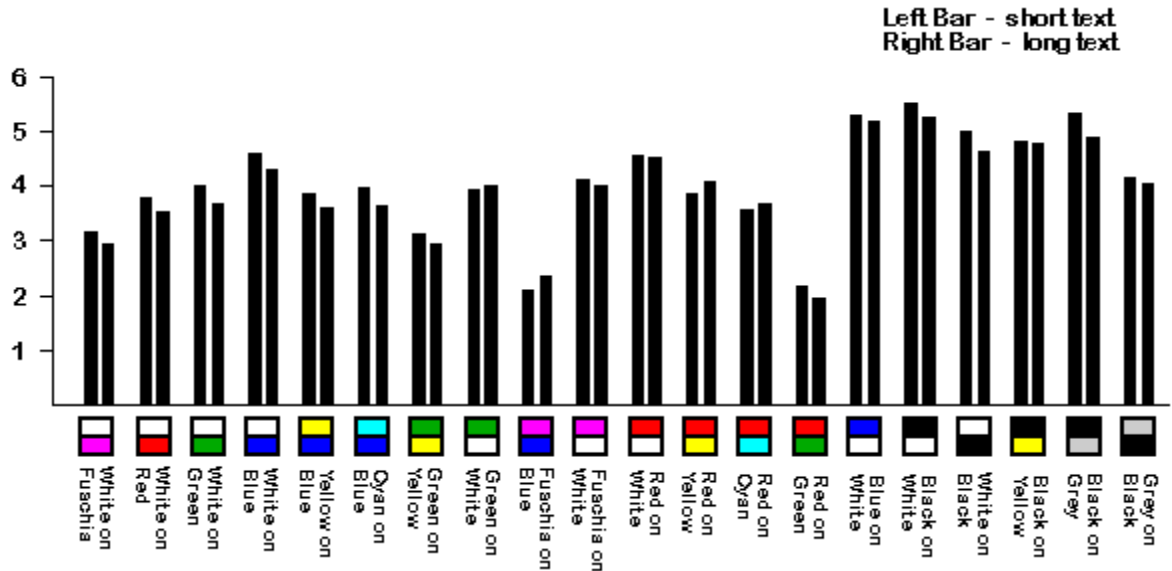


Figure 1 : Results of Color Test

Discussion of Results and Continuing Research

The survey of readability of different color combinations showed the following trends (see above figure). Not surprisingly, certain foreground/background color combinations were rated more highly than others.

As can be seen, the most readable color combination is black text on white background; overall, there is a stronger preference for any combination containing black. The two least readable combinations were red on green and fuchsia on blue. White on blue and red on yellow were ranked fairly high, while green on yellow and white on fuchsia were ranked fairly low. All others fell somewhere between these extremes.

Also, in every color combination surveyed, the darker text on a lighter background was rated more readable than its inverse (e.g. blue text on white background ranked higher than white text on blue background).

Chapter 3: Methodology

In this chapter, the methodology that shall be used for the project will be described briefly. In order to consider the satisfying of developing the system to the potential users and the timeframe of developing the system, I choose Rapid Application Development to develop the whole system.

3.1 Procedure Identification

3.1.1 Research

To collect information regarding this project, the followings approach were taken:

- Literature reviews – Finding out about suitable tools, system and architecture to incorporate the ideas proposed to the suitable development approaches
- Gather information from the IT department on how does they maintain all the access to a system or an application inside the intranet.
- Use the experience gain during the industrial internship to develop the User Access Control Management System.

3.1.2 Development Methodology

For the development of this project, RAD (Rapid Application Development) will be used considering the time frame given is quite short. The RAD category that best suits this project would be Iterative Development.

The iterative methodology breaks the overall system into a series of versions that are developed sequentially. The analysis phase identifies the overall concept, and the project team, users and system sponsor then categorize the requirement into a series of versions.

The basic idea behind iterative enhancement is to develop a software system incrementally, allowing developer to take advantage of what was being learned during the development

earlier, incremental, deliverable versions of the system. Key steps in the process were to start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving sequence of versions until the full system is implemented. At each iteration phase, design modifications are made and new functional capabilities are added.

3.1.2.1 Planning

In the planning stage, brainstorming about this project was done. The main objective of the system was outlined and how the system should be presented was determined at this stage. Once the system's objectives are clearly determined, the project plan is initiated. Project plan describes how the system will be developed and the list of tasks should be executed throughout the system development life cycle. The planning of this project is to be outline very carefully as it will determine whether the product will be able to be deliver at the end of the time frame given.

3.1.2.2 Analysis

For the analysis phase, research was done to find out more about the case being studied which is the user access to a certain application or system. Some researches were done in determining the feasibility of using Visual Basic.Net or ASP.Net latest version and Oracle or Microsoft SQL server to develop the proposed system. During this stage, problem statements of the current solution are collected to ensure the proposed system shall be able to overcome the limitation. All possible findings concerning the project are carefully reviewed to ensure the best solution is proposed.

3.1.2.3 Design

Design phase is also considered as one of the most challenging part in Software Development Life Cycle phase. In this phase, the design of the system must be carefully developed. This is to ensure that user's interactivity with the systems meet user's expectation. Every feature included will be reviewed and the functionality enhancement will be made during this phase.

3.1.2.4 Development and Implementation

This is the phase where the installation of proposed system takes place. After the installation, the prototype of the proposed system is constructed. The system is built and later tested to ensure it performs in conformance with the system objectives. For this system, test will be focusing more on the functionality assessment of the developed system. The random testing which involve small group of test user will be conducted to validate the functionality of the system and it is important to ensure the system developed is free from the bug or errors.

3.2 Tools Required

3.2.1 Software

- Visual Basic.Net / ASP.Net
- MySQL

3.2.2 Hardware

- Normal computers

3.3 System Architecture Design

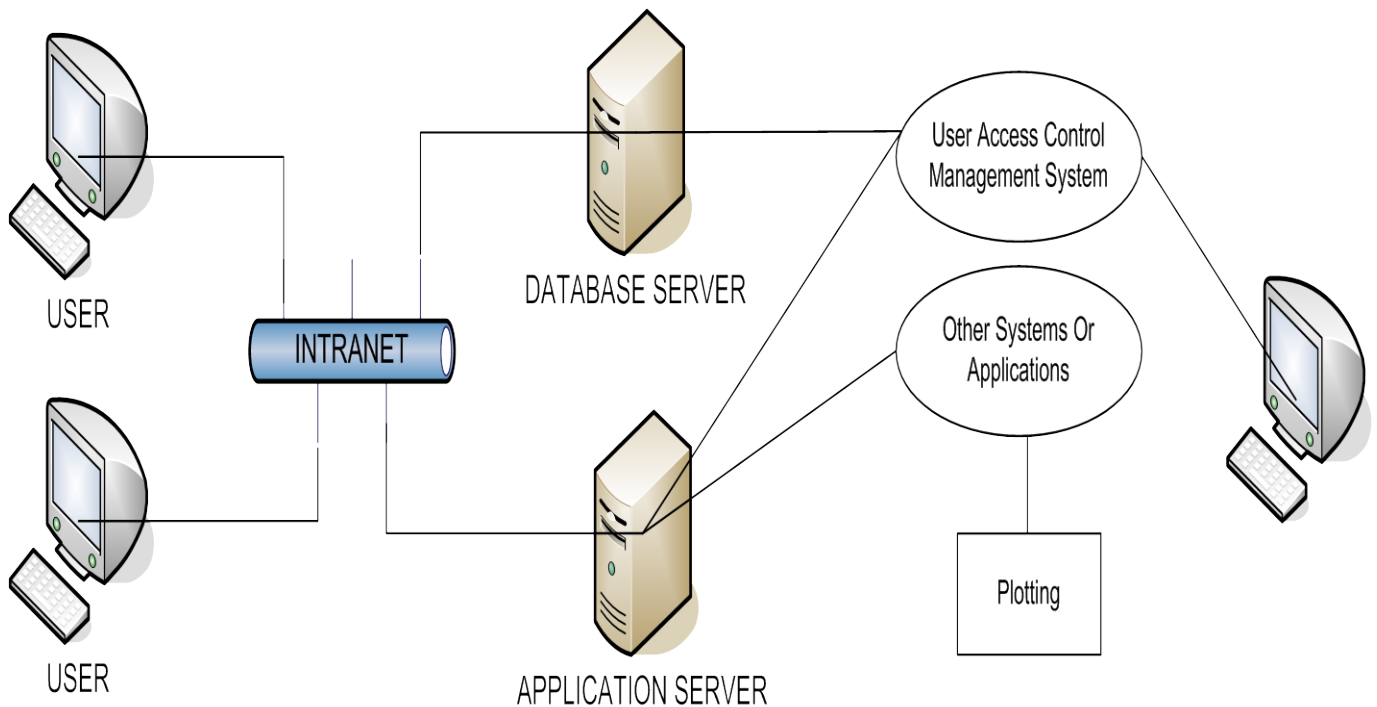


Figure 2: Web-Based User Access Control Management System

User Access Control Management System consists of 3 parts. First part is the Database part which is the most important part for the system. In this database is stored all the related information in the database server such as :-

- 1) Users information
- 2) User Group Information
- 3) System Information
- 4) Page Information
- 5) Function Information
- 6) User to Group Assignments Information
- 7) Page and Function to Group Assignments Information

Second Part is the Interface part which is located in the application server. In this part, there are several sections. One of the parts is the system section where user can choose to leave this system. Next section is the Setup section where User will do all the

setup for the User, Group, System, Page and Function. After all the information needed is setup, there is a section in the system interface where user can do maintenance on the created information. All the maintenance is such as edit and delete. Another section in the system is the Assignments section where user will assign users to a group in a system and user will also assign Page and Function to the group in a system. The final section is the Display section where user can actually choose some of the criteria information to be display. Below are all the section in the system:-

- 1) System
- 2) Setup
- 3) Maintenance
- 4) Display

The final part is the plotting where programmer will actually plot or add some information into the other system that wants to use this system as an Access Manager. In this part programmer will identify a system name, page name and the function name. Then all the exact name will be stored in the User Access Control Managements System database.

3.4 UML Diagram

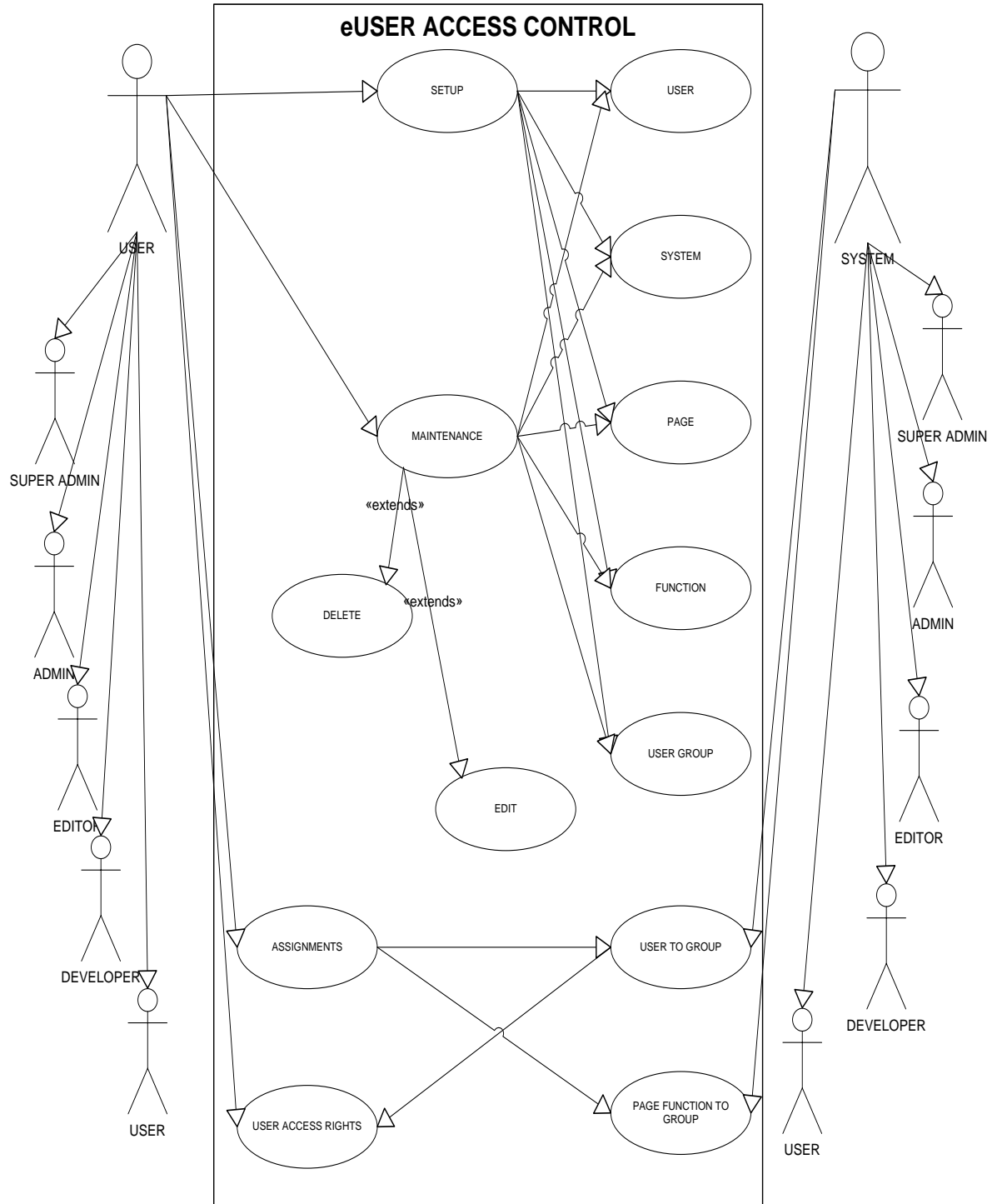


Figure 3 : Use Case Diagram for User Access Control System

Description

As shown in the Figure 1, it is a use case for the User Access Control System. As shown in the figure, there are two main actors that are involved directly inside the process. The first actor is the user and the second actor is the other system. Inside this use case there are few operations that are involved.

User can be group into several groups. This group is created by the higher user access or higher authority. This user will be using the User Access System directly. The other actor is the other system. Inside this system there are several group of user. This user is group according access level. Basically the groups are Administrator, Editor, Developer and User. Each group has own sets of access.

The actor User is the only actor that will be involve directly with the system. As for the actor, he can perform many operations on the system. However, when the user is group into several groups and automatically, each group will have different access to the system.

The user actor can directly use the setup section. In this setup section, user can actually setup user, setup system, setup page, setup function and setup user group. In this setup, user is required to enter all the information needed by the system.

Next, user can also use the maintenance section. In this section user can actually maintain all the data that user have setup. User can do the edit function and also the delete function on the data that they have setup.

Then user can use the assignment section where in this section user can assign or unassigned user to a system group and page function to a system group. In this section, user actually decides which group has the most access to a system.

Then in the user access rights section is where the user can actually search for another user and see their access rights to a system.

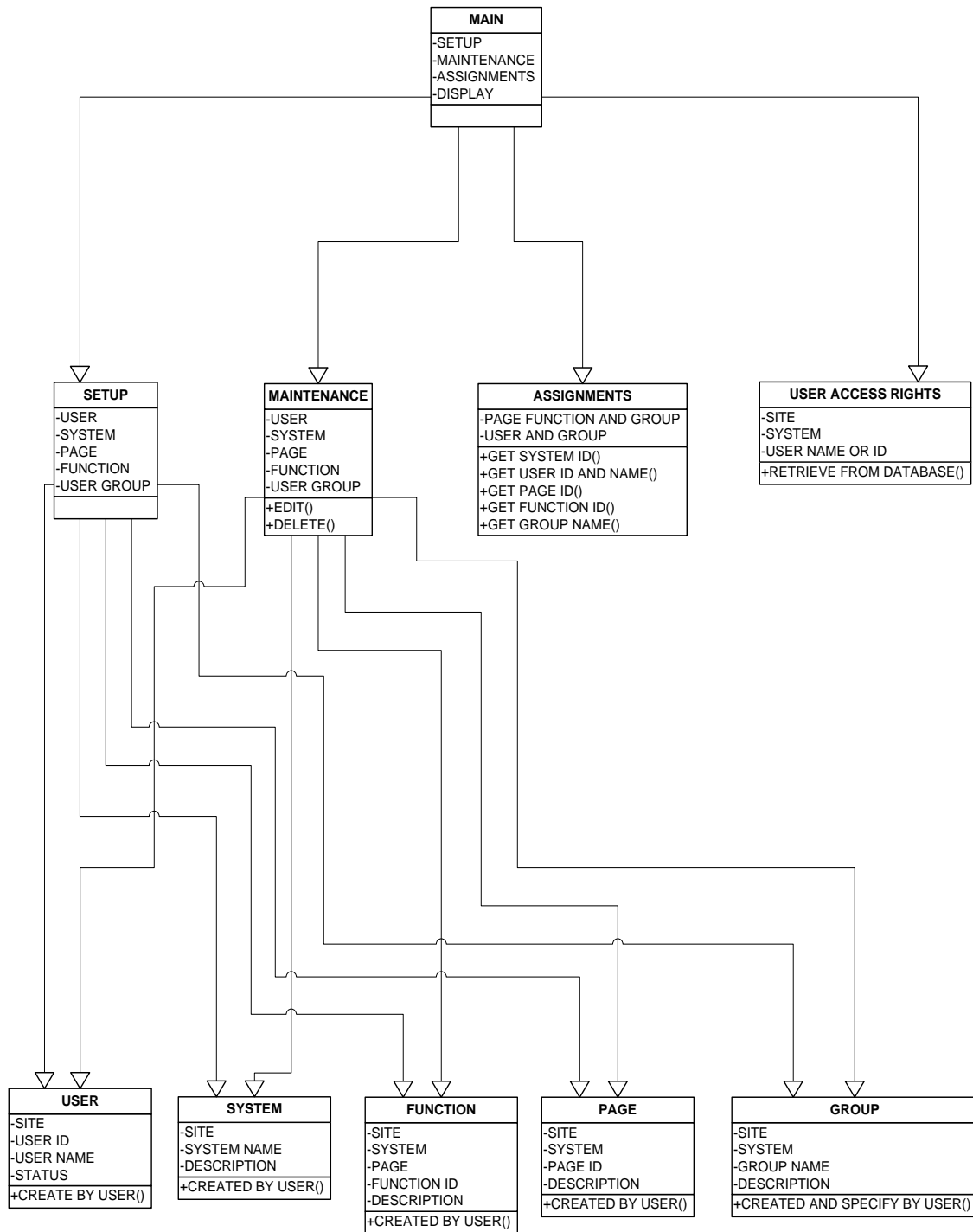


Figure 4 : Class Diagram for the User Access Control System

Description

As shown in the Figure 2, there are 10 objects that are in the class diagram. Firstly, there is the Main object. The Main object has several attributes. There are Setup, Maintenance, Assignments and Display. There is no operation in the Main Object.

Next is the Setup object. The Setup Object attributes are User, System, Page, Function and User Group. There is no operation inside the Main object.

Next is the Maintenance object. The Maintenance object attributes are User, System, Page, Function and User Group. The operations inside Maintenance object are Edit and Delete where user can edit or delete data in this object.

Next is the Assignments object. The Assignments object attributes are Page Function and Group and User and Group. The operation inside the Assignments object are Get User Id and Name, Get System Id, Get Page Id, Get Function Id and Get User Group Name where this object will get all of this info from the database.

Next is the User Access Rights object. The User Access Rights object attributes are Site, System, and User Name or Id. the object operation is Retrieve From Database() where the data to be display is retrieve from database.

Next is the User object. The User object attributes are Site, User ID, User Name, and Status. The operation inside the User object is Created By User() which means that the user created is created and maintained by user.

Next is the System object. The System object attributes are Site, System ID, and Description. The operation inside the System object is Created By User() which means that the system created is created and maintained by user.

Next is the Group object. The Group object attributes are Site, System, Group Name, and Description. The operation inside the Group object is Created By User() which means that the group created is created and maintained by user.

Next is the Page object. The Page object attributes are Site, System, Page Name, and Description. The operation inside the Page object is Created By User() which means that the page created is created and maintained by user.

Lastly is the Function object. The Function object attributes are Site, System ID, Page ID, Function ID, and Description. The operation inside the Function object is Created By User() which means that the page created is created and maintained by user.

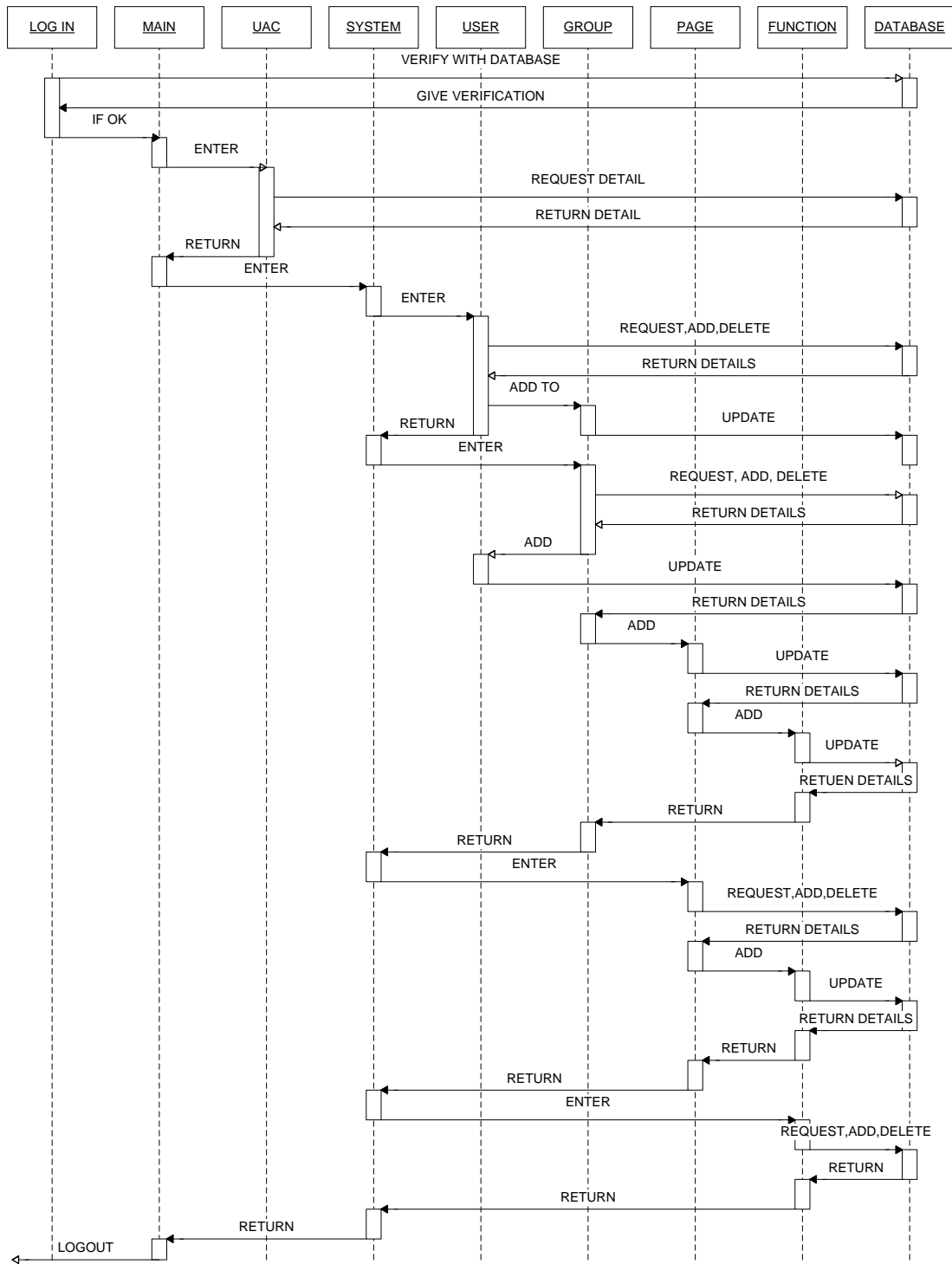


Figure 5 : Sequence Diagram for the User Access Control System

Description

In the Figure 3 show a sequence diagram of the User Access Control System. Firstly user needs to enter the Log In. In the Log In, user will need to enter an Employee Number and Password. Then the Log In will verify with the Database, and if the information matches, the user will be directed to Main. In Main, user can choose to enter the User Access Control (UAC) or the System. If user chooses to enter the UAC, then the system will display user access to any system by user providing the user name or the employee number. The system will retrieve the data from the database and display to user.

If user chooses to enter the system, user has to pick which system the user wants to enter. In the system, user can choose to add or delete a system in the User Access Control (UAC) System. Then the UAC system will update the database. Then user can choose to enter any system that the UAC system has.

In that particular system, user also can choose to enter the User (System User) or the Access Control Page. Inside this Access Control Page, user can choose to enter the Group, Page or the Function. In the User (System User), user can request the UAC system to display some or the entire user that are inside the particular system. User also can add, edit or delete users inside the particular system provided the user has a higher access. Then the UAC system will update the database and give a feedback to the user.

In the Group, user can request data for each group inside the particular system. In the group, user also can request, add or delete a user from a group. User also can specify which page and function should a group has an access to. Then the UAC system will update the database and notify the user.

In the Page, user can request the UAC system to display any page and all of its attributes. User also can add or delete a page from a particular system. Then the UAC system will update the database and notify the user.

In the Function, user can do as similar action in the Page where user can request the UAC system to display any function and all of its attributes. User also can add or delete a function from a particular system. Then the UAC system will update the database and notify the user. Finally user can logout from the UAC system.

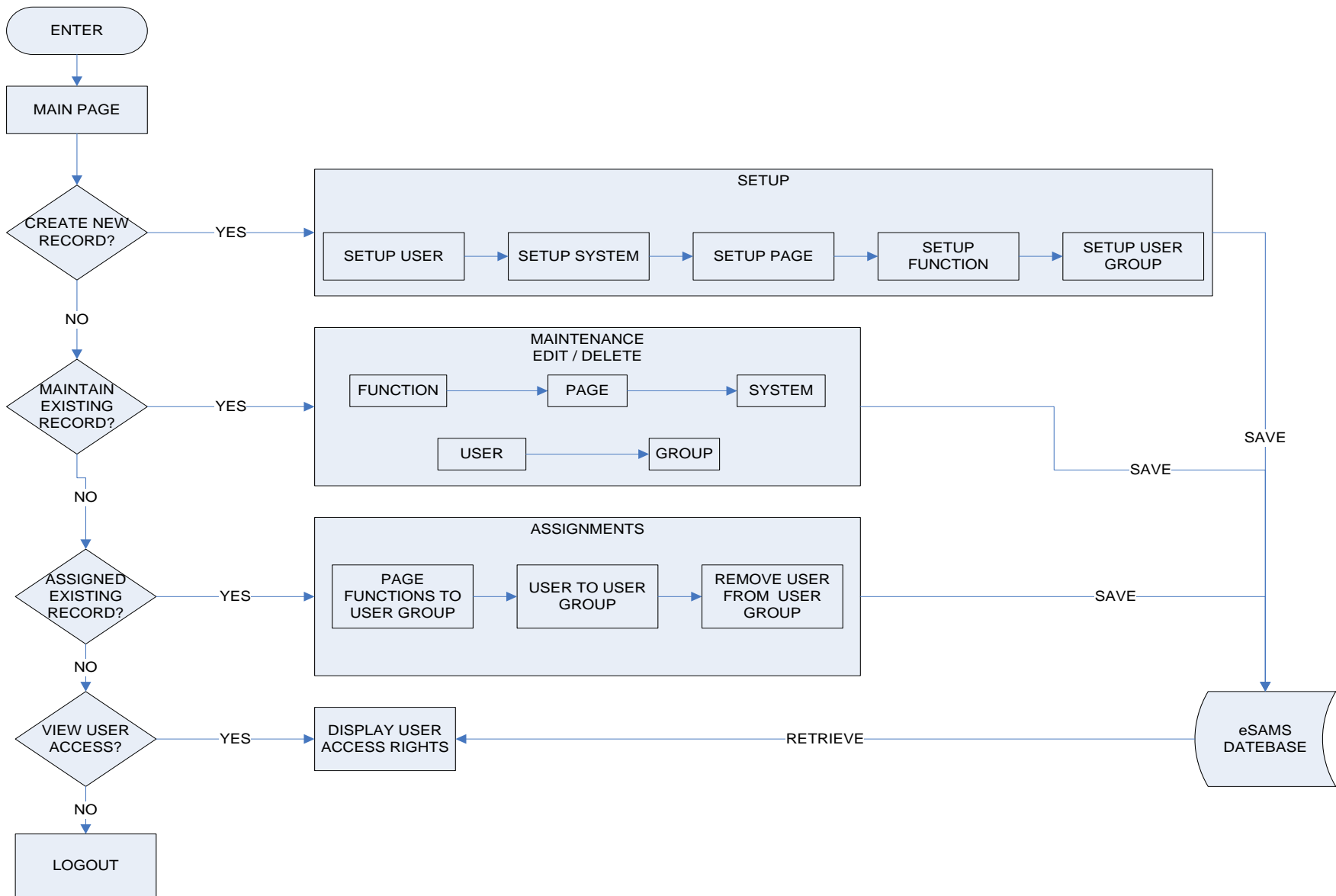


Figure 6 : Flow Chart Diagram for the User Access Control System

Description

Figure 4 is the Flow Chart Diagram for the User Access Control System. Flow chart is a schematic representation of an algorithm or a process. As shown above is the process of using the User Access Control Management System.

First, User wills Enter User Access Control Management System. Then User can choose to do one of the things that are inside the system. If User chooses to setup User, System, Page, Function or User Group, then user will enter the setup section. In this section User can choose to setup either one, however to setup a Page, it has to under existing or new System, or if user wants to setup a Function, it has to be under the existing or new System and Page.

If User chooses to maintain the Existing User, System, Page, Function of User Group details, User will enter the Maintenance Section where user can choose to do editing or deleting the details. However if User wants to delete the User Group detail, user have to make sure there are no user or function that are attached to the user group, or wants to delete Function detail, make sure that the Function is not attached to a user Group, or wants to delete Page detail, make sure there are no function in the Page, or wants to delete System detail, make sure there are no Page under the system, or wants to delete User detail, make sure the user is not attach to the User Group.

If User wants to Assign Page and Function to a System Group, then user have to make sure the System Page, Page Function, and System Group is already Setup or Created. If everything is already created, then User can choose to Assign Page and Function to a System Group. User can also choose to Unassigned Page and Function to a certain System Group. As usual, all the changes made will update the database.

If User chooses to Assign User to a System Group, User has to make sure that there are already User and System Group created in the System. If everything is already created, hen User can Assign User to System Group. User can also choose to Remove User from a Certain Group. All the changes made will update the database.

If User chooses to view other User Access Rights, User can choose to display User Access Rights. User only needs to know which system and also the User Id or User Name.

Lastly, if User chooses to Logout, User can do so at any time or any page.

3.5 Database Design

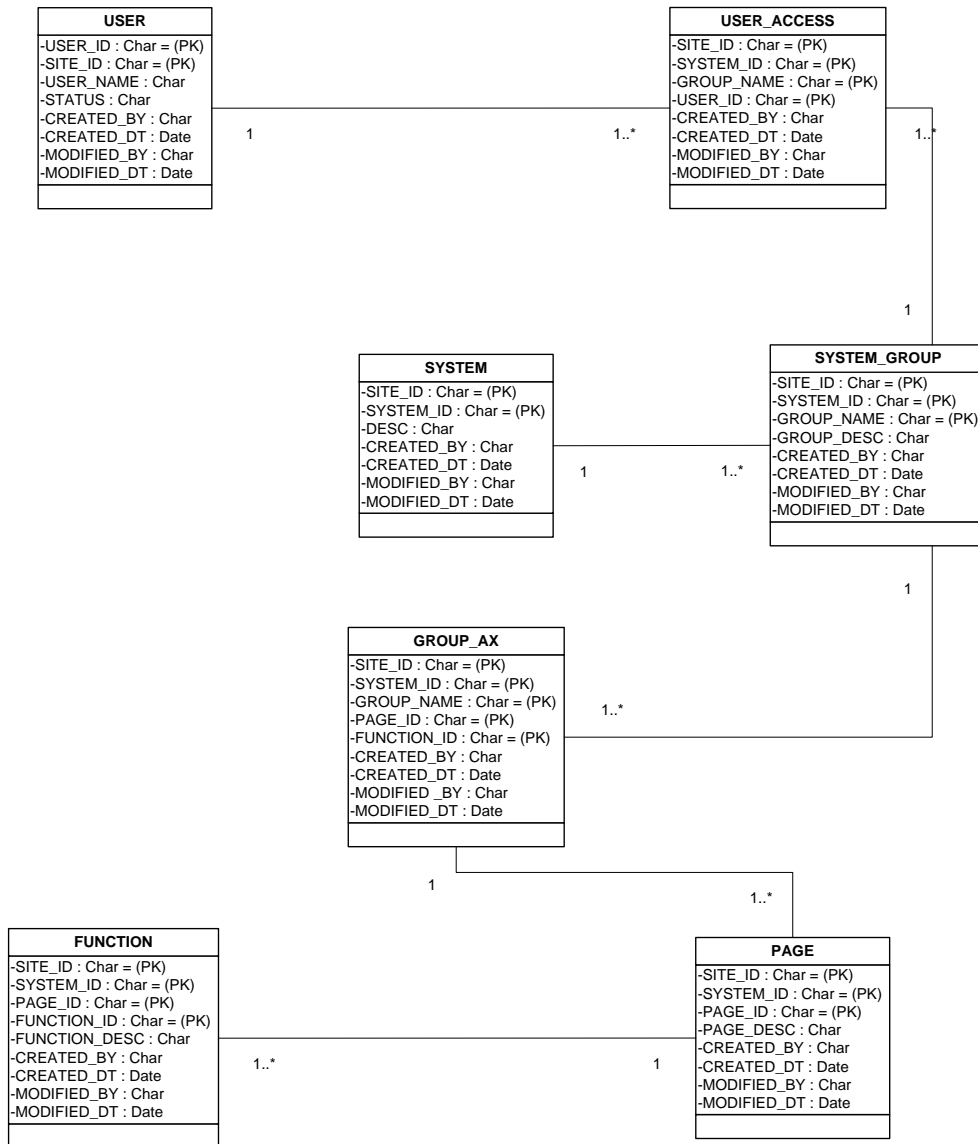


Figure 7 : Database Design

3.6 Interface Design

USER

SETUP USER

USER NAME:	A			
EMPLOYEE NUMBER :	1000	TO	1999	<input type="button" value="RETRIEVE"/>
SITE :	KM1			
STATUS :	ACTIVE			

	EMP ID	NAME	SITE	DEPARTMENT
<input checked="" type="checkbox"/>	001234	CRISTIANO RONALDO	KM1	IT
<input checked="" type="checkbox"/>	004231	WAYNE ROONEY	KM1	SPUTTER
<input checked="" type="checkbox"/>	005678	MICHEAL CARRICK	KM1	IT
<input checked="" type="checkbox"/>	008765	RIO FERDINAND	KM1	SPUTTER
<input checked="" type="checkbox"/>	009876	EDWIN VAN DER SAAR	KM1	IT
<input checked="" type="checkbox"/>	006789	HENNRICK LARRSON	KM1	IT
<input checked="" type="checkbox"/>	008765	NMENJA VIDIC	KM1	IT
<input checked="" type="checkbox"/>	002345	RYAN GIGGS	KM1	IT
<input checked="" type="checkbox"/>	005643	PAUL SCHOLES	KM1	IT
<input checked="" type="checkbox"/>	009879	LUIS SAHA	KM1	IT

Figure 8

MAINTAIN USER

SITE :	KM1			
USER NAME:	A			
EMPLOYEE NUMBER :	00	TO	00	<input type="button" value="RETRIEVE"/>

	EMP ID	NAME	SITE	STATUS	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	001234	CRISTIANO RONALDO	KM1	ACTIVE	WAYNE ROONEY	2/11/2006	WAYNE ROONEY	2/11/2006
<input checked="" type="checkbox"/>	004231	WAYNE ROONEY	KM1	ACTIVE	WAYNE ROONEY	1/7/2006	WAYNE ROONEY	1/7/2006
<input checked="" type="checkbox"/>	005678	MICHEAL CARRICK	KM1	ACTIVE	RYAN GIGGS	13/12/2005	RYAN GIGGS	13/12/2005
<input checked="" type="checkbox"/>	008765	RIO FERDINAND	KM1	ACTIVE	RYAN GIGGS	27/9/2006	RYAN GIGGS	27/9/2006
<input checked="" type="checkbox"/>	009876	EDWIN VAN DER SAAR	KM1	ACTIVE	RYAN GIGGS	15/3/2004	RYAN GIGGS	15/3/2004
<input checked="" type="checkbox"/>	006789	HENNRICK LARRSON	KM1	INACTIVE	RYAN GIGGS	31/12/2006	RYAN GIGGS	31/12/2006
<input checked="" type="checkbox"/>	008765	NMENJA VIDIC	KM1	ACTIVE	WAYNE ROONEY	24/8/2005	WAYNE ROONEY	24/8/2005
<input checked="" type="checkbox"/>	002345	RYAN GIGGS	KM1	ACTIVE	RYAN GIGGS	2/2/2004	RYAN GIGGS	2/2/2004
<input checked="" type="checkbox"/>	005643	PAUL SCHOLES	KM1	ACTIVE	WAYNE ROONEY	16/10/2005	WAYNE ROONEY	16/10/2005
<input checked="" type="checkbox"/>	009879	LUIS SAHA	KM1	INACTIVE	RYAN GIGGS	4/4/2004	RYAN GIGGS	4/4/2004

Figure 9

EDIT USER

SITE : KM1

USER NAME: CRISTIANO RONALDO

EMPLOYEE NUMBER : 001234

STATUS : ACTIVE

SAVE

Figure 10

MAINTAIN USER

SITE : KM1

USER NAME: A


EMPLOYEE NUMBER : 00 TO 00

RETRIEVE

	EMP ID	NAME	SITE	STATUS	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	001234	CRISTIANO RONALDO	KM1	ACTIVE	WAYNE ROONEY	24/8/2005	WAYNE ROONEY	2/11/2006
<input checked="" type="checkbox"/>	004231	WAYNE ROONEY	KM1	ACTIVE	WAYNE ROONEY	2/2/2004	WAYNE ROONEY	1/7/2006
<input checked="" type="checkbox"/>	005678	MICHEAL CARROLL	KM1	ACTIVE	RYAN GIGGS	16/10/2005	RYAN GIGGS	13/12/2005
<input checked="" type="checkbox"/>	008765	RIO FERDINAND	KM1	ACTIVE	RYAN GIGGS	4/4/2004	RYAN GIGGS	27/9/2006
<input checked="" type="checkbox"/>	009876	EDWIN VAN DER SAAR	KM1	ACTIVE	RYAN GIGGS	4/4/2004	RYAN GIGGS	15/3/2004
<input checked="" type="checkbox"/>	006789	HENNRICK LARSEN	KM1	ACTIVE	RYAN GIGGS	4/4/2004	RYAN GIGGS	31/12/2006
<input checked="" type="checkbox"/>	008765	NMENJA VIDIC	KM1	ACTIVE	WAYNE ROONEY	24/8/2005	WAYNE ROONEY	24/8/2005
<input checked="" type="checkbox"/>	002345	RYAN GIGGS	KM1	ACTIVE	RYAN GIGGS	2/2/2004	RYAN GIGGS	2/2/2004
<input checked="" type="checkbox"/>	005643	PAUL SCHOLES	KM1	ACTIVE	WAYNE ROONEY	16/10/2005	WAYNE ROONEY	16/10/2005
<input checked="" type="checkbox"/>	009879	LUIS SAHA	KM1	INACTIVE	RYAN GIGGS	4/4/2004	RYAN GIGGS	4/4/2004

EDIT DELETE

DELETE USER

 ARE YOU SURE YOU WANT TO DELETE USER **001234** (**CRISTIANO RONALDO**) ?

YES NO

Figure 11

SYSTEM

SETUP SYSTEM

SITE : KM1

SYSTEM ID: USER ACCESS CONTROL

SYSTEM DESCRIPTION : MANAGE DATA ON USER TO A SYSTEM

SAVE

Figure 12

MAINTAIN SYSTEM

SITE : KM1

SYSTEM ID: ALL RETRIEVE

	SYSTEM ID	DESCRIPTION	SITE	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	FINANCE SYSTEM	DO FINANCIAL CALCULATION	KM1	RYAN GIGGS	3-6-2006	WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	USER ACCESS CONTROL	DO USER MAINTIANANCE	KM1	RYAN GIGGS	4-7-2009	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	WEBPAGING	ABOUT USE PAGING	KM1	WAYNE ROONEY	12-12-2012	WAYNE ROONEY	14-2-2007

EDIT DELETE

Figure 13

EDIT SYSTEM

SITE : KM1

SYSTEM ID: eSPIM

DESCRIPTION : DO THIS AND THAT

SAVE

Figure 14

MAINTAIN SYSTEM

SITE : KM1

SYSTEM ID: ALL

DELETE SYSTEM

ARE YOU SURE YOU WANT TO DELETE SYSTEM eSPIM ?

YES NO

	SYSTEM ID					MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	FINANCE SYSTEM	DO F				WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	USER ACCESS CONTROL	DO				WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	WEBPAGING	ABOUT USE PAGING	KM1	WAYNE ROONEY	12-12-2012	WAYNE ROONEY	14-2-2007

EDIT DELETE

Figure 15

PAGE

SETUP PAGE

SITE : KM1

SYSTEM ID : KOMAG FINANCE

PAGE ID: FINANCIAL-001

PAGE DESCRIPTION : DO A FINANCIAL CALCULATION

SAVE/ADD

Figure 16

MAINTAIN PAGE

SITE : KM1

SYSTEM ID: ALL

PAGE ID : FINANCIAL

	PAGE ID	DESCRIPTION	SYSTEM ID	SITE	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	FINANCIAL	DO FINANCIAL CALCULATION	FINANCE SYSTEM	KM1	RYAN GIGGS	3-6-2006	WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	USER	DO USER MAINTIANANCE	FINANCE SYSTEM	KM1	RYAN GIGGS	4-7-2009	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	PAGING	ABOUT USE PAGING	WEB PAGING	KM1	RYAN GIGGS	12-12-2012	WAYNE ROONEY	14-2-2007

EDIT DELETE

Figure 17

EDIT PAGE

SITE :

SYSTEM ID:

PAGE ID :

PAGE DESCRIPTION :

Figure 18

MAINTAIN PAGE

SITE :

SYSTEM ID:

PAGE ID :

	PAGE ID	DESCRIPTION	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	FINANCIAL	DO FINANCIAL CALCULATION	WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	USER	DO USER MAINTIAN	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	PAGING	ABOUT USE PAGI	WAYNE ROONEY	14-2-2007

DELETE PAGE

ARE YOU SURE YOU WANT TO DELETE PAGE FINANCIAL ?

Figure 19

FUNCTION

SETUP FUNCTION

SITE :

SYSTEM ID :

PAGE ID :

FUNCTION ID :

FUNCTION DESCRIPTION :

Figure 20

MAINTAIN FUNCTION

SITE : KM1

SYSTEM ID: ALL

PAGE ID : ALL

FUNCTION ID : ALL

	FUNCTION ID	DESCRIPTION	PAGE ID	SYSTEM ID	SITE	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	ADD	DO ADDITION FUNCTION	FINANCIAL	FINANCE SYSTEM	KM1	RYAN GIGGS	3-6-2006	RYAN GIGGS	2-1-2007
<input checked="" type="checkbox"/>	SUBTRACT	DO SUBTRACTION FUNCTION	FINANCIAL	FINANCE SYSTEM	KM1	RYAN GIGGS	4-7-2009	RYAN GIGGS	17-1-2007
<input checked="" type="checkbox"/>	PAGE NO	DISPLAY PAGER NUMBER	PAGING	WEB PAGING	KM1	RYAN GIGGS	12-12-2012	RYAN GIGGS	14-2-2007

EDIT DELETE

Figure 21

EDIT FUNCTION

SITE : KM1

SYSTEM ID: FINANCIAL SYSTEM

PAGE ID : FINANCIAL

FUNCTION ID : ADD

FUNCTION DESCRIPTION : **ADDING TWO NUMBERS**

SAVE

Figure 22

MAINTAIN FUNCTION


SITE : KM1

SYSTEM ID: ALL

PAGE ID : ALL

FUNCTION ID : ALL

DELETE FUNCTION

 ARE YOU SURE YOU WANT TO DELETE FUNCTION ADD ?

YES NO

	FUNCTION ID	DESCRIPTION	PAGE ID	SYSTEM ID	SITE	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	ADD	DO ADDITION FUNCTI					2006	RYAN GIGGS	2-1-2007
<input checked="" type="checkbox"/>	SUBTRACT	DO SUBTRACTION FUNCTION	FINANCIAL	FINANCE SYSTEM	KM1	RYAN GIGGS	4-7-2009	RYAN GIGGS	17-1-2007
<input checked="" type="checkbox"/>	PAGE NO	DISPLAY PAGER NUMBER	PAGING	WEB PAGING	KM1	RYAN GIGGS	12-12-2012	RYAN GIGGS	14-2-2007

EDIT DELETE

Figure 23

USER GROUP

SETUP USER GROUP

SITE : KM1

SYSTEM ID : KOMAG FINANCE

GROUP NAME : ADMIN

GROUP DESCRIPTION : HIGHEST ACCESS

SAVE/ADD

Figure 24

MAINTAIN USER GROUP

SITE : KM1

SYSTEM ID: ALL

GROUP NAME : ALL

	GROUP NAME	DESCRIPTION	SYSTEM NAME	SITE	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	ADMIN	HIGHEST ACCESS	FINANCE SYSTEM	KM1	RYAN GIGGS	3-6-2006	WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	SUPPORT	SUPPORT ADMIN	FINANCE SYSTEM	KM1	RYAN GIGGS	4-7-2009	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	USER	NORMAL USER	FINANCE SYSTEM	KM1	WAYNE ROONEY	12-12-2012	WAYNE ROONEY	14-2-2007

EDIT DELETE

Figure 25

EDIT USER GROUP

SITE : KM1

SYSTEM ID: FINANCIAL SYSTEM

GROUP NAME : ADMIN

GROUP DESCRIPTION : HAVE 2ND HIGHEST ACCESS

SAVE

Figure 26

MAINTAIN USER GROUP

SITE : KM1

SYSTEM ID: ALL

GROUP NAME : ALL

DELETE USER GROUP

ARE YOU SURE YOU WANT TO DELETE GROUP ADMIN ?

YES NO

	GROUP NAME	DESCRIPTION	FINANCE SYSTEM	KM1	RYAN GIGGS	4-7-2009	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	ADMIN	HIGHEST					WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	SUPPORT	SUPPORT ADMIN	FINANCE SYSTEM	KM1	RYAN GIGGS	4-7-2009	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	USER	NORMAL USER	FINANCE SYSTEM	KM1	WAYNE ROONEY	12-12-2012	WAYNE ROONEY	14-2-2007

EDIT DELETE

Figure 27

ASSIGNMENT

ASSIGN PAGE AND FUNCTION TO SYSTEM USER GROUP

SITE : KM1

SYSTEM ID: FINANCIAL SYSTEM

GROUP NAME : ADMIN

PAGE ID : FINANCIAL

	FUNCTION ID	DESCRIPTION	CREATED BY	CREATED DATE
<input checked="" type="checkbox"/>	ADD	DO ADDITION WITH SEVERAL NUMBERS	RYAN GIGGS	3-6-2006
<input checked="" type="checkbox"/>	SUBTRACT	DO SUBTRACT WITH TWO NUMBER	RYAN GIGGS	4-7-2009
<input checked="" type="checkbox"/>	PAGE NO	DISPLAY PAGER NUMBER	RYAN GIGGS	12-12-2012

SAVE

Figure 28

ASSIGN USER TO SYSTEM USER GROUP

SITE : KM1

SYSTEM ID: FINANCIAL SYSTEM

GROUP NAME : ADMIN

USER NAME : AA

EMPLOYEE NUMBER : 1000 TO 1999

	EMPLOYEE NUMBER	NAME
<input checked="" type="checkbox"/>	001234	CRISTIANO RONALDO
<input checked="" type="checkbox"/>	001345	WAYNE ROONEY
<input checked="" type="checkbox"/>	001546	ERIC CANTONA

SAVE

Figure 29

REMOVE USER FROM SYSTEM USER GROUP

SITE : KM1

SYSTEM ID: FINANCIAL SYSTEM

GROUP NAME : ADMIN

	EMPLOYEE NUMBER	NAME	CREATED BY	CREATED DATE	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	001234	CRISTIANO RONALDO	RYAN GIGGS	3-6-2006	WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	001345	WAYNE ROONEY	RYAN GIGGS	4-7-2009	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	001546	ERIC CANTONA	WAYNE ROONEY	12-12-2012	WAYNE ROONEY	14-2-2007

REMOVE

Figure 30

NAVIGATION

MAIN

LOG OUT	SETUP	MAINTENANCE	ASSIGNING	INQUIRY
LOG OUT	USER SYSTEM PAGE FUNCTION USER GROUP	USER SYSTEM PAGE FUNCTION USER GROUP REMOVE USER FROM SYSTEM USER GROUP	PAGE FUNCTION TO USER GROUP USER TO USER GROUP	DISPLAY USER ACCESS

Figure 31

DISPLAY USER ACCESS RIGHTS

USER ACCESS RIGHT

SITE : KM1

SYSTEM ID: FINANCIAL SYSTEM

USER NAME : CRISTIANO RONALDO

EMPLOYEE NUMBER : 00 TO 00

SEARCH

	EMPLOYEE NUMBER	NAME	SITE	SYSTEM	USER GROUP	MODIFIED BY	MODIFIED DATE
<input checked="" type="checkbox"/>	001234	CRISTIANO RONALDO	KM1	eSPIM	ADMIN	WAYNE ROONEY	2-1-2007
<input checked="" type="checkbox"/>	001345	CRISTIANO RONALDO	KM1	WEB PAGING	USER	WAYNE ROONEY	17-1-2007
<input checked="" type="checkbox"/>	001546	CRISTIANO RONALDO	KM1	eLEAVE	SUPPORT	WAYNE ROONEY	14-2-2007

Figure 32

4.1 Results

All the interfaces have to be created according to the design planned. This is to make sure that the project will be on schedule and if there are any problems occurred during the implementation stage can be handle structurally.

Below are some of the pages that have been created according to the design.

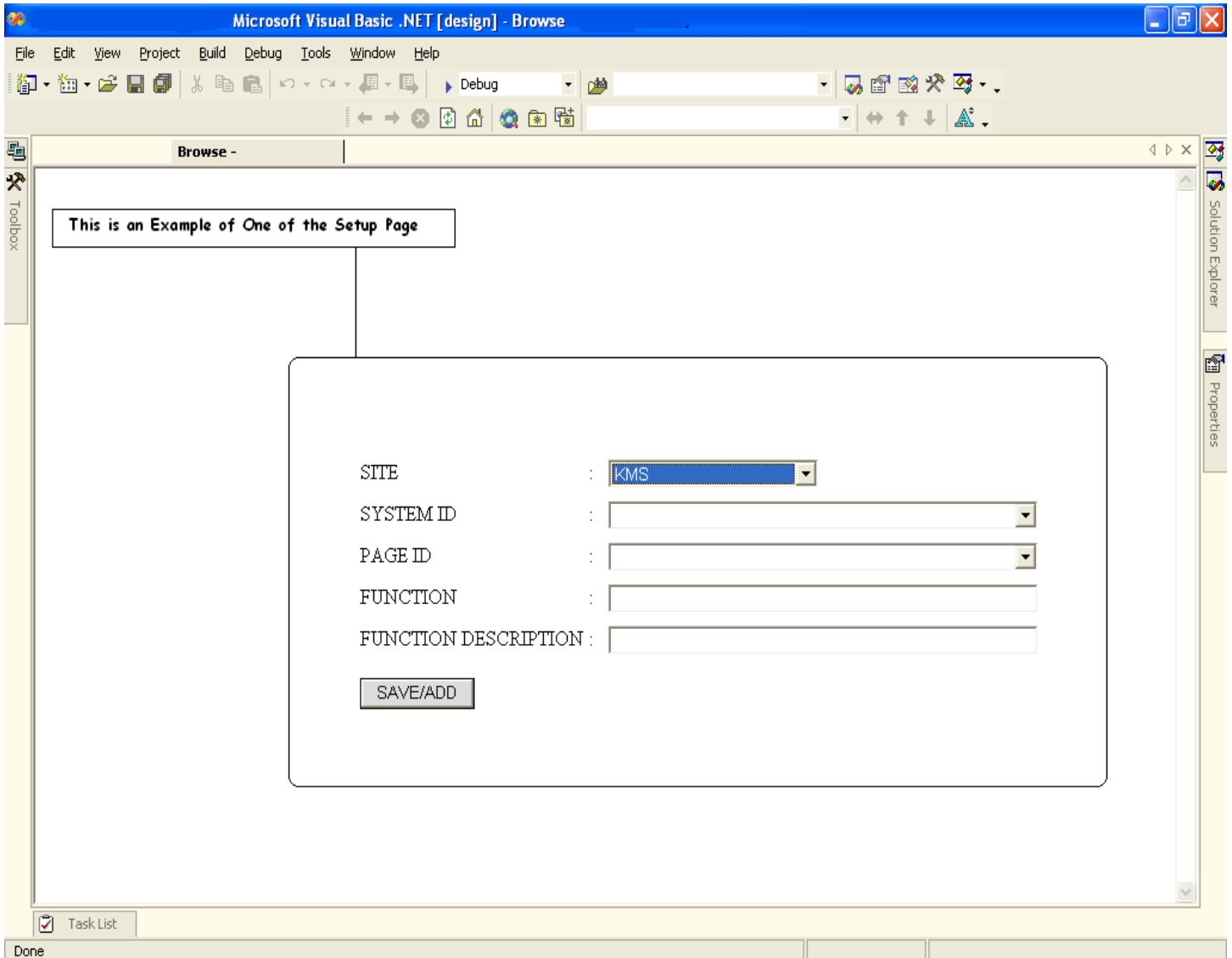


Figure 33: Example of Setup Page

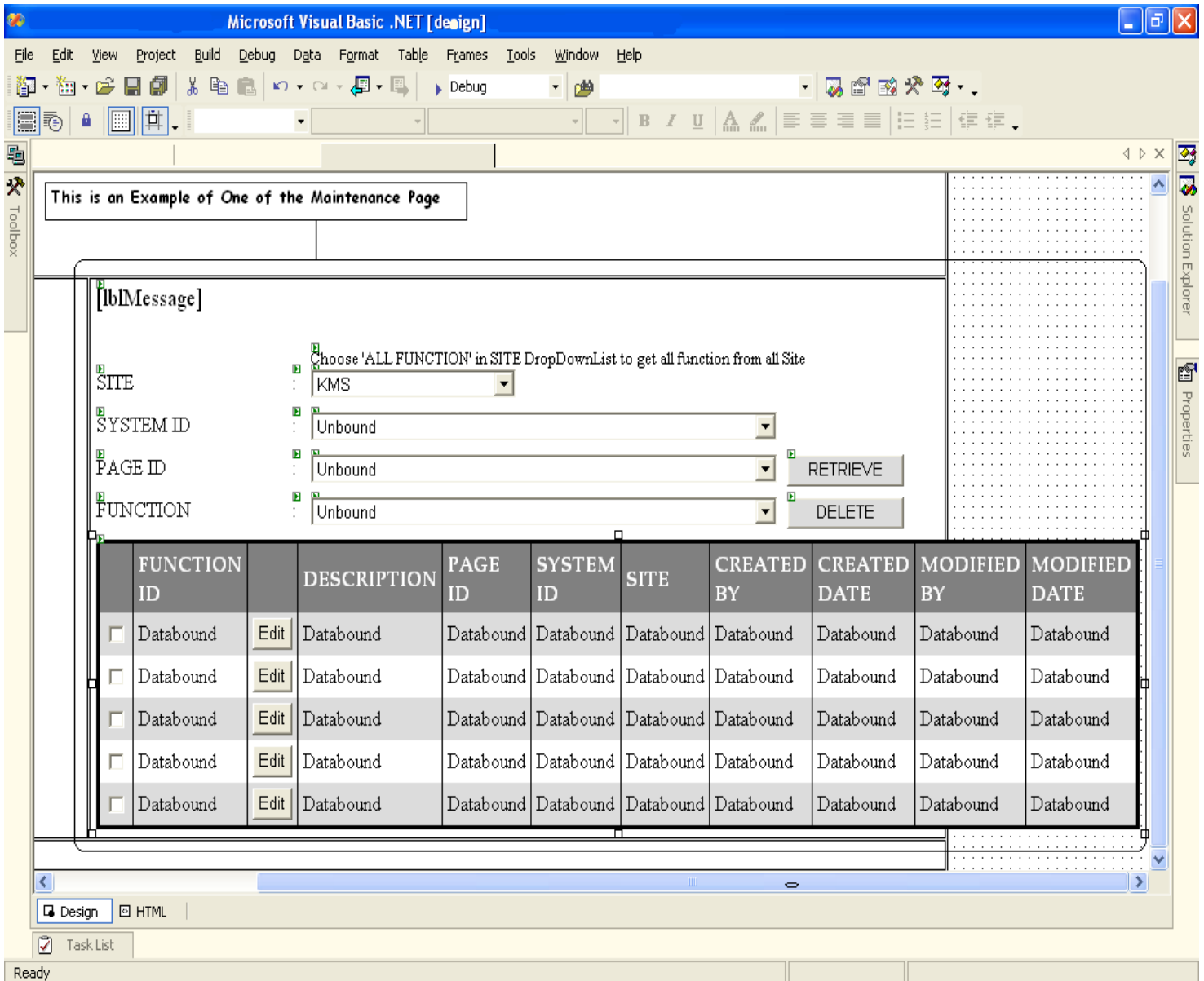


Figure 34: Example of Maintenance Page

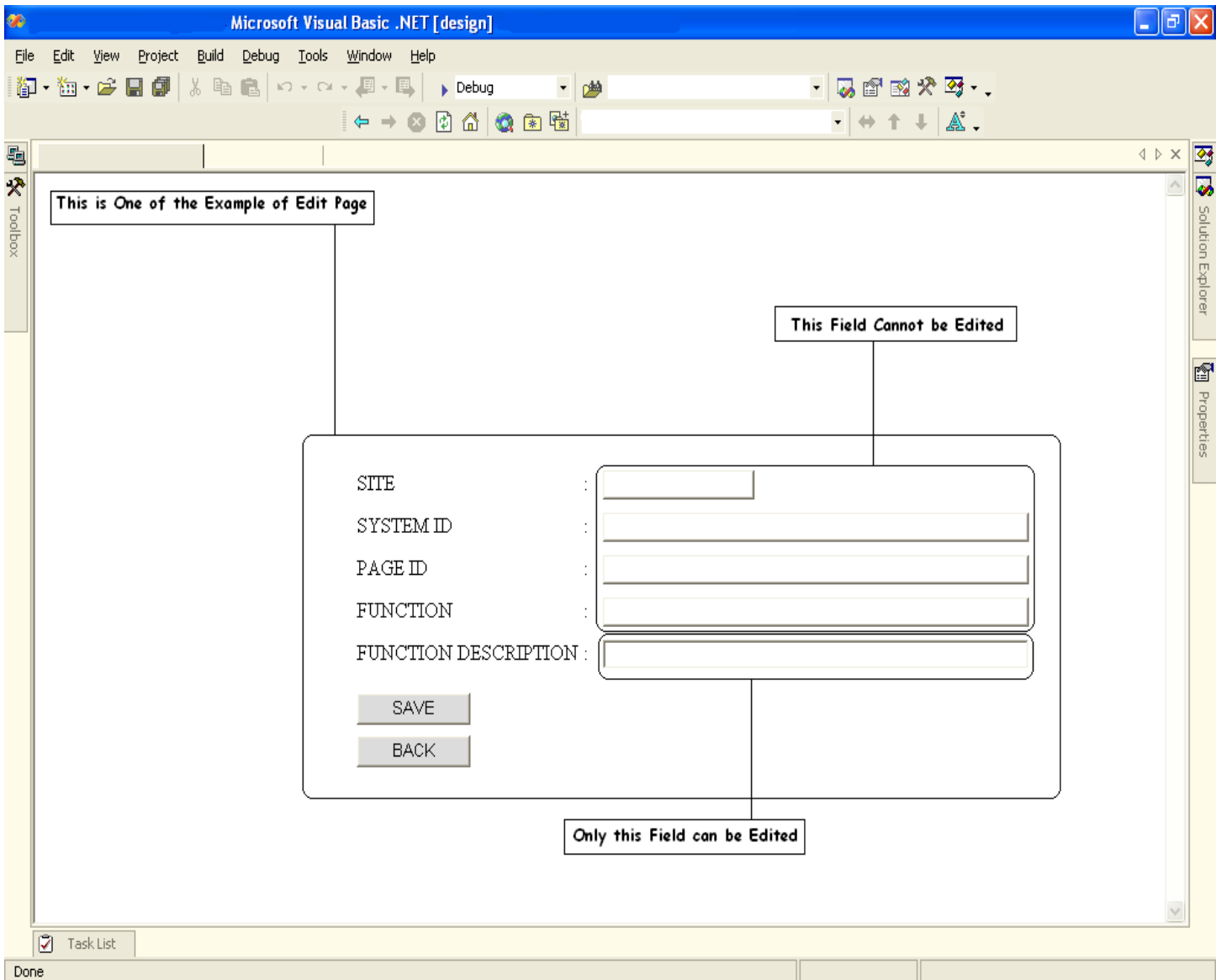


Figure 35: Example of Editing Page

Each of these 3 pages will be created repeatedly according to the User, System, Page, Function and User Group. The variation between each page is the information that each page will display or need. However the basics of each page is the same as show in the figure

4.2 Discussion

As has been explained in the chapter 2.4 – User Interface Design, to optimized user interface design, it requires a systematic approach to the design process. User interface design should make a product easy to understand and use, which results in greater user acceptance. Therefore the User Access Control System will try to provide the best User interface design.

The User Access Control System that has been created until now are actually trying to follow some other points that have been made in the chapter 2.5 - Ten Usability Heuristics. There are:-

1. Match between system and the real world

The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

2. Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions

3. Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility

Furthermore the system created up to this point also trying to implements some of the points in the chapter 2.6 - User Interface Design Tips and Technique, such as:-

1. Set standards and stick to them.

The only way to ensure consistency within application is to set user interface design standards, and then stick to them.

2. Align fields effectively.

When a screen has more than one editing field, the fields should be organized in a way that is both visually appealing and efficient. It is always found the best way to do so is to left-justify edit fields: in other words, make the left-hand side of each edit field line up in a straight line, one over the other. The corresponding labels should be right-justified and placed immediately beside the field. This is a clean and efficient way to organize the fields on a screen.

4.3 Difficulties Faced

Below are few of the difficulties faced during the development of the project as recorded:

- 1) Creating a connection to database(specifically the stored procedure)
- 2) Binding data to datagrid
- 3) Forwarding data to another page
- 4) Top navigational menu bar.

As the project progress, one by one of these difficulties are solved.

Solutions

- 1) Creating a connection to database (specifically the stored procedure)

When trying to connect the system to the database, there should be a connector to connect the coding side in the visual basic to the MySQL database. The connector needed is the Open Database Connectivity (ODBC). The goal of ODBC is to make it possible to access any data from any application, regardless of which database management system (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant -- that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them.

In the next page is the explanation on how to make a connection to the ODBC and retrieve data from the database in the application side in the coding.

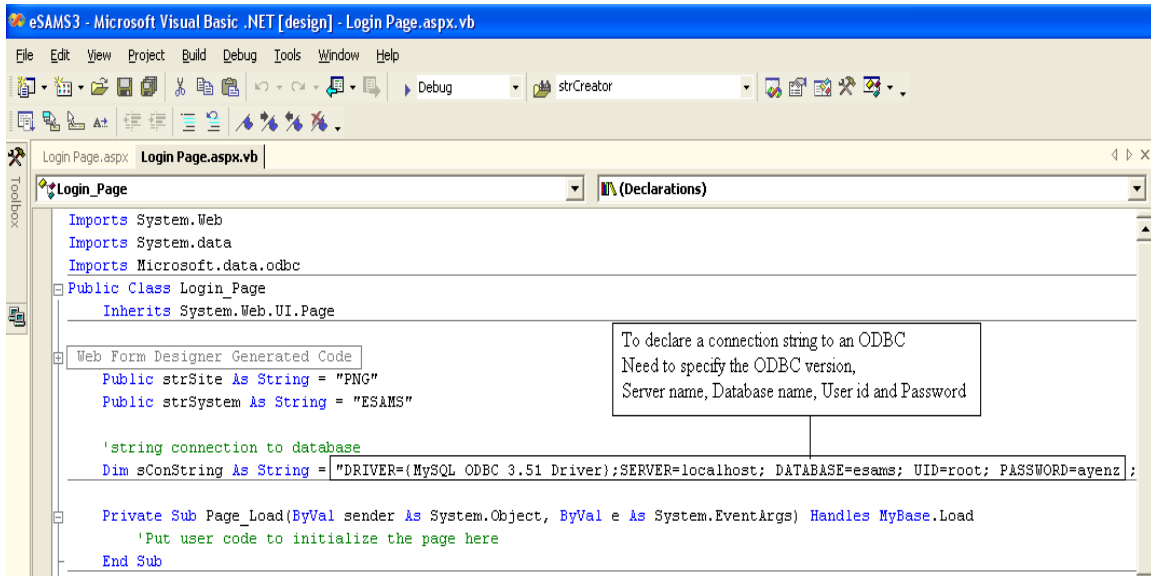


Figure 36

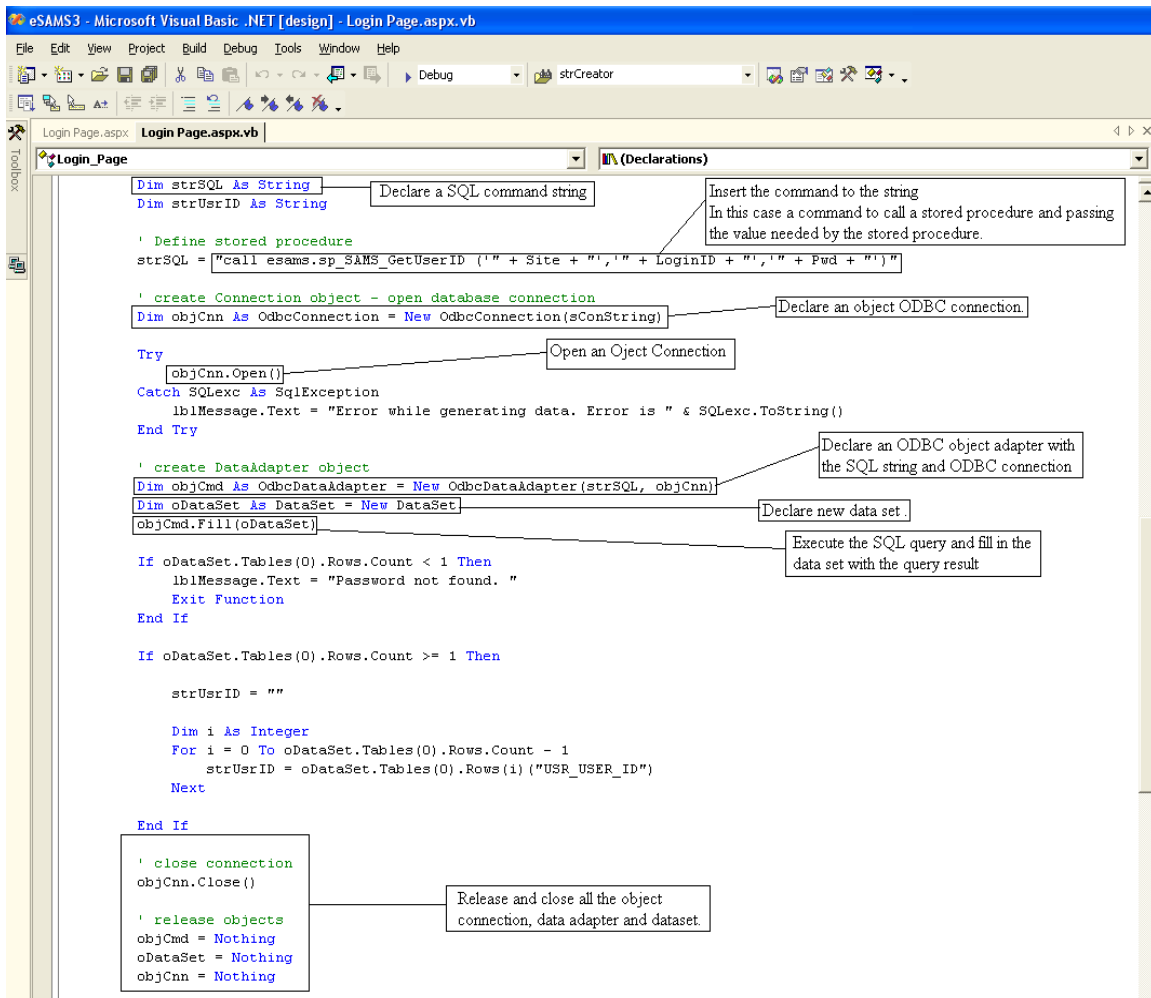
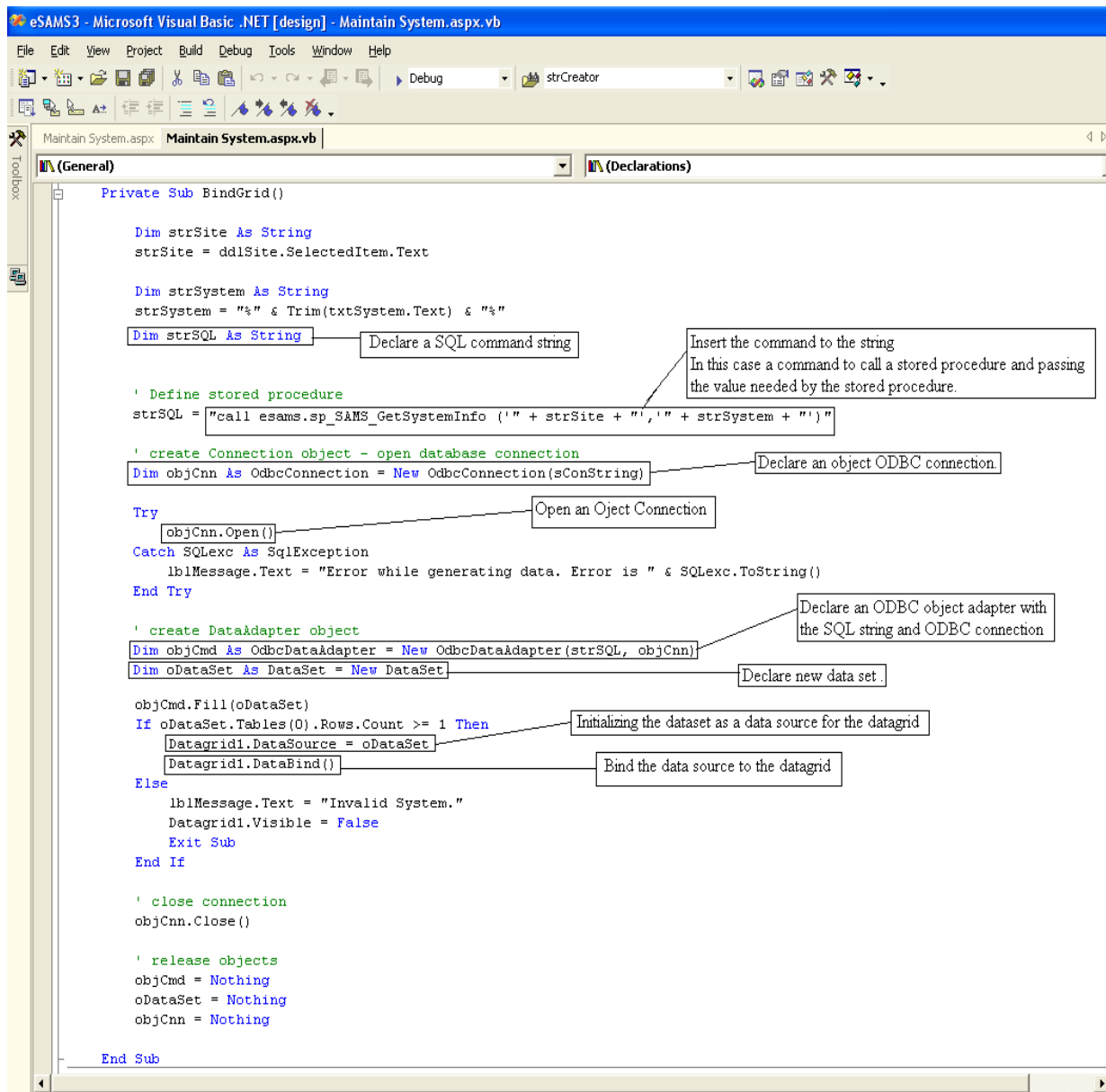


Figure 37

2) Binding data to datagrid

Inside the application, there are several time that the application will retrieve the data from the database and will bind it to a datagrid. This will make it easier the user to use this application.

Below is the explanation on the data binding to the datagrid in the coding.



```
Private Sub BindGrid()  
  
    Dim strSite As String  
    strSite = ddlSite.SelectedItem.Text  
  
    Dim strSystem As String  
    strSystem = "%" & Trim(txtSystem.Text) & "%"  
  
    Dim strSQL As String  
    ' Define stored procedure  
    strSQL = "call esams.sp_SAMS_GetSystemInfo ('" + strSite + "','" + strSystem + "'" )  
  
    ' create Connection object - open database connection  
    Dim objCnn As OdbcConnection = New OdbcConnection(sConString)  
  
    Try  
        objCnn.Open()  
    Catch SQLExc As SqlException  
        lblMessage.Text = "Error while generating data. Error is " & SQLExc.ToString()  
    End Try  
  
    ' create DataAdapter object  
    Dim objCmd As OdbcDataAdapter = New OdbcDataAdapter(strSQL, objCnn)  
    Dim oDataSet As DataSet = New DataSet  
  
    objCmd.Fill(oDataSet)  
    If oDataSet.Tables(0).Rows.Count >= 1 Then  
        Datagrid1.DataSource = oDataSet  
        Datagrid1.DataBind()  
    Else  
        lblMessage.Text = "Invalid System."  
        Datagrid1.Visible = False  
        Exit Sub  
    End If  
  
    ' close connection  
    objCnn.Close()  
  
    ' release objects  
    objCmd = Nothing  
    oDataSet = Nothing  
    objCnn = Nothing  
  
End Sub
```

The screenshot shows the Visual Basic code for the BindGrid() subroutine. Annotations explain the following steps:

- Declare a SQL command string:** Points to the declaration of `strSQL`.
- Insert the command to the string:** Points to the assignment of the SQL command to `strSQL`.
- Declare an object ODBC connection:** Points to the declaration of `objCnn`.
- Open an Oject Connection:** Points to the `objCnn.Open()` call.
- Declare an ODBC object adapter with the SQL string and ODBC connection:** Points to the declaration of `objCmd`.
- Declare new data set:** Points to the declaration of `oDataSet`.
- Initializing the dataset as a data source for the datagrid:** Points to the assignment of `oDataSet` to `Datagrid1.DataSource`.
- Bind the data source to the datagrid:** Points to the `DataBind()` call.

Figure 38

3) Forwarding data to another page

There is a lot of time in the application where the application will send a data to another page. This is very important when the application wants to send data about an access of a group inside the application.

Below is the coding on how to use a session.

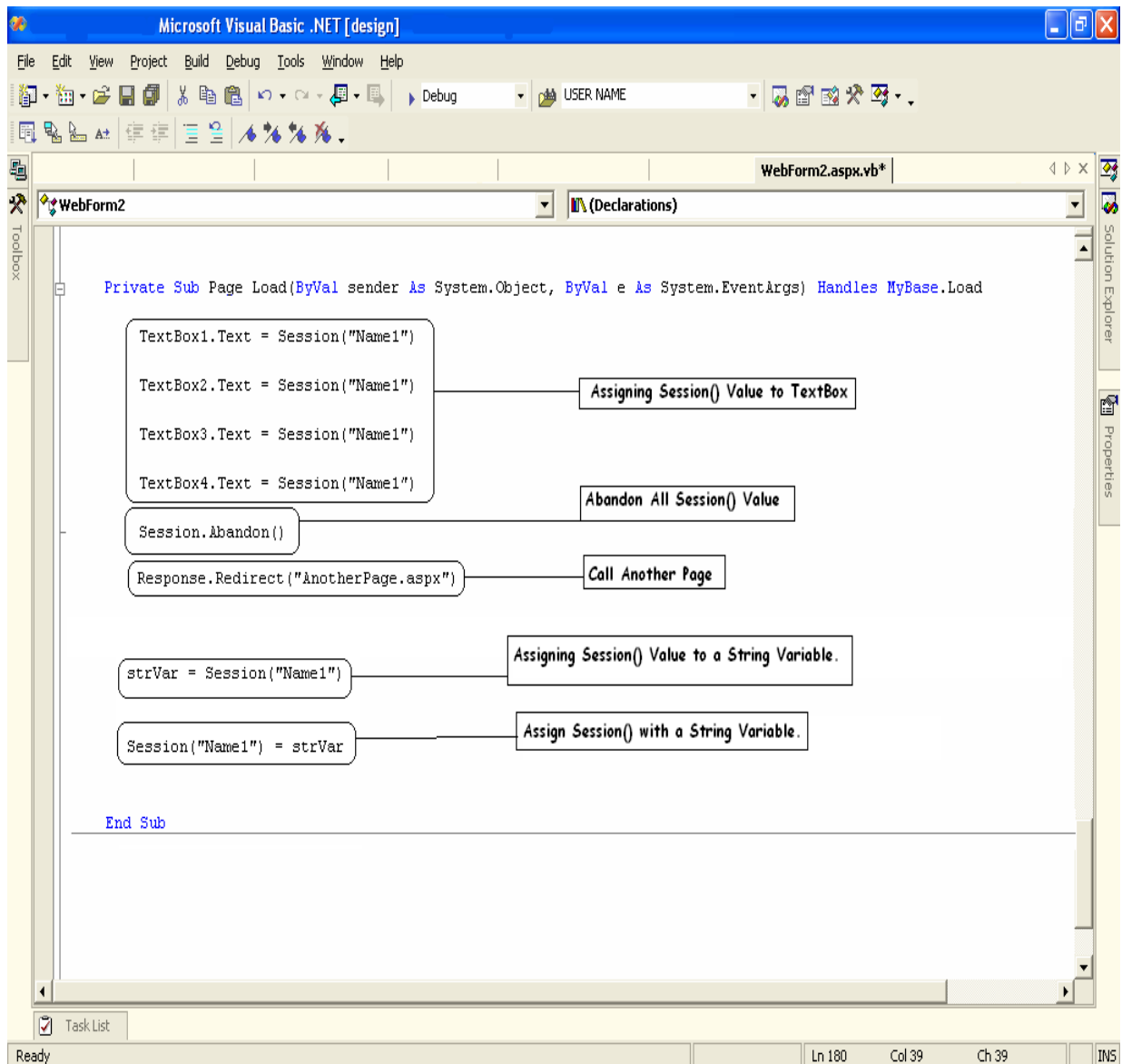


Figure 39

4) Top navigational menu bar.

In the ASP.NET 1.1, there is no top navigational menu bar. However there is a need for the system to have a top navigational menu bar so that user can view easily another page using it. In order to create a top navigational menu bar, there is a need to modify the coding inside the html and add few lines of java script. So far I found that it is the only way to create a top navigational menu bar in ASP.NET 1.1.

Below are few figures on the top navigational menu bar.

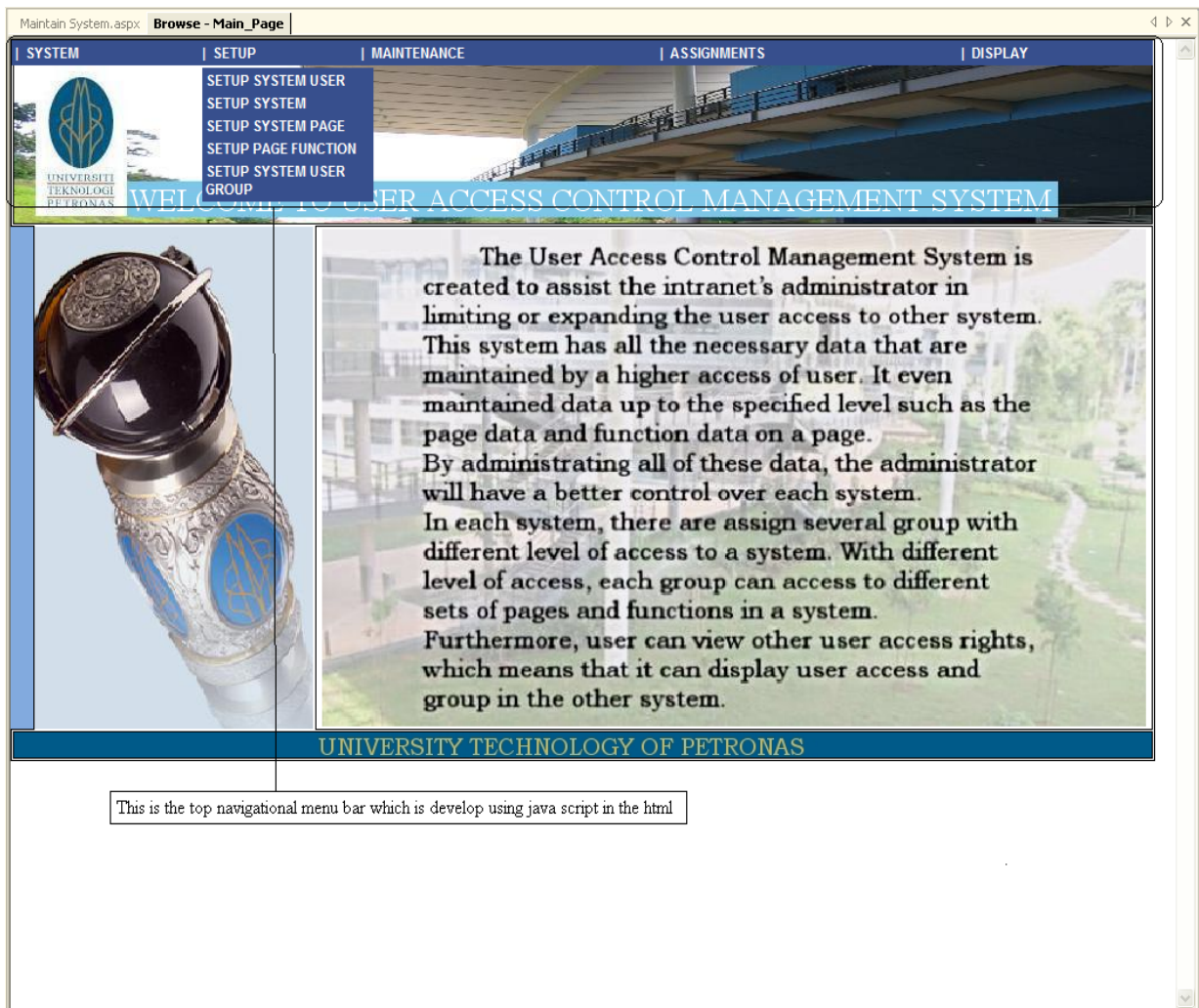


Figure 40

4.4 Project Scenario

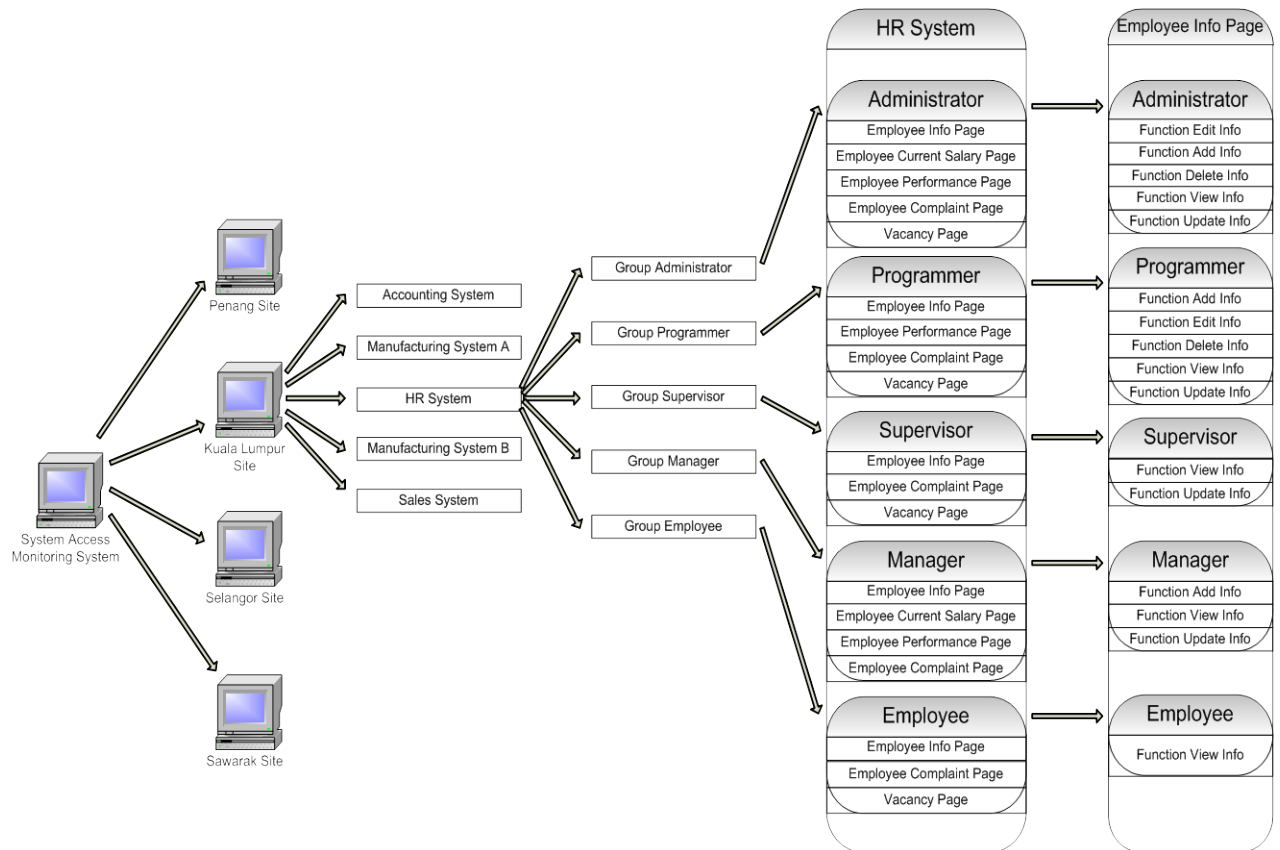


Figure 41

The Figure 41 is the scenario on how does this System Access Monitoring System (SAMS) works.

Firstly it is assuming that there are few branches of a company. Inside each branch, there are a lot of systems or applications. In each application, there are several groups. In the figure, there is an administrator group. The administrator group can access all the pages inside the application. As for the employee group, they can access only few pages. As the functions, given in the figure, in the Employee Info Page, the administrator can use all the functions inside the application and the employee group can only use the View Info function.

Chapter 5: Conclusion

There are many ways which computer technology can be improved to be a further assistance in maintaining and limiting user access to a certain system and one of it has been discussed in this documentation. It might also be considered a new level of access control where this system centralizes all the access to applications and systems in the intranet. Furthermore the control is up to the functionality.

Furthermore the system proposes and done is trying to fulfill all the requirements that have been published in the literature reviews as the guideline to finished this system. Therefore, hopefully with this system, administrator of an intranet will have a better control over the users that are using the applications and systems and the administrator will understand the use of the system.

Reference:

- [1] Alan Dennis, Barbara Haley Wixom, and David Tegarden (2002) , *Systems Analysis & Design An Object-Oriented Approach with UML*, John Wiley & Sons
- [2] http://en.wikipedia.org/wiki/Rapid_application_development
- [3] <http://csweb.cs.bgsu.edu/maner/domains/RAD.htm>
- [4] [http://en.wikipedia.org/wiki/Access control](http://en.wikipedia.org/wiki/Access_control) - Wikipedia, the free encyclopedia.mht
- [5] Andreas GEYER-SCHULZ And Karlsruhe, Germany - Implementation of Hierarchical Authorization For A Web Based Digital Library, February 2007
- [6] Ian Lurie, The Written Word, Inc. Spreading Out : Why Internet- Based Software is the future of corporate computing – December 31, 1998.
- [7] Pearson Education South Asia Pte Ltd, Software Engineering for IT Professionals, 2004.
- [8] <http://www.usernomics.com/user-interface-design.htm>
- [9] http://www.useit.com/papers/heuristic/heuristic_list.html
- [10] <http://www.ambysoft.com/essays/userInterfaceDesign.html>
- [11] <http://hubel.sfasu.edu/research/survreslts.html>