

A Mini Project Report on

PRIVACY-PRESERVING ATTRIBUTE-BASED ACCESS CONTROL MODEL FOR XML-BASED ELECTRONIC HEALTH RECORD SYSTEM

Submitted to

Jawaharlal Nehru Technological University, Hyderabad in partial
fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Anshul Nandiraju (17831A0509)

Anthati Yamini (17831A0510)

Anugu Vineeth Reddy (17831A0512)

Under the Esteemed Guidance of

Mrs. Nagabotu Vimala

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GURU NANAK INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUH-Hyderabad)

Ranga Reddy District -501506

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
GURU NANAK INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUH-Hyderabad)

Ranga Reddy District -501506



CERTIFICATE

This is to certify that the project entitled “**PRIVACY-PRESERVING ATTRIBUTE-BASED ACCESS CONTROL MODEL FOR XML-BASED ELECTRONIC HEALTH RECORD SYSTEM**” is being presented with report by **Anshul Nandiraju (17831A0509)**, **Anthathi Yamini (17831A0510)**, **Anugu Vineeth Reddy(17831A0512)** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, to **Jawaharlal Nehru Technological University, Hyderabad**.

INTERNAL GUIDE

PROJECT COORDINATOR

HEAD OF DEPARTMENT

EXTERNAL EXAMINER

PRINCIPAL



GURU NANAK INSTITUTE OF TECHNOLOGY

Ibrahimpattanam, R.R. Dist. – 501506.

VISION OF GNIT

To be a world – class educational and research institution in the service of humanity by promoting high quality Engineering and Management Education.

MISSION OF GNIT

M1: Imbibe soft skills and technical skills.

M2: Develop the faculty to reach the international standards.

M3: Maintain high academic standards and teaching quality that promotes the analytical thinking and independent judgment.

M4: Promote research, innovation and Product development by collaboration with reputed foreign universities.

M5: Offer collaborative industry programs in emerging areas and spirit of enterprise.



GURU NANAK INSTITUTE OF TECHNOLOGY

Ibrahimpattanam, R.R. Dist. – 501506.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION

To be a premier department of Computer Science and Engineering in the region.

MISSION

- Nurture young individuals into knowledgeable, skill-full and ethical professionals in their pursuit of Computer Science and Engineering.
- Nurture the faculty to expose them to world-class infrastructure.
- Sustain high performance by excellence in teaching, research and innovations.
- Extensive partnerships and collaborations with foreign universities for technology upgradation.
- Develop Industry-Interaction for innovation and product development.



GURU NANAK INSTITUTE OF TECHNOLOGY

Ibrahimpattam, R.R. Dist. – 501506.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program Educational Objectives (PEO's)

PEO-1: Graduates shall have the ability to apply knowledge and technical skills in emerging areas of Computer Science and Engineering for higher studies, research, employability, product development and handle realistic problems.

PEO-2: Graduates shall maintain ethical conduct, sense of responsibility to serve the society and protect the environment.

PEO-3: Graduates shall possess academic excellence with innovative insight, soft skills, managerial skills, leadership qualities, knowledge of contemporary issues for successful professional career.

Program Outcomes (PO's)

PO-1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.

PO-4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change



GURU NANAK INSTITUTE OF TECHNOLOGY

Ibrahimpattanam, R.R. Dist. – 501506.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAPPING WITH PO'S AND PEO'S

Sl.No.	Content	PO's	PEO's
1	Introduction	PO1, PO2	PEO1
2	Abstract	PO1, PO2	PEO1
3	System Specifications	PO3, PO5	PEO1
4	Architecture Design	PO3, PO4	PEO1, PEO2
5	HDFS CLI Commands	PO1	PEO3
6	Implementation	PO9, PO10, PO11	PEO2, PEO3
7	Results	PO4, PO11, PO12	PEO2, PEO3

DECLARATION

We hereby declare that mini project report entitled “**PRIVACY-PRESERVING ATTRIBUTE-BASED ACCESS CONTROL MODEL FOR XML-BASED ELECTRONIC HEALTH RECORDSYSTEM**” is the work done by ANSHUL NANDIRAJU, ANTHATI YAMINI, ANUGU VINEETH REDDY bearing the roll no. 17831A0509, 17831A0510, 17831A0512 towards the fulfillment of the requirement for the award of the Degree of **Bachelor of Technology in Computer Science and Engineering**, to **Jawaharlal Nehru Technological University, Hyderabad**, is the result of the work carried out under the guidance **Mrs. Nagabotu Vimala**, GuruNanak Institute of Technology, Hyderabad.

We further declare that this project report has not been previously submitted before either in part or full for the award of any degree or any diploma by any organization or any universities.

Anshul Nandiraju (17831A0509)

Anthati Yamini (17831A0510)

Anugu Vineeth Reddy (17831A0512)

ACKNOWLEDGEMENT

“**Task successful**” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the persons who have supported us to make it a success.

We would like to express our sincere thanks and gratitude to our Principal,

Dr. S. SREENATHA REDDY and Head of the Department **Dr. S. DEEPAJOTHI**, Department of Computer Science and Engineering, Guru Nanak Institute of Technology for having guided me in developing the requisite capabilities for taking up this project.

We thank Project Coordinator **Mr. CH. Balakrishna** CSE, GNIT for providing seamless support and right suggestions that are given in the development of the project.

We specially thank our internal guide **Mrs. Nagabotu Vimala** for her constant guidance in every stage of the project. We would also like to thank all our lecturers for helping me in every possible way whenever the need arose.

On a more personal note we thank our beloved parents and friends for their moral support during the course of our project.

**PRIVACY-PRESERVING ATTRIBUTE-
BASED ACCESS CONTROL MODEL FOR
XML-BASED ELECTRONIC HEALTH
RECORD SYSTEM**

ABSTRACT

Millions of private images are generated in various digital devices every day. The consequent massive computational workload makes people turn to cloud computing platforms for their economical computation resources. Meanwhile, the privacy concerns over the sensitive information contained in outsourced image data arise in public. In fact, once uploaded to cloud, the security and privacy of the image content can only presume upon the reliability of the cloud service providers. Lack of assuring security and privacy guarantees becomes the main barrier to further deployment of cloud-based image processing systems. This paper studies the design targets and technical challenges lie in constructing cloud-based privacy-preserving image processing system. We explore various image processing tasks, including image feature detection, digital watermarking, content-based image search. The state-of-the-art techniques, including secure multiparty computation, and homomorphic encryption are investigated. A detailed taxonomy of the problem statement and the corresponding solutions is provided.

.

TABLE OF CONTENTS

CONTENTS	PAGE NO.
CERTIFICATE	i
GNIT VISION AND MISSION	ii
DEPARTMENT VISION AND MISSION	iii
MAPPING WITH PO'S AND PEO'S	vi
DECLARATION	vii
ACKNOWLEDGEMENT	viii
ABSTRACT	x
LIST OF FIGURES	xiv
LIST OF SYMBOLS	xv
LIST OF ABBREVIATIONS	xviii
 CHAPTER 1: INTRODUCTION	
1.1 GENERAL	1
1.2 OBJECTIVE	1
1.3 EXISTING SYSTEM	1
1.3.1 EXISTINGSYSTEM DISADVANTAGES	2
1.3.2 LITERATURE SURVEY	2
1.4 PROPOSED SYSTEM	5
1.4.1 PROPOSED SYSTEM ADVANTAGES	5
 CHAPTER 2: PROJECT DESCRIPTION	
2.1 GENERAL	6
2.2 METHODOLOGIES	6
2.2.1 MODULES NAME	6
2.2.3 MODULE DIAGRAM	7
2.2.4GIVEN INPUTAND EXPECTED	11
OUTPUT	

CHAPTER 3: REQUIREMENTS

3.1 GENERAL	13
3.2 HARDWARE REQUIREMENTS	13
3.3 SOFTWARE REQUIREMENTS	14
3.4 FUNCTIONAL REQUIREMENTS	14
3.5 NON-FUNCTIONAL SPECIFICATIONS	15
3.5.1 EFFICIENCY	15
3.5.2 RELIABILITY	15

CHAPTER 4: SYSTEM DESIGN

4.1 GENERAL

4.1.1 ACTIVITY DIAGRAM	16
4.1.2 USE CASE DIAGRAM	17
4.1.3 DATA FLOW DIAGRAM	18
4.1.4 SEQUENCE DIAGRAM	19
4.1.5 COLLABORATION DIAGRAM	22
4.1.6 OBJECT DIAGRAM	23
4.1.7 CLASS DIAGRAM	24
4.1.8 ARCHITECTURE DIAGRAM	25
4.1.9 DEPLOYMENT DIAGRAM	26
4.1.10 E-R DIAGRAM	27
4.1.11 STATE DIAGRAM	28
4.1.12 COMPONENT DIAGRAM	29
4.2 CONCLUSION	30

CHAPTER 5: SOFTWARE SPECIFICATION

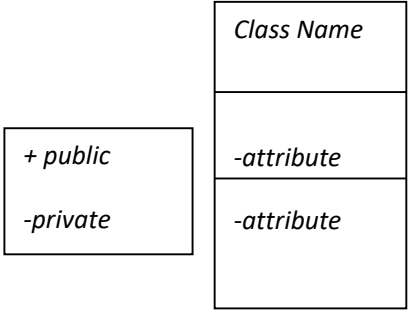
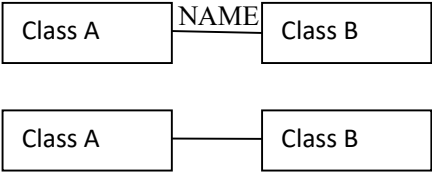
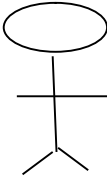
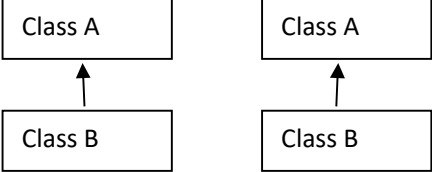
5.1 GENERAL	31
FRONT END	31
5.2 FEATURES OF JAVA	31
5.2.1 THE JAVA FRAMEWORK	31
5.2.2 OBJECTIVE OF JAVA	32
5.2.3 JAVA SERVER PAGES	33
5.2.4 EVOLUTION OF WEB APPLICATIONS	34

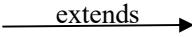

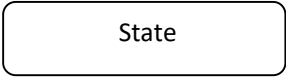
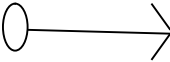
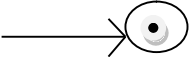
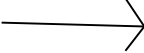
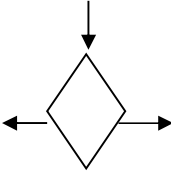
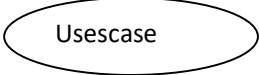
5.2.5 BENEFITS OF JSP	35
5.3 SERVLETS	36
5.4 JAVA SERVLETS	36
5.5 CONCLUSION	37
 CHAPTER 6: IMPLEMENTATION	
6.1 GENERAL	38
6.2 IMPLEMENTATION	38
 CHAPTER 7: SNAPSHOTS	
7.1 GENERAL	47
7.2 VARIOUS SNAPSHOTS	47
7.3 DATABASE STRUCTURE	51
 CHAPTER 8: SOFTWARE TESTING	
8.1 GENERAL	52
8.2 DEVELOPING METHODOLOGIES	52
8.3 TYPES OF TESTING	52
 CHAPTER 9: APPLICATIONS AND FUTURE ENHANCEMENTS	
9.1 GENERAL	55
9.2 APPLICATIONS	55
9.3 FUTURE ENHANCEMENTS	55
 CHAPTER 10: CONCLUSION AND REFERENCES	
10.1 CONCLUSION	56
REFERENCES	57

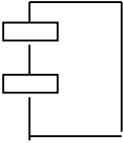
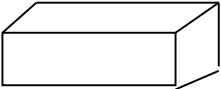
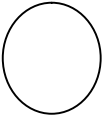
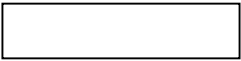
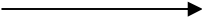

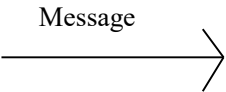
LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
2.2.3	Module Diagram	7
4.1.2	Activity Diagram	16
4.3	Use case Diagram	17
4.1.4	Data flow diagram	18
4.1.5	Sequence diagram	19
4.1.6	Collaboration diagram	22
4.1.7	Class diagram	24
4.1.8	Architecture Diagram	25
4.1.10	E-R Diagram	27
4.1.11	State Diagram	28
4.1.12	Component Diagram	29

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class		Represents a collection of similar entities grouped together.
2.	Association		Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
4.	Aggregation		Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Usecase		Interaction between the system and external environment.

14.	Component		Represents physical modules which is a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message		Represents the message exchanged.

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1.	DB	Database
2.	JVM	Java Virtual Machine
3.	JSP	Java Server Page
4.	LCA	Lowest Common Ancestor
5.	ELCA	Exclusive Lowest Common Ancestor
6.	JRE	Java Runtime Environment
7.	MCTs	Minimal Cost Trees
8.	SLCA	Smallest Lowest Common Ancestor

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Cloud-based electronic health record (EHR) systems enable medical documents to be exchanged between medical institutions; this is expected to contribute to improvements in various medical services in the future. However, as the system architecture becomes more complicated, cloud-based EHR systems may introduce additional security threats when compared to existing singular systems. Thus, patients may experience exposure of private data that they do not wish to disclose. In order to protect the privacy of patients, many approaches have been proposed to provide access control to patient documents when providing health services. However, most current systems do not support fine-grained access control or take into account additional security factors such as encryption and digital signatures. In this paper, we propose a cloud-based EHR model that performs attribute-based access control using extensible access control markup language. Our EHR model, focused on security, performs partial encryption and uses electronic signatures when a patient document is sent to a document requester. We use XML encryption and XML digital signature technology. Our proposed model works efficiently by sending only the necessary information to requesters who are authorized to treat the patient in question.

1.2 OBJECTIVE

We proposed a cloud-based EHR model that guarantees patient privacy. The proposed model is divided into two stages: access control, and the application of encryption and digital signatures. The proposed model uses an ABAC method built upon XACML. After performing access control on patient documents, encryption is performed and digital signatures are added using XML encryption and XML digital signatures as an added security measure. The proposed model provides more flexible and fine-grained control than existing RBAC systems and alleviates the risk of exposing patient privacy information by using partial encryption and electronic signatures. The implementation of a prototype demonstrated the feasibility of the proposed model. We compared the implemented security factors with

those used in other related studies and determined that the proposed method is superior to previous methods in terms of security.

1.3 EXISTING SYSTEM

Cloud-based electronic health record (EHR) systems enable medical documents to be exchanged between medical institutions;

Cloud-based EHR systems may introduce additional security threats when compared to existing singular systems.

1.3.1 EXISTING SYSTEM DISADVANTAGES

Hash collisions are practically unavoidable.

When hashing a random subset of a large set of possible keys.

Hash tables become quite inefficient when there are many collisions.

1.3.2 Literature Survey:

Title: Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption.

Author: Tang PC, Ash JS, Bates DW, Overhage JM, Sands DZ.

Year:2006

Description:

Recently there has been a remarkable upsurge in activity surrounding the adoption of personal health record (PHR) systems for patients and consumers. The biomedical literature does not yet adequately describe the potential capabilities and utility of PHR systems. In addition, the lack of a proven business case for widespread deployment hinders PHR adoption. In a 2005 working symposium, the American Medical Informatics Association's College of Medical Informatics discussed the issues surrounding personal health record systems and developed recommendations for PHR-promoting activities. Personal health

record systems are more than just static repositories for patient data; they combine data, knowledge, and software tools, which help patients to become active participants in their own care. When PHRs are integrated with electronic health record systems, they provide greater benefits than would stand-alone systems for consumers. This paper summarizes the College Symposium discussions on PHR systems and provides definitions, system characteristics, technical architectures, benefits, barriers to adoption, and strategies for increasing adoption.

Title: Inter-organizational future proof EHR systems: a review of the security and privacy related issues.

Author: van der Linden, H., Kalra, D., Hasman, A., & Talmon, J.

Year: 2009

Description:

Recently there has been a remarkable upsurge in activity surrounding the adoption of personal health record (PHR) systems for patients and consumers. The biomedical literature does not yet adequately describe the potential capabilities and utility of PHR systems. In addition, the lack of a proven business case for widespread deployment hinders PHR adoption. In a 2005 working symposium, the American Medical Informatics Association's College of Medical Informatics discussed the issues surrounding personal health record systems and developed recommendations for PHR-promoting activities. Personal health record systems are more than just static repositories for patient data; they combine data, knowledge, and software tools, which help patients to become active participants in their own care. When PHRs are integrated with electronic health record systems, they provide greater benefits than would stand-alone systems for consumers. This paper summarizes the College Symposium discussions on PHR systems and provides definitions, system characteristics, technical architectures, benefits, barriers to adoption, and strategies for increasing adoption.

Title: Key capabilities of an electronic health record system.

Author: Institute of Medicine (US) Committee on Data Standards for Patient Safety.

Washington (DC):

Year: 2003

Description:

Commissioned by the Department of Health and Human Services, *Key Capabilities of an Electronic Health Record System* provides guidance on the most significant

care delivery-related capabilities of electronic health record (EHR) systems. There is a great deal of interest in both the public and private sectors in encouraging all health care providers to migrate from paper-based health records to a system that stores health information electronically and employs computer-aided decision support systems. In part, this interest is due to a growing recognition that a stronger information technology infrastructure is integral to addressing national concerns such as the need to improve the safety and the quality of health care, rising health care costs, and matters of homeland security related to the health sector. *Key Capabilities of an Electronic Health Record System* provides a set of basic functionalities that an EHR system must employ to promote patient safety, including detailed patient data (e.g., diagnoses, allergies, laboratory results), as well as decision-support capabilities (e.g., the ability to alert providers to potential drug-drug interactions). The book examines care delivery functions, such as database management and the use of health care data standards to better advance the safety, quality, and efficiency of health care in the United States.

Title: The value of electronic health records in solo or small group practices.

Author: Miller RH, West C, Brown TM, Sim I, Ganchoff C.

Year:2005

Description:

We conducted case studies of fourteen solo or small-group primary care practices using electronic health record (EHR) software from two vendors. Initial EHR costs averaged \$44,000 per full-time-equivalent (FTE) provider, and ongoing costs averaged \$8,500 per provider per year. The average practice paid for its EHR costs in 2.5 years and profited handsomely after that; however, some practices could not cover costs quickly, most providers spent more time at work initially, and some practices experienced substantial financial risks. Policies should be designed to provide incentives and support services to help practices improve the quality of their care by using EHRs.

Title: Enhancing patient safety and quality of care by improving the usability of electronic health record systems: recommendations from AMIA.

Author: Middleton B, Bloomrosen M, Dente MA, Hashmat B, Koppel R, Overhage JM, Payne TH, Rosenbloom ST, Weaver C, Zhang J; American Medical Informatics Association.

Year:2013

Description:

In response to mounting evidence that use of electronic medical record systems may cause unintended consequences, and even patient harm, the AMIA Board of Directors convened a Task Force on Usability to examine evidence from the literature and make recommendations. This task force was composed of representatives from both academic settings and vendors of electronic health record (EHR) systems. After a careful review of the literature and of vendor experiences with EHR design and implementation, the task force developed 10 recommendations in four areas: (1) human factors health information technology (IT) research, (2) health IT policy, (3) industry recommendations, and (4) recommendations for the clinician end-user of EHR software. These AMIA recommendations are intended to stimulate informed debate, provide a plan to increase understanding of the impact of usability on the effective use of health IT, and lead to safer and higher quality care with the adoption of useful and usable EHR systems

1.4 PROPOSED SYSTEM

We propose a cloud-based EHR model that performs attribute-based access control using extensible access control markup language.

Our EHR model, focused on security, performs partial encryption and uses electronic signatures when a patient document is sent to a document requester.

1.4.1 PROPOSED SYSTEM ADVANTAGES

An encrypting algorithm scrambles the message and it can only be unscrambled with a key created at the same time.

Cipher algorithms are either symmetric or asymmetric for encryption security. For example: Symmetric - the exact same key is used to encrypt and decrypt data.

CHAPTER 2

PROJECT DESCRIPTION

2.1 GENERAL

Cloud-based electronic health record (EHR) systems enable medical documents to be exchanged between medical institutions; this is expected to contribute to improvements in various medical services in the future. However, as the system architecture becomes more complicated, cloud-based EHR systems may introduce additional security threats when compared to existing singular systems. Thus, patients may experience exposure of private data that they do not wish to disclose. In order to protect the privacy of patients, many approaches have been proposed to provide access control to patient documents when providing health services. However, most current systems do not support fine-grained access control or take into account additional security factors such as encryption and digital signatures. In this paper, we propose a cloud-based EHR model that performs attribute-based access control using extensible access control markup language. Our EHR model, focused on security, performs partial encryption and uses electronic signatures when a patient document is sent to a document requester. We use XML encryption and XML digital signature technology. Our proposed model works efficiently by sending only the necessary information to requesters who are authorized to treat the patient in question.

2.2 METHODOLOGIES

2.2.1 MODULES NAME

In this project we have the following modules.

1. User Interface Design
2. Electronic Health Record System
3. User
4. Admin

2.2.3 MODULE DIAGRAM

1. USER INTERFACE DESIGN

This is the first module of our project. In this the application user's first create their account properly which are stored at the back end for verification or for providing security to the accounts. If user wants to get into his account first they have to submit their constraints such as username, password and so on...otherwise can't able to access the account. In our project according to actions they are performing we disperse the users as admin or normal application user.

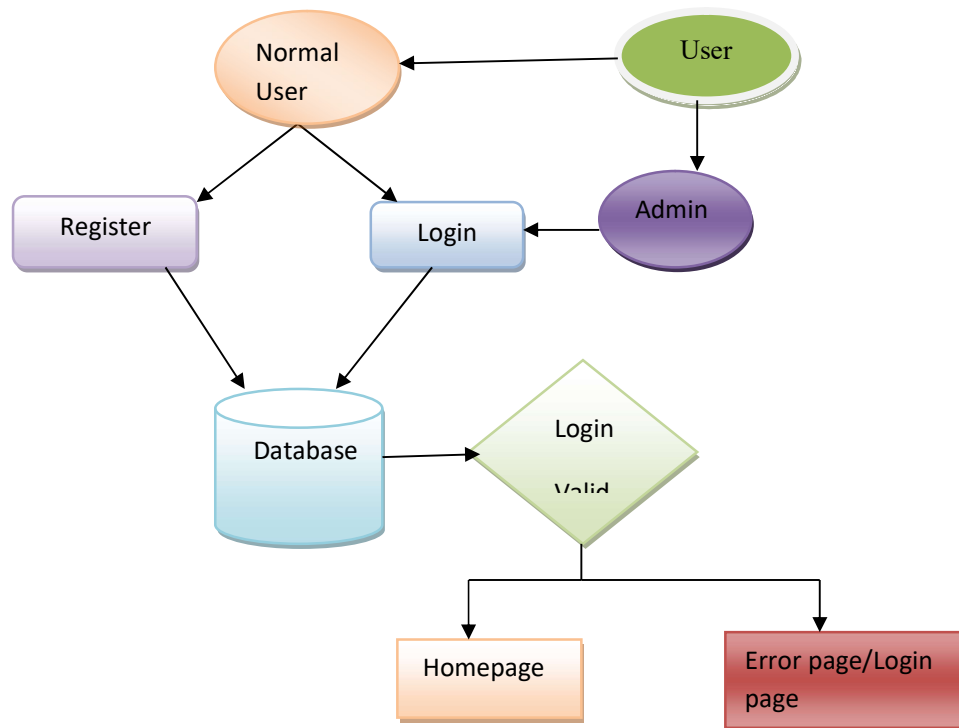


Fig 1. User Interface Design

2. Electronic Health Record System:

Cloud-based electronic health record (EHR) systems enable medical documents to be exchanged between medical institutions; this is expected to contribute to improvements in various medical services in the future. In this project the electronic records are stored in xml files and we are store encrypted xml files to server.

A electronic health record have title, reference name, keywords, data.

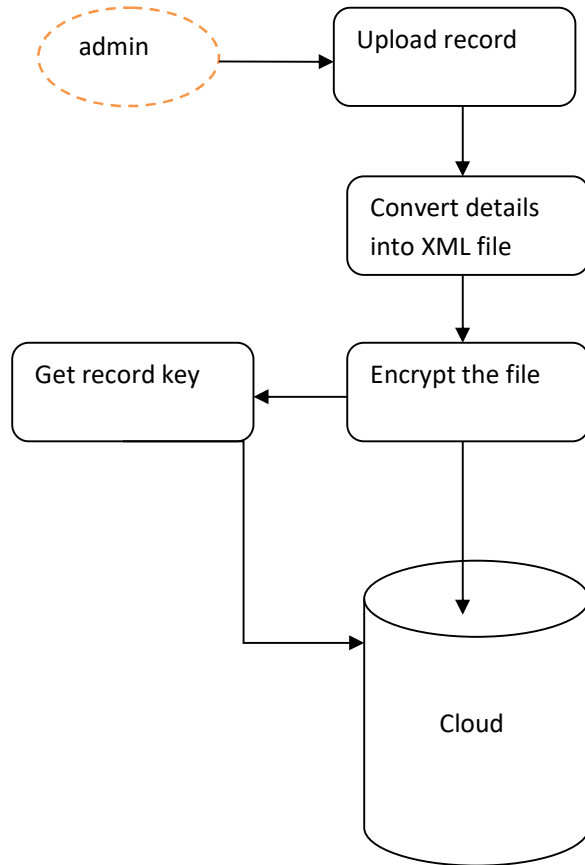


Fig 2. Electronic Health Record System

3. User:

The third party users have the permissions for searching the data related to electronic file to view the data the user must need a key to decrypt a file due to we are storing the files in the form of encryption in cloud. And the key are stored at admin, So when we need a file data we must need get key from the admin, for that we need to send key request to admin for particular file.

- Register.
- Login.
- Searching records by keyword based.
- Select record.
- Send record key request for admin.
- Get record key from admin.
- View the record data by decrypting that by using key.
- Logout.

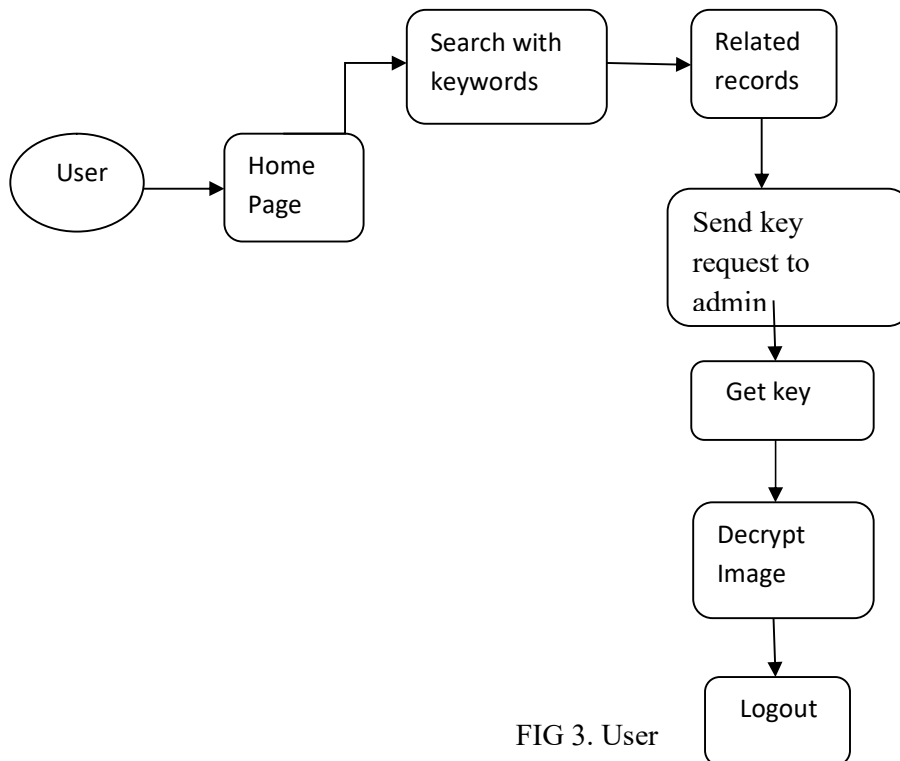


FIG 3. User

4. Admin:

Here the admin will handle whole the site. The admin will add the electronic health records into cloud, when he upload the record he get a key. And the admin want to maintain the key with users to decrypt the record.

- Login.
- Upload records.
- View user key requests.
- Give response (Accept/decline) to the user key request.
- View records details.

He had his unique username and password apart from those he can't be able to perform any operation why because he can't get into his home page where these operations are maintained.

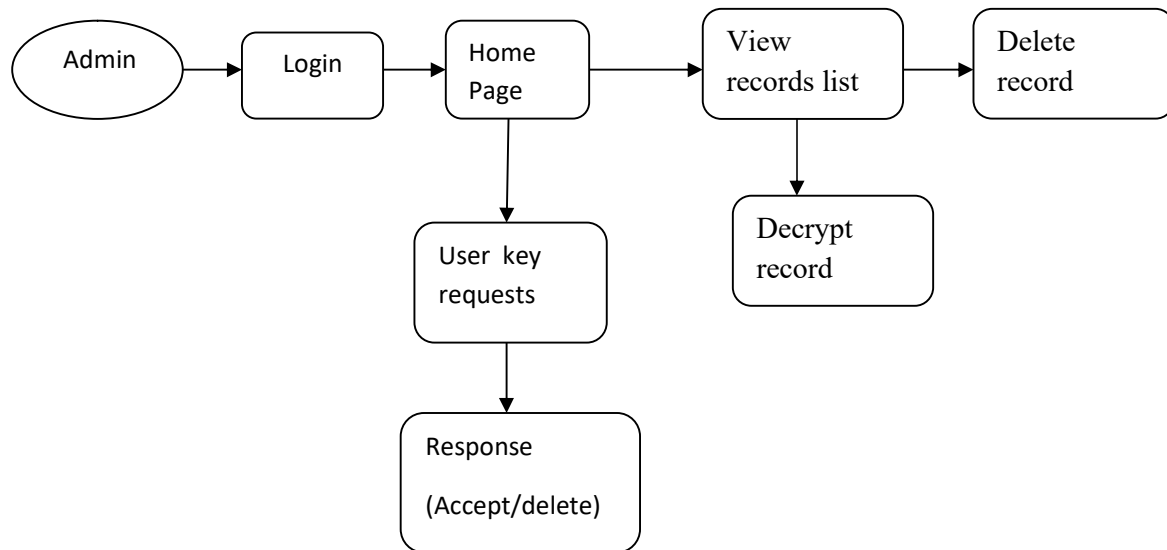


Fig 4. Admin

XML Security for Medical Document Security

Since the proposing of homomorphic properties, Fully Homomorphic Encryption (FHE) has been considered as the Holy Grail in cryptography. After Gentry's breakthrough on lattice-based FHE [11], a general solution has been shown to allow homomorphic evaluations over ciphertext domain. However, applying existing general fully homomorphic encryption scheme to image processing applications would be far from practical, due to their huge computation complexity. Different from FHE, SHE schemes can only support limited times of homomorphic operations. Considering the design targets of secure image processing mechanisms, SHE schemes seem to be suitable for some image processing applications. Here, we first briefly introduce the framework of the state-of-the-art practical SHE scheme before discussing its merits and drawbacks.

2.2.4 GIVEN INPUT AND EXPECTED OUTPUT

1) User Interface

Input: Enter login name and password.

Output: If valid user means directly open the home page otherwise show the error message and redirect to the registration page.

2) Admin upload record

Input: Give record details and data.

Output: If record was successfully uploaded the admin will get record key as well as successful message. Otherwise he will get error message.

3) User searching by keywords

Input: user will enter keywords related to record

Output: The user will get related records depending upon keywords which are entered by user.

4) Key requesting and responding

Input: user send key request to admin

Output: Admin will send particular key to the user to decrypt the file.

5) Decrypt record

Input: Enter a record key and click decrypt

Output: If the user entered a valid key of particular record the record will displayed in the form of decrypt otherwise the user will get null data.

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL

MOTIVATED by the rapid growth of image processing and data mining techniques, more and more image processing based applications are deployed in various end-users' devices. For example, content-based image search, digital watermark verification and so on. The consequent massive image processing tasks bring enormous computation overhead to data owners. To solve this problem, more and more users are outsourcing the “expensive” tasks to cloud computing platforms. In such cloud computing platform, Cloud Service Provider (CSP) offers a pay-per-use business model, which enables individual user to use robust computation power in cloud while saving time and System Model cost on setting up corresponding infrastructures.

In fact, not only individual or small business data owners refer to, Internet giants like Microsoft and Yahoo are also attracted by the benefits brought by cloud computing and authorize some services to third-party cloud computing platforms. For example, several types of data searching tasks in Microsoft Bing have been outsourced to Wolfram.

3.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

HARDWARE REQUIREMENTS

- PROCESSOR : PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
- RAM : 4GB DD RAM
- MONITOR : 15” LCD,LED MONITOR
- HARD DISK : 40 GB

3.3 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

SOFTWARE

- Front End : Core Java, J2EE (Servlets, Jsp)
- Back End : My sql 5.5
- Operating System : Windows 07
- IDE : Eclipse

3.4 FUNCTIONAL REQUIREMENTS

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behavior, and outputs. The proposed system is achieved by suppression-based and generalization-based k-anonymous and confidential databases. The protocols rely on well-known cryptographic assumptions, and we provide theoretical analyses to proof their soundness and experimental results to illustrate their efficiency.

3.5 NON-FUNCTIONAL REQUIREMENTS

3.5.1 EFFICIENCY

To address the scalability issue, we propose an edge-centric clustering scheme to extract sparse social dimensions. In sparse social dimensions, the social dimension based approach can efficiently handle networks of millions of actors while demonstrating comparable prediction performance as other non-scalable methods.

3.5.2 RELIABILITY

The dynamic nature of networks entails efficient update of the model for collective behavior prediction.

CHAPTER 4

SYSTEM DESIGN

4.1 GENERAL

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

4.1.1 ACTIVITY DIAGRAM

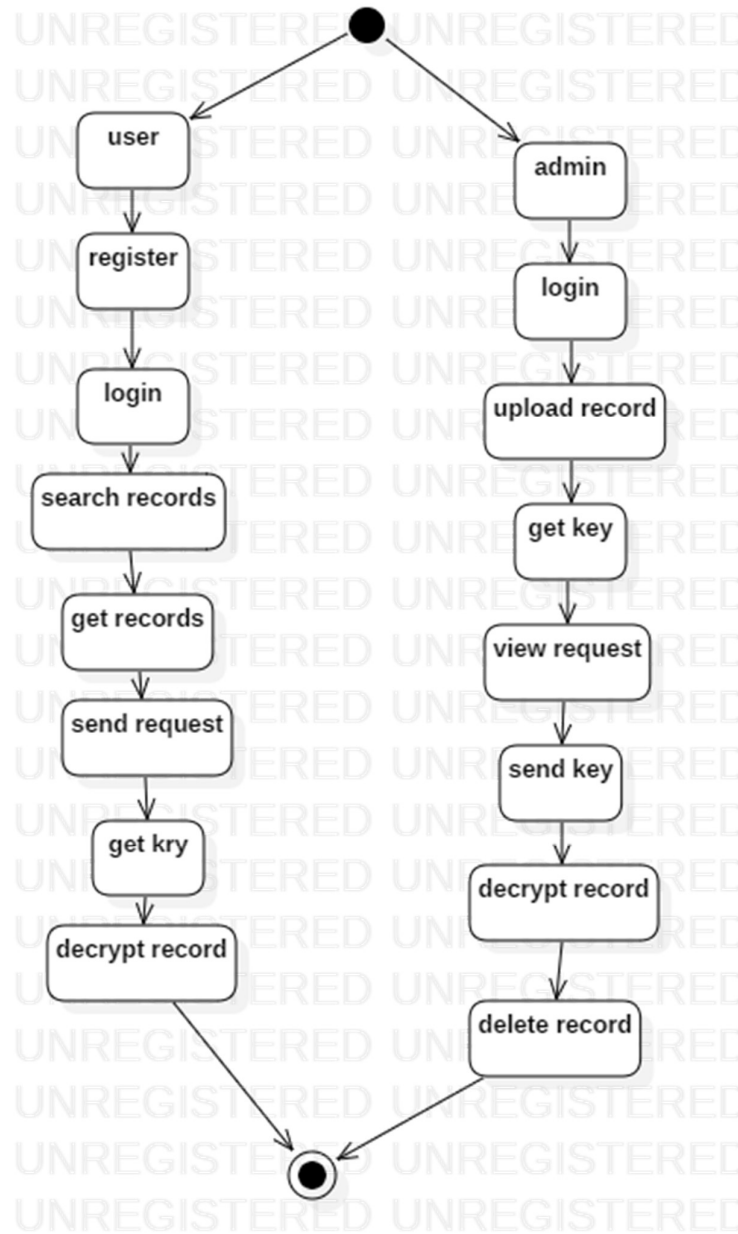


Fig 4.1.1 Activity Diagram

EXPLANATION:

. Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

4.1.2 USE CASE

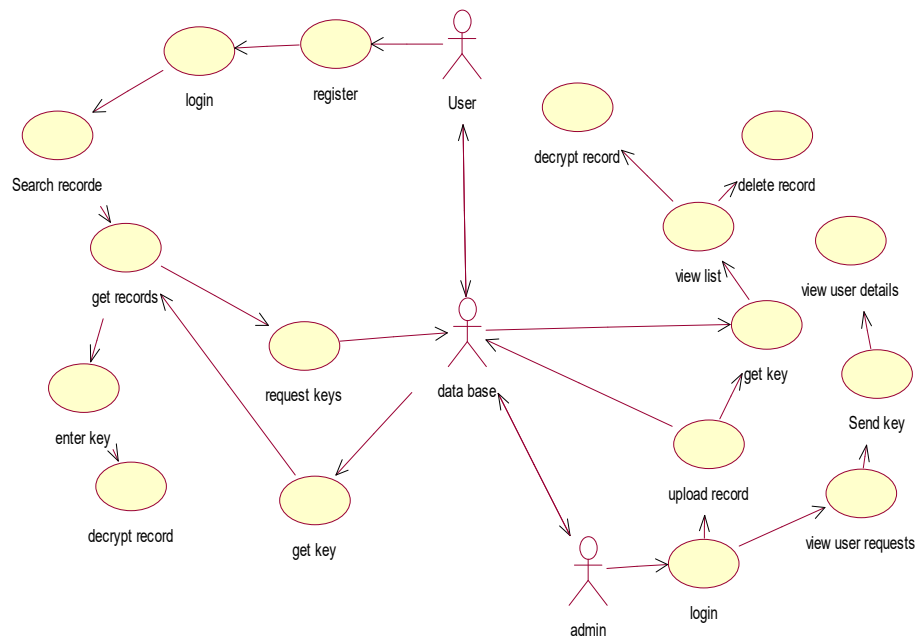


Fig 4.1.2 Use Case Diagram

EXPLANATION:

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

4.1.3 DATAFLOW DIAGRAM

Level-0:

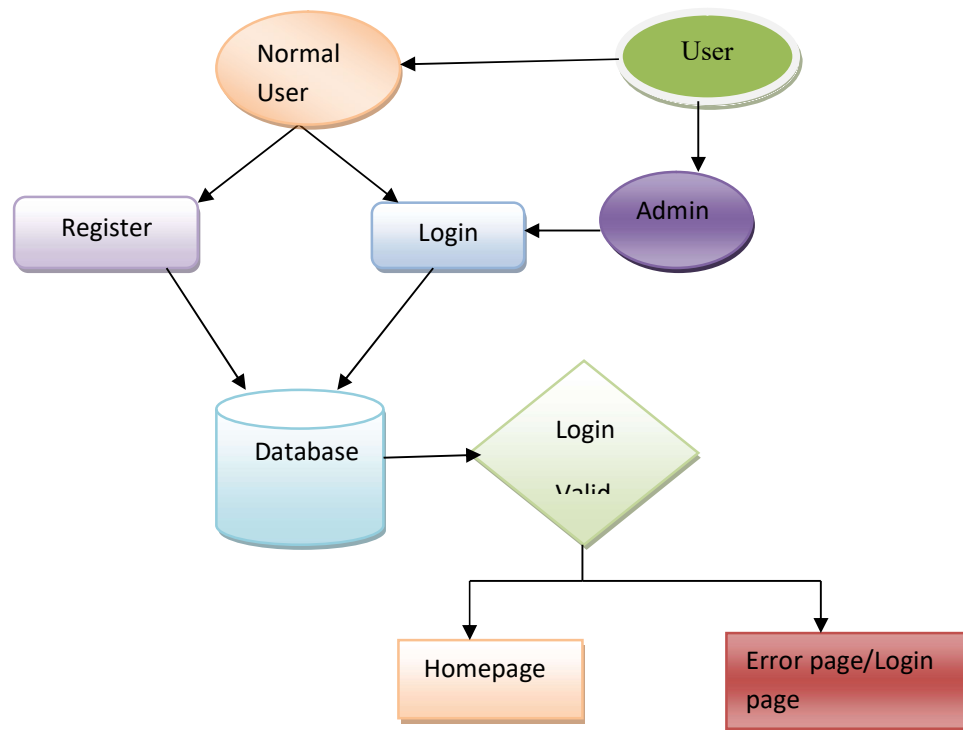


Fig 4.1.3 Dataflow Diagram level-0

Level-1:

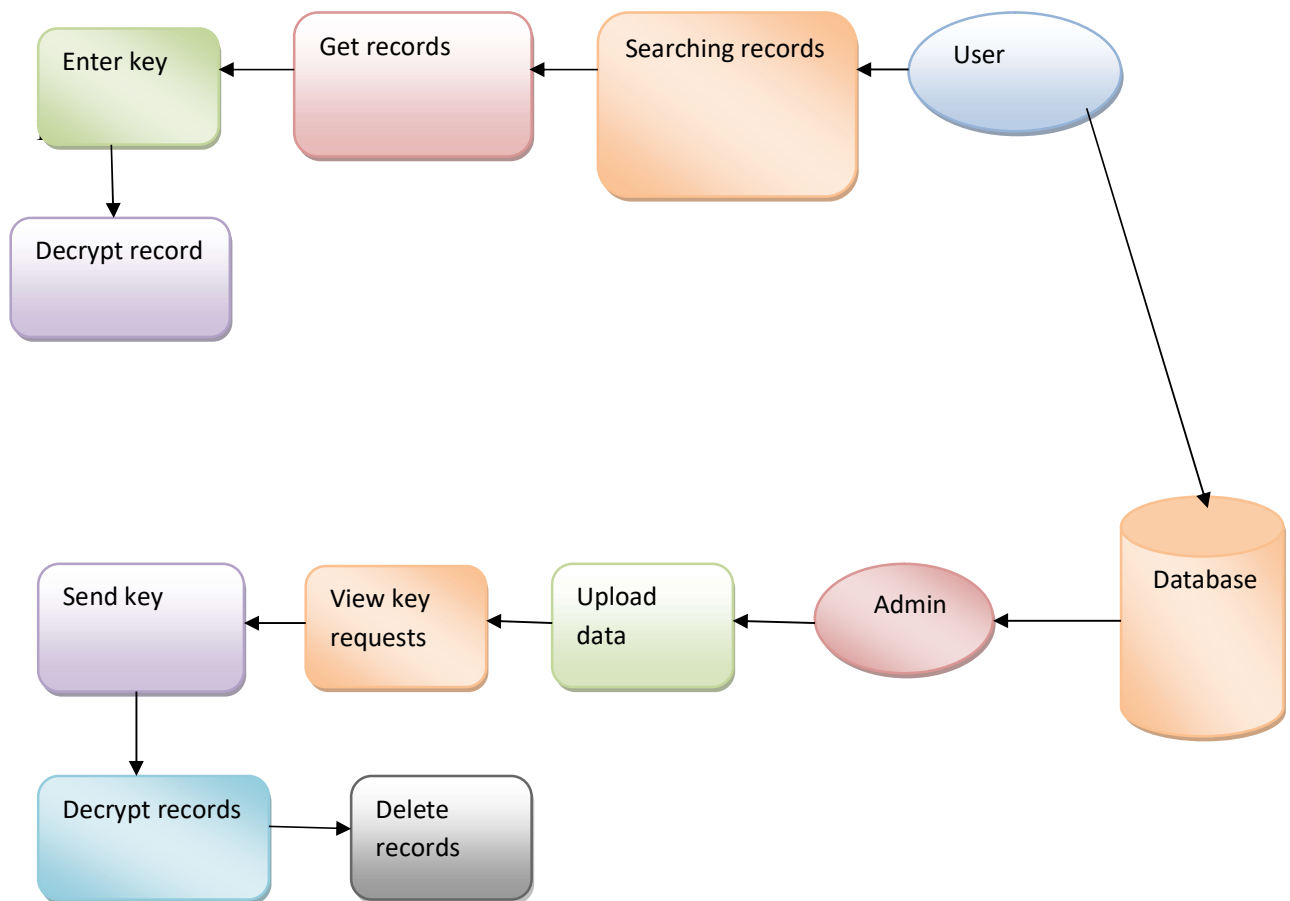


Fig 4.1.3 Dataflow Diagram level-1

Explanation

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel

4.1.4 SEQUENCE DIAGRAM

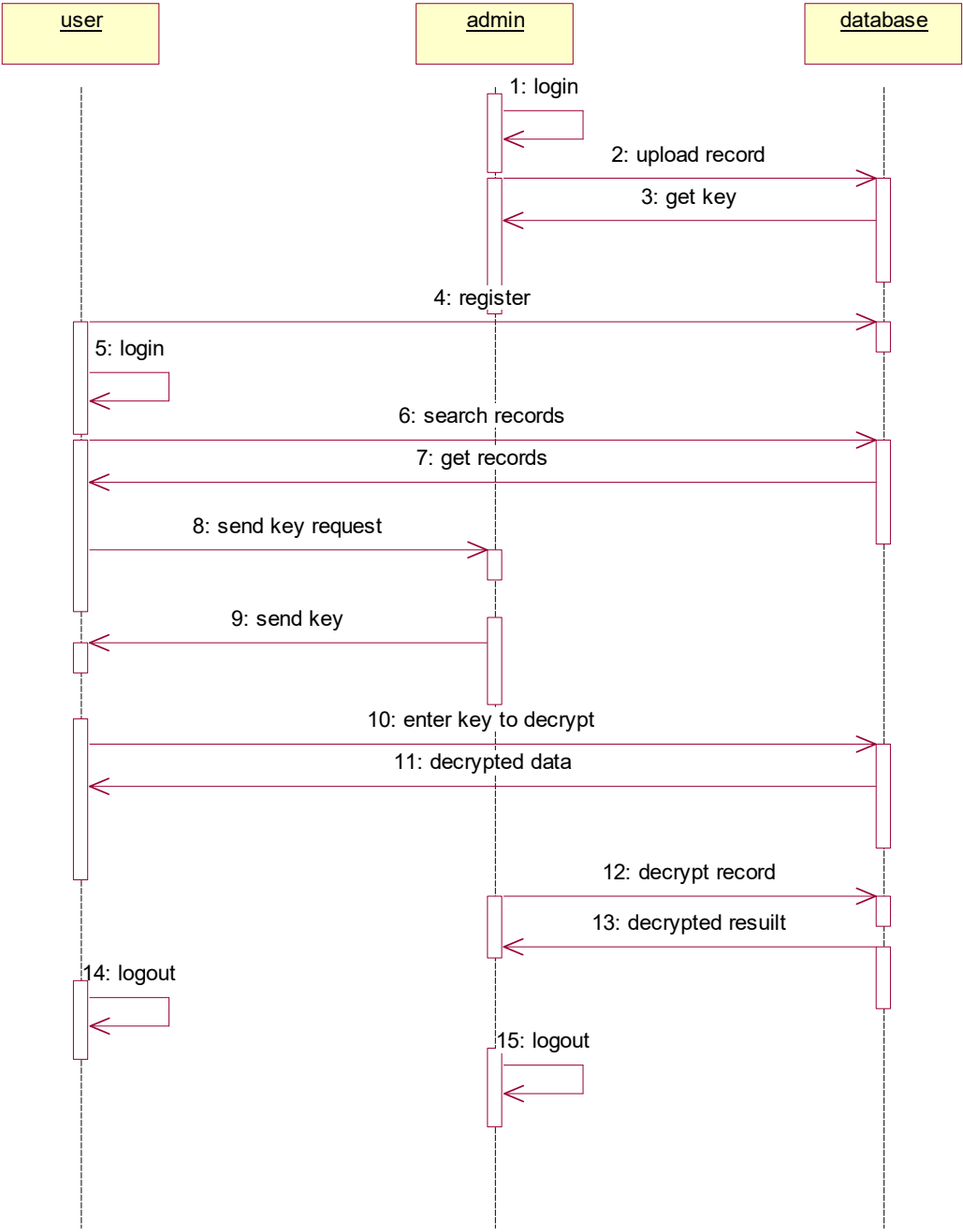


Fig 4.1.4 Sequence Diagram

EXPLANATION:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

4.1.5 COLLABORATION DIAGRAM:

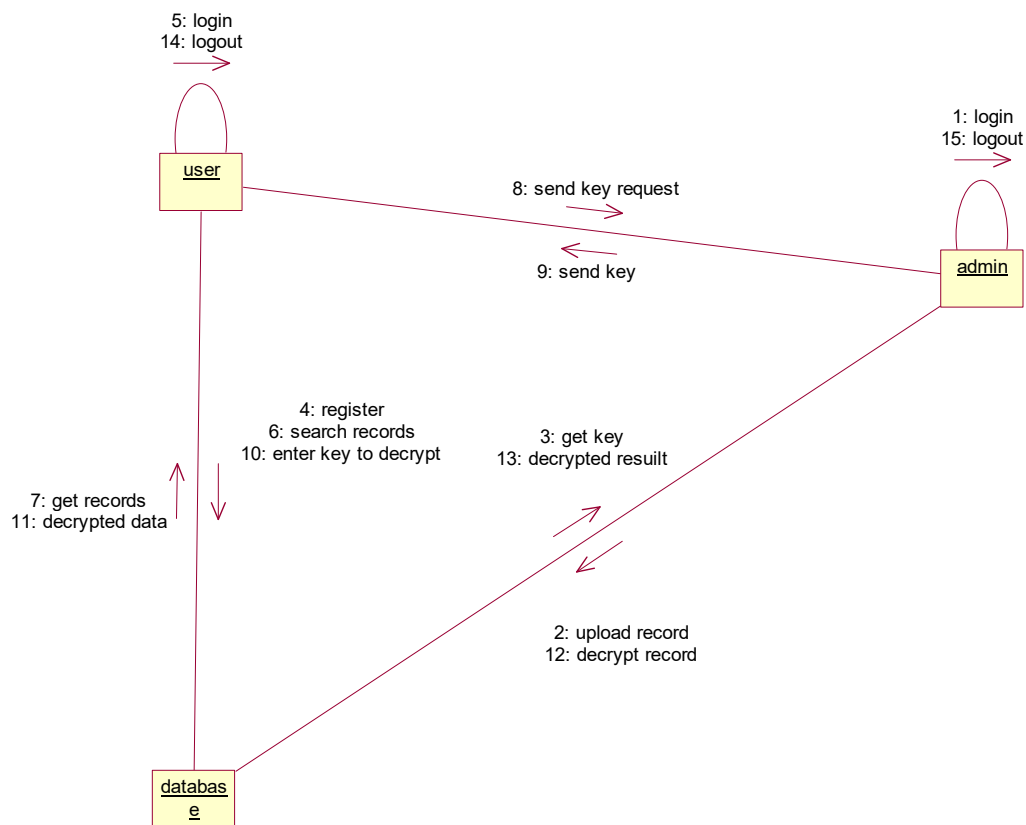


Fig 4.1.5 Collaboration Diagram

EXPLANATION:

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

4.1.6 Object diagram

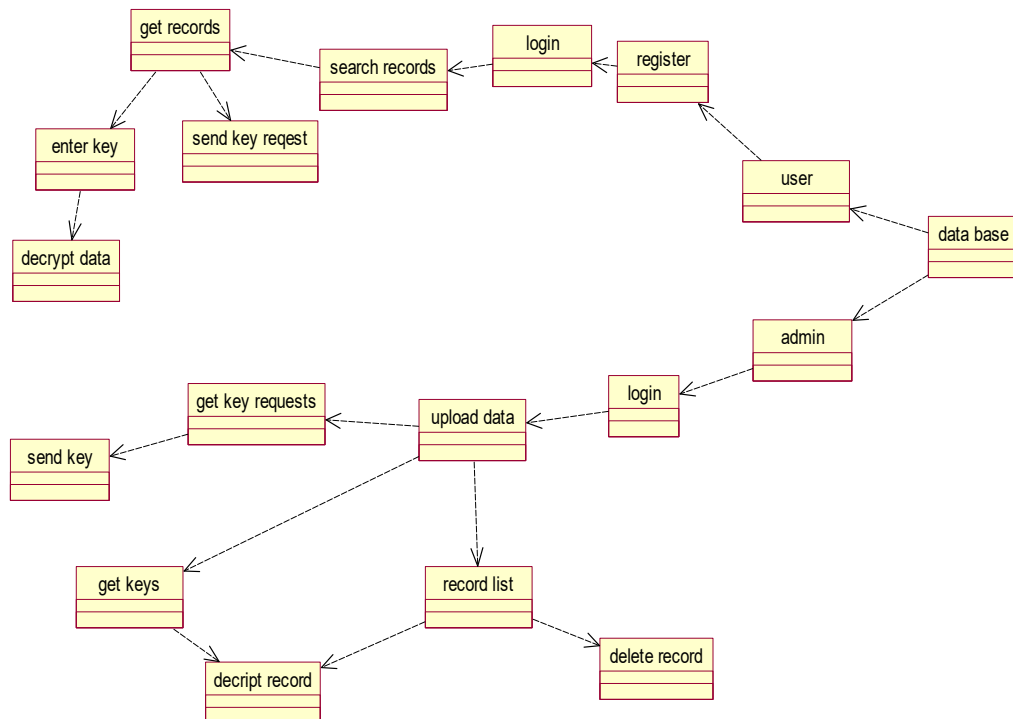


Fig 4.1.6 Object Diagram

EXPLANATION:

In the above diagram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

4.1.7 CLASS DIAGRAM

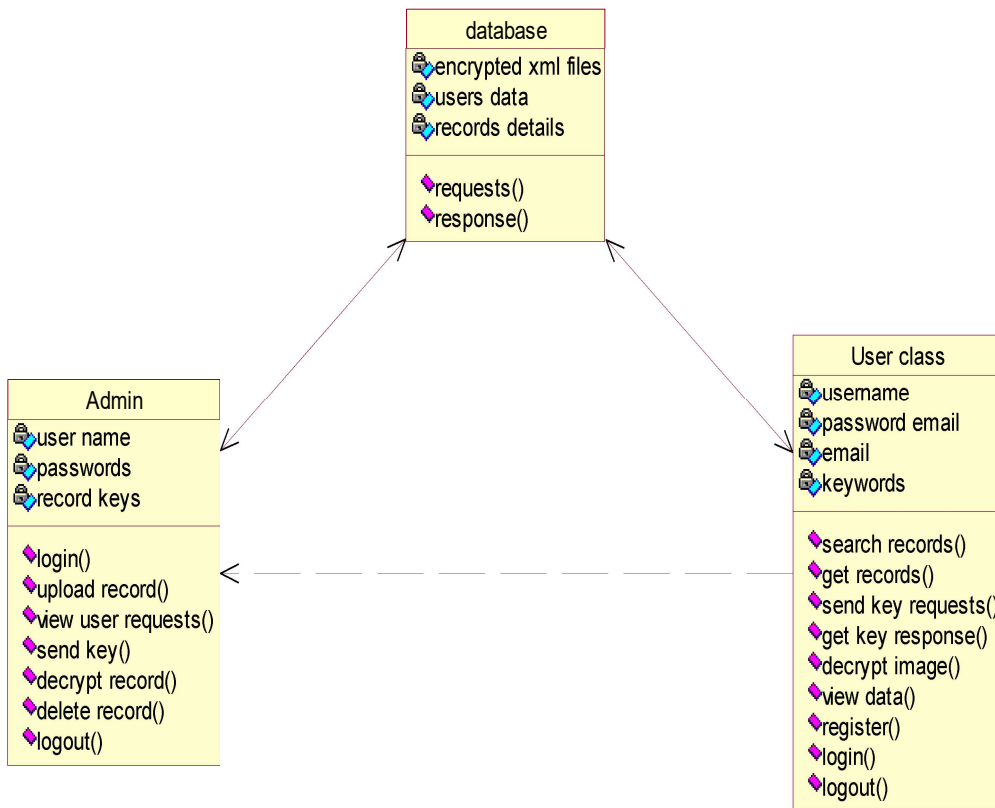


Fig 4.1.7

EXPLANATION:

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project.

4.1.8 ARCHITECTURE DIAGRAM

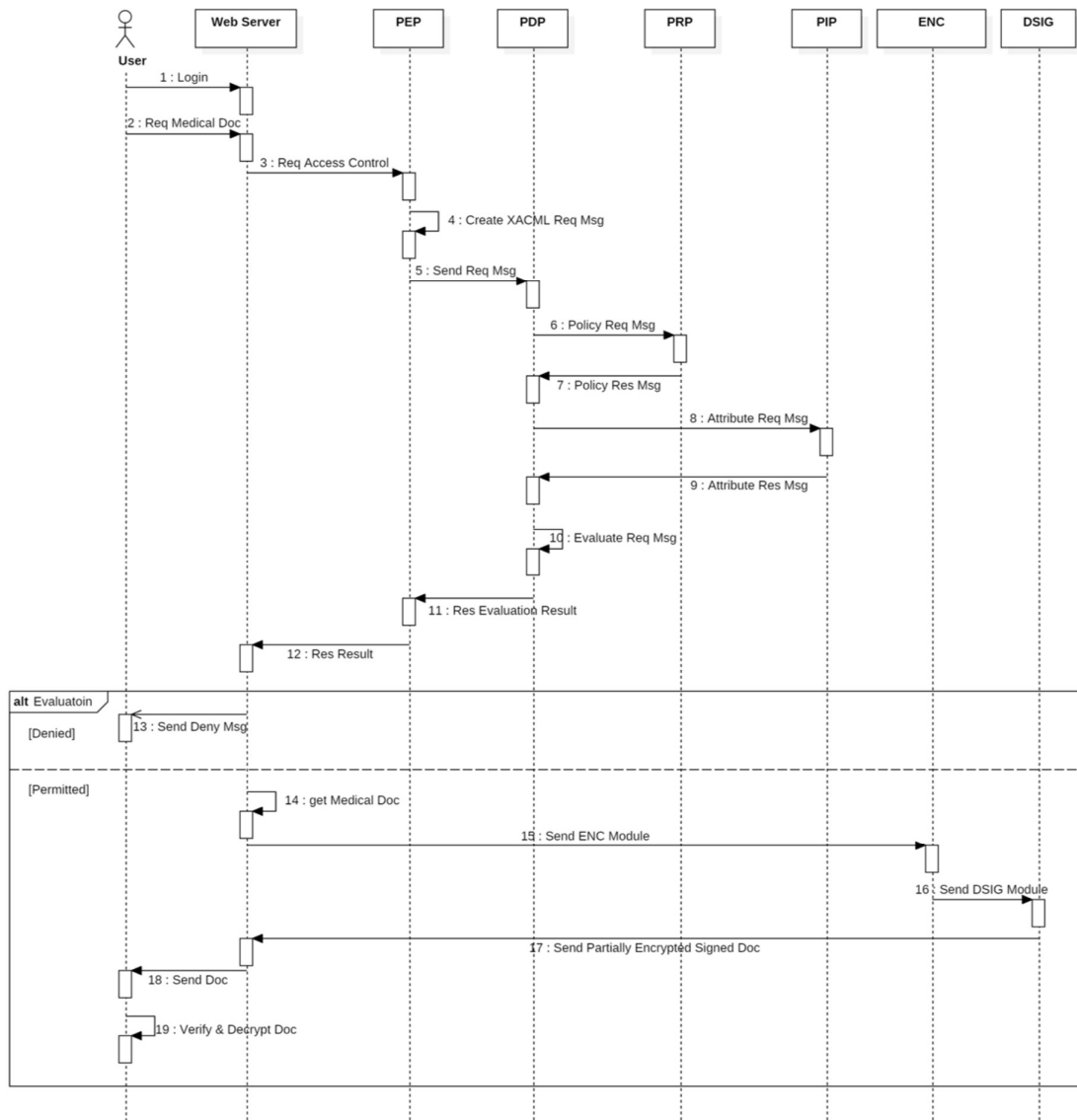


Fig 4.1.8 Architecture Diagram

EXPLANATION:

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages (ADLs).

4.1.9 Deployment Diagram:

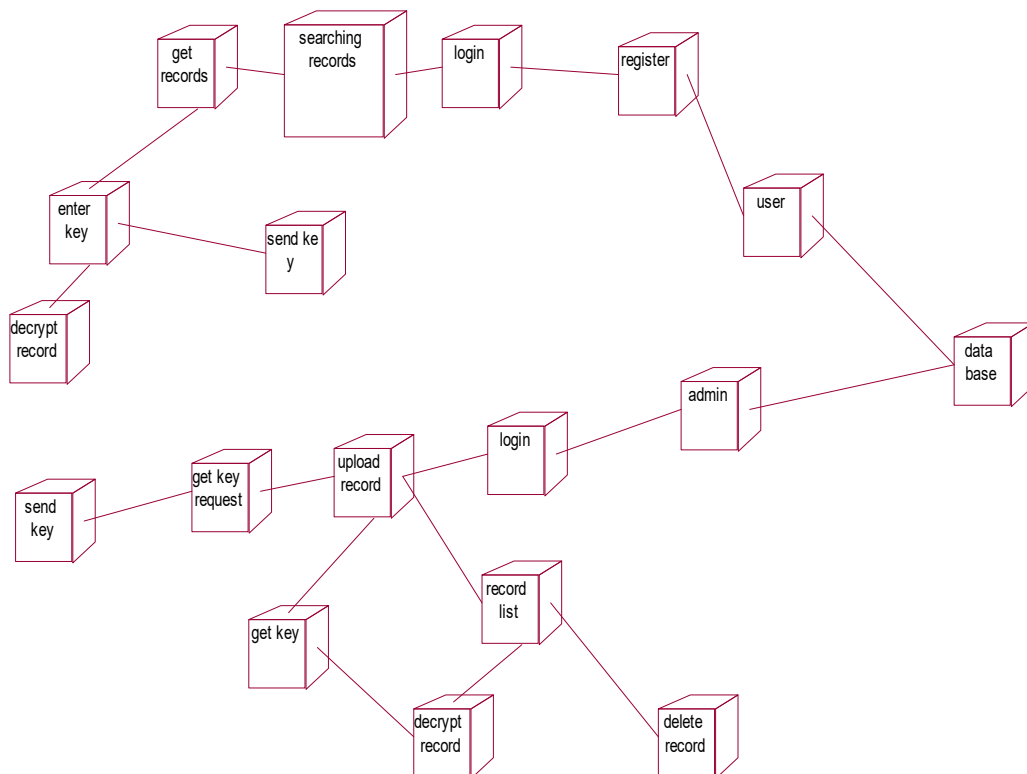


Fig 4.1.9 Deployment Diagram

4.1.10 E –R DIAGRAM

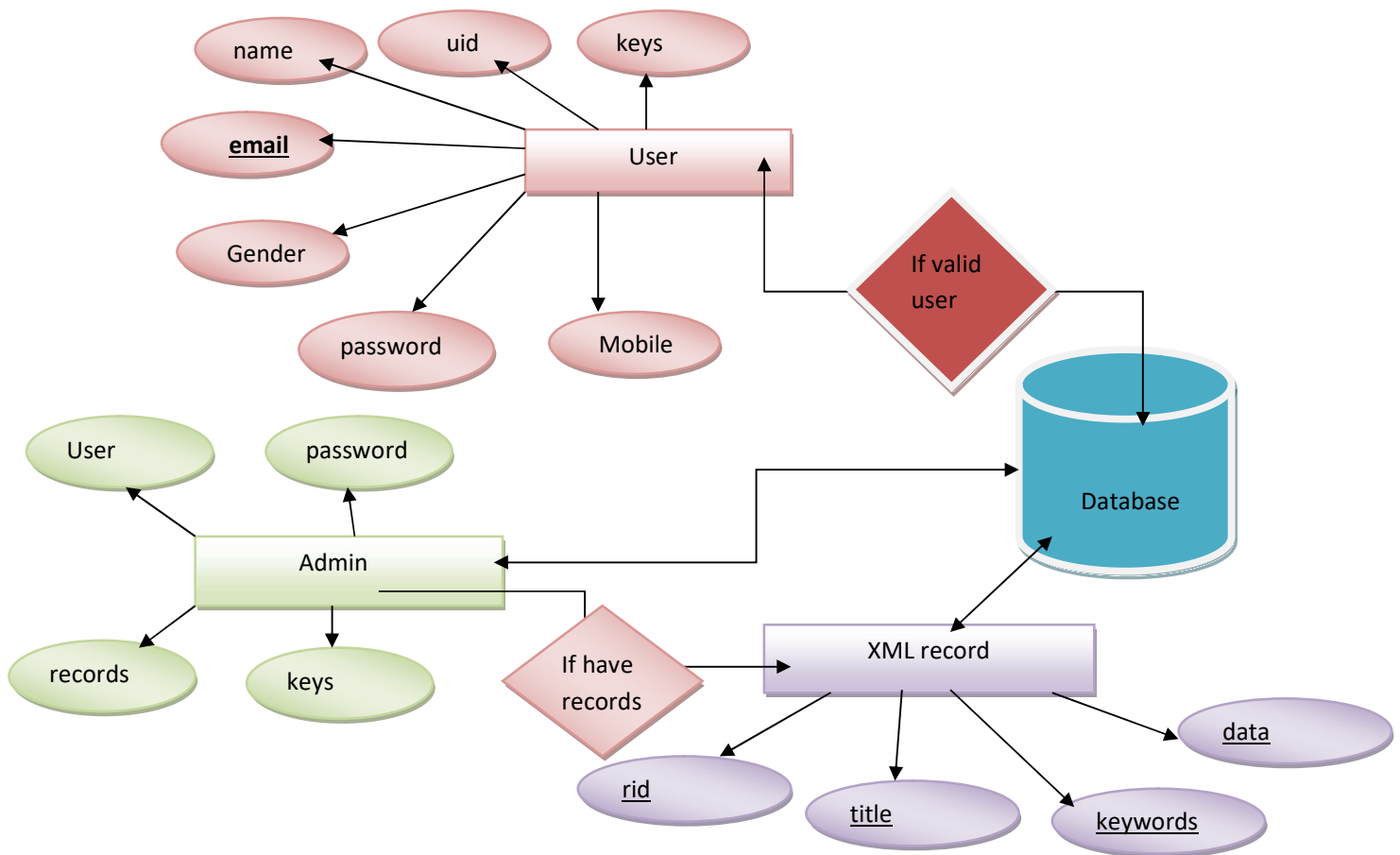


Fig 4.1.10 E-R Diagram

EXPLANATION:

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. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators.

4.1.11 State Diagram

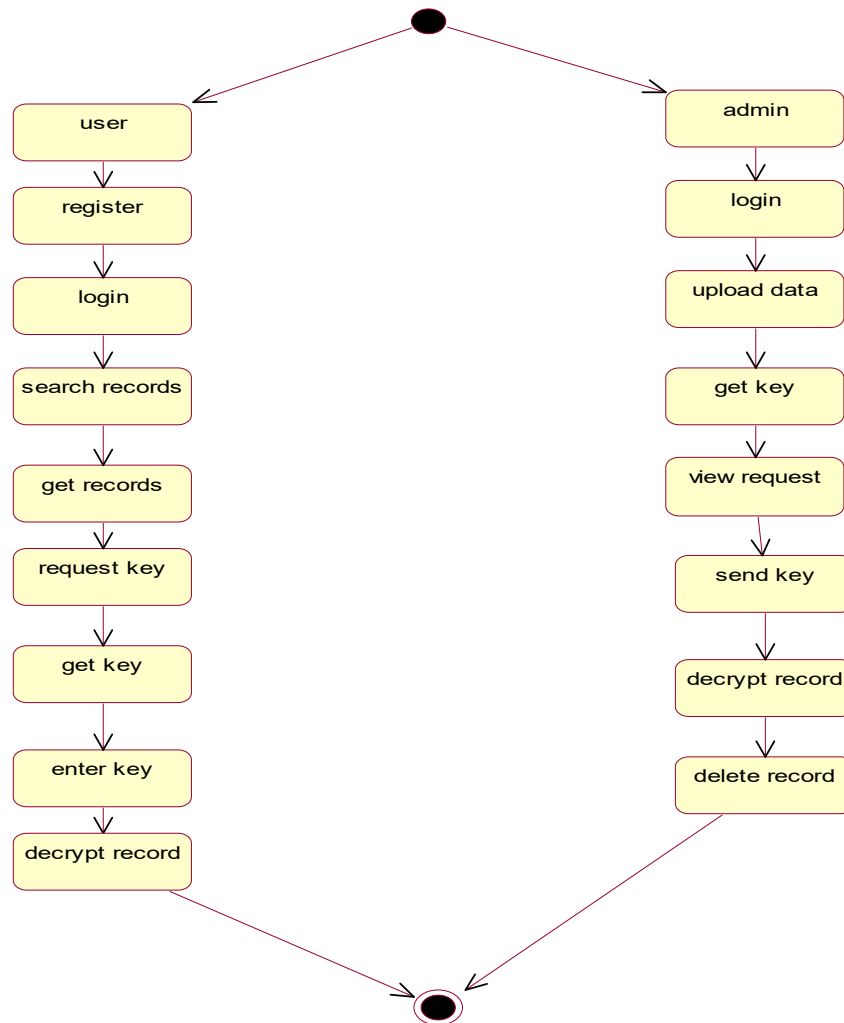


Fig 4.1.11 State Diagram

Explanation

State 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. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structural development.

4.1.12 COMPONENT DIAGRAM:

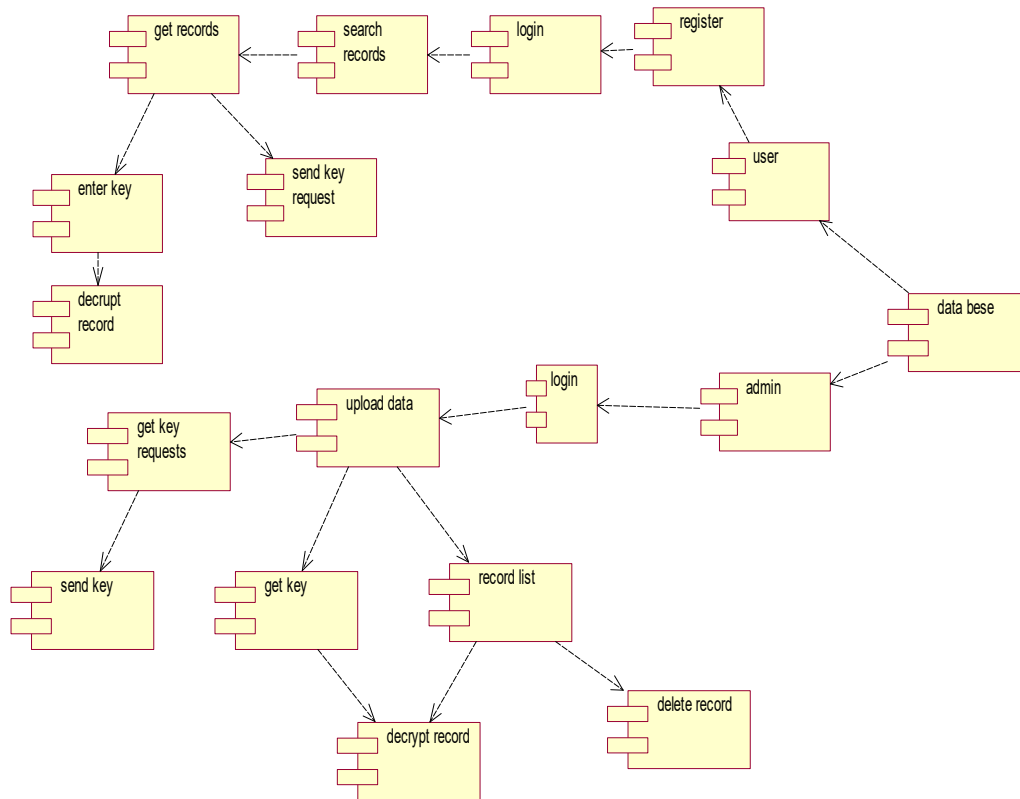


Fig 4.1.12 Component Diagram

Explanation

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

4.2 CONCLUSION

This paper studies the problem of privacy-preserving image processing on cloud computing platform, which could enable any fancy image processing based applications on devices with limited computation power. For example, a variety of instant image processing apps on the lens, watch or other personal devices. Comparing with other outsourced computation tasks, image processing algorithms are relatively complicated and have high computation complexity. To solve the problem, we start with building system model and formulating design targets. After that, the state-of-the-art techniques are introduced, including homomorphic encryption, secure multiparty computation and so on. We also present several case studies for different techniques and analyze their merits and drawbacks. Through the analysis, we find that the balance among design targets: functionality, security, and efficiency makes it difficult to solve the problem by applying only one technique.

CHAPTER 5

SOFTWARE SPECIFICATIONS

5.1 GENERAL

This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA. The Primary languages are JAVA,J2EE and J2ME. In this project J2EE is chosen for implementation.

5.2 FEATURES OF JAVA

5.2.1 THE JAVA FRAMEWORK

Java is a programming language originally developed by James Gosling at Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to bytecode that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications. The java framework is a new platform independent that simplifies application development internet. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

5.2.2 OBJECTIVES OF JAVA

To see places of Java in Action in our daily life, explore java.com.

Why Software Developers Choose Java

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:

- Write software on one platform and run it on virtually any other platform
- Create programs to run within a Web browser and Web services
- Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
- Combine applications or services using the Java language to create highly customized applications or services
- Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

Some Ways Software Developers Learn Java

Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

Object Oriented

To be an Object Oriented language, any language must follow at least the four characteristics.

1. **Inheritance** : It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse the existing code and adding addition a features as needed.

2. Encapsulation: It is the mechanism of combining the information and providing the abstraction.

3. Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.

4. Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

5.2.3 Java Server Pages - An Overview

Java Server Pages or JSP for short is Sun's solution for developing dynamic web sites. JSP provide excellent server side scripting support for creating database driven web applications. JSP enable the developers to directly insert java code into jsp file, this makes the development process very simple and its maintenance also becomes very easy.

JSP pages are efficient, it loads into the web servers memory on receiving the request very first time and the subsequent calls are served within a very short period of time.

In today's environment most web sites servers dynamic pages based on user request. Database is very convenient way to store the data of users and other things. JDBC provide excellent database connectivity in heterogeneous database environment. Using JSP and JDBC its very easy to develop database driven web application.

Java is known for its characteristic of "write once, run anywhere." JSP pages are flat Java Server Pages

Java Server Pages (JSP) technology is the Java platform technology for delivering dynamic content to web clients in a portable, secure and well-defined way. The Java Server Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML, and XML templates, and Java code, which is secure, fast, and independent of server platforms.

JSP has been built on top of the Servlet API and utilizes Servlet semantics. JSP has become the preferred request handler and response mechanism. Although JSP technology is going to be a powerful successor to basic Servlets, they have an evolutionary relationship and can be used in a cooperative and complementary manner.

Servlets are powerful and sometimes they are a bit cumbersome when it comes to generating complex HTML. Most servlets contain a little code that handles application logic and a lot more code that handles output formatting. This can make it difficult to separate and reuse portions of the code when a different output format is needed. For these reasons, web application developers turn towards JSP as their preferred servlets environment.

5.2.4 Evolution of Web Applications

Over the last few years, web server applications have evolved from static to dynamic applications. This evolution became necessary due to some deficiencies in earlier web site design. For example, to put more of business processes on the web, whether in business-to-consumer (B2C) or business-to-business (B2B) markets, conventional web site design technologies are not enough. The main issues, every developer faces when developing web applications, are:

1. Scalability - a successful site will have more users and as the number of users is increasing fastly, the web applications have to scale correspondingly.
2. Integration of data and business logic - the web is just another way to conduct business, and so it should be able to use the same middle-tier and data-access code.
3. Manageability - web sites just keep getting bigger and we need some viable mechanism to manage the ever-increasing content and its interaction with business systems.
4. Personalization - adding a personal touch to the web page becomes an essential factor to keep our customer coming back again. Knowing their preferences, allowing them to configure the information they view, remembering their past transactions or frequent search keywords are all important in providing feedback and interaction from what is otherwise a fairly one-sided conversation.

Apart from these general needs for a business-oriented web site, the necessity for new technologies to create robust, dynamic and compact server-side web applications has been realized. The main characteristics of today's dynamic web server applications are as follows:

1. Serve HTML and XML, and stream data to the web client
2. Separate presentation, logic and data
3. Interface to databases, other Java applications, CORBA, directory and mail services
4. Make use of application server middleware to provide transactional support.
5. Track client sessions.

5.2.5 Benefits of JSP

One of the main reasons why the Java Server Pages technology has evolved into what it is today and it is still evolving is the overwhelming technical need to simplify application design by separating dynamic content from static template display data. Another benefit of utilizing JSP is that it allows to more cleanly separating the roles of web application/HTML designer from a software developer. The JSP technology is blessed with a number of exciting benefits, which are chronicled as follows:

1. The JSP technology is platform independent, in its dynamic web pages, its web servers, and its underlying server components. That is, JSP pages perform perfectly without any hassle on any platform, run on any web server, and web-enabled application server. The JSP pages can be accessed from any web server.
2. The JSP technology emphasizes the use of reusable components. These components can be combined or manipulated towards developing more purposeful components and page design. This definitely reduces development time apart from the At development time, JSPs are very different from Servlets, however, they are precompiled into Servlets at run time and executed by a JSP engine which is installed on a Web-enabled application server such as BEA WebLogic and IBM WebSphere.

5.3 Servlets

Earlier in client- server computing, each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that are mostly supported by all the browsers and web pages are displayed to the client as static documents.

A web page can merely displays static content and it also lets the user navigate through the content, but a web application provides a more interactive experience.

Any computer running Servlets or JSP needs to have a container. A container is nothing but a piece of software responsible for loading, executing and unloading the Servlets and JSP. While servlets can be used to extend the functionality of any Java- enabled server.

They are mostly used to extend web servers, and are efficient replacement for CGI scripts. CGI was one of the earliest and most prominent server side dynamic content solutions, so before going forward it is very important to know the difference between CGI and the Servlets.

5.4 Java Servlets

Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server. Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server so these are safe and portable.

Unlike applets they do not require support for java in the web browser. Unlike CGI, servlets don't use multiple processes to handle separate request. Servlets can be handled by separate threads within the same process. Servlets are also portable and platform independent.

A web server is the combination of computer and the program installed on it. Web server interacts with the client through a web browser. It delivers the web pages to the client and to an application by using the web browser and the HTTP protocols respectively.

The define the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using File Transfer Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

5.5 Conclusion

JSP and Servlets are gaining rapid acceptance as means to provide dynamic content on the Internet. With full access to the Java platform, running from the server in a secure manner, the application possibilities are almost limitless. When JSPs are used with Enterprise JavaBeans technology, e-commerce and database resources can be further enhanced to meet an enterprise's needs for web applications providing secure transactions in an open platform. J2EE technology as a whole makes it easy to develop, deploy and use web server applications instead of mingling with other technologies such as CGI and ASP. There are many tools for facilitating quick web software development and to easily convert existing server-side technologies to JSP and Servlets.

CHAPTER 6

IMPLEMENTATION

6.1 GENERAL

This chapter describes implementation of the project.

6.2 IMPLEMENTAION

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>VTJCC04_2018</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
  <meta name="description" content="Free HTML5 Website Template by uicookies.com"
/>
  <meta name="keywords" content="free bootstrap 4, free bootstrap 4 template, free
website templates, free html5, free template, free website template, html5, css3,
mobile first, responsive" />
  <meta name="author" content="uicookies.com" />

  <link href="https://fonts.googleapis.com/css?family=Work+Sans" rel="stylesheet">

  <link rel="stylesheet" href="css/bootstrap.min.css">
  <link rel="stylesheet" href="css/open-iconic-bootstrap.min.css">

  <link rel="stylesheet" href="css/owl.carousel.min.css">
  <link rel="stylesheet" href="css/owl.theme.default.min.css">

  <link rel="stylesheet" href="css/icomoon.css">
  <link rel="stylesheet" href="css/flaticon.css">
  <link rel="stylesheet" href="css/animate.css">
  <link rel="stylesheet" href="css/bootstrap-datepicker.css">
  <link rel="stylesheet" href="css/style.css">

</head>
<body>

  <nav class="navbar navbar-expand-lg navbar-dark bg-dark probootstrap-navbar-
dark">
    <div class="container">
      <!-- <a class="navbar-brand" href="index.html">Health</a> -->
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#probootstrap-nav" aria-controls="probootstrap-nav" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
```

```

    <div class="collapse navbar-collapse" id="probootstrap-nav">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active"><a href="#" class="nav-link pl-
0">Home</a></li>
        <li class="nav-item"><a href="#reg" class="nav-link">Register</a></li>
        <li class="nav-item"><a href="#uLogin" class="nav-link">User
Login</a></li>
        <li class="nav-item "><a href="#aLogin" class="nav-link">Admin
Login</a></li>
      </ul>
      <div class="ml-auto">
        <form action="#" method="get" class="probootstrap-search-form mb-sm-0
mb-3">
          <div class="form-group">
            <button class="icon submit"><span class="icon-magnifying-
glass"></span></button>
            <input type="text" class="form-control" placeholder="Search">
          </div>
        </form>
      </div>
    </div>
  </nav>
  <!-- END nav -->
  <header role="banner" class="probootstrap-header py-5">

  <section class="mb-5">
    <div class="container">
      <div class="row">
        <div class="col-md-12 mb-5">
          <h1 class="display-6">Privacy-preserving Attribute-based Access Control
Model for XML-based Electronic Health Record System</h1>
        </div>
      </div>
    </div> </section>
  </header>

  <section class="probootstrap-services">
    <div class="container">
      <div class="row no-gutters">
        <div class="col-md-3 pb-5 probootstrap-aside-stretch-left probootstrap-
inside">
          <div class="mb-3 pt-5">
            <h2 class="h6">Authentication</h2>
            <ul class="list-unstyled probootstrap-light mb-4">
              <li class="active"><a href="#reg">User register</a></li>
              <li><a href="#uLogin">User Login</a></li>
              <li><a href="#aLogin">Admin Login</a></li>
            </ul>
          </div>
        </div>
        <div class="col-md-9 pl-md-5 pb-5 pl-0 probootstrap-inside" id="reg">
          <h1 class="mt-4 mb-4">Registration</h1>
          <div class="row">
            <div class="col-md-12">
              <form action="Reg" method="post" class="probootstrap-form">
                <div class="form-group">

```



```

        </div>
        <div class="col-md-4">
            <input type="submit" value="User Login" class="btn btn-primary btn-
block">
        </div>

    </div>
</form>
</div>
</section>

<section class="probootstrap-subscribe" id="aLogin">
    <div class="container">
        <div class="row mb-5">
            <div class="col-md-12">
                <h2 class="h1 text-white">Admin Login</h2>
            </div>
        </div>
        <form action="ALogin" method="post">
            <div class="row">
                <div class="col-md-4">
                    <input type="text" class="form-control" name="email"
placeholder="Email">
                </div>
                <div class="col-md-4 mb-md-0 mb-3">
                    <input type="password" class="form-control" name="pass"
placeholder="Password">
                </div>
                <div class="col-md-4">
                    <input type="submit" value="Admin Login" class="btn btn-primary btn-
block">
                </div>
            </div>
        </form>
    </div>
</section>

<footer class="probootstrap-footer">
    <div class="container">
        <div class="row mb-5">
            <div class="col-md-3">
                <h3 class="heading">Latest Blog</h3>
                <ul class="list-unstyled probootstrap-footer-recent-post">
                    <li>
                        <a href="#"><span class="date">November 15, 2017</span> The practice
of medicine is a lot like parenting </a>
                    </li>
                    <li>
                        <a href="#"><span class="date">November 15, 2017</span> Physicians:
Want to overcome burnout? Start studying business. </a>
                    </li>
                </ul>
            </div>
            <div class="col-md-3">
                <h3 class="heading">Head Office</h3>
                <p class="mb-5">98 West 21th Street, Suite 721 New York NY 10016</p>
                <h3 class="heading text-white">Open</h3>
            </div>
        </div>
    </div>

```

```

    <p>
      Mon-Fri 7:30-18:00 <br>
      Sat 7:30-18:00 <br>
      Sun 7:30-18:00
    </p>
  </div>
  <div class="col-md-3">
    <h3 class="heading">Quick Links</h3>
    <ul class="list-unstyled probootstrap-footer-links">
      <li><a href="#">Home</a></li>
      <li><a href="#">Department</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </div>
  <div class="col-md-3">
    <h3 class="heading">Follow us</h3>
    <ul class="probootstrap-footer-social">
      <li><a href="#"><span class="icon-twitter"></span></a></li>
      <li><a href="#"><span class="icon-facebook"></span></a></li>
      <li><a href="#"><span class="icon-linkedin"></span></a></li>
    </ul>
  </div>
</div>
<!-- END row -->
<div class="row probootstrap-copyright">
  <div class="col-md-12">
    <p><small>&copy; 2017 <a href="https://uicookies.com/"
target="_blank">uiCookies Health</a>. All Rights Reserved. Designed &
Developed by <a href="https://uicookies.com/" target="_blank">uicookies.com</a>
(please don't remove credit, see <a href="https://uicookies.com/license/"
target="_blank">license</a>) <br> Demo Images <a
href="https://pexels.com/">Pexels</a> </small></p>
  </div>
</div>
</div>
</footer>

<!-- loader -->
<div id="probootstrap-loader" class="show fullscreen"><svg class="circular"
width="48px" height="48px"><circle class="path-bg" cx="24" cy="24" r="22"
fill="none" stroke-width="4" stroke="#eeeeee"/><circle class="path" cx="24"
cy="24" r="22" fill="none" stroke-width="4" stroke-miterlimit="10"
stroke="#32609e"/></svg></div>

<script src="js/jquery-3.2.1.min.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/jquery.waypoints.min.js"></script>
<script src="js/bootstrap-datepicker.js"></script>
<script src="js/jquery.animateNumber.min.js"></script>

<script src="js/main.js"></script>
</body>

</html>

```

Login.java

```
package com.servlets;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.SQLException;


import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


import com.beans.UserBean;


/**
 * Servlet implementation class Login
 */
@WebServlet("/Login")

public class Login extends HttpServlet {
```

```

    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Login() {

        super();

        // TODO Auto-generated constructor stub

    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {

        // TODO Auto-generated method stub

    }

    /**

```

```
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
*/
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {
```

```
// TODO Auto-generated method stub
```

```
PrintWriter out=response.getWriter();
```

```
String uname=request.getParameter("email");
```

```
String pass=request.getParameter("pass");
```

```
UserBean u=new UserBean();
```

```
u.setEmail(uname);
```

```
u.setPass(pass);
```

```
try {
```

```
if(com.controller.DbConnection.checkLog(u))
```

```
{
```

```
HttpSession h=request.getSession();
```

```
h.setAttribute("email", uname);
```

```
response.sendRedirect("userhome.jsp");
```



```

    }else{

        out.println("<script type=\"text/javascript\">");

        out.println("alert('failed to login');");

out.println("window.location='index.html'</script>");

    }

    } catch (SQLException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    }

}

}

```

CHAPTER 7

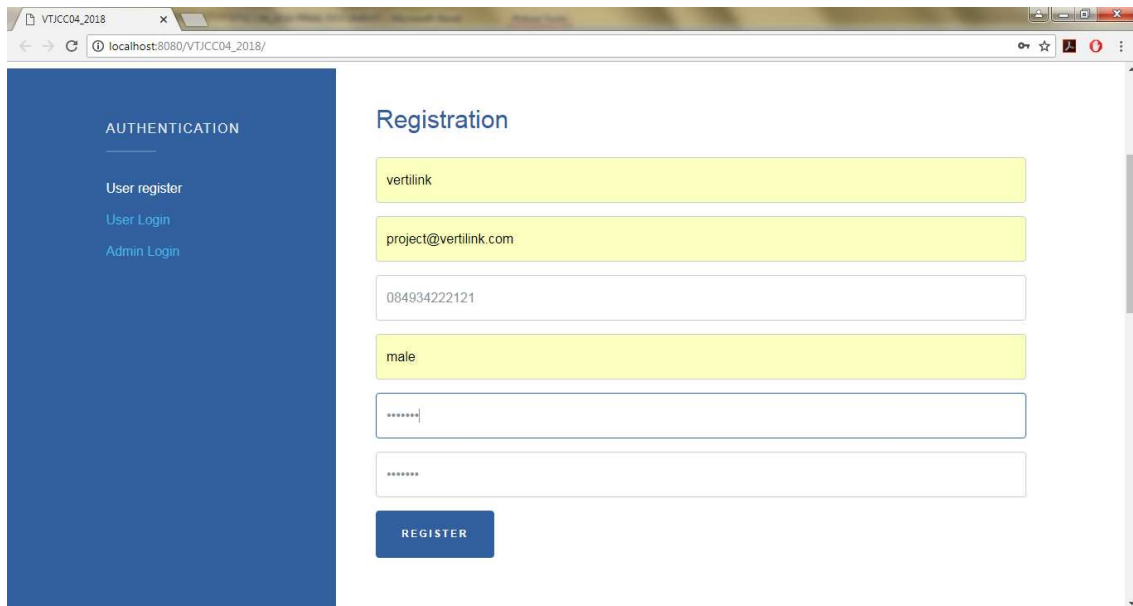
SNAPSHOTS

7.1 GENERAL

Snapshot is nothing but every moment of the application while running. It gives the clear elaborated of application. It will be useful for the new user to understand for the future steps.

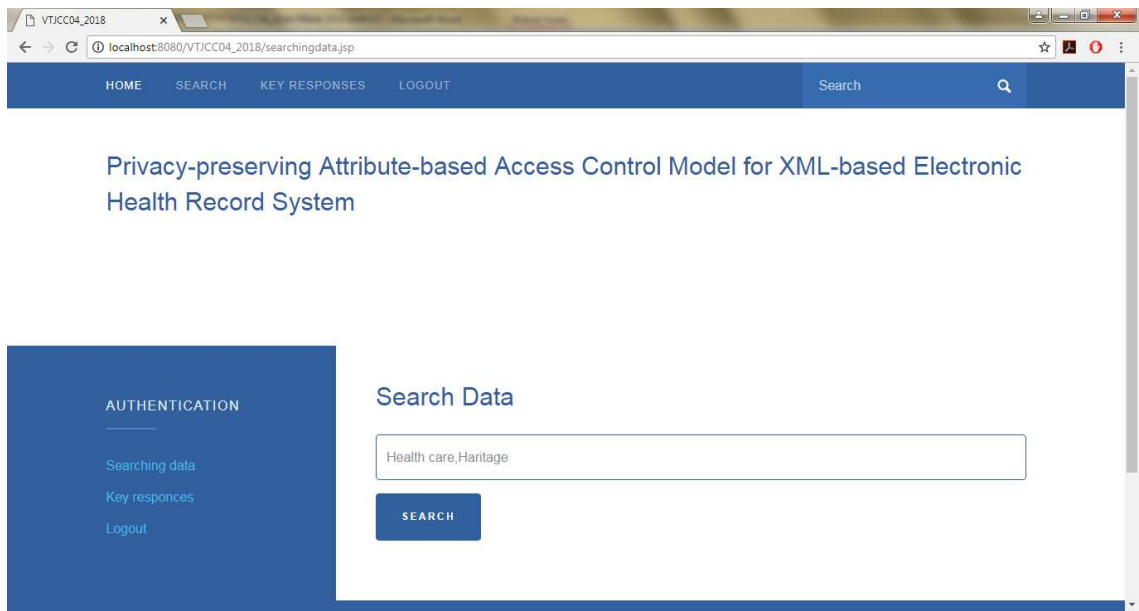
7.2VARIOUS SNAPSHOTS

Screenshot 1

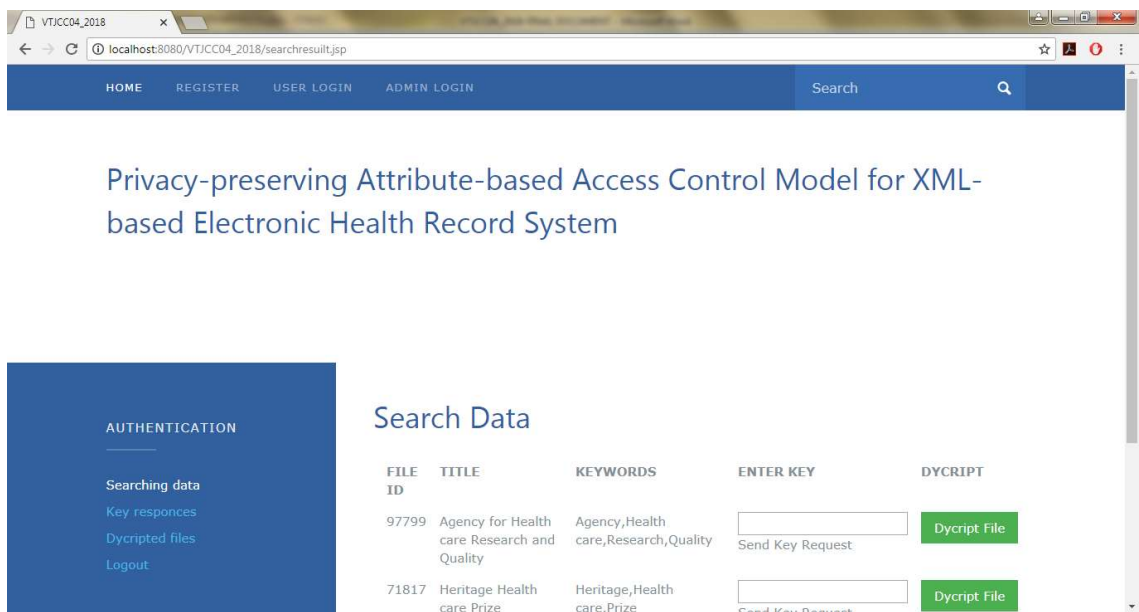


The screenshot displays a web browser window with the address bar showing 'localhost:8080/VTJCC04_2018/'. The page features a dark blue sidebar on the left with the heading 'AUTHENTICATION' and three links: 'User register', 'User Login', and 'Admin Login'. The main content area is white and titled 'Registration'. It contains a form with the following fields: a text input for 'vertiink', a text input for 'project@vertiink.com', a text input for '084934222121', a text input for 'male', a password input (masked with '*****'), and another password input (masked with '*****'). A blue 'REGISTER' button is positioned at the bottom of the form.

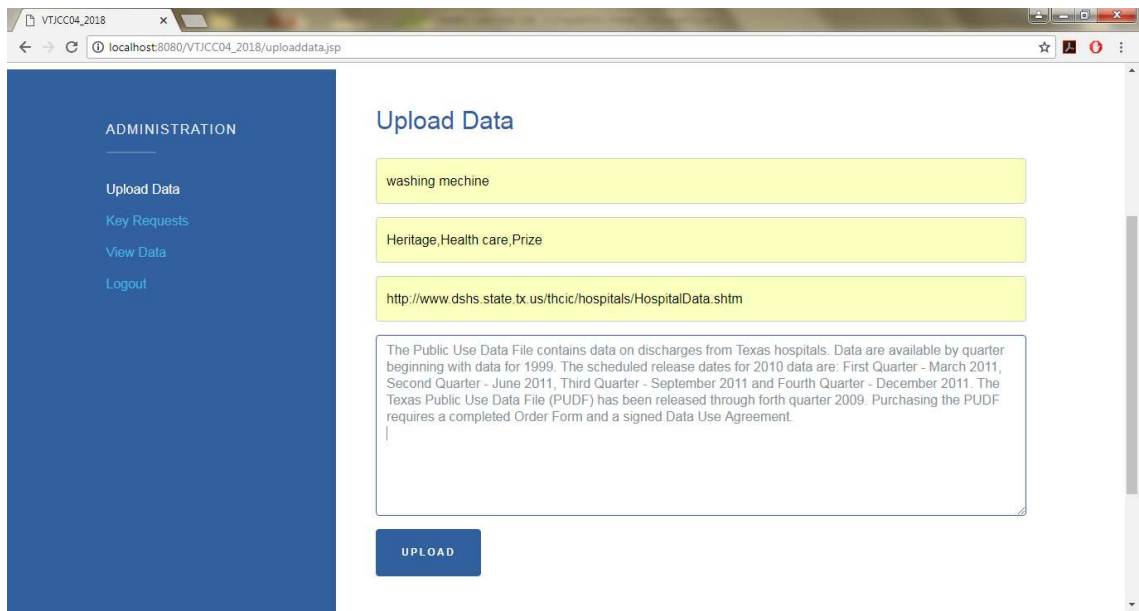
Screenshot 2



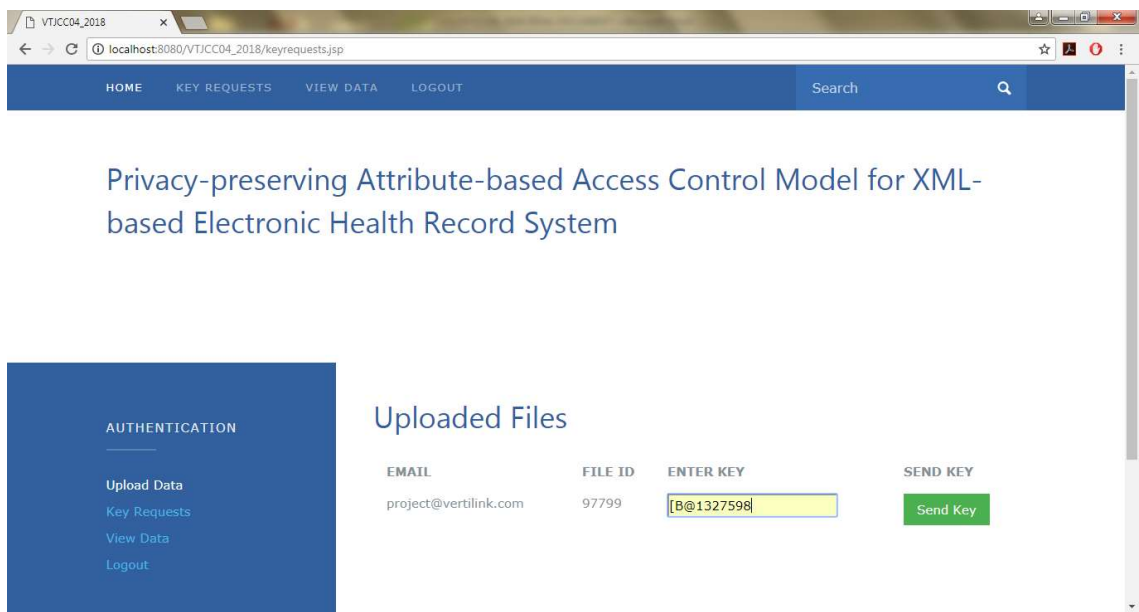
Screenshot 3



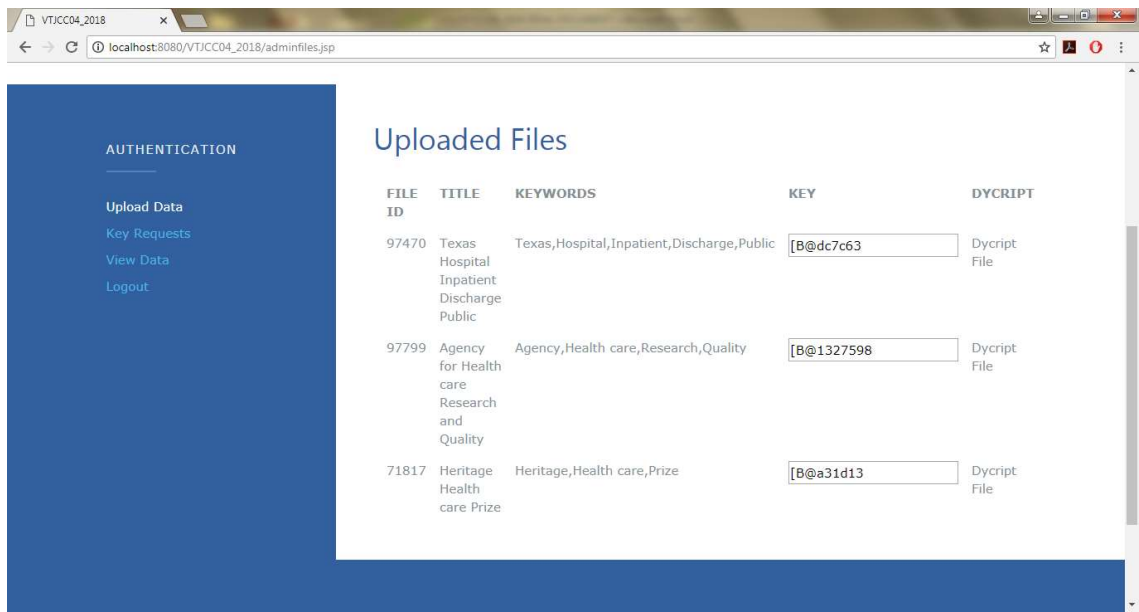
Screenshot 4



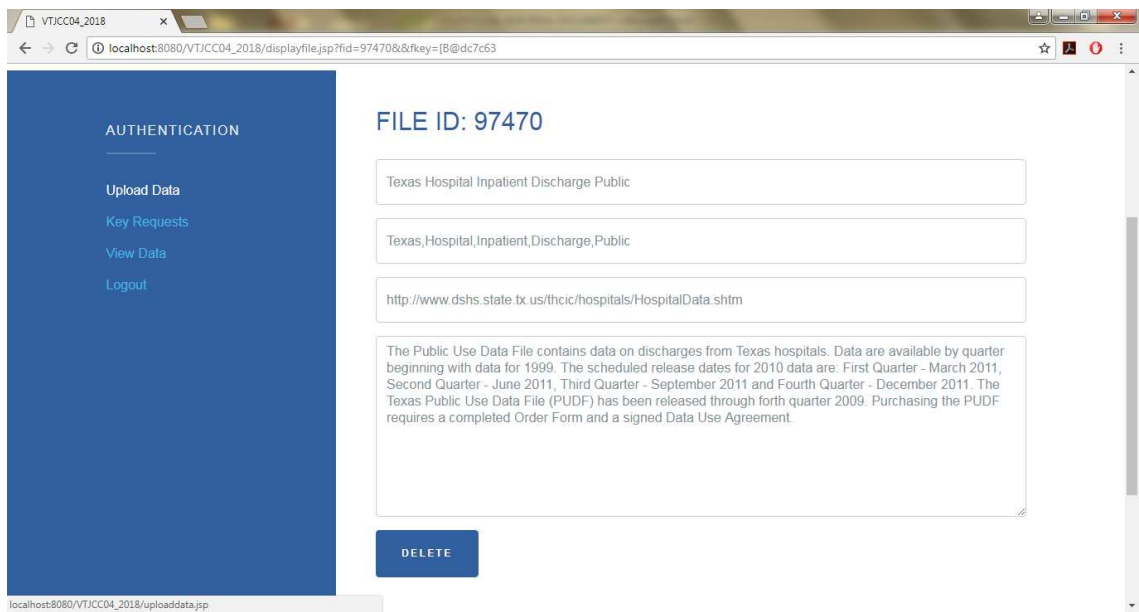
Screenshot 5



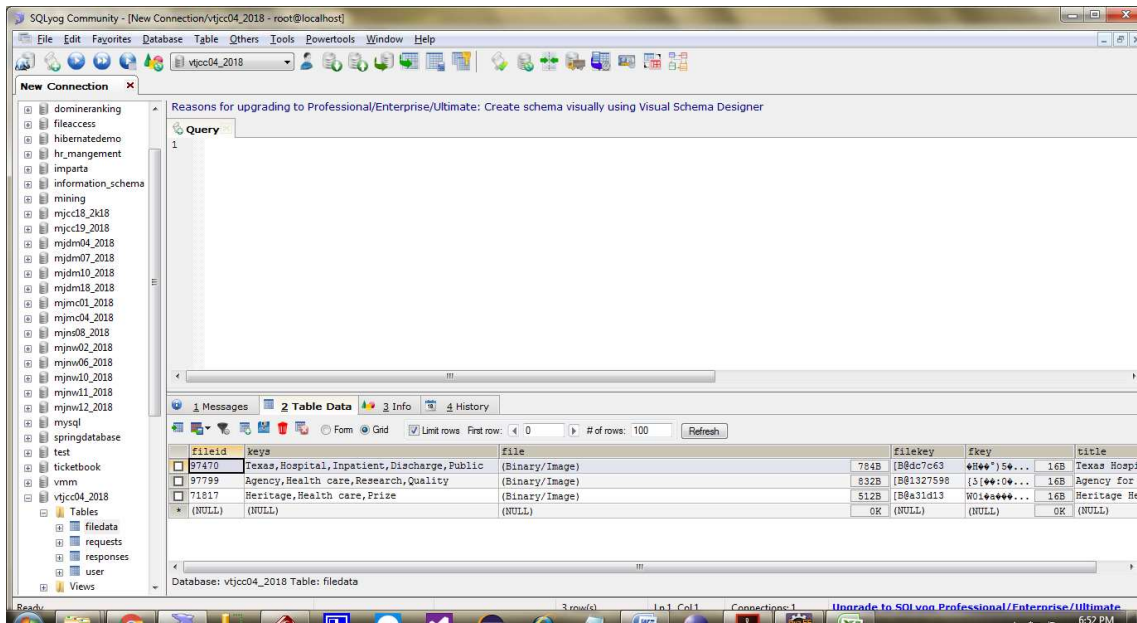
Screenshot 6



Screenshot 7



7.3 Database Structure



CHAPTER 8

SOFTWARE TESTING

8.1 GENERAL

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. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

8.2 DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

8.3 Types of Tests

8.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.3.2 Functional test

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

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

8.3.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

8.3.4 Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

8.3.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

8.3.6 User Registration test

Test Scenarios	Test Steps	Test Data	Expected Results	Success/Failure
Check user registration with valid data	1. Enter username 2. Enter Email 3. Phone no. 4. Gender 5. Password 6. Confirm password	7. Anshul 8. Anshul123@gmail.com 9. 9676124490 10. Male 11. 123 12. 123	User registration successful	Success
Check user registration with invalid data	Invalid Data	Invalid Data	Registration Failed	Success

CHAPTER 9

APPLICATIONS AND FUTURE ENHANCEMENTS

9.1 GENERAL

Cloud-based electronic health record (EHR) systems enable medical documents to be exchanged between medical institutions; this is expected to contribute to improvements in various medical services in the future. However, as the system architecture becomes more complicated, cloud-based EHR systems may introduce additional security threats when compared to existing singular systems. Thus, patients may experience exposure of private data that they do not wish to disclose. In order to protect the privacy of patients, many approaches have been proposed to provide access control to patient documents when providing health services. However, most current systems do not support fine-grained access control or take into account additional security factors such as encryption and digital signatures. In this paper, we propose a cloud-based EHR model that performs attribute-based access control using extensible access control markup language. Our EHR model, focused on security, performs partial encryption and uses electronic signatures when a patient document is sent to a document requester. We use XML encryption and XML digital signature technology. Our proposed model works efficiently by sending only the necessary information to requesters who are authorized to treat the patient in question.

9.2 APPLICATION

- Public service applications
- content based secure cloud application

9.3 FUTURE ENHANCEMENTS

In the future, we will further refine the processes used in the proposed model and implement additional security features. We will also expand the implementation of the prototype to implement a more refined system and perform quantitative performance evaluation.

CHAPTER 10

CONCLUSION AND REFERENCE

10.1 CONCLUSION:

In this paper We proposed a cloud-based EHR model that guarantees patient privacy. The proposed model is divided into two stages: access control, and the application of encryption and digital signatures. The proposed model uses an ABAC method built upon XACML. After performing access control on patient documents, encryption is performed and digital signatures are added using XML encryption and XML digital signatures as an added security measure. The proposed model provides more flexible and fine-grained control than existing RBAC systems and alleviates the risk of exposing patient privacy information by using partial encryption and electronic signatures. The implementation of a prototype demonstrated the feasibility of the proposed model. We compared the implemented security factors with those used in other related studies and determined that the proposed method is superior to previous methods in terms of security.

REFERENCES:

- [1] Tang, P. C., Ash, J. S., Bates, D. W., Overhage, J. M., & Sands, D. Z. (2006). Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption. *Journal of the American Medical Informatics Association*, [Online]. 13(2), 121-126. Available: <https://academic.oup.com/jamia/article/13/2/121/729326/Personal-Health-Records-Definitions-Benefits-and>
- [2] Waegemann, C. P. (2003). Ehr vs. cpr vs. emr. *Healthcare Informatics Online*, [Online]. 1, 1-4. Available: <https://pdfs.semanticscholar.org/ce2f/cf783c1fa2afdaa81c5a46c317e7edff04bc.pdf>
- [3] van der Linden, H., Kalra, D., Hasman, A., & Talmon, J. (2009). Inter-organizational future proof EHR systems: a review of the security and privacy related issues. *International journal of medical informatics*, [Online]. 78(3), 141-160. Available: <http://www.sciencedirect.com/science/article/pii/S1386505608001081>
- [4] Tang, P. C. (2003). Key capabilities of an electronic health record system. Washington, DC, Institute of Medicine of the National Academies. [Online]. Available: <http://www.nationalacademies.org/hmd/Reports/2003/Key-Capabilities-of-an-Electronic-Health-Record-System.aspx>
- [5] Miller, R. H., West, C., Brown, T. M., Sim, I., & Ganchoff, C. (2005). The value of electronic health records in solo or small group practices. *Health Affairs*, [Online]. 24(5), 1127-1137. Available: <http://content.healthaffairs.org/content/24/5/1127.short>
- [6] Middleton, B., Bloomrosen, M., Dente, M. A., Hashmat, B., Koppel, R., Overhage, J. M., ... & Zhang, J. (2013). Enhancing patient safety and quality of care by improving the usability of electronic health record systems: recommendations from AMIA. *Journal of the American Medical Informatics Association*, [Online]. 20(e1), e2-e8. Available: <https://academic.oup.com/jamia/article/20/e1/e2/692244/Enhancing-patient-safety-and-quality-of-care-by>
- [7] Simon, S. R., Kaushal, R., Cleary, P. D., Jenter, C. A., Volk, L. A., Poon, E. G., ... & Bates, D. W. (2007). Correlates of electronic health record adoption in office practices: a statewide survey. *Journal of the American Medical Informatics Association*, [Online]. 14(1), 110-117. Available: <https://academic.oup.com/jamia/article/14/1/110/746202/Correlates-of-Electronic-Health-Record-Adoption-in>

- [8] Ratnam, K. A., & Dominic, P. D. D. (2012, June). Cloud services-Enhancing the Malaysian healthcare sector. In Computer & Information Science (ICCIS), 2012 International Conference on. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/6297101/>
- [9] Zhang, R., & Liu, L. (2010, July). Security models and requirements for healthcare application clouds. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/5557983/>
- [10] Benaloh, J., Chase, M., Horvitz, E., & Lauter, K. (2009, Nov.). Patient controlled encryption: ensuring privacy of electronic medical records. In Proceedings of the 2009 ACM workshop on Cloud computing security. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1655024>
- [11] Ray, P., & Wimalasiri, J. (2006, Aug.). The need for technical solutions for maintaining the privacy of EHR. In Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/4462848/>
- [12] Abbas, A., & Khan, S. U. (2014). A review on the state-of-the-art privacy-preserving approaches in the e-health clouds. IEEE Journal of Biomedical and Health Informatics, [Online]. 18(4), 1431-1441. Available: <http://ieeexplore.ieee.org/abstract/document/6714376/>
- [13] eXtensible Access Control Markup Language (XACML) Version 3.0, OASIS Standard, 22 Jan. 2013, Available: <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>
- [14] XML Encryption Syntax and Processing, W3C Recommendation, 10 Dec 2002, Available: <http://www.w3.org/TR/xmlenc-core/>.
- [15] Standards for Privacy of Individually Identifiable Health Information: Final Rule. Dec. 28, 2000.