SECURE AND SCALABLE SHARING OF PERSONAL HEALTH RECORDS USING ATTRIBUTE BASED ENCRYPTION IN CLOUD

##### A PROJECT REPORT

###### ***Submitted by***

**T.SRI HARI REDDY (11310104107)**

**C.P.SUBODH KRISHNA (11310104109)**

**VARUN RAPARLA (11310104114)**

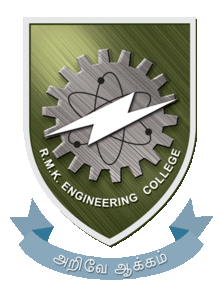
***in partial fulfillment for the award of the degree***

***of***

##### BACHELOR OF TECHNOLOGY

IN

**COMPUTER SCIENCE AND ENGINEERING**

[](http://en.wikipedia.org/wiki/File:RMK_En) 

**R.M.K ENGINEERING COLLEGE,KAVARAIPETTAI**

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2014

**BONAFIDE CERTIFICATE**

Certified that this project report **“SECURE AND SCALABLE SHARING OF PERSONAL HEALTH RECORDS USING ATTRIBUTE BASED ENCRYPTION IN CLOUD”** is the bonafide work of **“T.SRI HARI REDDY(11310104107) , C.P.SUBODH KRISHNA(11310104109) , VARUN RAPARLA(11310104114)”** who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**DR.K.L.SHANMUGANATHAN N.INDIRA**

**M.E.,M.S.,F.I.E.,Ph.D** (ASST. PROFESSOR)

**HEAD OF THE DEPARTMENT SUPERVISOR**

Professor & Head, Assistant Professor,

Computer Science & Engineering Computer Science & Engineering

R.M.K Engineering College, R.M.K Engineering College,

R.S.M Nagar, R.S.M Nagar,

Kavaraipettai-601 206 Kavaraipettai- 601 206

Submitted for the Project Viva-voce held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EXTERNAL EXAMINER INTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We convey our thanks to our Chairman **Thiru.R.S.Munirathinam** and Vice-chairman **Mr.R.M.Kishore** who took keen interest on us and encouraged throughout the course of study and for their attention and valuable suggestions offered throughout the course of study.We convey our sincere thanks to our Director **Thiru.R.Jothi Naidu** for being a great support to our college.

We express our sincere thanks to our principal **Dr.Elwin Chandra Monie** for his whole hearted and kind co-operation.

We are highly thankful to our Head of the Department , our project supervisor and our project internal guide **N.INDIRA B.E,M.E,** for supporting and rendering us with all the facilities for undertaking the project. We are also thankful for her support.

Our heartful thanks to our project coordinator who always encouraged **Dr.C.Jayakumar M.E., Ph.D.,** for his great support and guidance.We extend our thanks to all the faculties of the Department of Computer Science and Engineering who were behind us throughout the course of study.

**ABSTRACT**

Personal health record (PHR) is an emerging patient-centric model of health information exchange, which is often outsourced to be stored at a third party, such as cloud providers. However, there have been wide privacy concerns as personal health information could be exposed to those third party servers and to unauthorized parties. To assure the patients’ control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing. Yet, issues such as risks of privacy exposure, scalability in key management, flexible access and efficient user revocation, have remained the most important challenges toward achieving fine-grained, cryptographically enforced data access control. In this paper, we propose a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semi-trusted servers. To achieve fine-grained and scalable data access control for PHRs, we leverage attribute based encryption (ABE) techniques to encrypt each patient’s PHR file. Different from previous works in secure data outsourcing, we focus on the multiple data owner scenario, and divide the users in the PHR system into multiple security domains that greatly reduces the key management complexity for owners and users. A high degree of patient privacy is guaranteed simultaneously by exploiting multi-authority ABE. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE**  **NO.** |
|  | **ABSTRACT** | 4 |
|  | **LIST OF FIGURES** | 7 |
|  | **LIST OF ABBREVIATIONS** | 8 |
| **1** | **INTRODUCTION**    1. 1 OVERVIEW OF THE PROJECT  1. 2 EXISTING SYSTEM 1. 3 PROPOSED SYSTEM 1. 4 FEASIBILITY STUDIES  1. 5 REQUIREMENT SPECIFICATION | 9  9  9  10  11  12 |
| **2** | **DESIGN**  2.1 OVERALL SYSTEM DESIGN  2.1.1 ARCHITECTURE DIAGRAM  2.1.2 LOGICAL DATABASE DESIGN  2.1.3 E-R DIAGRAM  2.1.4 DATAFLOW DIAGRAM  2.2 DETAILED DESIGN  UML DIAGRAMS  2.2.1 USE CASE DIAGRAMS  2.2.2 CLASS DIAGRAM  2.2.3 ACTIVITY DIAGRAM  2.2.4 SEQUENCE DIAGRAM  2.2.5 COLLABORATION DIAGRAM | 14  14  14  16  19  20  21  22  23  24  26  27 |
| **3** | **IMPLEMENTATION**  3.1 MODULES  3.2 MODULE DESCRIPTION  3.2.1 PHR OWNER MODULE  3.2.2 CLOUD SERVER MODULE  3.2.3 ATTRIBUTE BASED ACCESS POLICY MODULE  3.2.4 DATA CONFIDENTIALITY MODULE | 28  28  28  28  29  29  30 |
| **4** | **TESTING**  4.1 TYPES OF TESTS  4.1.1 UNIT TESTING  4.1.2 INTEGRATION TESTING  4.1.3 FUNCTIONAL TESTING  4.1.4 SYSTEM TESTING  4.1.5 WHITE BOX TESTING  4.1.6 BLACK BOX TESTING | 31  31  31  31  31  31  32  32 |
| **5** | **APPENDIX A- SOFTWARE DESCRIPTION** | 33 |
| **6** | **APPENDIX B-SAMPLE CODE** | 37 |
| **7** | **APPENDIX C-SCREEN SHOTS** | 53 |
| **8** | **REFERENCES** | 58 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 2.1.1 | ARCHITECTURE DIAGRAM  CLASS DIAGRAM | 14 |
| 2.1.3 | E-R DIAGRAM | 19 |
| 2.1.4 | DATAFLOW DIAGRAM | 20 |
| 2.2.1 | USE CASE DIAGRAMS | 21 |
| 2.2.2 | CLASS DIAGRAM | 22 |
| 2.2.3 | ACTIVITY DIAGRAM | 23 |
| 2.2.4 | SEQUENCE DIAGRAM | 26 |
| 2.2.5 | COLLABORATION DIAGRAM | 27 |

**LIST OF ABBREVIATIONS**

**S.NO ABBREVIATIONS EXPANSION**

1. **UP** Utility pattern
2. **UML**  Unified modeling

language

1. **DFD** Data flow diagram
2. **ERM** Entity relations model
3. **TCP** transmission

control protocol

1. **UDP** User datagram

protocol

**7. PHR** personal health records

**INTRODUCTION**

**1. 1 OVERVIEW OF THE PROJECT**

Personal health information could be exposed to those third party servers and to unauthorized parties to assure the patients' control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing. Yet, issues such as risks of privacy exposure, scalability in key management, flexible access, and efficient user revocation, have remained the most important challenges toward achieving fine-grained, cryptographically enforced data access control. In this paper, we propose a novel patient-centric framework and a suite of mechanisms for data

**1.2 EXISTING SYSTEM:**

personal health record (PHR) has emerged as a patient-centric model of health information exchange. A PHR service allows a patient to create, manage, and control her personal health data in one place through the web, which has made the storage, retrieval, and sharing of the the medical information more efficient. Especially, each patient is promised the full control of her medical records and can share her health data with a wide range of users, including healthcare providers, family members or friends. Due to the high cost of building and maintaining specialized data centers, many PHR services are outsourced to or provided by third-party service providers, The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the other hand, due to the high value of the sensitive personal health information (PHI), the third-party storage servers are often the targets of various malicious behaviors which may lead to exposure of the PHI and security numbers and health problems was stolen by an employee who took the data home without authorization.

**DISADVANTAGES:**

1. patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. Because a third-party server inside hackers can able to leak the patients information and security records to other peoples so this scheme is not fully trust.
2. In Existing attribute-based encryption important issues such as key management scalability, dynamic policy updates, and efficient on-demand revocation are non-trivial to solve, and remain largely open up-to-date. To this end, we make the following main contributions:

**1.3 PROPOSED SYSTEM:**

We propose a novel ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multi-owner settings. To address the key management challenges, we conceptually divide the users in the system into two types of domains, namely *public* and *personal domains*. In particular, the majority professional users are managed distributively by attribute authorities in the former, while each owner only needs to manage the keys of a small number of users in her personal domain. In this way, our framework can simultaneously handle different types of PHR sharing applications’ requirements, while incurring minimal key management overhead for both owners and users in the system. In addition, the framework enforces write accesscontrol, handles dynamic policy updates, and provides break-glass access to PHRs under emergence scenarios.

In the public domain, we use multi-authority ABE (MA-ABE) to improve the security

and avoid key escrow problem. Each attribute authority (AA) in it governs a disjoint subset of user role attributes, while none of them alone is able to control the security of the whole system. We propose mechanisms for key distribution and encryption so that PHR owners can specify personalized fine-grained role-based access policies during file encryption. In the personal domain, owners directly assign access privileges for personal users and encrypt a PHR file under its data attributes. Furthermore, we enhance MA-ABE by putting forward an efficient and on-demand user/attribute revocation scheme, and prove its security under standard security assumptions. In this way, patients have full privacy control over their PHRs.

**ADVANTAGES:**

Our proposed secure PHR sharing solution, in terms of multiple metrics in computation, communication, storage and key management. We also compare our scheme to several previous ones in complexity, scalability and security. Furthermore, we demonstrate the efficiency of our scheme by implementing it on a modern workstation and performing experiments/simulations.

**1. 4 FEASIBILITY STUDIES**

A feasibility study is a study made to decide whether or not the proposed system is worthwhile.

**TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**OPERATIONAL FEASIBILITY**

The operational feasibility is based on the human factors and political aspects. It can be performed by answering following questions,

* What change will be brought with the system?
* What are the factors that are disturbing organizational structure?
* Which are the new skills that are required for the improvements in operation?

The operational infeasibility does not lead to rejection of the project. The operational feasibility study is made by small group of people who are involved in the project analysis and design activity.

**ECONOMIC FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**1. 5 REQUIREMENT SPECIFICATION**

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirement analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements.

**SYSTEM REQUIREMENTS**

**HARDWARE USED**

|  |  |  |  |
| --- | --- | --- | --- |
| * Processor | | : Pentium IV | |
| * Hard Disk | | : 40GB | |
| * RAM | | : 512MB or more | |
| **SOFTWARE USED** | | |  | |
|  | Operating System | | : Windows XP/2007 or Linux | |
|  | User Interface | | : HTML, CSS | |
|  | Client-side Scripting | | : JavaScript | |
|  | Programming Language | | : Java | |
|  | Web Applications | | : JDBC, Servlets, JSP | |
|  | IDE/Workbench | | : Netbeans 7.3 | |
|  | Database | | : MySql 5.1 | |
|  | Server Deployment | | : Tomcat 7 | |

**CHAPTER 2**

**DESIGN**

**2.1 OVERALL SYSTEM DESIGN**

**2.1.1 ARCHITECTURE DIAGRAM**

Owner Data

Data Organized by Cloud

Providers

The Architecture explains the process involved in the system. The architecture of the project consists of three core end.

1. Front End.
2. Middle-ware.
3. Back End.

**FRONT END**

The Front end comprises of the client browser such as HTML where the client request for the content based retrieval from the SERVLET API.

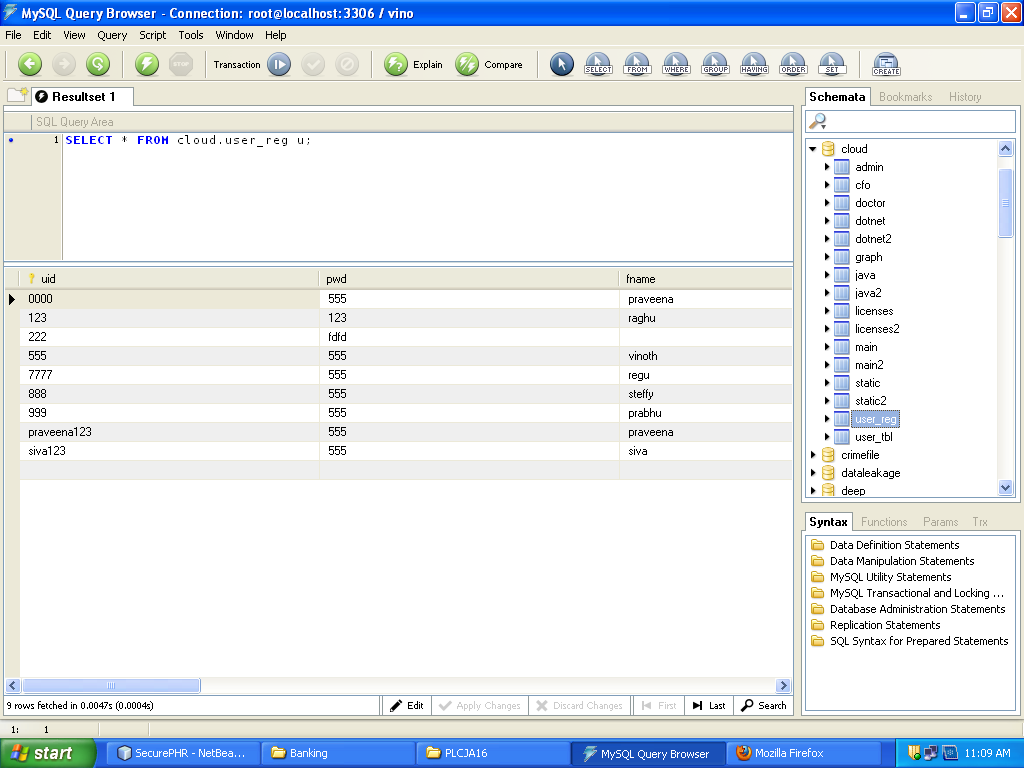
**MIDDLE-WARE**

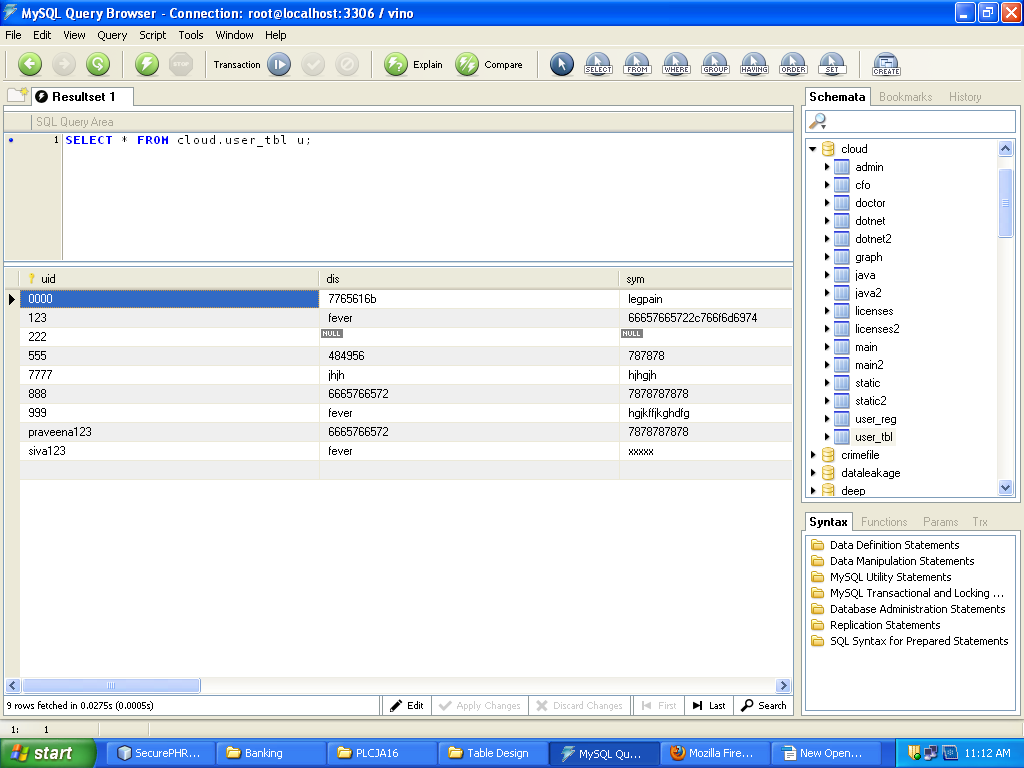
The middle-ware is the core-part of the architecture encompasses of SERVLET API and interaction among the webpage. The webpage request for the data from the back end and it is displayed in the specified webpage.

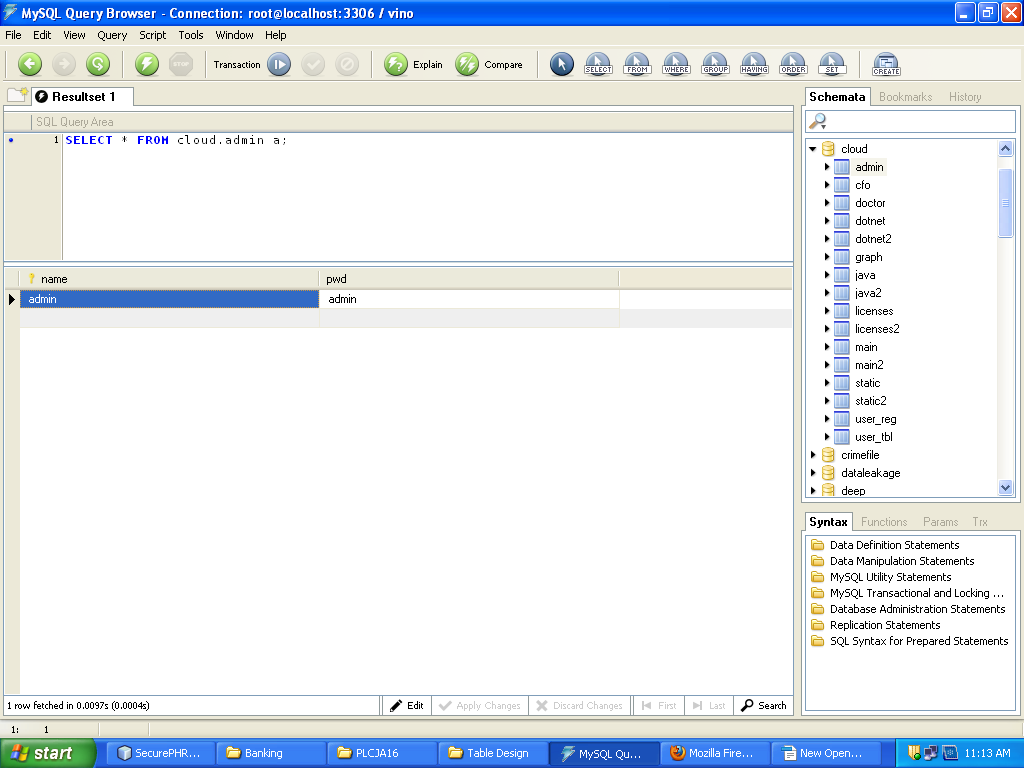
**BACK END**

The Back end consist of the MY SQL database in which the Data are send and retrieve from the database. The database transactions are carried out with the help of SQL Query.

**2.1.2 LOGICAL DATABASE DESIGN**



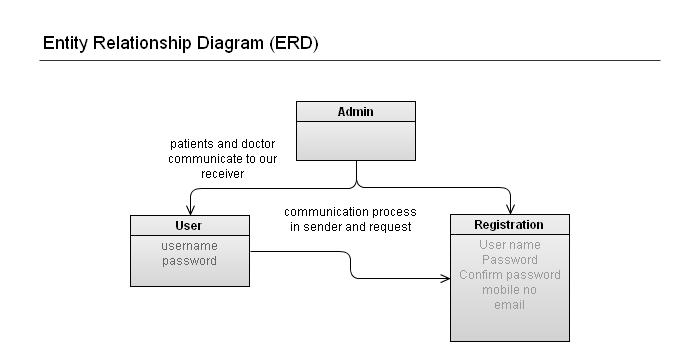




**2.1.3 E-R DIAGRAM**

In software engineering, an entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs.

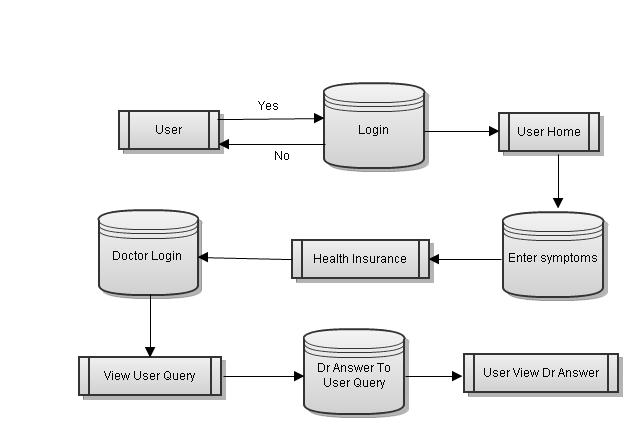
An entity-relationship (ER) diagram is a specialized graphic that illustrates the relationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes



**2.1.4 DATAFLOW DIAGRAM**

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

Data Flow Diagram (DFD) is an important technique for modeling a system’s high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs reveal relationships among and between the various components in a program or system. DFD consists of four major components: entities, processes, data stores and dataflow.



**2.2 DETAILED DESIGN**

**UML DIAGRAMS**

The Unified Modeling Language (UML) is the industry-standard modeling language used for specifying, visualizing, constructing and documenting the artifacts of a software system. The UML is also effective for modeling business and other Non-software systems.

One of the major goals of UML is to present a common modeling language that all developers can use. It is a language whose vocabulary and rules focus on the conceptual and physical representation of the system.

The UML uses Diagram to represent different views of the system being modeled. The purpose of diagram is to present a set of modeled elements, which are rendered as shapes and connectors.

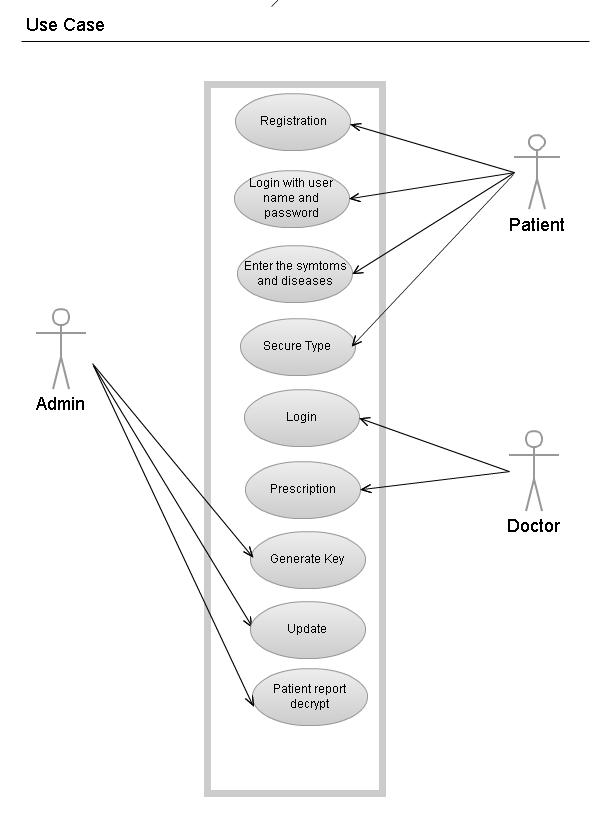
`The UML defines nine graphical diagrams:

1. Use-Case Diagram
2. Class Diagram
3. Activity Diagram
4. Sequence Diagram
5. Collaboration Diagram

**2.2.1 USE CASE DIAGRAMS**

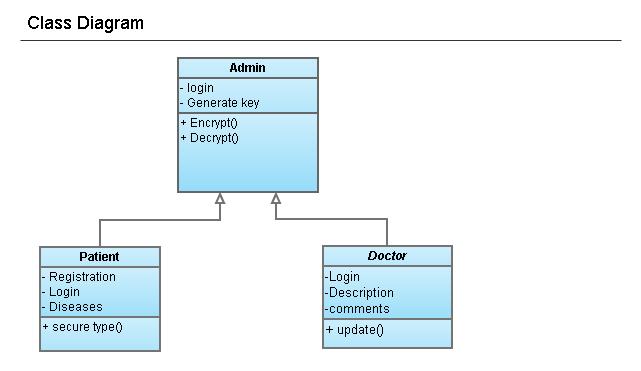
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.

Providers, admin and customer are actors. Providers add the items and also used to find the frequent items. admin is used to view the registered users and also

View the details.customer is connect with the provider and the customer is used to purchase.

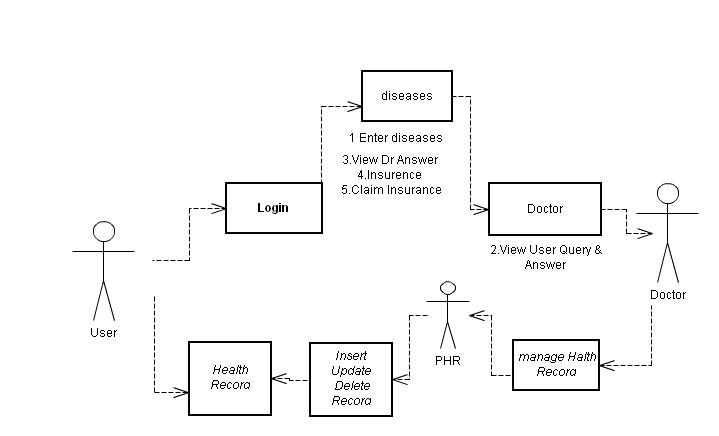
**2.2.2 CLASS DIAGRAM**

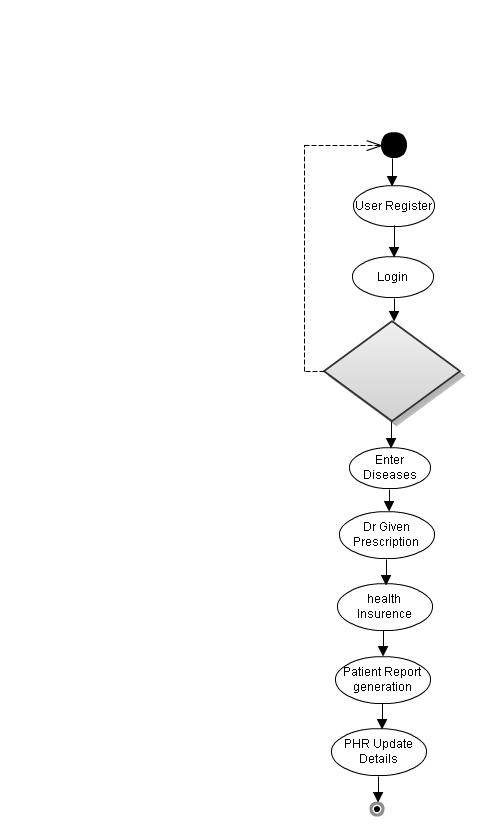
A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class. In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.



**2.2.3 ACTIVITY DIAGRAM**

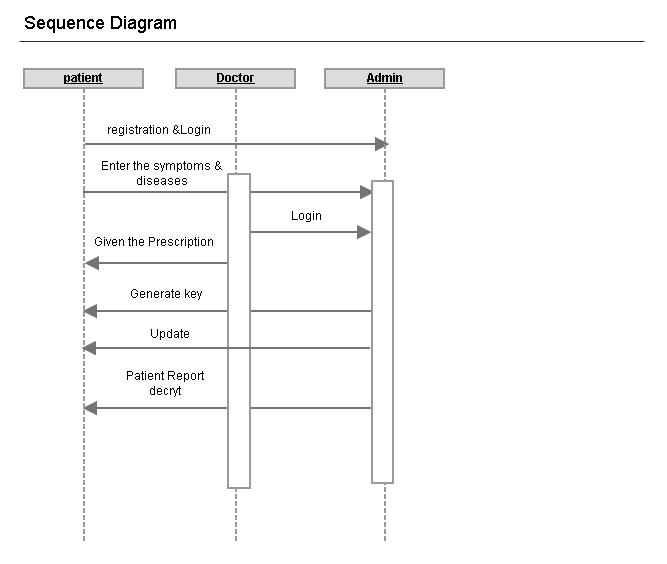
Activity diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.

The provider can login and login informations are validated. If the informations are validated correctly then he can select the option that he can view the item and add the item. If the informations are wrong it exits the loop. 



**2.2.4 SEQUENCE DIAGRAM**

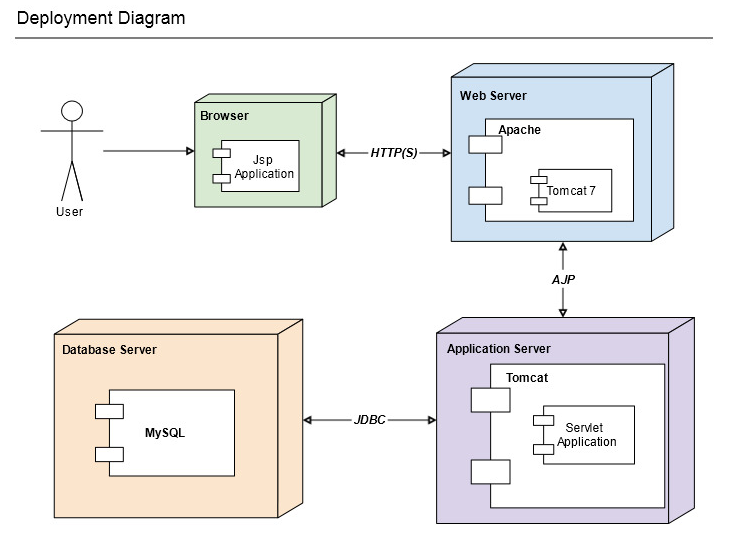
A sequence diagram in UML is a kind of interaction diagram that shows how processes operate with one another and in what order.

Provider login the page and he can add the item and view the item. Similarly the admin can also view the items. Customer after login the page he can view the products and purchase the products. Based on the customer purchase,provider can find the frequent item

**2.2.5 COLLABORATION DIAGRAM**

A collaboration diagram show the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction. The collaboration diagram can be a decomposition of a class, class diagram, or part of a class diagram.

Provider login the page and he can add the item and view the item. Similarly the admin can also view the items. Customer after login the page he can view the products and purchase the products. Based on the customer purchase,provider can find the frequent items.



**CHAPTER 3**

**IMPLEMENTATION**

**3.1 MODULES**

The system after careful analysis has been identified to be presented with the following modules:

* PHR Owner Module
* Cloud Server Module
* Attribute based Access Policy Module
* Data confidentiality Module

**3.2 MODULE DESCRIPTION:**

**3.2.1 PHR OWNER MODULE**

The main goal of our framework is to provide secure patient-centric PHR access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, *public domains* (PUDs) and *personal domains* (PSDs)) according to the different users’ data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors, nurses and medical researchers. In practice, a PUD can be mapped to an independent sector in the society, such as the health care, government or insurance sector. For each PSD, its users are personally associated with a data owner (such as family members or close friends), and they make accesses to PHRs based on access rights assigned by the owner.

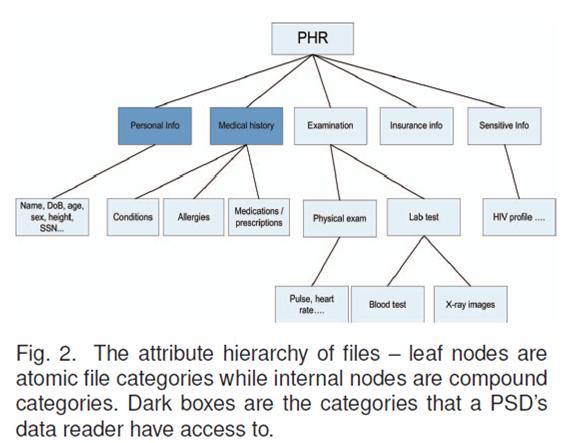
Each data owner (e.g., patient) is a trusted authority of her own PSD, who uses a KP-ABE system to manage the secret keys and access rights of users in her PSD. Since the users are personally known by the PHR owner, to realize patient-centric access, the owner is at the best position to grant user access privileges on a case-by-case basis. For PSD, *data attributes* are defined which refer to the intrinsic properties of the PHR data, such as the category of a PHR file. For the purpose of PSD access, each PHR file is labeled with its data attributes, while the key size is only linear with the number of file categories a user can access. Since the number of users in a PSD is often small, it reduces the burden for the owner. When encrypting the data for PSD, all that the owner needs to know is the intrinsic data properties.

**3.2.2 CLOUD SERVER MODULE**

In this paper, we consider the server to be semi-trusted, i.e., honest but curious as those in [28] and [15]. That means the server will try to find out as much secret information in the stored PHR files as possible, but they will honestly follow the protocol in general. On the other hand, some users will also try to access the files beyond their privileges. For example, a pharmacy may want to obtain the prescriptions of patients for marketing and boosting its profits. To do so, they may collude with other users, or even with the server. In addition, we assume each party in our system is preloaded with a public/private key pair, and entity authentication can be done by traditional challenge-response protocols.

**3.2.3 ATTRIBUTE BASED ACCESS POLICY MODULE**

In our framework, there are multiple SDs, multiple owners, multiple AAs, and multiple users. In addition, two ABE systems are involved. We term the users having read and write access as data readers and contributors, respectively.



**3.2.4 DATA CONFIDENTIALITY MODULE**

The owners upload ABE-encrypted PHR files to the server. Each owner’s PHR file is encrypted both under a certain fine grained and role-based access policy for users from the PUD to access, and under a selected set of data attributes that allows access from users in the PSD. Only authorized users can decrypt the PHR files, excluding the server.

**CHAPTER 4**

**TESTING**

The project is tested to verify its correction and efficiency. The test plan includes following test.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product.

**4.1 TYPES OF TESTS**

**4.1.1 UNIT TESTING**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs.

**4.1.2 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**4.1.3 FUNCTIONAL TESTING**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

**4.1.4 SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements.System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**4.1.5 WHITE BOX TESTING**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose.

**4.1.6 BLACK BOX TESTING**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

**APPENDIX A**

**SOFTWARE DESCRIPTION**

**A.1 JAVA TECHNOLOGY**

Java technology is both a programming language and a platform.

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

-Simple

-Architecture neutral

-Object oriented

-Portable

-Distributed

-High performance

-Interpreted

-Multithreaded

-Robust

-Dynamic

-Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called

**Java bytecodes**—the platform-independent codes interpreted by the interpreteron the Java platform. The interpreter parses and runs each Java byte code instruction on the computer.

Java byte codes as the machine code instructions for the Java Virtual Machine(Java VM). Every Java interpreter, whether it’s a developmenttool or a Web browser that can run applets, is an implementation of the Java VM.

**A.2 ODBC**

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database.

**A.3 JDBC**

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

**SQL Level API**

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently.

**A.4 NETWORKING**

**HTTP PROTOCOL**

The **Hypertext Transfer Protocol** (**HTTP**) is an [application protocol](http://en.wikipedia.org/wiki/Application_protocol) for distributed, collaborative, [hypermedia](http://en.wikipedia.org/wiki/Hypermedia) information system.HTTP is the foundation of data communication for the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web).

[Hypertext](http://en.wikipedia.org/wiki/Hypertext) is structured text that uses logical links ([hyperlinks](http://en.wikipedia.org/wiki/Hyperlinks)) between [nodes](http://en.wikipedia.org/wiki/Node_(computer_science)) containing text. HTTP is the protocol to exchange or transfer hypertext.The standards development of HTTP was coordinated by the [Internet Engineering Task Force](http://en.wikipedia.org/wiki/Internet_Engineering_Task_Force) (IETF) and the [World Wide Web Consortium](http://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C).

**A.5 DATABASE**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching, and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

**MySQL DATABASE:**

MySQL is a fast, easy-to-use RDBMS used being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB.

**APPENDIX B**

**SAMPLE CODING**

**HOME.JSP**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<% session.invalidate(); %>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>PHR</title>

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td><table border="0" align="center" cellpadding="0" cellspacing="0" id="outer">

<tr>

<td id="header" valign="top"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td height="10" colspan="2"></td>

</tr>

<tr>

<td valign="top"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td height="143"><table width="150%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td class="name"><pre><b> Secure Sharing of Personal Health Records

using Attribute-based Encryption</b> </pre></td>

</tr>

</table></td>

</tr>

<tr>

<td height="38"><table border="0" cellspacing="0" cellpadding="0">

</table></td>

</tr>

</table></td>

<td width="224" valign="top">&nbsp;</td>

</tr>

</table></td>

</tr>

<tr>

<td valign="top"><table width="942" border="0" cellspacing="0" cellpadding="0">

<div style="margin: 0px 0px 0px 200px">

<a href="index.jsp"> <h1> Patients</h1></a><br></br>

<a href="index\_1.jsp"> <h1> Doctors</h1></a><br></br>

<a href="index\_2.jsp"> <h1> Admin</h1></a><br></br>

<a href="cloud.jsp"> <h1> Health care</h1></a><br></br>

<% String msg=request.getParameter("msg");

if(msg != null){

%>

<%= msg %>

<%

}

%>

</div>

</table></td>

</tr>

</table></td>

</tr>

</table>

</body>

</html>

**CLOUD.JSP**

<html xmlns="http://www.w3.org/1999/xhtml">

<%@ page import="java.sql.\*" %>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title></title>

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td><table border="0" align="center" cellpadding="0" cellspacing="0" id="outer">

<tr>

<td id="header" valign="top"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td height="10" colspan="2"></td>

</tr>

<tr>

<td valign="top"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td height="143"><table width="150%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td class="name"><pre><b> Secure Sharing of Personal Health Records

using Attribute-based Encryption</b> </pre></td>

</tr>

</table></td>

</tr>

<tr>

<td height="38"><table border="0" cellspacing="0" cellpadding="0">

<td class="toplinks"><a href="Home.jsp">Homepage</a></td>

</table></td>

</tr>

</table></td>

<td width="224" valign="top">&nbsp;</td>

</tr>

</table></td>

</tr>

<tr>

<td valign="top"><table width="942" border="0" cellspacing="0" cellpadding="0">

<div style="">

<table style="overflow: auto;height: 100px;"border="1" > <tbody>

<label> users</label><br></br>

<tr>

<td style="height: 50px;width:100px;color: #A6EF00"><h2>UID</h2> </td>

<td style="height: 50px;width:100px;color: #A6EF00""><h2> Disease</h2> </td>

<td style="height: 50px;width:100px;color: #A6EF00"><h2> Symptoms</h2> </td>

<td style="height: 50px;width:100px;color: #A6EF00"><h2> Date </h2> </td>

<td style="height: 50px;width:100px;color: #A6EF00"><h2> Blood Group</h2> </td>

<td style="height: 50px;width:100px;color: #A6EF00"><h2> Doctor Name</h2></td>

<td style="height: 50px;width:100px;color: #A6EF00"><h2> Prescription</h2> </td>

<td style="height: 50px;width:100px;color: #A6EF00"> <h2> comments</h2> </td>

<td style="height: 50px;width:100px;color: #A6EF00"><h2> Date </h2> </td>

</tr>

<%

String uid="555";

Statement st;

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/cloud","root","admin");

String qry="select \* from user\_tbl";

st = con.createStatement();

ResultSet rs = st.executeQuery(qry);

while(rs.next())

{%>

<tr>

<td style="height: 50px;width:100px"><b><%=rs.getString(1)%></b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(2)%></b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(3)%></b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(4)%> </b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(5)%> </b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(6)%></b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(7)%></b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(8)%></b> </td>

<td style="height: 50px;width:100px"><b><%=rs.getString(9)%></b> </td>

</tr><tr></tr>

<% }

}

catch (Exception e) {

e.printStackTrace();

}

%>

</tbody>

</table>

</div>

</table></td>

</tr>

</table></td>

</tr>

</table>

</body>

</html>

**ENCRYPT.JSP**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<%@page contentType="text/html" import="java.util.\*" %>

<%@ page import="java.sql.\*" %>

<%@page import="java.io.InputStream"%>

<%@page import="java.io.File"%>

<script type="text/javascript" src="includes/JBAJSencrypt.js"></script>

<head>

<script language="JavaScript">

var decode='function dc(e){var ds="";for(i=0;i<e.length;i+=2){' +

'ds+=String.fromCharCode(parseInt(e.substring(i,i+2), 16));}return ds;}document.write(dc(ep));';

function ec(s){

//Function to encrypt any string with byte ranges from 0 To 255

var es="", ct="", nc;

//Build ASCII table

for(c=0;c<=255;c++) ct+=String.fromCharCode(c);

for(i=0;i<s.length;i++){

//Convert the ASCII key code of the

//current char to its hexadecimal equivalent

nc=ct.indexOf(s.charAt(i)).toString(16);

//If hex string is not 2 chars then add a 0 before it

if(nc.length!=2) nc="0"+nc;

es+=nc;

}

return es;

}

function dc(e){

//Function to decrypt string encrypted with HEX encryption

var ds="";

for(i=0;i<e.length;i+=2){

//Get two chars and then convert them

//from hexadecimal to an ASCII char

ds+=String.fromCharCode(parseInt(e.substring(i,i+2), 16));

}

return ds;

}

function encrypt(){

window.status="Encrypting page...";

if( document.getElementById("chk2").checked==true){

var pagecontent=document.pageprotector.dis.value;

document.pageprotector.dis.value=ec(pagecontent);}

if( document.getElementById("chk3").checked==true){

var pagecontent=document.pageprotector.sym.value;

document.pageprotector.sym.value=ec(pagecontent);}

if( document.getElementById("chk4").checked==true){

var pagecontent=document.pageprotector.bg.value;

document.pageprotector.bg.value=ec(pagecontent);}

if( document.getElementById("chk5").checked==true){

var pagecontent=document.pageprotector.dname.value;

document.pageprotector.dname.value=ec(pagecontent);}

if( document.getElementById("chk6").checked==true){

var pagecontent=document.pageprotector.pres.value;

document.pageprotector.pres.value=ec(pagecontent);}

if( document.getElementById("chk7").checked==true){

var pagecontent=document.pageprotector.cmd.value;

document.pageprotector.cmd.value=ec(pagecontent);}

if( document.getElementById("chk8").checked==true){

var pagecontent=document.pageprotector.ddate.value;

document.pageprotector.ddate.value=ec(pagecontent);}

window.status="";

}

function decrypt(){

window.status="Decrypting page...";

if( document.getElementById("chk2").checked==true){

var pagecontent=document.pageprotector.dis.value;

document.pageprotector.dis.value=dc(pagecontent);}

if( document.getElementById("chk3").checked==true){

var pagecontent=document.pageprotector.sym.value;

document.pageprotector.sym.value=dc(pagecontent);}

if( document.getElementById("chk4").checked==true){

var pagecontent=document.pageprotector.bg.value;

document.pageprotector.bg.value=dc(pagecontent);}

if( document.getElementById("chk5").checked==true){

var pagecontent=document.pageprotector.dname.value;

document.pageprotector.dname.value=dc(pagecontent);}

if( document.getElementById("chk6").checked==true){

var pagecontent=document.pageprotector.pres.value;

document.pageprotector.pres.value=dc(pagecontent);}

if( document.getElementById("chk7").checked==true){

var pagecontent=document.pageprotector.cmd.value;

document.pageprotector.cmd.value=dc(pagecontent);}

if( document.getElementById("chk8").checked==true){

var pagecontent=document.pageprotector.ddate.value;

document.pageprotector.ddate.value=dc(pagecontent);}

window.status="";

}

function protectpage(){

var pagecontent=document.pageprotector.pagehtml.value;

//Build the encrypted page, also note that the decryption

//code is also encrypted using the escape method

var protected="<html><body><script>var ep=\"" + pagecontent + "\";eval(unescape(\"" + escape(decode) + "\"));<\/script><\/body><\/html>";

document.pageprotector.pagehtml.value=protected;

document.pageprotector.pagehtml.focus();

window.status="";

alert("Page was protected succesfully");

}

function selectall(){

document.pageprotector.pagehtml.focus();

document.pageprotector.pagehtml.select();

}

function update(){

var uid1= document.getElementById('uid1').value;

var dis= document.getElementById('dis').value;

var sym= document.getElementById('sym').value;

var bg= document.getElementById('bg').value;

var dname= document.getElementById('dname').value;

var pres= document.getElementById("pres").value;

var cmd= document.getElementById("cmd").value;

var ddate= document.getElementById("ddate").value;

alert(combo);

window.location ="Admin\_update.jsp?uid1="+uid1+"&dis="+dis+"&sym="+sym+"&bg="+bg+"&dname="+dname+"&pres="+pres+"&cmd="+cmd+"&ddate="+ddate;

}

function getRandomNum(lbound, ubound) {

return (Math.floor(Math.random() \* (ubound - lbound)) + lbound);

}

function getRandomChar(number, lower, upper, other, extra) {

var numberChars = "0123456789";

var lowerChars = "abcdefghijklmnopqrstuvwxyz";

var upperChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

var otherChars = "`~!@#$%^&\*()-\_=+[{]}\\|;:'\",<.>/? ";

var charSet = extra;

if (number == true)

charSet += numberChars;

if (lower == true)

charSet += lowerChars;

if (upper == true)

charSet += upperChars;

if (other == true)

charSet += otherChars;

return charSet.charAt(getRandomNum(0, charSet.length));

}

function getPassword(length, extraChars, firstNumber, firstLower, firstUpper, firstOther,

latterNumber, latterLower, latterUpper, latterOther) {

var rc = "";

if (length > 0)

rc = rc + getRandomChar(firstNumber, firstLower, firstUpper, firstOther, extraChars);

for (var idx = 1; idx < length; ++idx) {

rc = rc + getRandomChar(latterNumber, latterLower, latterUpper, latterOther, extraChars);

}

return rc;

}

</script>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<title>Secue</title>

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<%

String uid=request.getParameter("uid");

Statement st;

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/cloud","root","admin");

String qry="select \* from user\_tbl where uid='"+uid+"'";

st = con.createStatement();

ResultSet rs = st.executeQuery(qry);

if(rs.next())

{%>

<table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td><table border="0" align="center" cellpadding="0" cellspacing="0" id="outer">

<tr>

<td id="header" valign="top"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td height="10" colspan="2"></td>

</tr>

<tr>

<td valign="top"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td height="143"><table width="100%" border="0" cellspacing="0" cellpadding="0">

<tr>

<td class="name"><pre><b> Secure Sharing of Personal Health Records

using Attribute-based Encryption</b> </pre></td>

</tr>

</table></td>

</tr>

<tr>

<td height="38"><table border="0" cellspacing="0" cellpadding="0">

<td class="toplinks"><a href="Home.jsp">Homepage</a></td>

<td class="sap">|</td>

<td class="toplinks"><a href="">Inbox</a></td>

<td class="sap">|</td>

<td class="toplinks"><a href="Home.jsp">Signout</a></td>

<td class="sap">|</td>

<td class="toplinks"><a href="Admin\_search.jsp">Admin</a></td>

<td class="sap">|</td>

<td class="toplinks"><a href="admin\_view.jsp?uid=<%= uid %>">Profile</a></td>

<td class="sap">| </td>

<td class="toplinks"> <a href="compose.jsp">Compose</a></td>

<td class="sap">| </td>

</table></td>

</tr>

</table></td>

<td width="224" valign="top">&nbsp;</td>

</tr>

</table> </td>

</tr>

<tr>

<td valign="top"><table width="942" border="0" cellspacing="0" cellpadding="0">

<form name="pageprotector"action="Admin\_update.jsp">

<div style="margin: 0px 0px 0px 300px">

<label>Enter your Problems </label><br>

<pre><input type="checkbox" name="chk1"id="chk1" value=""></input><label>UID </label> <input type=text value="<%= uid %>" size="30"id="uid1"name=uid1><br></pre>

<pre><input type="checkbox" name="chk2"id="chk2" value=""></input><label>Disease </label><input type=text value="<%= rs.getString(2) %>"style="height: 50px;width: 200px"id="dis"name=dis><br></pre>

<pre><input type="checkbox" name="chk3"id="chk3" value=""></input><label>Symptoms </label><input type=text style="height: 50px;width: 200px" value="<%= rs.getString(3) %>"id="sym"name=sym><br></pre>

<pre><input type="checkbox" name="chk4"id="chk4" value=""></input><label>Blood Group </label><input type=text value="<%= rs.getString(4) %>" size="30"id="bg"name=bg><br></pre>

<pre><input type="checkbox" name="chk5"id="chk5" value=""></input><label>Doctor Name </label> <input type=text id="dname"size="30"value="<%= rs.getString("dname") %>"name=dname><br></pre>

<pre><input type="checkbox" name="chk6"id="chk6" value=""></input><label>Prescription </label> <input style="height: 50px;width: 200px" value="<%= rs.getString(7) %>"id="pres"type=text name=pres><br></pre>

<pre><input type="checkbox" name="chk7"id="chk7" value=""></input><label>comments </label><input type=text value="<%= rs.getString("cmd") %>"style="height: 50px;width: 200px"id="cmd"name=cmd><br></pre>

<pre><input type="checkbox" name="chk8"id="chk8" value=""></input> <label>Date </label> <input type=text size="30"name=ddate id="ddate" value=" <%= new java.util.Date() %> "<br></pre>

<pre><input type="checkbox" name="chk9"id="chk9" value=""></input><label>Generate Key </label> <input type=text value="" size="30"id="key"name=key><br> </pre>

<table width=80% border=0>

<tr align=center>

<td>

<table border=0>

<tr>

<td>

</td>

<td>

<input type=hidden name=firstNumber checked>

<input type=hidden name=firstLower checked>

<input type=hidden name=firstUpper checked>

<input type=hidden name=firstOther>

</td>

</tr>

<tr>

<td>

</td>

<td>

<input type=checkbox name=latterNumber checked>

<input type=checkbox name=latterLower checked>

<input type=checkbox name=latterUpper checked>

<input type=checkbox name=latterOther>

</td>

</tr>

<tr>

<td>

</td>

<td>

<input type=hidden name=passwordLength value="5" size=3>

</td>

</tr>

<tr>

<td>

</td>

<td>

<input type=hidden name=extraChars size=20>

</td>

</tr>

</table>

</td>

</tr>

<tr align=center>

<td>

<p><input type="button" value=" Encrypt " name="B1" onClick="encrypt();" class="form\_s kin">

<input type="button" value="Decrypt" name="B2" onClick="decrypt();" class="form\_skin"></input>

<font size=2><input type=submit name=submit value="UPDATE"></input></font>

</div>

</form>

<input type=submit value="Generate Key" onClick="document.pageprotector.key.value =

getPassword(document.pageprotector.passwordLength.value, document.pageprotector.extraChars.value,

document.pageprotector.firstNumber.checked, document.pageprotector.firstLower.checked,

document.pageprotector.firstUpper.checked, document.pageprotector.firstOther.checked,

document.pageprotector.latterNumber.checked, document.pageprotector.latterLower.checked,

document.pageprotector.latterUpper.checked, document.pageprotector.latterOther.checked);">

</table></td>

</tr>

</table></td>

</tr>

</table>

<% }

else{

}}

catch (Exception e) {

e.printStackTrace();

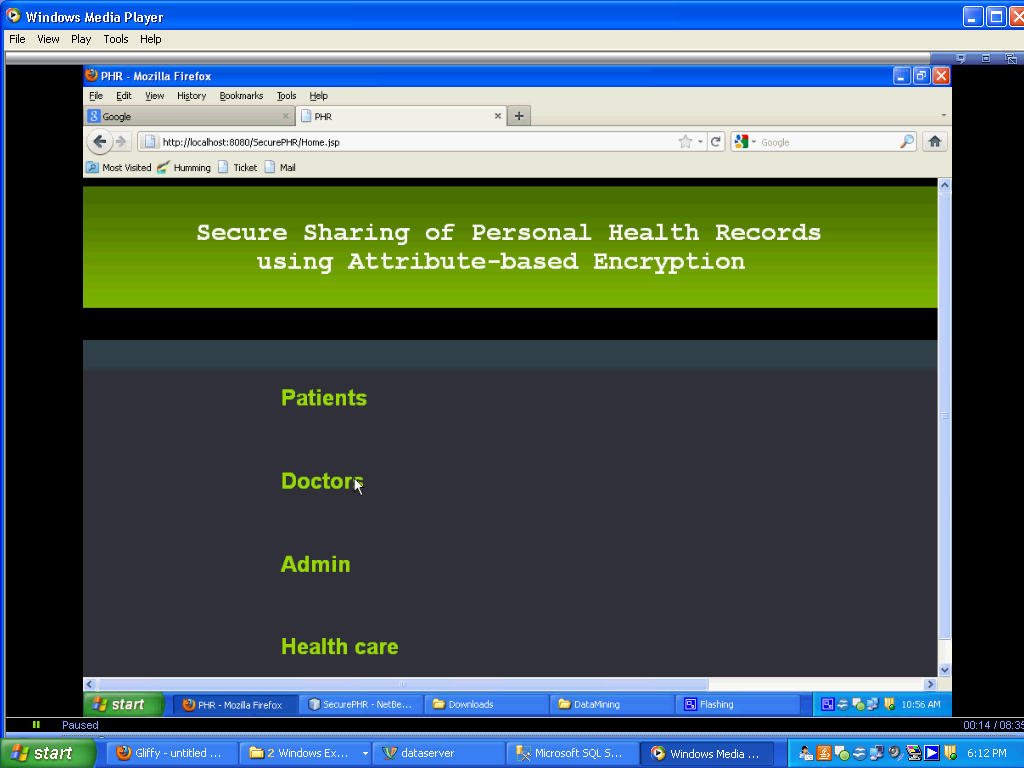
}

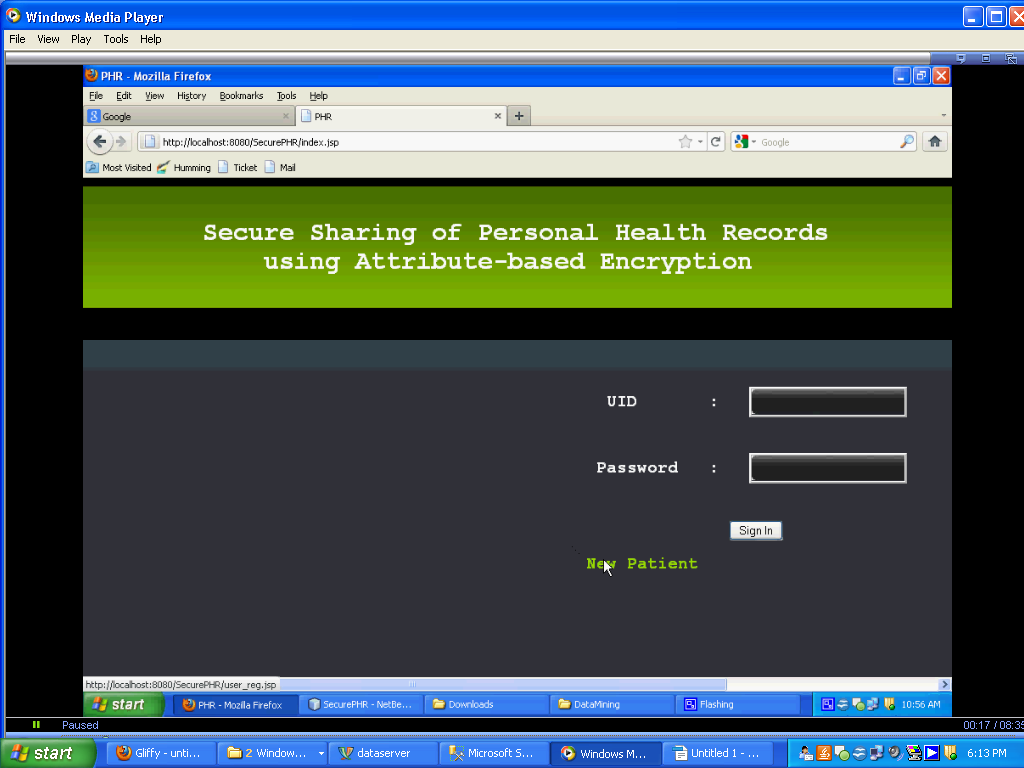
%>

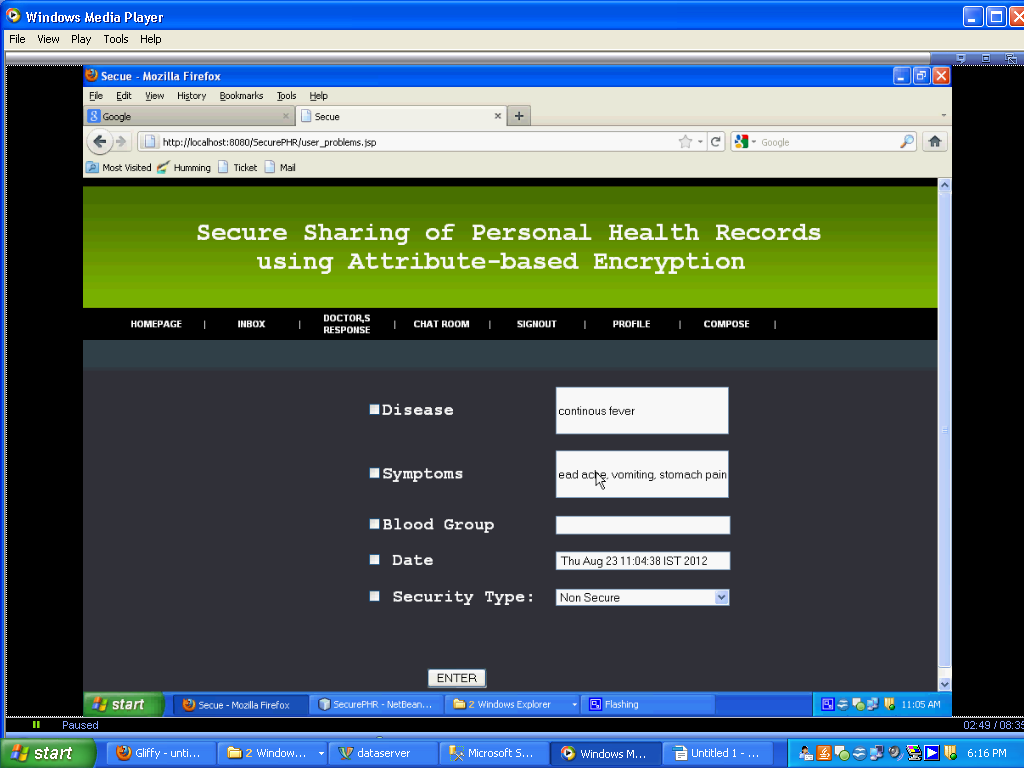
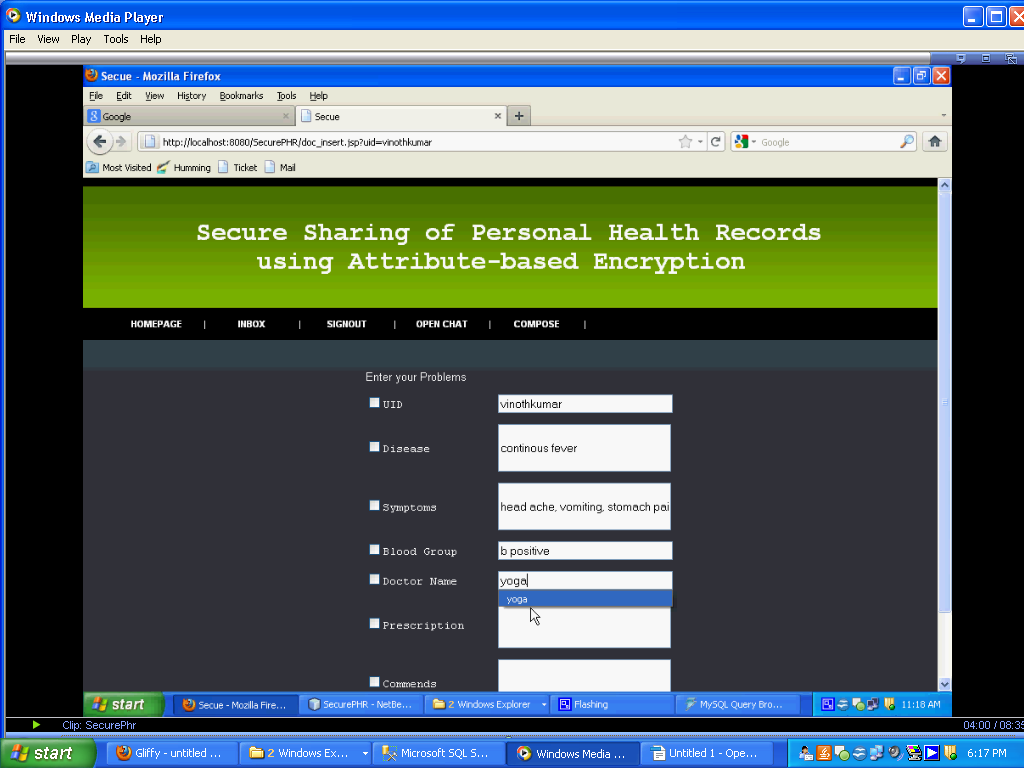
</body>

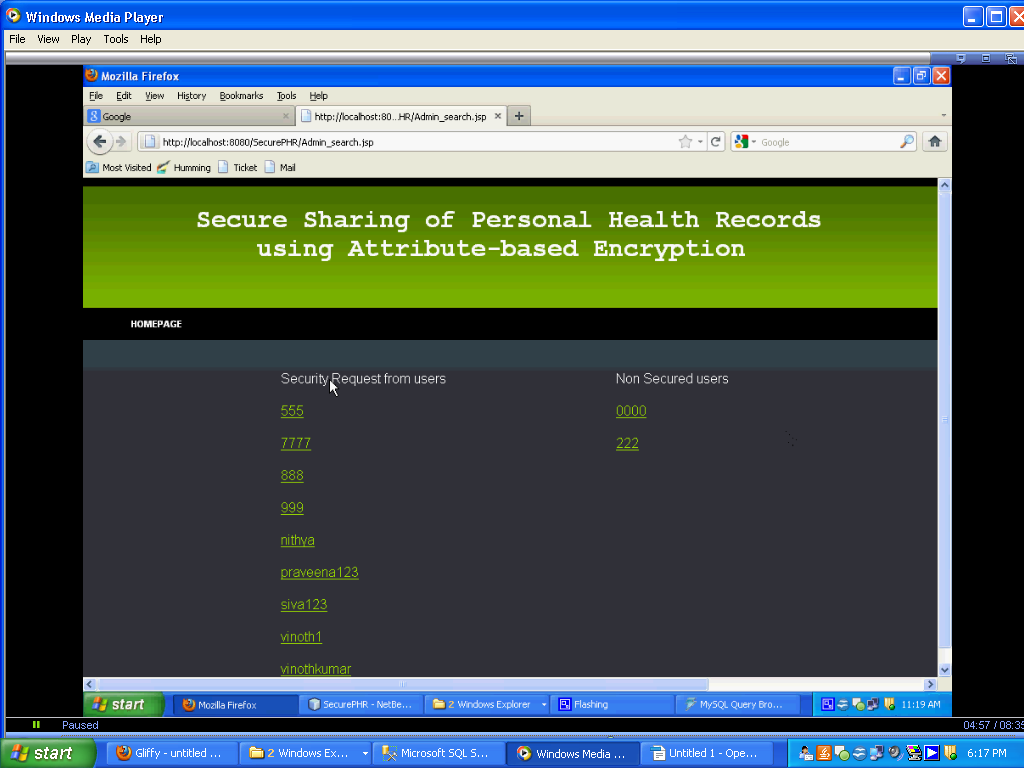
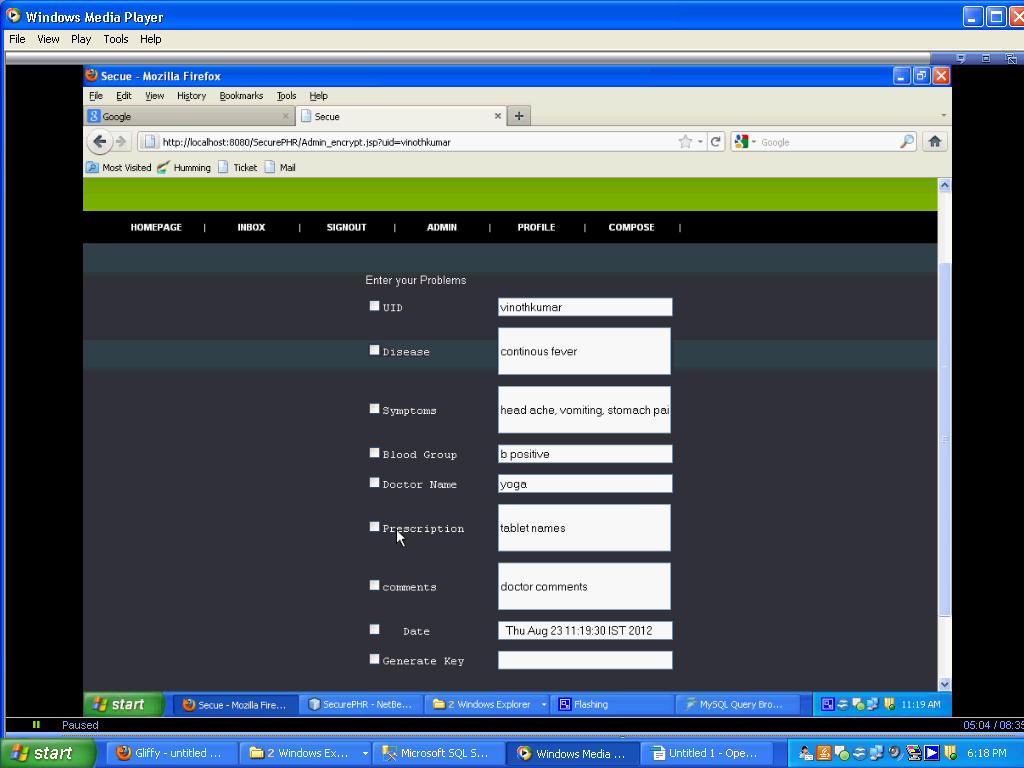
</html>

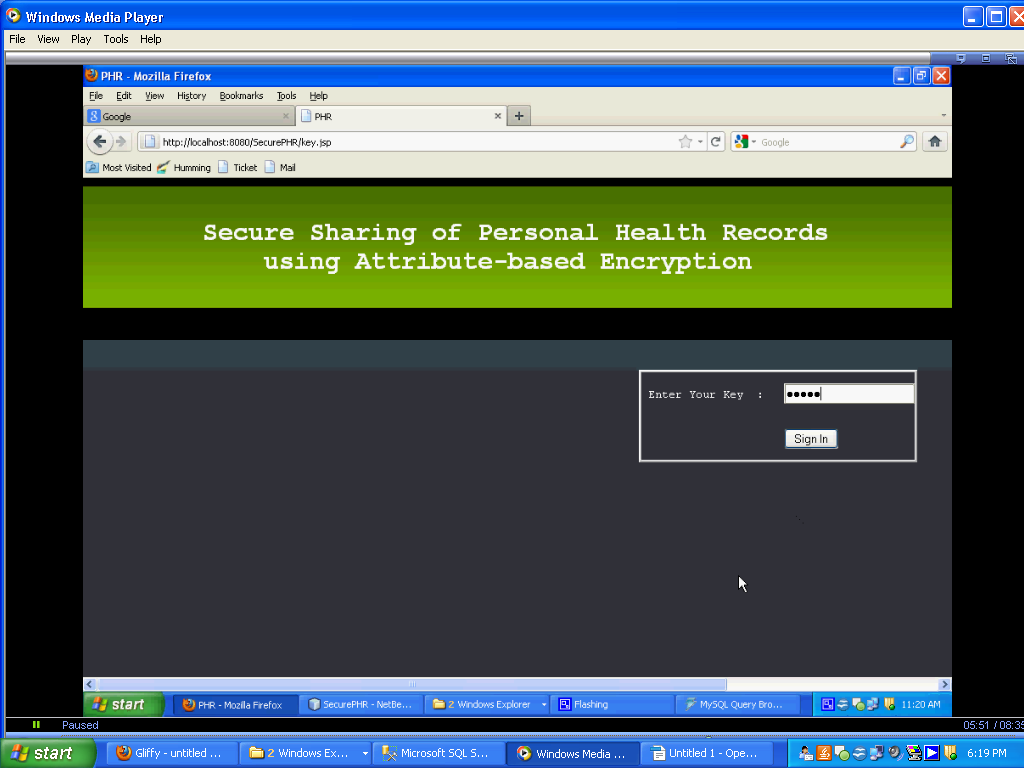
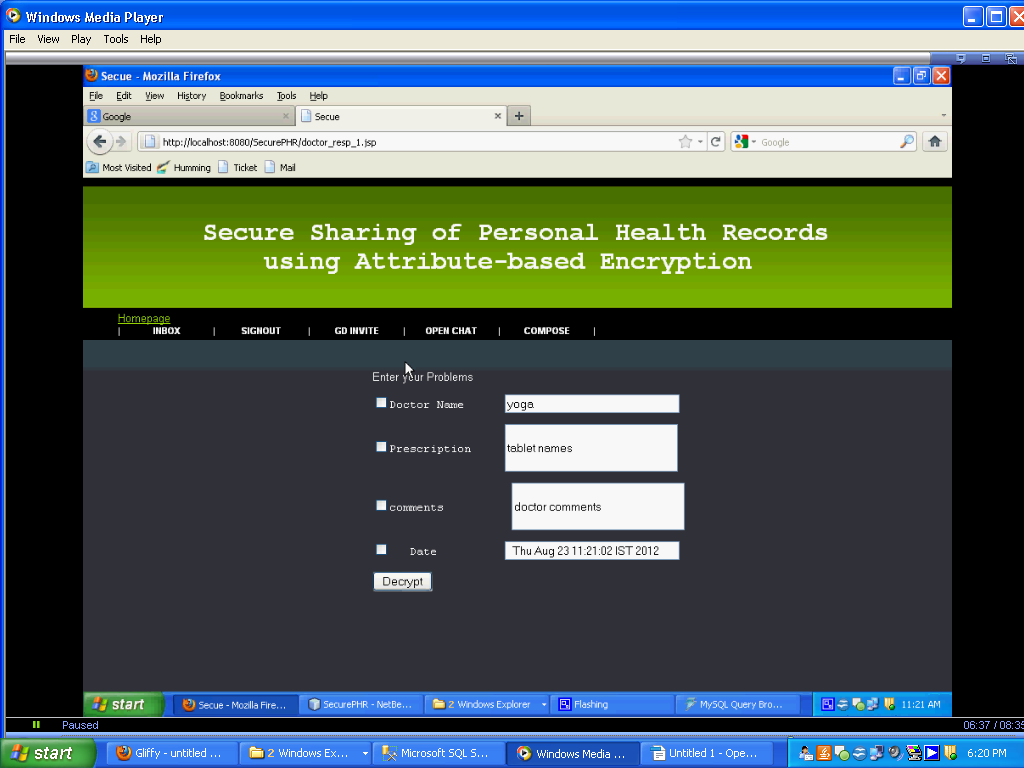
**APPENDIX C**

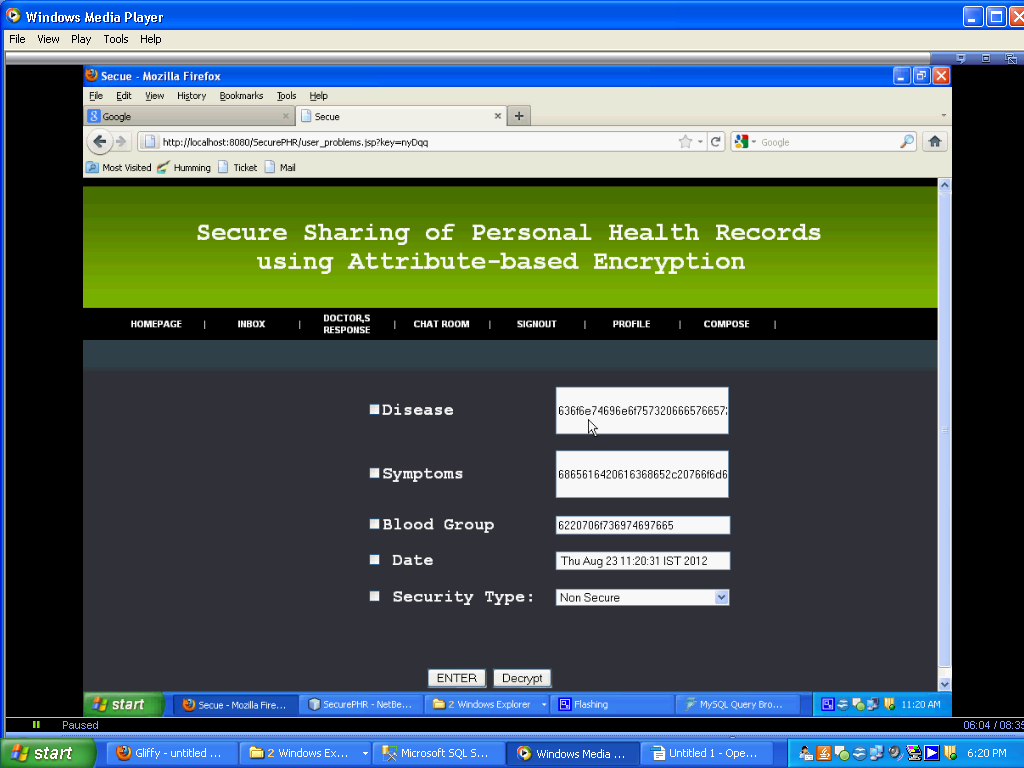
**SCREEN SHOTS**

****

****

****

****

****

**REFERENCES**

[1] M. Li, S. Yu, K. Ren, and W. Lou, “Securing personal health records in cloud computing: Patient-centric and ﬁne-grained data access control in multi-owner settings,” in SecureComm’10, Sept. 2010, pp. 89–106.

[2] H. Lohr, A.-R. Sadeghi, and M. Winandy, “Securing the e-health ¨ cloud,” in Proceedings of the 1st ACM International Health Informatics Symposium, ser. IHI ’10, 2010, pp. 220–229.

[3] M. Li, S. Yu, N. Cao, and W. Lou, “Authorized private keyword search over encrypted personal health records in cloud computing,” in ICDCS ’11, Jun. 2011.

[4] “The health insurance portability and accountability act.” [Online]. Available:

http://www.cms.hhs.gov/HIPAAGenInfo/01 Overview.asp

[5] “Google, microsoft say hipaa stimulus rule doesn’t apply to them,” http://www.ihealthbeat.org/Articles/2009/4/8/. [6] “At risk of exposure – in the push for electronic medical records, concern is growing about how well privacy can be safeguarded,” 2006. [Online]. Available: http://articles.latimes.com/2006/jun/26/health/he-privacy26

[6] K. D. Mandl, P. Szolovits, and I. S. Kohane, “Public standards and patients’ control: how to keep electronic medical records accessible but private,” BMJ, vol. 322, no. 7281, p. 283, Feb. 2001. [8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, “Patient controlled encryption: ensuring privacy of electronic medical records,” in CCSW ’09, 2009, pp. 103–114.