International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 3, Issue 9 (September 2014), PP.53-67

Role-Based Access Control (Rbac) Based In Hospital Management

[1]Edwin Okoampa Boadu, [2]Gabriel Kofi Armah
[1]State Key Laboratory of Electronic Thin films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, P. R. China.
[2]School of Computer Science and Technology University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, P. R. China.

Abstract:- A key issue in any information security is to protect information about all forms against unauthorized access. Innovation access control model is now becoming a need for application on systems due to emerging acts. Role based access control (RBAC) is a feasible alternative to traditional Discretionary Access Control (DAC) and Mandatory Access Control (MAC). RBAC has been presented to be cost operative and is being employed in different application domains on account of its characteristics: policy neutrality, separation of duty relations, rich specification, and principal of least privilege and ease of managerial issues. Managing the hospital workers, assigning duties and keeping confidential health records effectively is a big hurdle these days. Accordingly, the administration of different security levels, resources, users, tasks etc. is indispensable. This study focused on a Hospital management system by using Role Based Access Control (RBAC). The design architecture is based on RBAC concepts, according to the concept, only the administrator has the privilege to manage or administer the data. She/he provides all types of privileges required to maintain users, their authorization and access, and the authorized resources. The administrator controls the largest information, including access to health care workers' files and has the sole access to all potential workers and their assigned duties. This study took into the account the security access control, and security policies and methods integrated into the RBAC which is appropriate for hospital management system. RBAC is used to control the access to patients' medical records, files and the hospital resources and eliminates security violations.
In the design, the language C++/Qt framework is used to implement the security workflow for different LOGIN processes.

Keywords:- RBAC Concept, Security Policy, Hospital Management, Architecture.

I. INTRODUCTION

Intrusion and unwavering security issues in the management of firms, organizations and private workflow environments is becoming so common these days, there is the need to have a better security system to manage and to cut out these intrusions. Any information management needs to protect their resources, and data against such an unauthorized revelation at the same time ensuring their accessibility to potential work use. Access control policy is one of the most popular and security mechanism these days applied in most security systems of management.

Threats to the Hospital management, security resources have grown dramatically in proportion to the ever growing number and kind of users – not only employees, but also partners, patients and customers –who have access to resources. The problem in hospital management is more complex than a simple case of keeping unauthorized external users out, it's about ensuring that authorized, and legitimate users have access only to specific resources to which they are entitled to. The constantly changing nature of roles, users doesn't make this any easier.

Role-Based Access Control has been improved by the National Institute of Standard and Technology (NIST) after DAC and the MAC. It is an authorized model that defines the access to the objects which are the resources. It decouples users and permissions by introducing roles. RBAC now defines:
$M \subseteq Users\ X\ Permissions$ (1)
A permission represents an authorization to perform an operation on an object. A declaration access control is the specification of an authorized by a relation. A user has access to the objects only if he/she has the appropriate permission.
$U \in User\ has\ P \in Permissios: (U, P) \in AC$ (2)

The essence of Role-based access control (RBAC) [1]is that permissions are assigned to roles. This simplifies the security management and helps to determine efficiently, which permissions are authorized for what Users in a large organizationwhich in this case Hospital Management is the case study. The nature of roles is static, so RBAC shows no flexibility and responsiveness. RBAC does not encompass the overall context associated with any collaborative activity [2]; it is a passive security system that serves the function of maintaining permission assignments. It lacks the ability to specify a fine-grained control of individual users in certain roles and on individual object instances. Despite all the identified flaws, RBAC is appropriate for consideration in systems that process unclassified but sensitive information as well as those that process classified information.

A properly managed RBAC system usually enables users to carry out many range of authorized operations, and also provides much flexibility and broad applications. The system Manager / Administrator can control access at a level of abstraction that is natural to the way that management typically conduct its activities. This is achieved by strategically using statistical and dynamical means to regulate users' activities throughout the system and definition of roles, role hierarchies, relations and restrictions. The moment the RBAC framework is established for an organization or a firm, the administrative actions are then granted and revoking of users into and out of roles are defined. In addition, it is also possible to associate the concept of an RBAC operation with the concept of "method" in Object Technology. This association leads to the approach where Object Technology can be used in applications and operating systems to implement an RBAC operation [3].

II. USERS AND ROLES AS USED IN RBAC

In the RBAC framework, users are granted membership into roles based on their qualifications and responsibilities in the firm or organization. The activities that a user is authorized to perform are usually based on the user's role.

The User Membership into role(s) can be revoked and new memberships established especially when new operations are introduced, and old activities can be deleted as the duties and organizational functions changes and evolves in the system. Therefore, administration and management of privileges are simplified in the process, roles can be updated without the privileges for every user on an individual basis. When a role is assigned to a User the user can be given no more privilege than is necessary to perform the job. This RBAC concept of least privilege needs, identifying the user's activity functions, which determines the least set of privileges needed to perform that function, and restricting the user to a domain where those privileges are.

Role based management isa very important aspect of distributing systems management. As the system grows bigger in size it is therefore necessary to decentralize the management activities amongst multiple administrators and probably automated agents. Again, it must be possible to dynamically load and retract policies from agents to change the behavior and strategy of the management system without re-coding or interrupting their activities [4].

III. ROLES & ROLE HIERARCHIES

With RBAC, roles can have sometimes overlapping responsibilities and privileges; i.e. Users belonging to different roles might need to perform common activities. Some general operations might be performed by all employees. In such a situation, it would be inefficient and administratively difficult to specify repeatedly these general activities for each role that gets created. Role hierarchies can be established to provide for the natural structure of the organization. A role hierarchy defines roles that have unique attributes and that might contain other roles. In health care case, a role specialist could contain the roles of Doctor and Intern. This implies that members of the role specialist are implicitly associated with the activities associated with the roles Doctor and Intern without the administrator having list the Doctor and intern activities, Moreover, the roles Cardiologist and Rheumatologist could each contain the specialist role [3]. Role hierarchies are a natural way of organizing roles to reflect authority, responsibility, and competency.

In the case ofhospital management, the role in which the User is gaining membership is not exclusive mutual, with another role for which the User already possesses membership. These activities and roles are usually subjected to management or administrative policies and constraints. When these activities intersect, hierarchies of roles can be established. Rather than costly auditing to monitor access, management can put in constraints on access through the useof RBAC. This might seem sufficient to allow physicians to have access to all patients' data records if their access is monitored carefully. Applying RBAC, constraints can be placed on physician access so that only those records that are related to the physician can be accessed.

IV. MOTIVATION

In this model of RBAC for the Hospital Management, we want to prove that this program is secured that is to say that it does not leak any secret information when it is run on a private device. Imagine for example that an external application running on your personal computer, explores your personal data, and leak this information to a third party. This is the property that non-interference captures: that secret data does not impact the public data. It ensures that running a program in a particular environment, especially an environment containing secret data such as passwords etc. does not leak any information about it into the public world like the internet. We wish to define security in terms of information flow in a system. Information is considered to flow from a domain U to a domain V, if the actions performed by domain U caused the behaviors of the system be perceived differently by domain V from those (behaviours) when these actions were not present in the system. Informally, a system is secure if a domain V is not able to distinguish between the states of the system after it has processed a sequence of actions. To fulfil this property, it is required to purge or delete all subsequent actions that are not allowed to interfere with the domain V.

V. HOSPITAL MANAGEMENT ENVIRONMENT

The hospital environment is a complex mixture of Medical professionals (Doctors, Nurses), electronic and non-electronic systems, patients, administrator, Receptionist etc. Data processed in this environment are valuable and might have a negative outcomes attached to them if not handled well enough. The security of this sensitive data is of most importance to hospital staff, management, patients and probably the supporting IT specialists. In this paper, we focus on the basic privacy, security mechanism (RBAC application) that are fundamental of Hospital information management.Hospital electronic and non-electronic records keeping are particularly complex due to the highly sensitive nature of the records and the need to provide maximum protection, while allowing access to the data by a large number of users in the hospital who might need access for specific purposes. Records in hospitals is heavily regulated; due to the sensitive and privacy implications [5].

In this paper the following are defined in relation to the hospital environment:

An Administrator: Is the only one who, has access to control the security system and implement an assignment: He can assign a role to a user, and authorizes or defines a password for users.

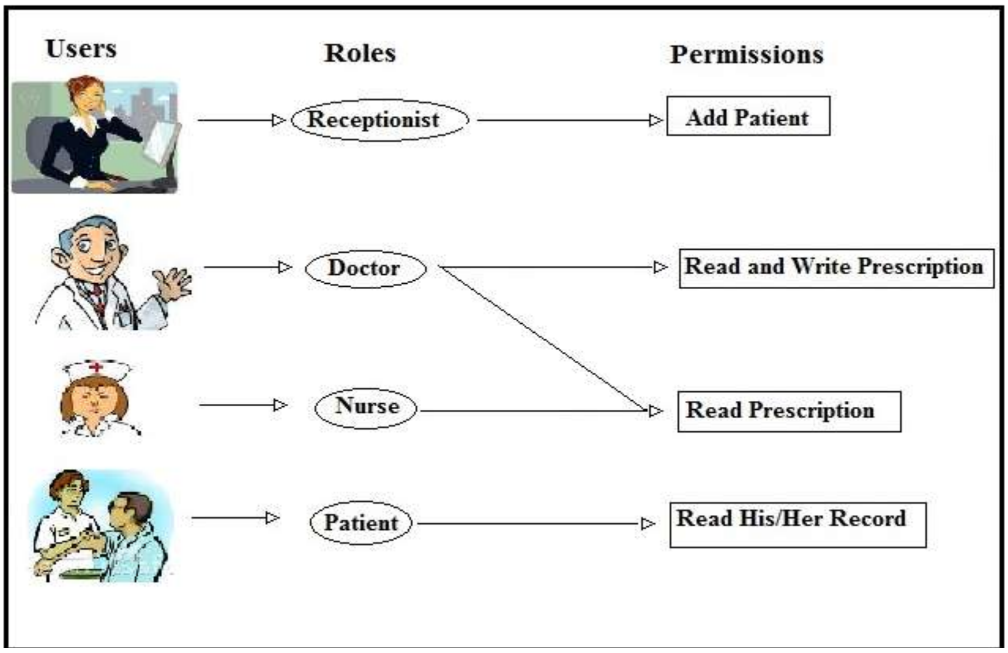
A doctor can read his/her patient's medical record and can also write a prescription, modify information on patients.

A nurse can read the prescription and give an injection where necessary, but he/she have no access to the patients' entire medical records.

The receptionist can register a new patient, write basic information about the patients and save them in the system, but cannot have access to the patients' medical records.

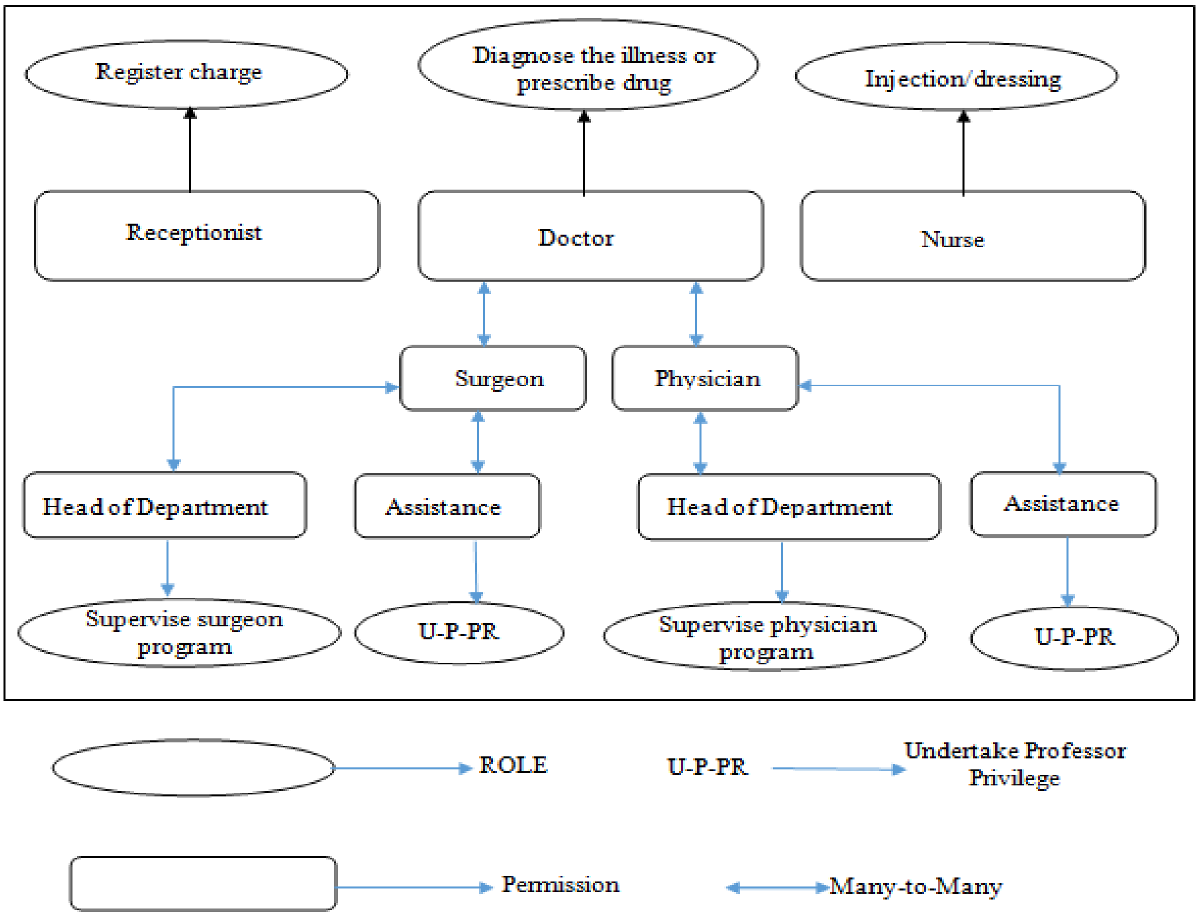
Policy Implementation: enforcing security in the hospital, access control policies define the user's rights on objects. It also defines the identification and authentication of each role. In this model, policies define which permissions are established to roles figure 1.1 in the hospital management. Permission related to role allows the appearance of access authorization in the generic way. Consequently, it is not required that only Doctor X who has access to patient Y records. Instead the Doctor is identified to have right access to Y medical records. The hierarchy of hospital organization, roles and associated permission of RBAC comprises the organization confidential policies. In this context, the hospital management structure is introduced, the implementation and illustrations of the role, sub-system function by data flow diagram.

Figure 1.1: Relationship between Hospital Workers and Privilege



Users are defined as Human figure 1.1 (Doctor, Nurses, Patients, etc.)

Model Description
Figure 2.1: Hierarchy in the hospital



This model illustrates how workers in the hospital are managed, according to their role or job function, the privilege that is assigned. As explained earlier hospital management involves thorough analysis of how the management operates and include input from a wide spectrum of users in the hospital. It would be desirable to have a simpler solution that is easier to configure, maintained and reliably executed.

The figure 2.1 shows the hierarchy of assigning roles and partitioning hospital management into activities as required by hospital management in the implementation of Role-Based-Access Control. It also illustrates the relationship between the working partners (users), subjects, roles which operate in RBAC system. The basic idea underlying hospital management technology is a system that explains the roles, and manages the activities of workflow through the use of designing software. In this model, an administratormanages the system in such a way that it can control the Receptionist, Nurse and Doctors using the RBAC method as shown in the figure… hierarchy[6]. The workflow specified in this model, describes and managed by a workflow management system which enacts each segment in the order specified by the process definition input. The RBAC is used to define Hospital membership of the individual working in the hospital by assigning individuals to roles, assign permissions to roles, and now activate the job function or role with respect to the appropriate points in the sequence. Each user is assigned one or more "ROLES" and each 'ROLE' is assigned one or more 'PERMISSION' that is authorized for the users in that role by the administrator. In this model, permission consist principally of the opportunity to perform operations within an activity of the hospital workflow. Objects, such as files and processes, can be organized into hierarchies. In such object hierarchies, it is important to know not only the access of role group to an object, but also to know whether the path in the hierarchy could be traversed [7].

VI. MODEL DATABASE FOR THE HOSPITAL

In the hospital, all the medical information and patient information is personal and are stored in the database system. The data must only be accessed by the users who are defined or authorized. This is the first step for any information stored and any secure data. To make the queries simple and provide an easy to administer, a decent database must reduce data redundancy.

Figure 3.1, their role tables, user tables and user role table. In the user table, every user has a unique User ID and User Name. Also, in Role Table, every role is determined by a unique Role ID and Role Name.

The relationship between a role and a user is managed by the User Role Table. It may include that one role can comprise many users. If the user decides to have two or more different roles, he/she should have two or more different User Name's, because one user with a single User Name can only be accepted for one role [8].

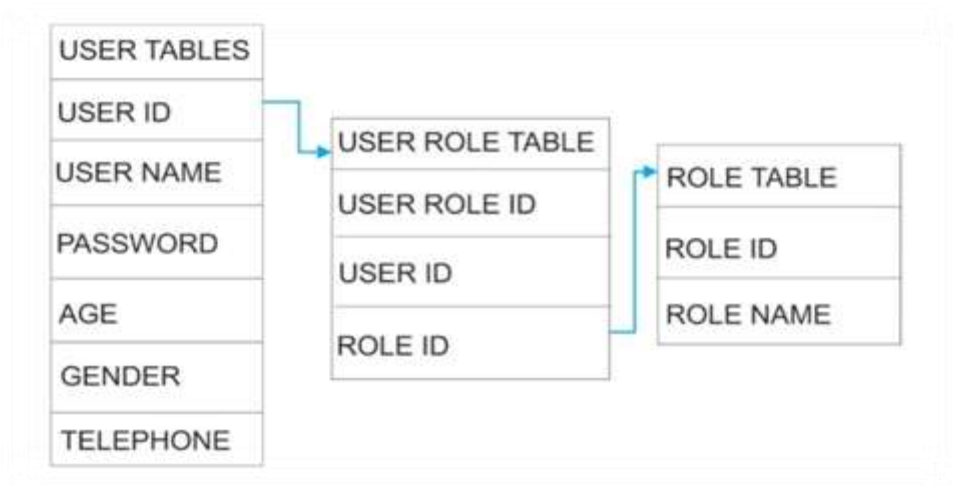


Figure3.1 RBAC database

According to the **figure 3.1,** we employed three roles: Receptionist, Doctor, and Nurses. With the different roles, when the user tries to log into the system, the overall system will check first the user's role, then his/her username and his/her password. If one user tries to log in one domain that he does not have an access, one message will appear directly (the system denied your request because of lack of rights) according to a program that will be installed in the database. A user is able or has the access to change his password once he is already accepted by the system in the data. According to the request imputed correctly, means username and password accepted, the system will appear in the windows. Different functionality is provided by each window. In order to advance the security, the machine that the user log in will turn automatically off if there is no action within on period time (15min). This action can be defined by the administrator at any time. It avoids the leaking of certain information that is stored in the data as personal information and Patient's private record. For instance, at the moment that one user forgot to log off in the system and left his/her office, other can have access by using his/her machine to get some information or to alter a document. There is some information on hospital data that is private, only who has right access should modify.

VII. PARTITIONING OF ROLES IN IMPLEMENTATION

In this section, the typical role of personnel in hospital management is outlined according the model in figure 2.1:

Administrator Role

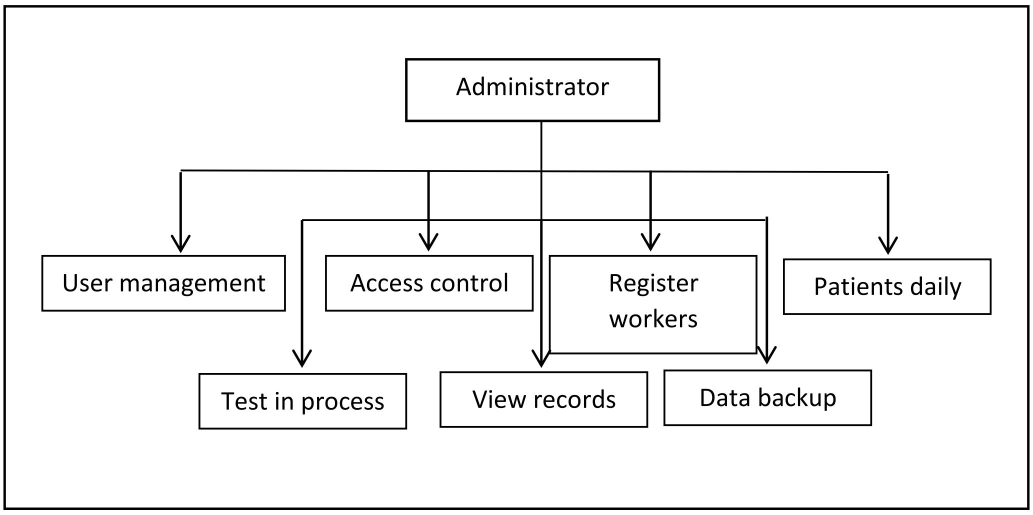


Figure 4.1: Flow Chart for Administrator

VII.1 User Management

In the administrator panel, User management has multiple functions. An administrator can create or add a new user, has the right to modify user's information, and an overview of patient information. The function "add" in the database system allow him to fill in or delete "A new user information" and "save" any action. He can fill directly on his home page all information about users. The administrator has the access to add a User ID, Name, Age, Gender, user type, and telephone number. If it is necessary to modify user's information he can do it. It is only the user who had been successfully assigned to a role can log into the system. Once logged in, he can view the entire inactive patient, their personal information or medical record stored in the data, and the whole program of workers in hospitals.

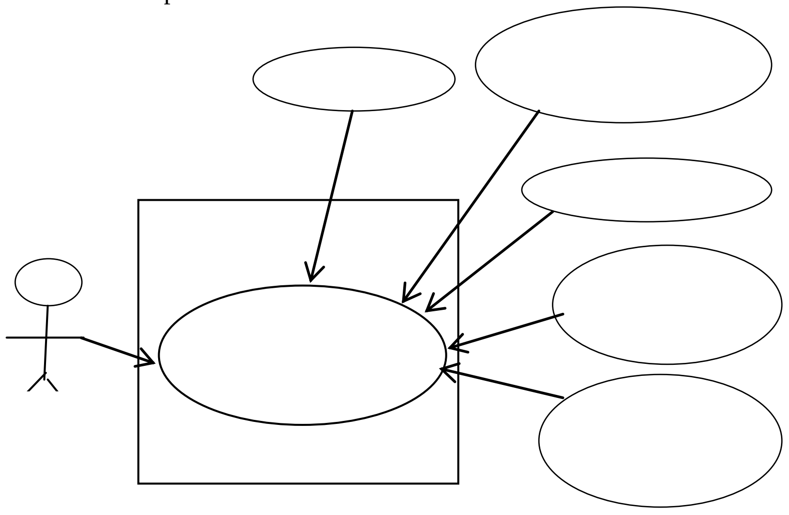


Figure 5.1: User management flow chart

VII. 2. Access control

The workflow includes multiple function, within the administrator one can define first the name andusers types, second assignprivilegese to users. He can define which user has access to which section. The delegate function is the primary function. He can give access, as an example, if user type "Nurse" is allowed or authorized to view, the sys-logs, patient-record, view-all-workers, etc.…

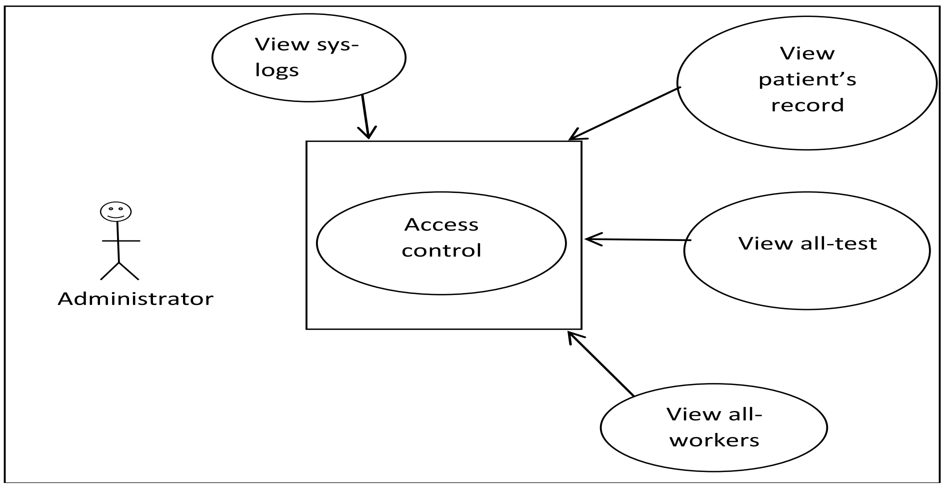


Figure 6.1: Managing Access Control

VII.3. Receptionist role:

The receptionist is the first person that the patient will see when they arrive at the hospital. Usually, he is typically stationed at the front desk.

In the management of the hospital, they have for privilege to register a new patient. They have to fill the information such as name, surname, age, gender, etc.… according to the role and the privilege that will be already defined by the administrator, the receptionist can add a new patient and modify his information if it is necessary. He is also in charge of answering phones, assisting callers with any information they need, etc.… Receptionists have access to open the schedule of appointment times to record the patient information and details regarding the reason for the appointment. Furthermore, he can provide politeness calls to patients to remind them of an upcoming appointment.

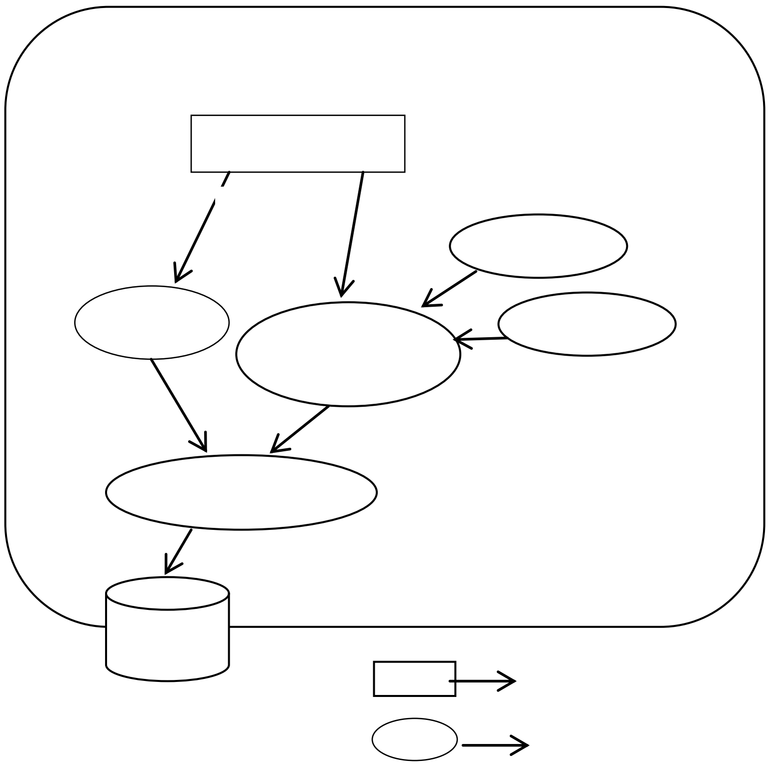


Figure 7.1: Receptionist Role

VII.4. Doctor's Role

In this model, we designate two types of Doctors, Physician and Surgeon, though they are all called 'Doctors' but they work in different domains. The hierarchy in each department is almost the same. Senior roles inherit permission from junior roles through role hierarchy relation as shown in figure 2.1. In figure 8.1 two cases are implemented 'Patient List' and 'Medical Records Form'

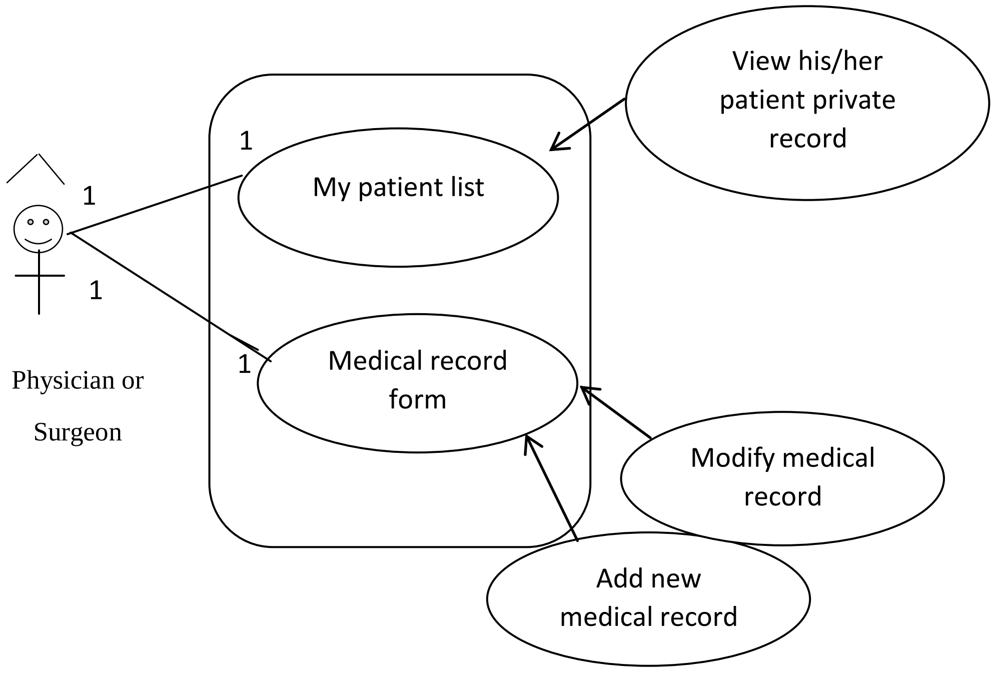


Figure 8.1a: Flow Chart for Doctor (Physician or Doctor)

VII.5. Function of Physician/Surgeon

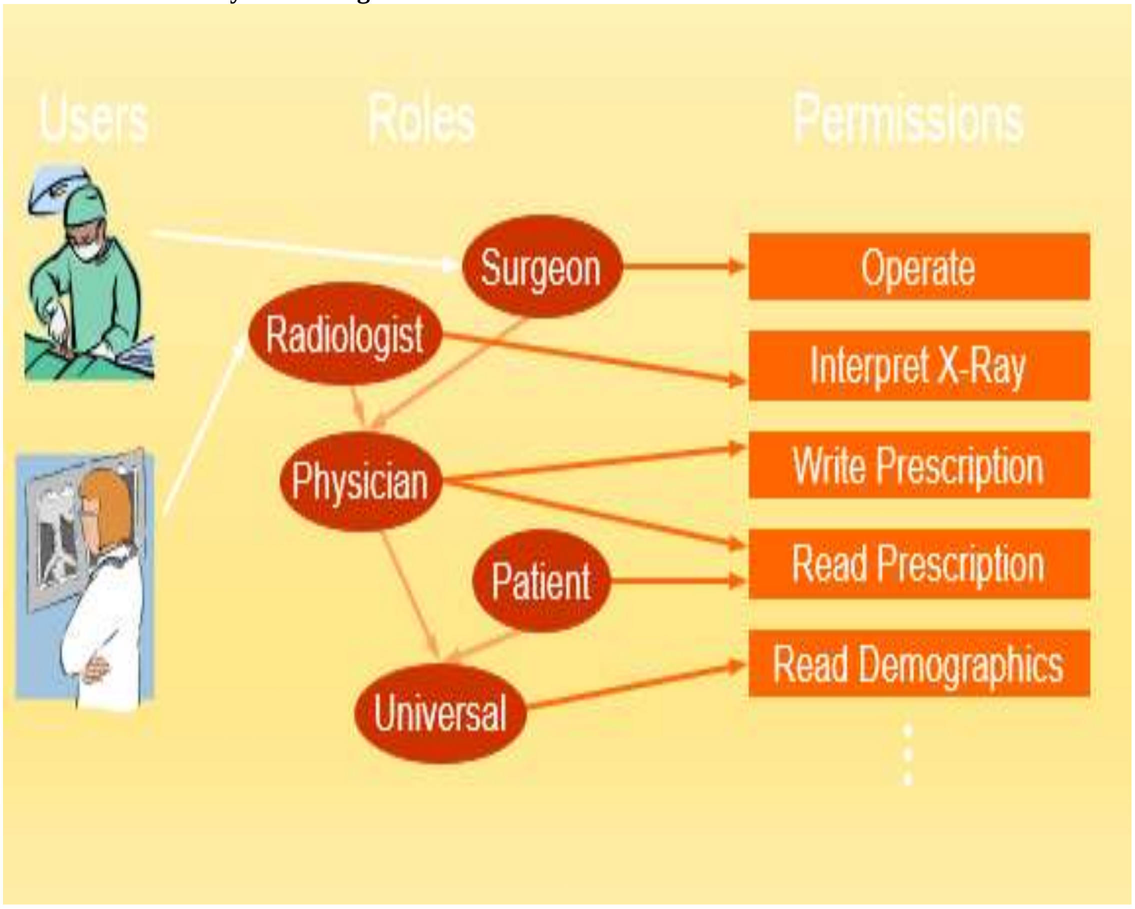


Figure 8.1b: Some general Roles of Doctors in RBAC

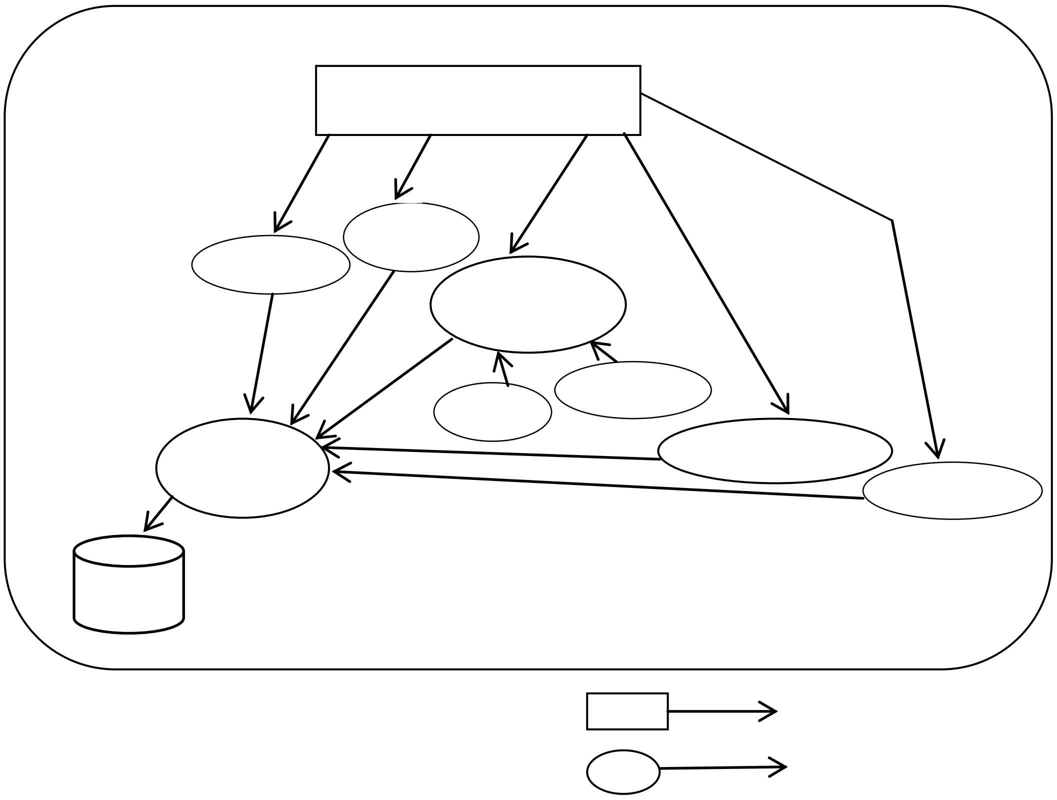


Figure 9.1: Physician/Surgeon Role

VII.6. Nurse Role & Function

In model, the role of the nurse is to offer adequate care for patients to make them feel good while in the hospital. With this role, nurses carry out frequent procedures including patient monitoring patient's vital signs, evaluations, and managing medications. The physician/surgeon prescribes the drug, the nurse does not have that privilege to do it, but he or she has to know when an order, a medication, a dosage, etc. According to figure 10.1 nurses are authorized to give injection/dressing for the patient. Once he/she logs onto the system in the hospital, he/she can open the schedule of patients.

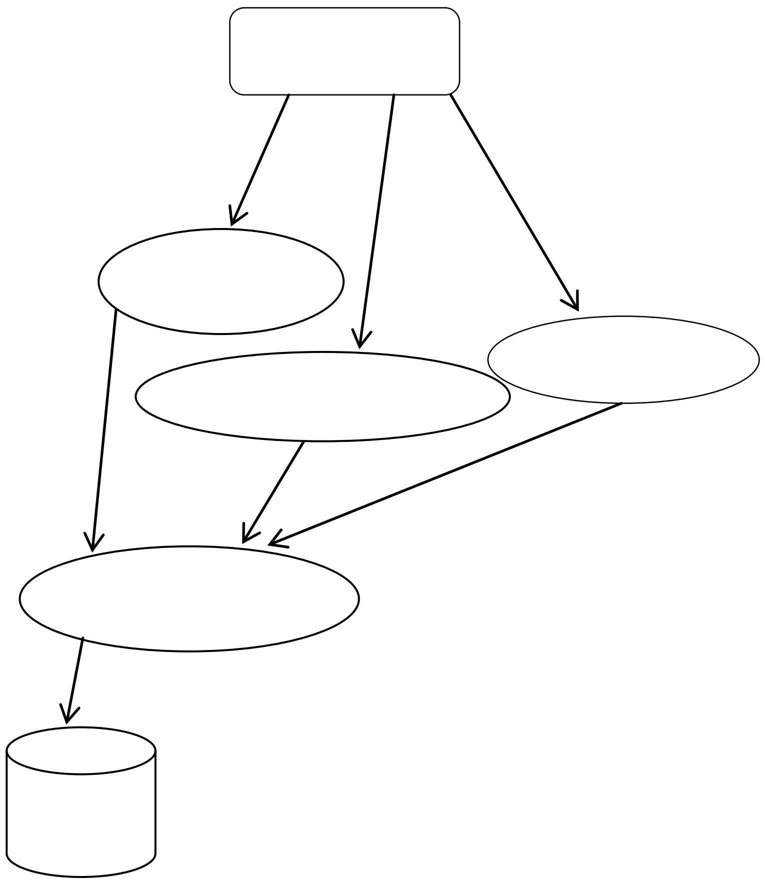


Figure 10.1: Outline of the Nurses Role

VIII. LANGUAGE

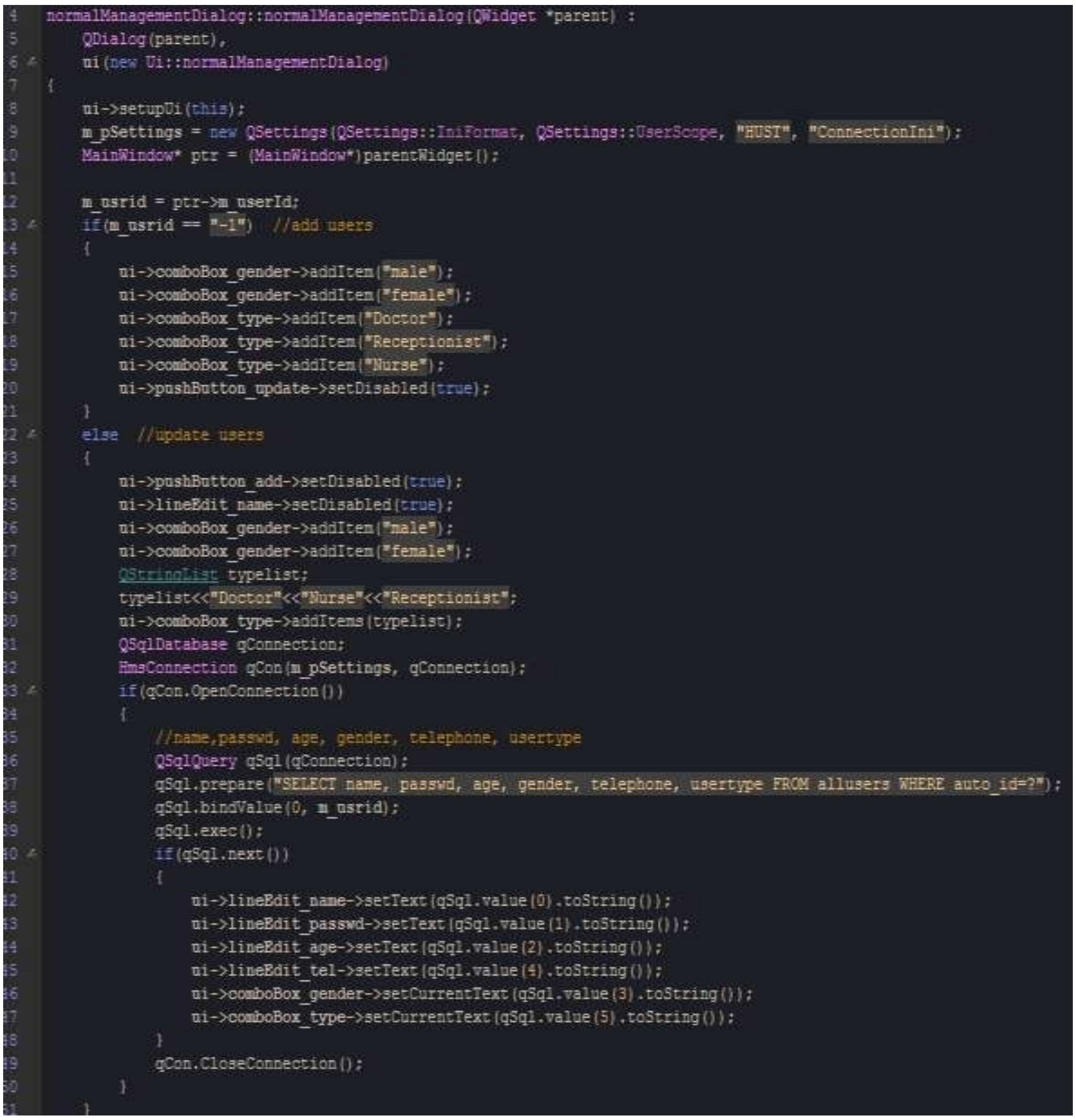
In order to tackle the proposed management structure of RBAC integrations within object data models, we propose a generic language MySQL and C++, allowing expression of RBAC authorizations and integrating access control mechanism [9]. The declaration part of the language is composed of: the body, which relies on C++ syntax while adding access authorization; the header which defines the roles which are to be used within the definition access authorization. In addition, C++ code is used to set up a connection to MySQL database and access stored function.

X. RESULTS

Source program for Administrator

Register Workers, Patients daily, Test in process, view records and back up some details of users and view the saved process. The figure is a construction of the dialog class:

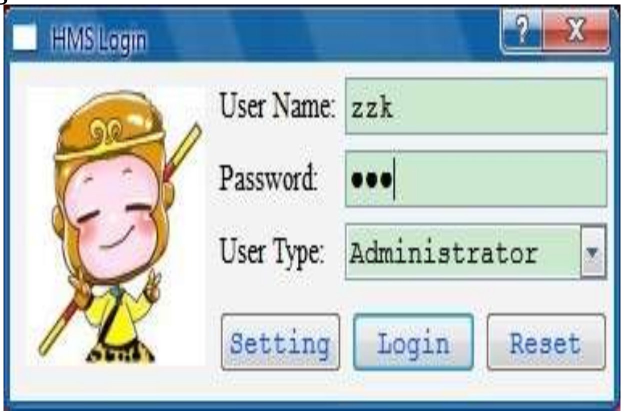
In the administrator personal interface the home page shows user's personal information. In the source code, the method permits much of the generality of the RBAC concept of 'action' to be realized. When the basic actions on have been established, any conditions permitting actions specified in an RBAC policy will be implemented. In addition, this approach allows roles to be created, removed, and modified without having to recompile either the application or the basic access methods class.



```
normalManagementDialog::normalManagementDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::normalManagementDialog)
{
    ui->setupUi(this);
    m_pSettings = new QSettings(QSettings::IniFormat, QSettings::UserScope, "HUST", "ConnectionIni");
    MainWindow* ptr = (MainWindow*)parentWidget();

    m_usrid = ptr->m_userId;
    if(m_usrid == "-1")  //add users
    {
        ui->comboBox_gender->addItem("male");
        ui->comboBox_gender->addItem("female");
        ui->comboBox_type->addItem("Doctor");
        ui->comboBox_type->addItem("Receptionist");
        ui->comboBox_type->addItem("Nurse");
        ui->pushButton_update->setDisabled(true);
    }
    else  //update users
    {
        ui->pushButton_add->setDisabled(true);
        ui->lineEdit_name->setDisabled(true);
        ui->comboBox_gender->addItem("male");
        ui->comboBox_gender->addItem("female");
        QStringList typelist;
        typelist<<"Doctor"<<"Nurse"<<"Receptionist";
        ui->comboBox_type->addItems(typelist);
        QSqlDatabase qConnection;
        HmsConnection qCon(m_pSettings, qConnection);
        if(qCon.OpenConnection())
        {
            //name,passwd, age, gender, telephone, usertype
            QSqlQuery qSql(qConnection);
            qSql.prepare("SELECT name, passwd, age, gender, telephone, usertype FROM allusers WHERE auto_id=?");
            qSql.bindValue(0, m_usrid);
            qSql.exec();
            if(qSql.next())
            {
                ui->lineEdit_name->setText(qSql.value(0).toString());
                ui->lineEdit_passwd->setText(qSql.value(1).toString());
                ui->lineEdit_age->setText(qSql.value(2).toString());
                ui->lineEdit_tel->setText(qSql.value(4).toString());
                ui->comboBox_gender->setCurrentText(qSql.value(3).toString());
                ui->comboBox_type->setCurrentText(qSql.value(5).toString());
            }
            qCon.CloseConnection();
        }
    }
```

Figure 11.1: Construction dialog Class

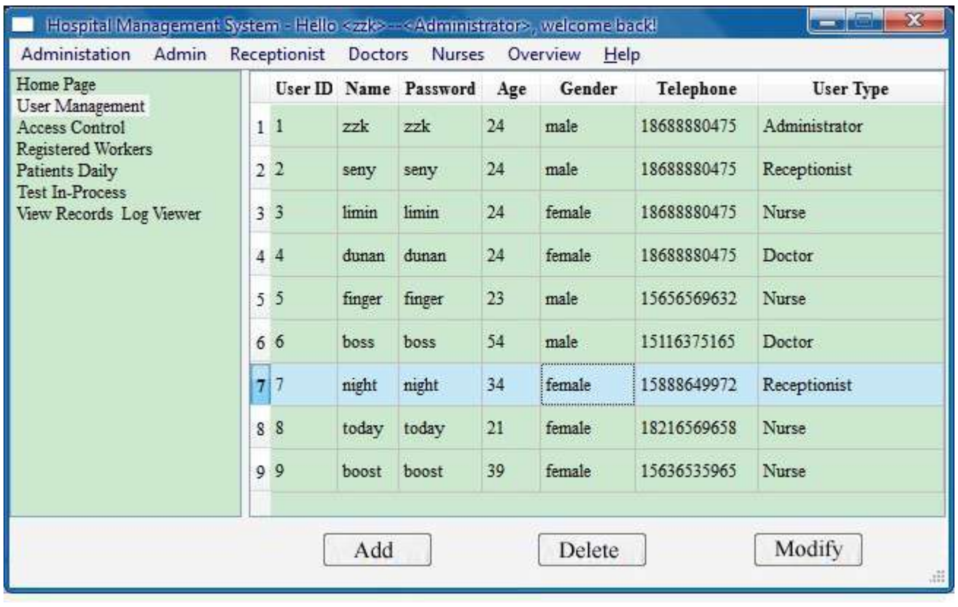
VIV.1. Administrator Login Result



User Name: zzk
Password: •••
User Type: Administrator

Figure 12.1: Process of Login

Figure 12.1 shows the administrator Login Page, it has the User Name, Password and User Type. In the User type one can select from these four options: Administrator, Receptionist, Doctor, and Nurse etc. On bottom Panel, one can select these function bars: Setting, Login and Reset.



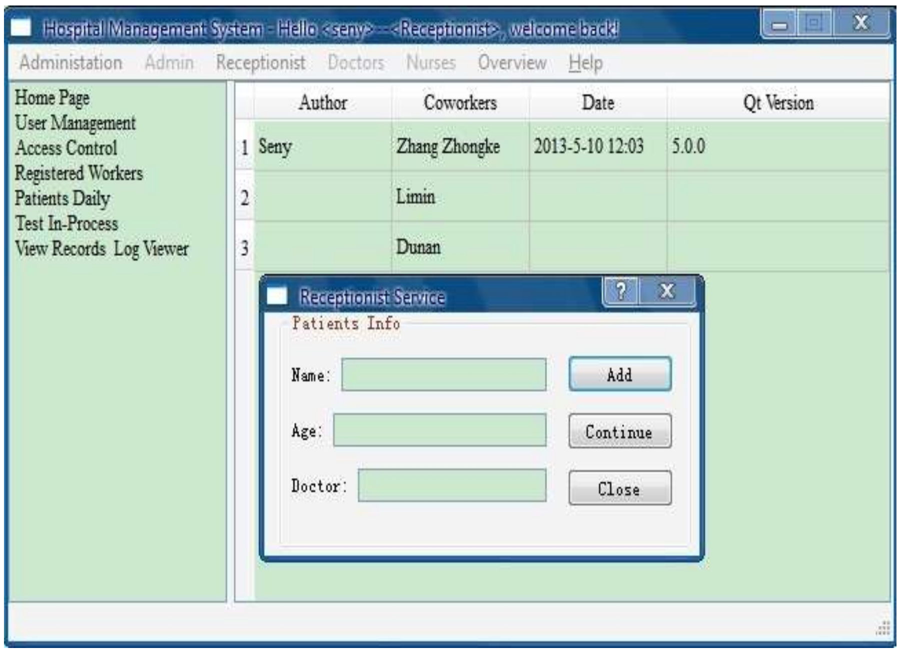
	User ID	Name	Password	Age	Gender	Telephone	User Type
1	1	zzk	zzk	24	male	18688880475	Administrator
2	2	seny	seny	24	male	18688880475	Receptionist
3	3	limin	limin	24	female	18688880475	Nurse
4	4	dunan	dunan	24	female	18688880475	Doctor
5	5	finger	finger	23	male	15656569632	Nurse
6	6	boss	boss	54	male	15116375165	Doctor
7	7	night	night	34	female	15888649972	Receptionist
8	8	today	today	21	female	18216569658	Nurse
9	9	boost	boost	39	female	15636535965	Nurse

Figure 13.1: Administrator Login Interface

Figure 13.1 shows the administrator Login Interface, some data tables like User ID, Name, Password, Age, Gender, Telephone number and User Type can be accessed by the administrator. He can add, Delete and modify existing user information.

VIV.2. GUI of Receptionist (Graphical User Interface)

Limitation of the receptionist by adding new patient information and as is stored in its database, the receptionist can no longer access.



	Author	Coworkers	Date	Qt Version
1	Seny	Zhang Zhongke	2013-5-10 12:03	5.0.0
2		Limin		
3		Dunan		

Name:

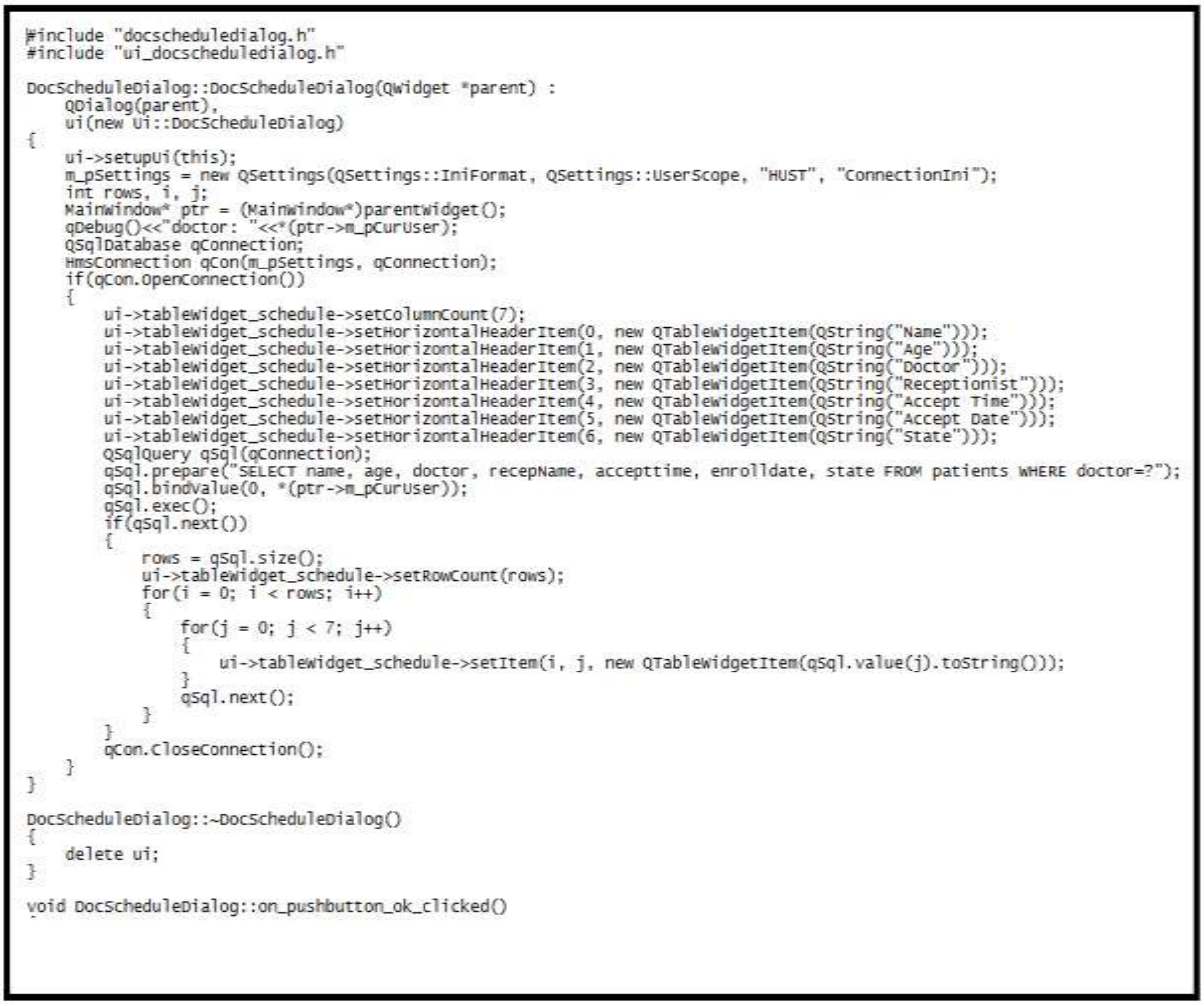
Age:

Doctor:

Figure 14.1: GUI of Receptionist

VIV.3. Doctor GUI

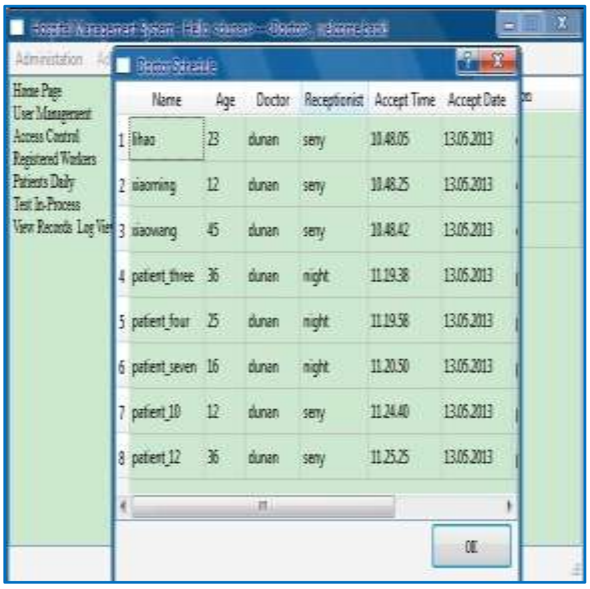
In this source code, there are two methods used which provides hospital application access to patients records: GetIdinfo( ) – this provides a list of patients names and their IDs; GetPr(pid) – Given a patient ID, returns to patient records. Figure 15.1 is the role classes associated with patients (Pat_PRDBO) and Doctor role (Doc_PRDBO). These role classes inherit from a base class(Role_PRDBO) which defines the types, names, and parameters corresponding to the methods in the basic access method class. The doctor and patient role classes together implement the following RBAC policy. Only the doctor is permitted to read and modify the patients' records. Also authorized to diagnose the patient and prescribe the drug.



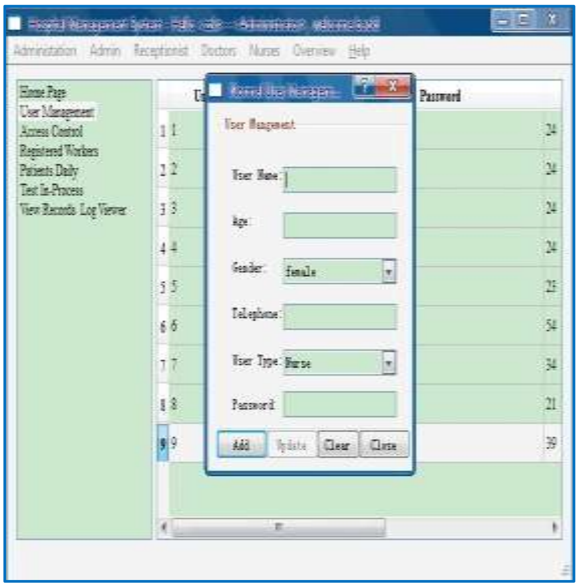
```
#include "docscheduledialog.h"
#include "ui_docscheduledialog.h"

DocScheduleDialog::DocScheduleDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::DocScheduleDialog)
{
    ui->setupUi(this);
    m_pSettings = new QSettings(QSettings::IniFormat, QSettings::UserScope, "HUST", "ConnectionIni");
    int rows, i, j;
    MainWindow* ptr = (MainWindow*)parentWidget();
    qDebug()<<"doctor: "<<*(ptr->m_pCurUser);
    QSqlDatabase qConnection;
    HmsConnection qCon(m_pSettings, qConnection);
    if(qCon.openConnection())
    {
        ui->tableWidget_schedule->setColumnCount(7);
        ui->tableWidget_schedule->setHorizontalHeaderItem(0, new QTableWidgetItem(QString("Name")));
        ui->tableWidget_schedule->setHorizontalHeaderItem(1, new QTableWidgetItem(QString("Age")));
        ui->tableWidget_schedule->setHorizontalHeaderItem(2, new QTableWidgetItem(QString("Doctor")));
        ui->tableWidget_schedule->setHorizontalHeaderItem(3, new QTableWidgetItem(QString("Receptionist")));
        ui->tableWidget_schedule->setHorizontalHeaderItem(4, new QTableWidgetItem(QString("Accept Time")));
        ui->tableWidget_schedule->setHorizontalHeaderItem(5, new QTableWidgetItem(QString("Accept Date")));
        ui->tableWidget_schedule->setHorizontalHeaderItem(6, new QTableWidgetItem(QString("State")));
        QSqlQuery qSql(qConnection);
        qSql.prepare("SELECT name, age, doctor, recepName, accepttime, enrolldate, state FROM patients WHERE doctor=?");
        qSql.bindValue(0, *(ptr->m_pCurUser));
        qSql.exec();
        if(qSql.next())
        {
            rows = qSql.size();
            ui->tableWidget_schedule->setRowCount(rows);
            for(i = 0; i < rows; i++)
            {
                for(j = 0; j < 7; j++)
                {
                    ui->tableWidget_schedule->setItem(i, j, new QTableWidgetItem(qSql.value(j).toString()));
                }
                qSql.next();
            }
        }
        qCon.closeConnection();
    }
}

DocScheduleDialog::~DocScheduleDialog()
{
    delete ui;
}

void DocScheduleDialog::on_pushButton_ok_clicked()
```

Figure 15.1 Source Code for Doctor

Once the doctor Logs onto the system, a data schedule dialog pops up, where he/she can view the new and old patient medical records, where he can prescribe drugs, diagnose and assign lab test for patients. Only the administrator and the Doctor have access to the patients' medical records.



	Name	Age	Doctor	Receptionist	Accept Time	Accept Date
1	lihao	23	dunan	seny	10.48.05	13.05.2013
2	xiaoming	12	dunan	seny	10.48.25	13.05.2013
3	xiaowang	45	dunan	seny	10.48.42	13.05.2013
4	patient_three	36	dunan	night	11.19.38	13.05.2013
5	patient_four	25	dunan	night	11.19.58	13.05.2013
6	patient_seven	16	dunan	night	11.20.50	13.05.2013
7	patient_10	12	dunan	seny	11.24.40	13.05.2013
8	patient_12	36	dunan	seny	11.25.25	13.05.2013



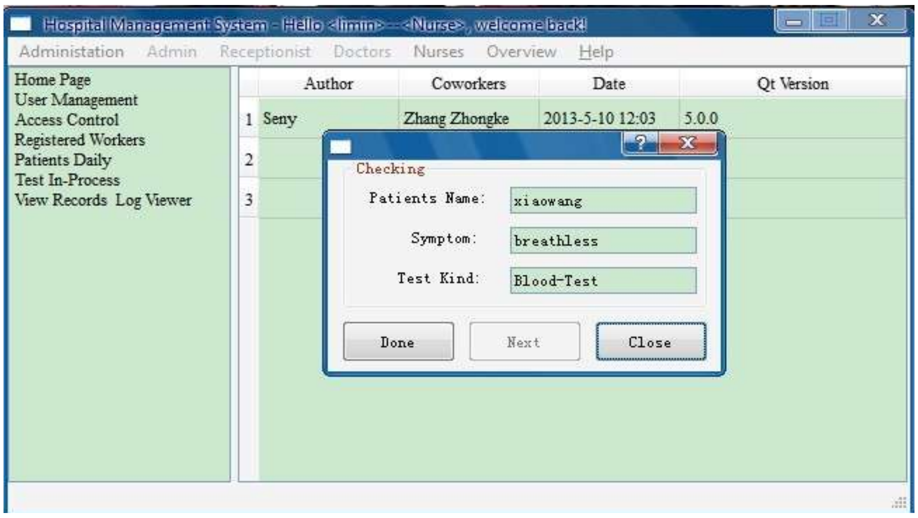
User Management

User Name:	
Age:	
Gender:	female
Telephone:	
User Type:	Nurse
Password:	

Figure 16.1: Interface of Doctors Schedule and Interface of filling medical Records

GUI of Nurse

When a Nurse logsinn successfully, the interface of theNurse'se page looks like figure 18.1. The nurse can access the patient record only if it has been administered by the doctor. A dialogue pops up showing the patients Name, Symptom and Test. The moment the medical record is saved, the nurse no longer has access to the patient information.

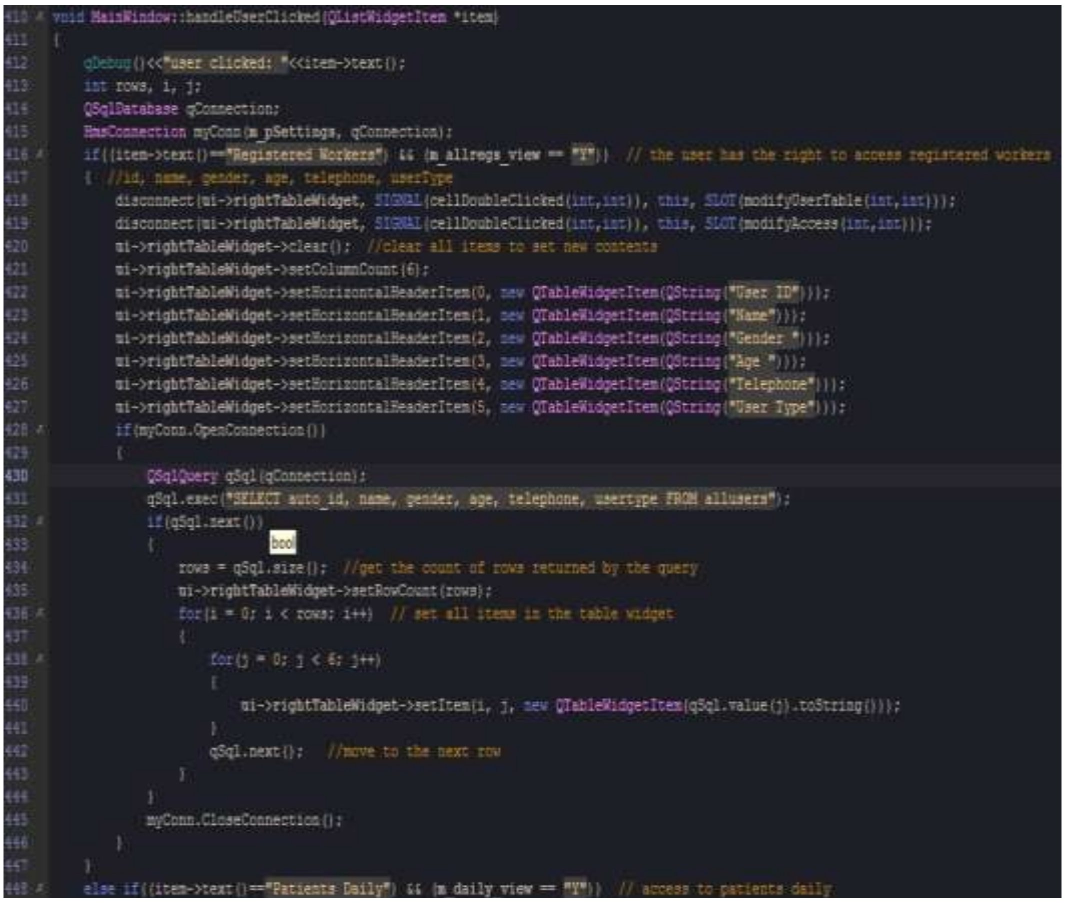


	Author	Coworkers	Date	Qt Version
1	Seny	Zhang Zhongke	2013-5-10 12:03	5.0.0
2				
3				

Patients Name: xiaowang
Symptom: breathless
Test Kind: Blood-Test

Figure 17.1: RBAC Policy Code for Hospital Management

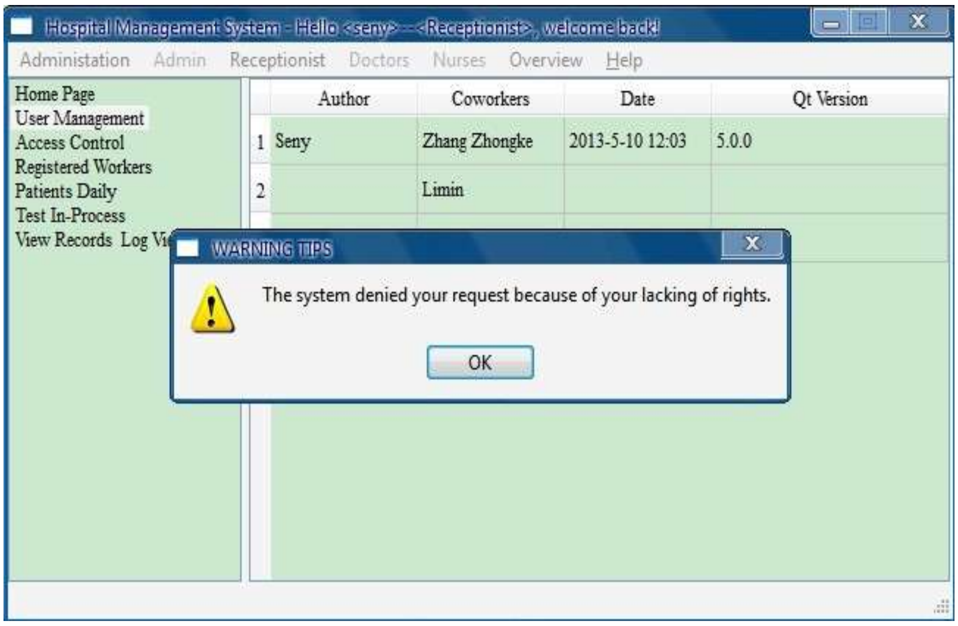
Figure 17.1 is the Access control Policy Code that contains a handle User Clicked () function. This function is designed to implement the access control policy, whatever is typed in by the User, the program will show a warning to tell the user whether or not he/she can access the item because of the privilege is denied by the administrator or authorized to manage the page.



```
void MainWindow::handleUserClicked(QListWidgetItem *item)
{
    qDebug()<<"user clicked: "<<item->text();
    int rows, i, j;
    QSqlDatabase qConnection;
    HmsConnection myConn(m_pSettings, qConnection);
    if((item->text()=="Registered Workers") && (m_allregs_view == "Y"))  // the user has the right to access registered workers
    {  //id, name, gender, age, telephone, userType
        disconnect(ui->rightTableWidget, SIGNAL(cellDoubleClicked(int,int)), this, SLOT(modifyUserTable(int,int)));
        disconnect(ui->rightTableWidget, SIGNAL(cellDoubleClicked(int,int)), this, SLOT(modifyAccess(int,int)));
        ui->rightTableWidget->clear();  //clear all items to set new contents
        ui->rightTableWidget->setColumnCount(6);
        ui->rightTableWidget->setHorizontalHeaderItem(0, new QTableWidgetItem(QString("User ID")));
        ui->rightTableWidget->setHorizontalHeaderItem(1, new QTableWidgetItem(QString("Name")));
        ui->rightTableWidget->setHorizontalHeaderItem(2, new QTableWidgetItem(QString("Gender ")));
        ui->rightTableWidget->setHorizontalHeaderItem(3, new QTableWidgetItem(QString("Age ")));
        ui->rightTableWidget->setHorizontalHeaderItem(4, new QTableWidgetItem(QString("Telephone")));
        ui->rightTableWidget->setHorizontalHeaderItem(5, new QTableWidgetItem(QString("User Type")));
        if(myConn.OpenConnection())
        {
            QSqlQuery qSql(qConnection);
            qSql.exec("SELECT auto_id, name, gender, age, telephone, usertype FROM allusers");
            if(qSql.next())
            {
                rows = qSql.size();  //get the count of rows returned by the query
                ui->rightTableWidget->setRowCount(rows);
                for(i = 0; i < rows; i++)  // set all items in the table widget
                {
                    for(j = 0; j < 6; j++)
                    {
                        ui->rightTableWidget->setItem(i, j, new QTableWidgetItem(qSql.value(j).toString()));
                    }
                    qSql.next();  //move to the next row
                }
            }
            myConn.CloseConnection();
        }
    }
    else if((item->text()=="Patients Daily") && (m_daily_view == "Y"))  // access to patients daily
```

Figure 18.1: RBAC Policy Code

GUI for a Non Authorized User



Hospital Management System - Hello <seny>---<Receptionist>, welcome back!

	Author	Coworkers	Date	Qt Version
1	Seny	Zhang Zhongke	2013-5-10 12:03	5.0.0
2		Limin		

WARNING TIPS

The system denied your request because of your lacking of rights.

Figure 19.1: GUI Non-Authorized

X. CONCLUSION

Information systems in Hospital Management have unique specific security and privacy requirements. If management would like to apply RBAC in this information system to reduce the administrative tasks and manage the smooth running of the hospital, then several context awareness, strong personalization of access control policies that is efficient, flexible and fairly generic must be adopted. Other issues that should be looked at is control of data sharing in an open distributed environment. We therefore propose this model of RBAC, this approach increases the data availability, confidentiality, integrity, accountability, which is the most important requirement in the hospital management. In addition, we have also proposed a program that takes permission evaluation when conflicting roles are present.

Managing the hospital have come a long way from serving principally as a means of making it easier to manage access to applications. As the growing number of employees' roles-driven projects indicates, RBACs are increasingly likely to address critical management objectives such as greater cost efficiencies, improved compliance, and reduce security exposure. Working as part of an integrated, automated role-management and identity-management solution, RBAC can go a long way toward helping avert potential management catastrophes in increasing collaborative and complex hospital environments.

REFERENCES

[1]. John A. Miller, Mei Fan, ShengliWu,IsmailcemB. Arpinar, Amit P.Sheth, Krys J. Kochut, "Security for the METEOR Workflow Management System", Large Scale Distributed Information Systems Lab (LSDIS), Department of Computer Science, the University ofGeorgia, http://LSDIS.cs.uga.edu.

[2]. Patrick Brézillon1 and GhitaKouadriMostéfaoui, "Context-Based Security Policies: A New Modeling Approach", Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW'04), IEEE, 2004, pages 154 Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual, International, vol. 1, 2004, 72-77.

[3]. http://csrc.nist.gov/groups/SNS/rbac/documents/design_implementation/ Intro. Role-based.

[4]. DSOM (1994). Proceedings of the IEEE/IFIP Distributed Systems operations and Management, Toulouse (France).

[5]. Dick, R., Steen, S., Elaine, B., &Detmer, D. E. (1997). The computer-based patient record: An essential technology for health care.

[6]. Hongmei. Chi, Edward, I., and Landz "Implementation of security Access Control for inter-organization Healthcare Information System" IEEE 2008, pp 692 – 699.

[7]. Lin Y. and Ma, Y. (1990). 'Remarks on controllabilities of general systems', *International J. General Systems*, Vol.17 No. 4, pp. 317-27.

[8]. John Barkley, "Application Engineering in Health Care," *Proceedings for the 2nd Annual CHIN Summit, 1995.*

[9]. ISO/IEC 9075, (Working Draft) Database, Language SQL – Part 2: Foundation, Document ISO/IEC

JTC/SC21 N9463, March 1995.

[10]. Wang, J., Osborn, S.L. "A Role-Based Approach to Access Control for XML Databases", ACM SACMAT, 2004, page 4 – 7.

[11]. http//www.esat.kuleuven.ac.be/cosic/sesame/doc-txt/overview.txt.

[12]. American National Standard, 359 – 2004, Information technology-role based access control. National Academy Press-Book. ISBN 0309055326. Washington, D. C

[13]. D. Simons, T. Egami, J. Perry, "Remote Patient Monitoring Solution", in G. Spekowius and T. Wendler (Eds*), Advances in Healthcare Technology, Spring, 2006, pp. 505 – 516.*

[14]. XieYumin;, "An Access Rights Administration Model in Role-Based Security Systems", Information Engineering and Computer Science, 2009. ICIECS 2009, International Conference, pp. 1 – 4, Dec. 2009