

Clinical Management Support System in the Cloud

at

Sathyabama Institute of Science and Technology

(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of Bachelor
Engineering Degree in Computer Science and Engineering By

SANABOYINA JASWANTH (38110055)

PUTHINEEDI HARISH (38110195)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC I 12B

Status by UGC I Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600 119

March – 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Sanaboyina Jaswanth (38110055), Puthineedi Harish (38110195)** who carried out the project entitled “CLINICAL MANAGEMENT SUPPORT SYSTEM IN THE CLOUD” under our supervision from November 2021 to March 2022.

Internal Guide

Dr Cruz Antony

Head of the Department

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, **Sanaboyina Jaswanth (38110107) and Puthineedi Harish (38110195)** hereby declare that the project entitled “**CLINICAL MANAGEMENT SUPPORT SYSTEM IN THE CLOUD**” done by me under the guidance of **Dr Cruz Antony** at **Sathyabama Institute of Science and Technology** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean**, School of Computing and **Dr. S. VIGNESHWARI., M.E., Ph.D., and Dr. Lakshmanan M.E., Ph.D.,** Heads of the Department, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

-

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Cruz Antony**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Cloud computing is a new way of delivering computing resources by online base. This new technique conveys enormous impact to the society, especially the business world. The services of cloud computing set the clients free from worrying about data processing problems, so that they can focus on their major businesses. This new economic model for computing is increasingly becoming popular and is seeing massive global investment. Despite the surge in activity and interest, there are significant, persistent concerns about cloud computing that are the major barriers for cloud adoption. Data will be generated in various situation in many kinds of format. However, collecting the details from the various place, analysis will give the solution to upgrade their process. These reports can focus on a few areas, such as productivity in the workplace, efficiency levels, how effective a marketing campaign has been, and a number of other essential areas in business. When staff is outdoor service will take up the data from the people directly. If staff is inside the office, will give the appointment to the people such as online service.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF FIGURES	v
	LIST OF SYMBOLS	vii
	LIST OF ABBREVIATIONS	xi
	LIST OF TABLES	xii
1.	CHAPTER 1: INTRODUCTION 1.1 GENERAL 1.2 OBJECTIVE 1.3 EXISTING SYSTEM 1.3.1 EXISTING SYSTEM DISADVANTAGES 1.3.2 LITERATURE SURVEY 1.4 PROPOSED SYSTEM 1.4.1 PROPOSED SYSTEM ADVANTAGES	
2.	CHAPTER 2: PROJECT DESCRIPTION 2.1 GENERAL 2.2 PROBLEM DEFINATION	

	2.3 METHODOLOGIES 2.3.1 MODULES 2.3.2 MODULES EXPLANATION 2.3.3 MODULE DIAGRAM 2.4 TECHNIQUE OR ALGORITHM	
3.	CHAPTER 3 : REQUIREMENTS 3.1 GENERAL 3.2 HARDWARE REQUIREMENTS 3.3 SOFTWARE REQUIREMENTS	
4.	CHAPTER 4 : SYSTEM DESIGN 4.1 GENERAL 4.1.1 USE CASE DIAGRAM 4.1.2 STATE DIAGRAM 4.1.3 CLASS DIAGRAM 4.1.4 SEQUENCE DIAGRAM 4.1.5 COLLABORATION DIAGRAM 4.1.6 DATA FLOW DIAGRAM	

	4.1.7 E-R DIAGRAM 4.1.8 SYSTEM ARCHITECTURE	
5.	CHAPTER 5: SOFTWARE SPECIFICATION 5.1 GENERAL FRONT END 5.2 FEATURES OF JAVA 5.2.1 THE JAVA FRAMEWORK 5.2.2 OBJECTIVE OF JAVA 5.2.3 JAVA SERVER PAGES 5.2.4 EVOLUTION OF WEB APPLICATIONS 5.2.5 BENEFITS OF JSP 5.3 SERVLETS 5.4 JAVA SERVLETS 5.5 CONCLUSION	
6.	CHAPTER 6 : IMPLEMENTATION 6.1 GENERAL 6.2 IMPLEMENTATION	

7.	CHAPTER 7 : SNAPSHOTS 7.1 GENERAL 7.2 VARIOUS SNAPSHOTS	
8.	CHAPTER 8 : SOFTWARE TESTING 8.1. FEASIBILITY STUDY 8.1.1. ECONOMICAL FEASIBILITY 8.1.2. TECHNICAL FEASIBILITY 8.1.3. OPERATIONAL FEASIBILITY 8.2. SYSTEM TESTING 8.2.1. VARIOUS LEVELS OF TESTING 8.2.1.1. WHITE BOX TESTING 8.2.1.2. BLACK BOX TESTING 8.2.1.3. UNIT TESTING 8.2.1.4. FUNCTIONAL TESTING 8.2.1.5. PERFORMANCE TESTING 8.2.1.6. INTEGRATION TESTING 8.2.1.7. VALIDATION TESTING 8.2.1.8. SYSTEM TESTING 8.2.1.9. OUTPUT TESTING 8.2.1.10. USER ACCEPTANCE TESTING	
9.	CHAPTER 9: APPLICATIONS AND FUTURE ENHANCEMENT 9.1 GENERAL 9.2 APPLICATIONS	

	9.3 FUTURE ENHANCEMENTS	
10	CHAPTER 10 : 10.1 CONCLUSION 10.2 REFERENCES	
	GANTT CHART	

LIST OF FIGURES

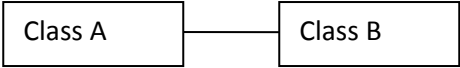
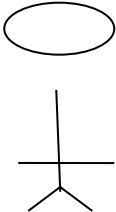
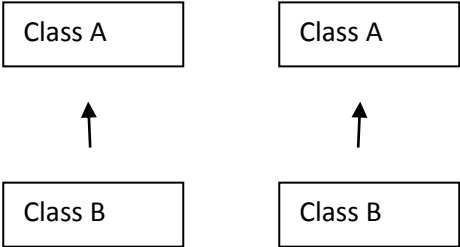
FIGURE NO	NAME OF THE FIGURE	PAGE NO.
2.3.2	Module Diagram	
4.2	Activity Diagram for Anonymous database Management	
4.3	Use case Diagram for Anonymous database Management	
4.4	Data flow diagram for Anonymous database Management	
4.5	Sequence diagram for Anonymous database	

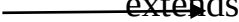
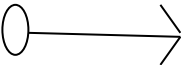
	Management	
4.6	Collaboration diagram for Anonymous database Management.	
4.7	Class diagram for Anonymous database Management	
4.8	Architecture Diagram for Anonymous database Management	
4.9	Object Diagram	
4.10	State Diagram	
4.11	Component Diagram	
4.12	E-R Diagram	
7.1	Medical home page	
7.2	Patient Registration Page	
7.3	Doctor Registration page	
7.4	Patient get Doctor Appointment Page	

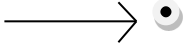
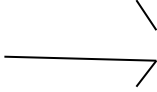
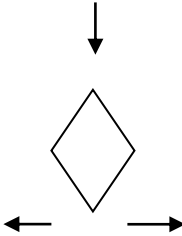
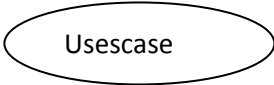
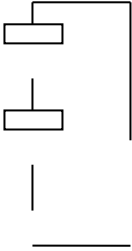
7.5	Doctor Appointment Page	

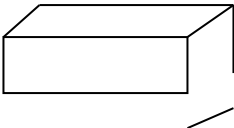
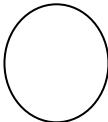
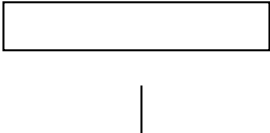
LIST OF SYSMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	<i>+ public</i> <i>-private</i> <i>Class Name</i> <i>-attribute</i> <i>-attribute</i>	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.	<i>Relation</i> (uses)	<i>uses</i>	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 processs.
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		Interact ion 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.	SMC	Secure Multiparty Computation
3.	MDA	Medical Admin
4.	DBC	Data Base Confidentiality
10	JVM	Java Virtual Machine
11.	JSP	Java Server Page

CHAPTER 1:

1.1 INTRODUCTION:

Improving the quality of health care is a national priority in many countries to help reduce the unacceptable levels of variation in health system practices. Despite the potential benefits that can be realized by the deployment of cloud computing, Clinical decision support systems (CDSS) are computer-based programs that data within provide prompts and reminders to assist health care providers in implementing evidence-based clinical guidelines at the point of care. Data security and privacy remain areas of ongoing focus. This Clinical Specialist position will involve training physicians, nutrition and supporting clinical procedures across the region. The client takes the train in doctor and specialist consulting.

1.2 OBJECTIVE:

Various types of clinical Management support systems include specialist support such as weightless and physically, alerts and reminders based on patient management systems that use computer representations of patient care guidelines. By defining a set of goals and objectives for the development of a CDS intervention, a practice can make use of the five rights to determine the what (information), who (recipient), how (intervention), where (format), and when (workflow) for a proposed intervention.

1.3 EXISTING SYSTEM:

There is need to find characteristics of an individual patient are matched to a computerized clinical knowledge base and patient-specific assessments or recommendations are then presented to the clinician for a decision.

1.3.1 EXISTING SYSTEM DISADVANTAGES:

Not robust method for processing the data

1.3.2 LITERATURE SURVEY

TITLE: Optimized Search-and-Compute Circuits and Their Application to Query Evaluation on Encrypted Data

AUTHOR NAME: Jung Hee Cheon, Miran Kim

PAPER YEAR: 28 September 2015

PAPER EXPLANATION:

Private query processing on encrypted databases allows users to obtain data from encrypted databases in such a way that the users' sensitive data will be protected from exposure. Given an encrypted database, users typically submit queries similar to the following examples: 1) How many employees in an organization make over U.S. \$100000? 2) What is the average age of factory workers suffering from leukemia? Answering the questions requires one to search and then compute over the relevant encrypted data sets in sequence. In this paper, we are interested in efficiently processing queries that require both operations to be performed on fully encrypted databases. One immediate solution is to use several special-purpose encryption schemes simultaneously; however, this approach is associated with a high computational cost for maintaining multiple encryption contexts. Another solution is to use a privacy homomorphic scheme. However, no secure solutions have been developed that satisfy the efficiency requirements. In this paper, we construct a unified framework to efficiently and privately process queries with search and compute operations. For this purpose, the first part of our work involves devising several underlying circuits as primitives for queries on encrypted data. Second, we apply two optimization techniques to improve the efficiency of these circuit primitives. One technique involves exploiting single-instruction-multiple-data (SIMD) techniques to accelerate the basic circuit operations.

TITLE: Data Mining: Random Swapping based Data Perturbation Technique for Privacy Preserving in Data Mining

AUTHOR NAME: Ajmeera Kiran Dr. D. Vasumathi

PAPER YEAR: June 2019

PAPER EXPLANATION:

Data mining is a process of collecting unknown data from different data sources and such data are very much useful for various decision-making Processes. Data mining process utilizes such sensitive information for analyzing purpose but, privacy preservation of such sensitive data is very much important in every data mining application. For Example, inpatient Health records some of the sensitive attributes like PID, Age, and Disease Name should not be disclosed to the third party which will lead to privacy violation of the individuals. Hence, a new model should be designed to preserve the privacy of such Sensitive data before it make publicly available. In this paper, an accurate and efficient PPDM (Privacy Preserving Data Mining) technique is implemented in order to preserve the private information about individuals. In the Existing System, traditional Geometric data perturbation (Gaussian Noise Based) technique preserved the individual privacy with some information loss. In the Proposed Paper, an efficient and effective Random Swapping based data perturbation technique is proposed which is mainly focuses on preserving the sensitive attributes and also attaining accurate classification results with minimum Information loss. In Proposed Framework, the accuracy, error rates is compared with a Naïve Bayes classification algorithm and J48 decision tree Algorithm and results are analyzed using Weka 3.8 tool. Proposed Random Swapping Based perturbation technique improved the Accuracy and reduced the error rates with minimum information loss with compared to the existing system.

TITLE: Identity-based leveled fully homomorphic encryption over ideal lattices

AUTHOR NAME: Wang Weili, Hu Bin, Zhao Xiufeng.

PAPER YEAR: 10-12 March 2017

PAPER EXPLANATION:

This paper proposes an efficient identity-based leveled fully homomorphic encryption (IBFHE) scheme over ideal lattice. First, we achieve an identity-based encryption (IBE) scheme in dual-LWE cryptosystem. Then, we take advantage of the eigenvector method to compile the IBE scheme into an IBFHE scheme. Compared with the previous scheme, our scheme is of shorter key size, smaller storage, and higher efficiency, owing to a new trapdoor generation function and algebraic structure of ideal lattice. The security analysis shows that our scheme is selective-identity secure against chosen-plaintext attacks in the standard model.

TITLE: Towards Pattern-Based Change Verification Framework for Cloud-Enabled Healthcare Component-Based

AUTHOR NAME: SADIA AL, MAMOONA HUMAYUN

PAPER YEAR: August 21, 2020

PAPER EXPLANATION: To survive in the competitive environment, most organizations have adopted component-based software development strategies in the rapid technology advancement era and the proper utilization of cloud-based services. To facilitate the continuous configuration, reduce complexity, and faster system delivery for higher user satisfaction in dynamic scenarios. In cloud services, customers select services from web applications dynamically. Healthcare body sensors are commonly used for diagnosis and monitoring patients continuously for their emergency treatment. The healthcare devices are connected with mobile or laptop etc. on cloud environment with network and frequently change applications. Thus, organizations rely on regression testing during changes and implementation to validate the quality and reliability of the system after the alteration. However, for a large application with limited resources and frequently change component management activities in the cloud computing environment, component-based system verification is difficult and challenging due to irrelevant and redundant test cases and faults. In this study, proposed a test case selection and prioritization framework using a design pattern to increase the faults detection rate. First, we select test cases on frequently accessed components using observer patterns and, secondly, prioritize test cases on adopting some strategies. The proposed framework was validated by an experiment and compared with other techniques (previous faults based and random priority). Hence, experimental results show that the proposed framework successfully verified changes. Subsequently, the proposed framework increases the fault detection rate (i.e., more

than 90%) than previous faults based and random priority (i.e., more than 80% respectively).

TITLE: Secured Medical Decision Support System based on SVM

AUTHOR NAME: Akshay Muley, 2 S. A. Kinariwala

PAPER YEAR: January 2018

PAPER EXPLANATION: The Secure Medical Decision Support System (SMDSS), which employs several data mining techniques, is used to help diagnose and treat patients with similar symptoms. The advantages of safe decision support systems include not only privacy preservation but also improves diagnostic accuracy. It also reduces the diagnostic time. To speed up the diagnosis and improve the accuracy of the diagnosis in the current health care system, it is important to make the diagnosis faster and cheaper. This system is called Secure Medical Decision Support System (SMDSS). The data mining techniques used to assist physicians in diagnosing patients with similar symptoms have been very popular in recent years. Past The advantages of medical decision support systems include improved diagnostic accuracy. It also reduces the diagnostic time. In this article, we use the Support Vector Machine (SVM) mining technique, which is used as a classifier that has advantages over traditional mining methods and is a new way of predicting a patient's disease. Because the system was built on the cloud platform, it was necessary to add some features that meet the security requirements. Particularly with the many data related to health care being created daily, the classification can be used to dig up valuable information to improve medical decision support systems. Homomorphic cryptographic techniques are useful for keeping the privacy of the patient in the cloud. Here, using the SVM Classifier and Encryption Techniques, and trying to make the clinical decision support system more useful in providing some important, accurate and effective death information.

1.4 PROPOSED SYSTEM:

A clinical decision support system is intended to improve healthcare delivery by enhancing medical decisions with targeted clinical knowledge, patient information, and other health information patient to specialist.

1.4.1 PROPOSED SYSTEM ADVANTAGES:

It provides robust processing for data processing.

CHAPTER 2

2.1 GENERAL

Thus, there search database used by the doctor is anonymous. Suppose that certain data concerning visitors are related to the use of a drug over a period of four years and certain side effects have been observed and recorded by the doctors in the doctor's database. These data (even if anonymized) need to be kept confidential and accessible only to the few specialists of the institution working on this project, until further evidence is found about the drug. If these anonymous data were to be disclosed, privacy of the visitors would not be at risk; however, the company manufacturing the drug may be adversely affected.

2.2 PROBLEM DEFINITION:

It is very important to maintain efficient software to handle information of a hospital. This application provides **a way** to record this information and to access these in a simple way.

2.3 METHODOLOGIES

Methodologies is the process of analyzing the principles or procedure of a Progressive Anonymous Database management system.

2.3.1 MODULES:

- VISITOR LOGIN
- VISITOR DETAILS
- DOCTOR SUGGEST
- SPECIALIST UPLOAD
- DOCTOR VIEW
- DOWNLOAD

2.3.2 MODULE DISCRIPTION:

VISITOR LOGIN:

This module gives the way to enter into Visitor main page after login with valid input such as username or email id and password. Visitor can enter along with valid text.

VISITOR DEATAILS:

This module gives the way to visitor enter the details in form. What type of Specialist wants?

DOCTOR SUGGEST:

This module doctor checks the visitor details and suggest specialist.

SPECIALIST UPLOAD:

Specialist view the doctor suggest and upload the document.

DOCTOR VIEW:

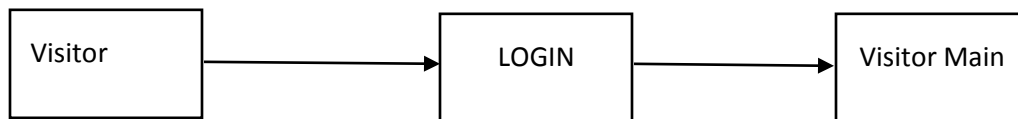
Visitor send the specialist document to doctor view and check the details.

Download:

After doctor once check visitor to download the document.

2.3.3 MODULE DIAGRAM:

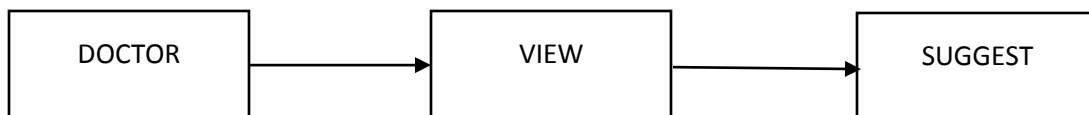
VISITOR LOGIN



VISITOR DEATIL



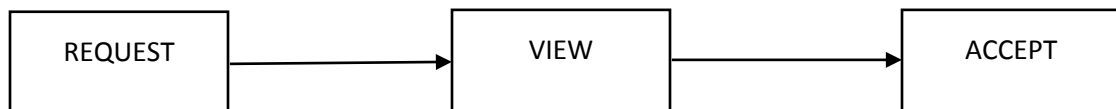
DOCTOR SUGGEST



SPECIALIST UPLOAD:



DOCTOR VIEW:



DOWNLOAD:

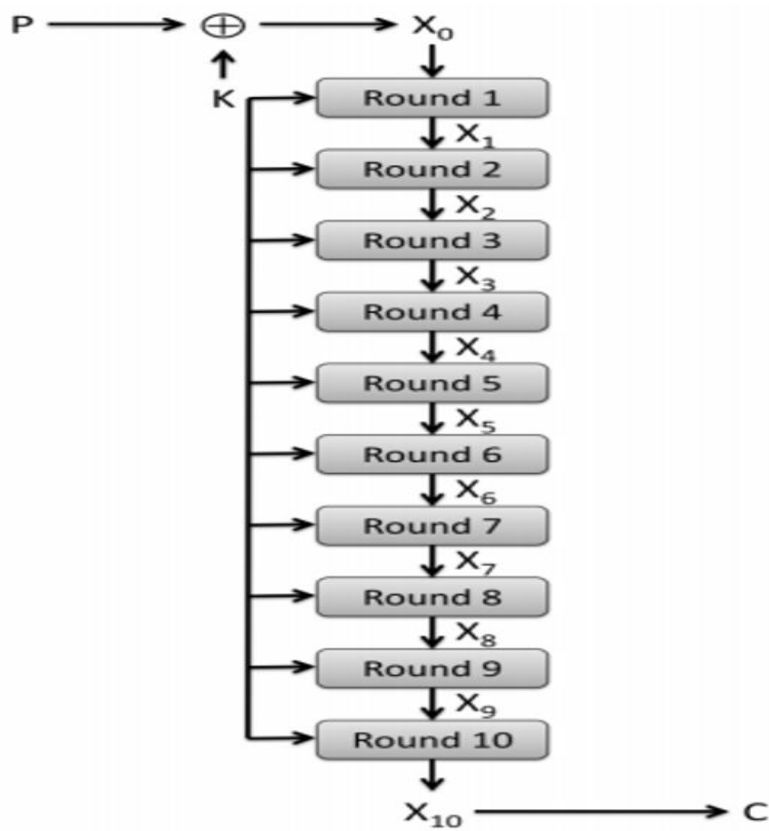


2.4 ALGORITHM USED:

The **AES algorithm** (also known as the **Rijndael algorithm**) is a symmetrical block cipher algorithm that takes plain text in blocks of 128 bits and converts them to cipher text using keys of 128, 192, and 256 bits. Since the AES algorithm is considered secure, it is in the worldwide standard.

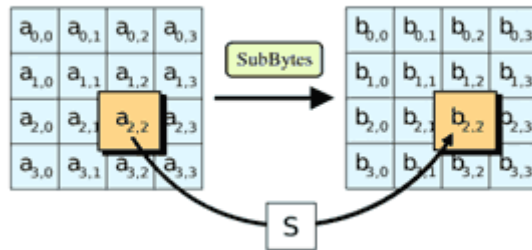
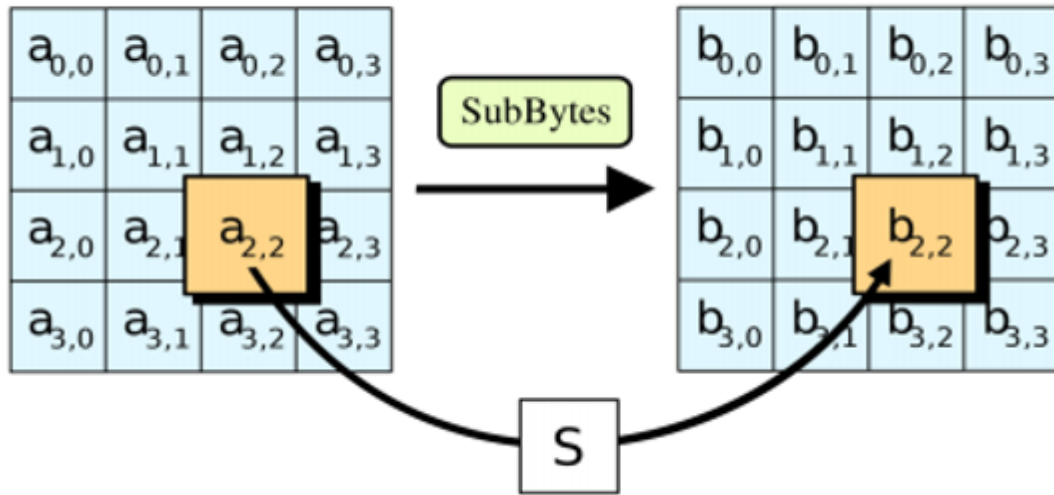
How does AES work?

The AES algorithm uses a substitution-permutation, or SP network, with multiple rounds to produce cipher text. The number of rounds depends on the key size being used. A 128-bit key size dictates ten rounds, a 192-bit key size dictates 12 rounds, and a 256-bit key size has 14 rounds. Each of these rounds requires a round key, but since only one key is inputted into the algorithm, this key needs to be expanded to get keys for each round, including round 0.



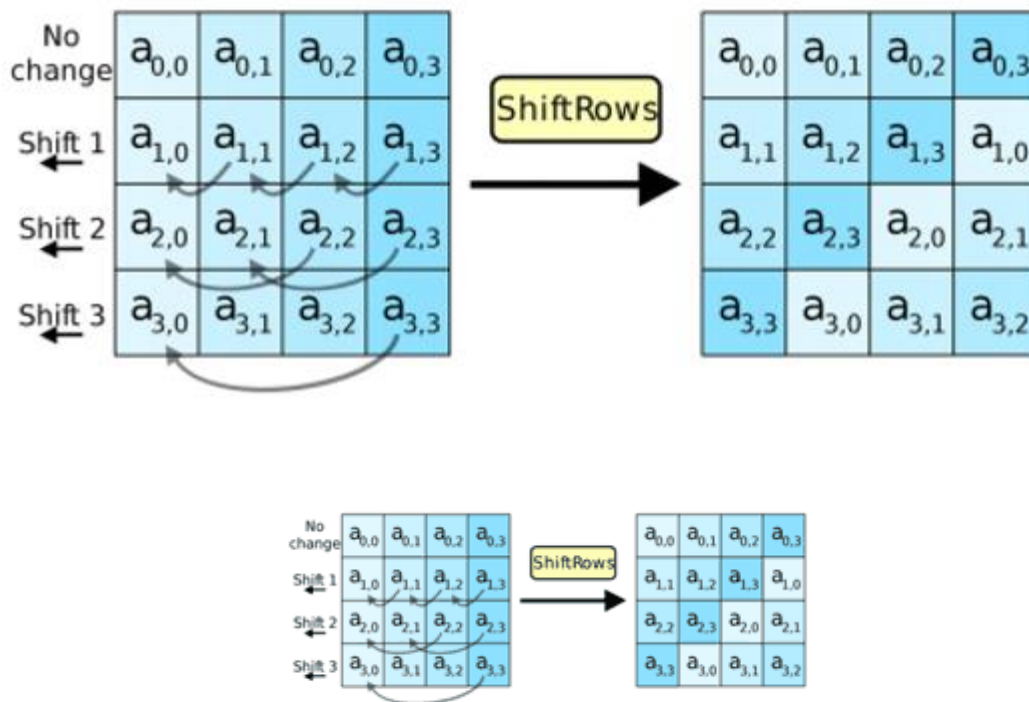
1. Substitution of the bytes

In the first step, the bytes of the block text are substituted based on rules dictated by predefined S-boxes (short for substitution boxes).



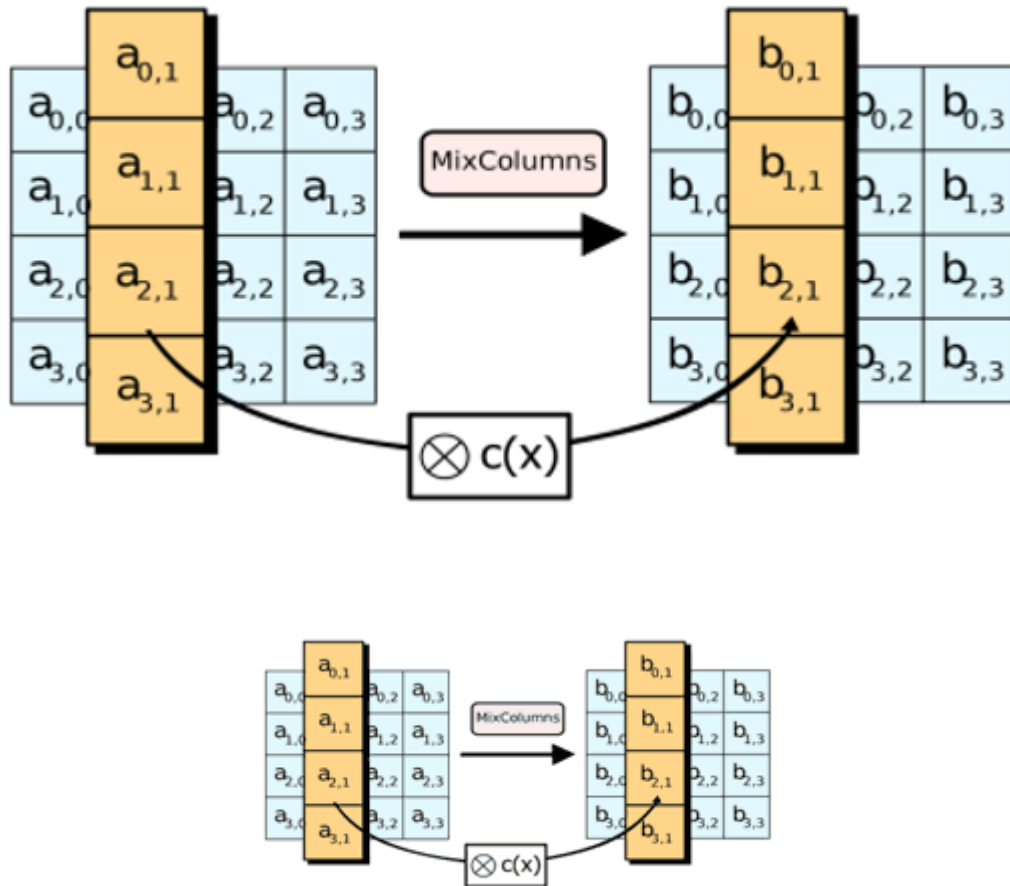
2. Shifting the rows

Next comes the permutation step. In this step, all rows except the first are shifted by one, as shown below.



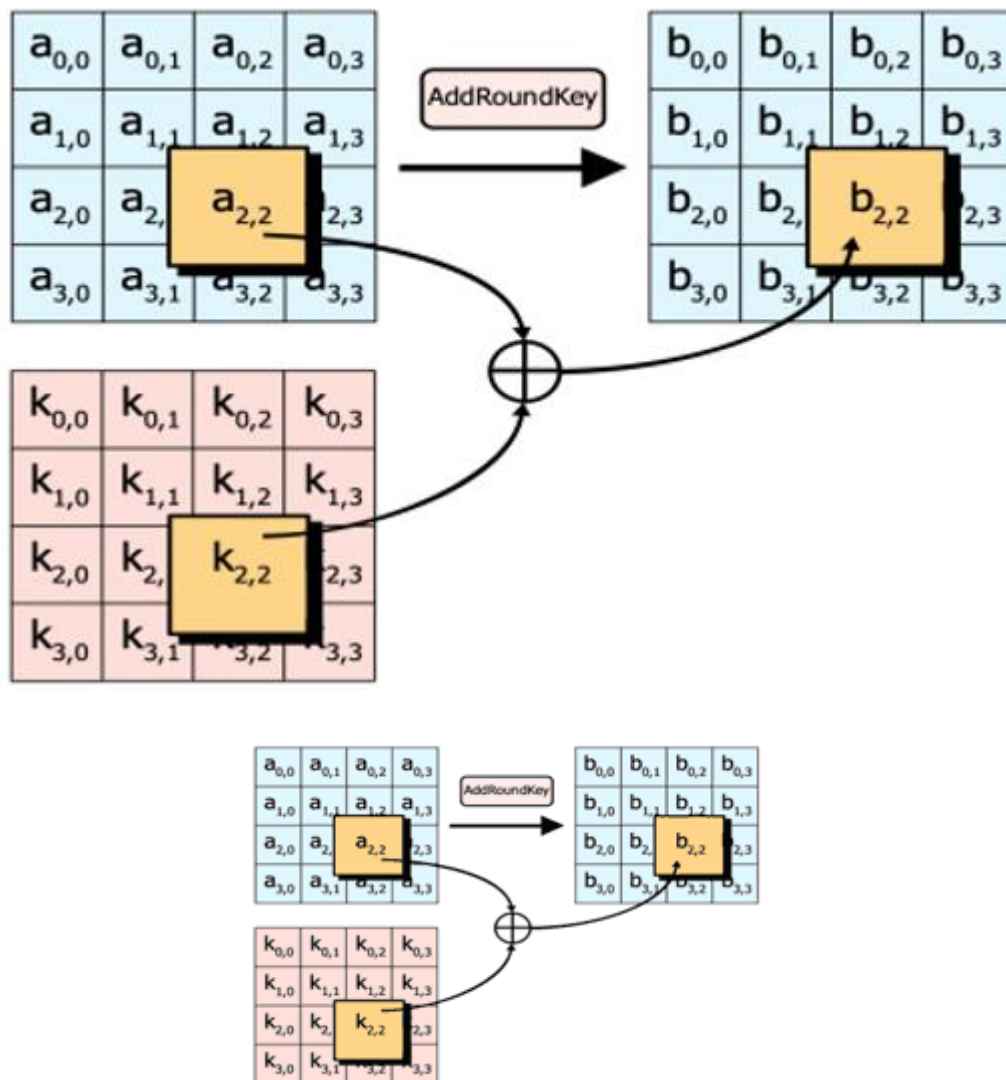
3. Mixing the columns

In the third step, the Hill cipher is used to jumble up the message more by mixing the block's columns.



4. Adding the round key

In the final step, the message is XORed with the respective round key.



When done repeatedly, these steps ensure that the final ciphertext is secure.

CHAPTER 3:

REQUIREMENTS ENGINEERING

3.1 GENERAL:

These are the requirements for doing the project. Without using these tools and software's we can't do the project. So, we have two requirements to do the project. They are

1. Hardware Requirements.
2. Software Requirements.

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.

PROCESSOR	:	PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
RAM	:	4GB DD RAM
MONITOR	:	15" COLOR
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 team's and tracking the team's progress throughout the development activity.

Front End	:	J2EE (JSP, SERVLETS) JAVASCRIPT
Back End	:	MY SQL 5.5
Operating System	:	Windows 07
IDE	:	Eclipse

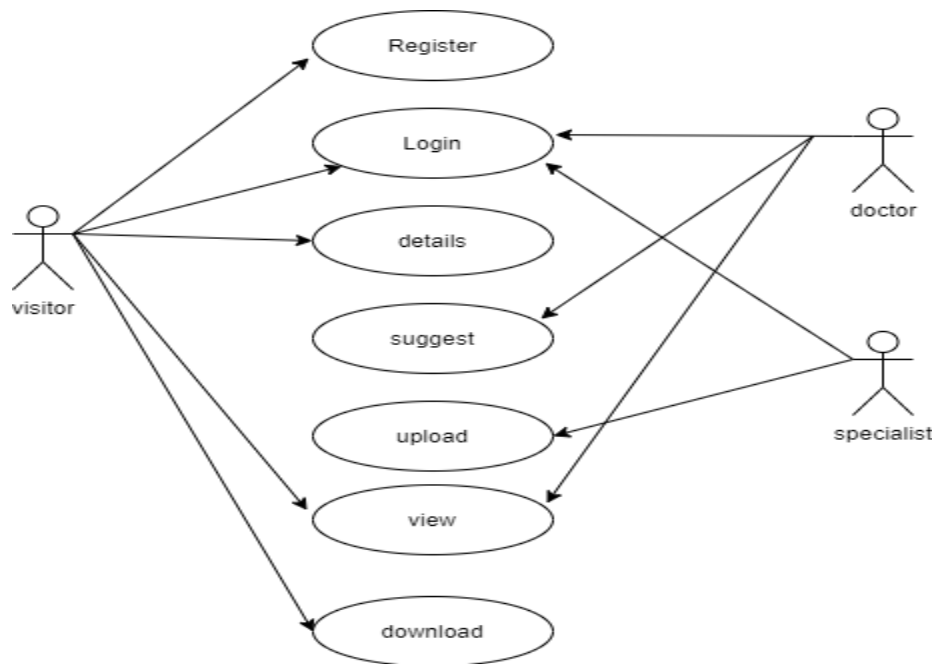
CHAPTER 4

DESIGN ENGINEERING

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.

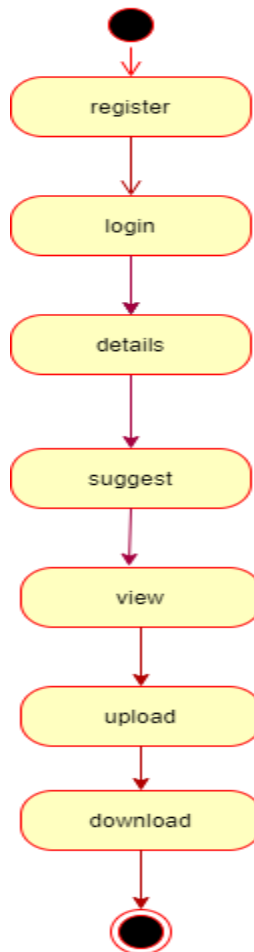
4.1.1 USE CASE:



EXPLANATION:

The use case diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. For this in our component diagram first propose a data In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

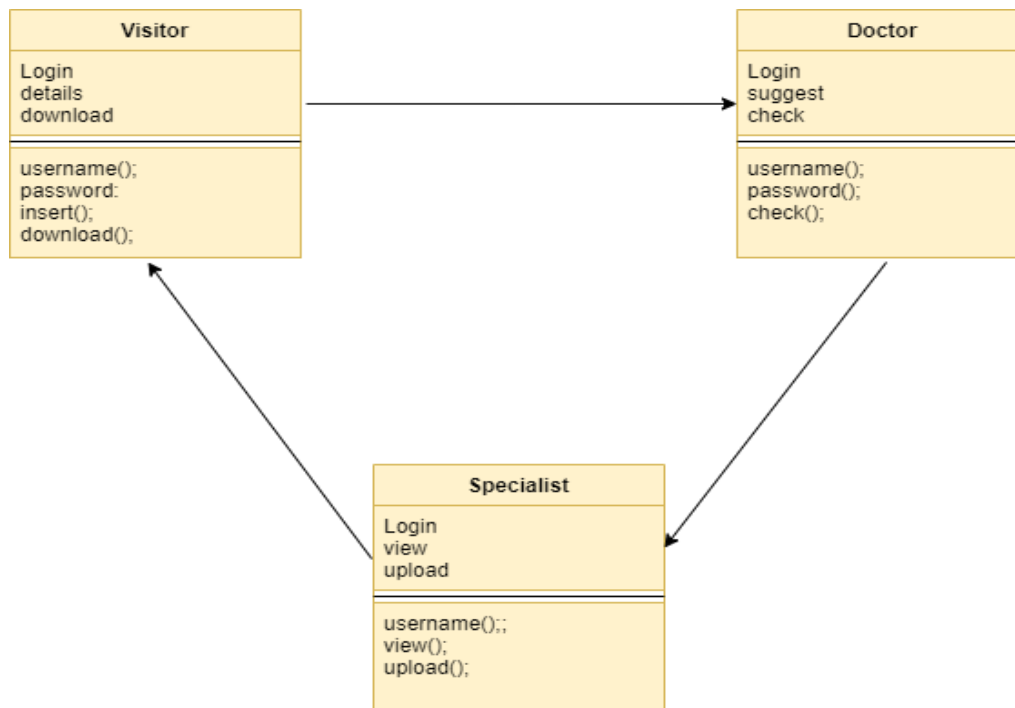
4.1.2 STATE DIAGRAM:



EXPLANATION:

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. In our state diagram first propose. For this in our component diagram first propose a data in this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

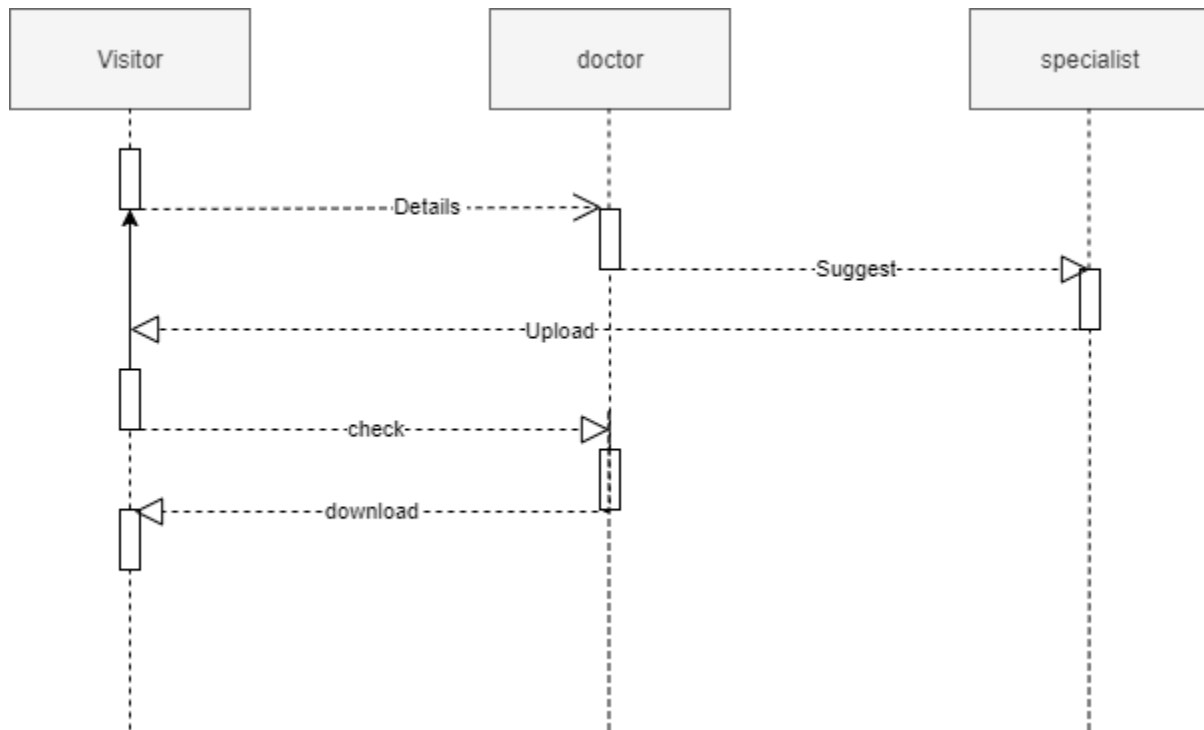
4.1.3 CLASS DIAGRAM:



EXPLANATION:

Class diagram 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. The classes in a class diagram represent both the main objects and or interactions in the application and the objects.

4.1.4 SEQUENCE DIAGRAM:



EXPLANATION:

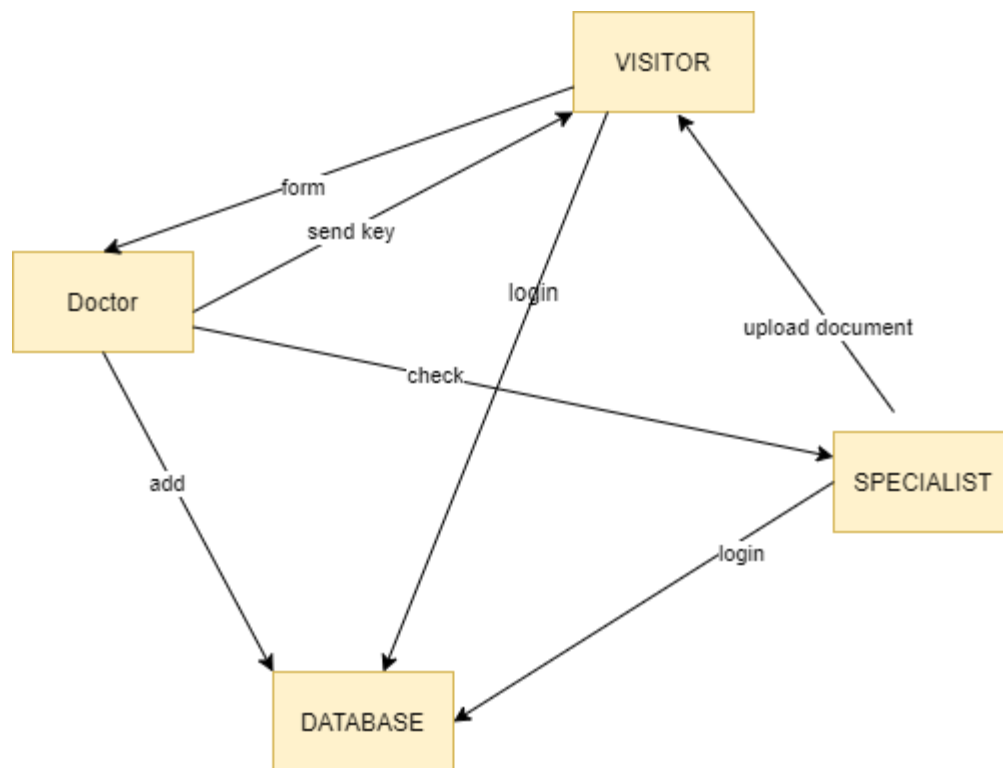
In our sequence diagram specifying processes operate with one another and in order. In our sequence diagram first propose for this in our component diagram first propose a data in this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

4.1.5 COLLABORATION DIAGRAM:

A collaboration diagram shows 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. It can be the decomposition of a use case, use case diagram, or part of a use case diagram.

The collaboration diagram shows messages being sent between classes and object (instances). A diagram is created for each system operation that relates to the current development cycle (iteration).

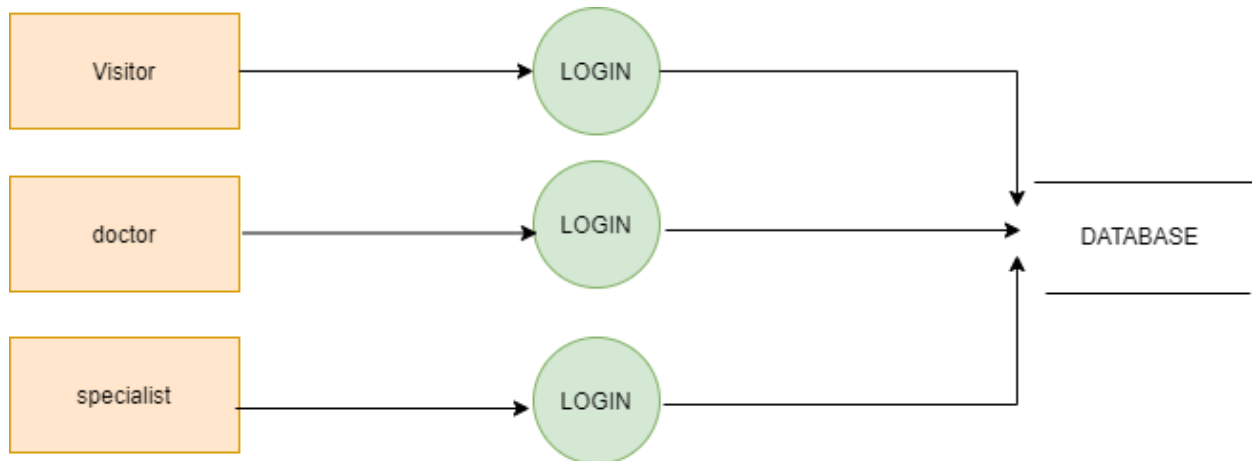


4.1.6 DATAFLOW DIAGRAM:

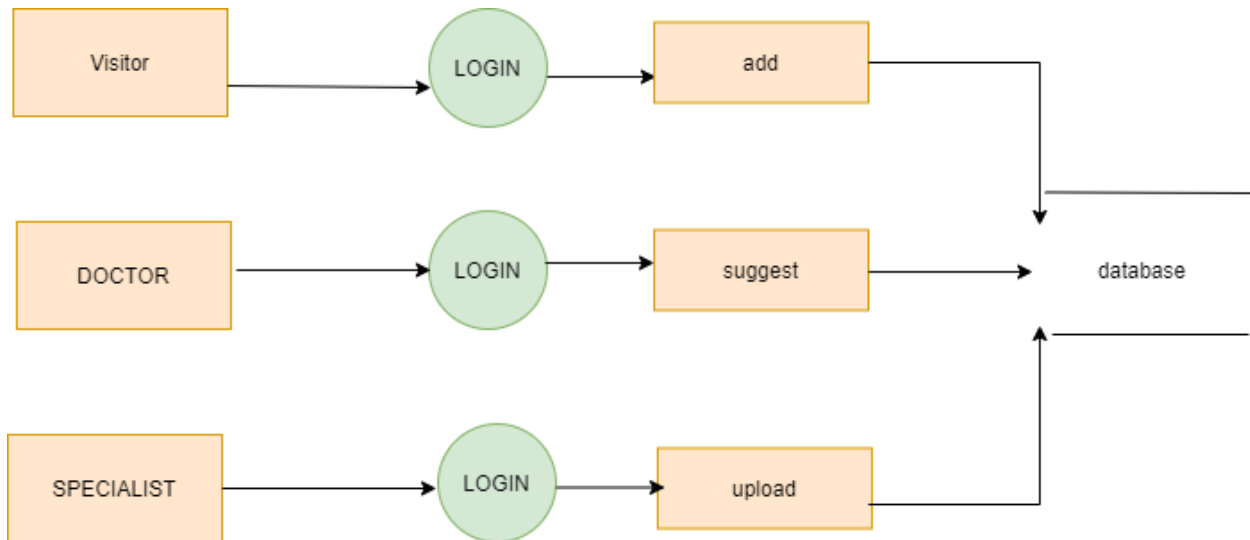
LEVEL 0:



LEVEL 1:



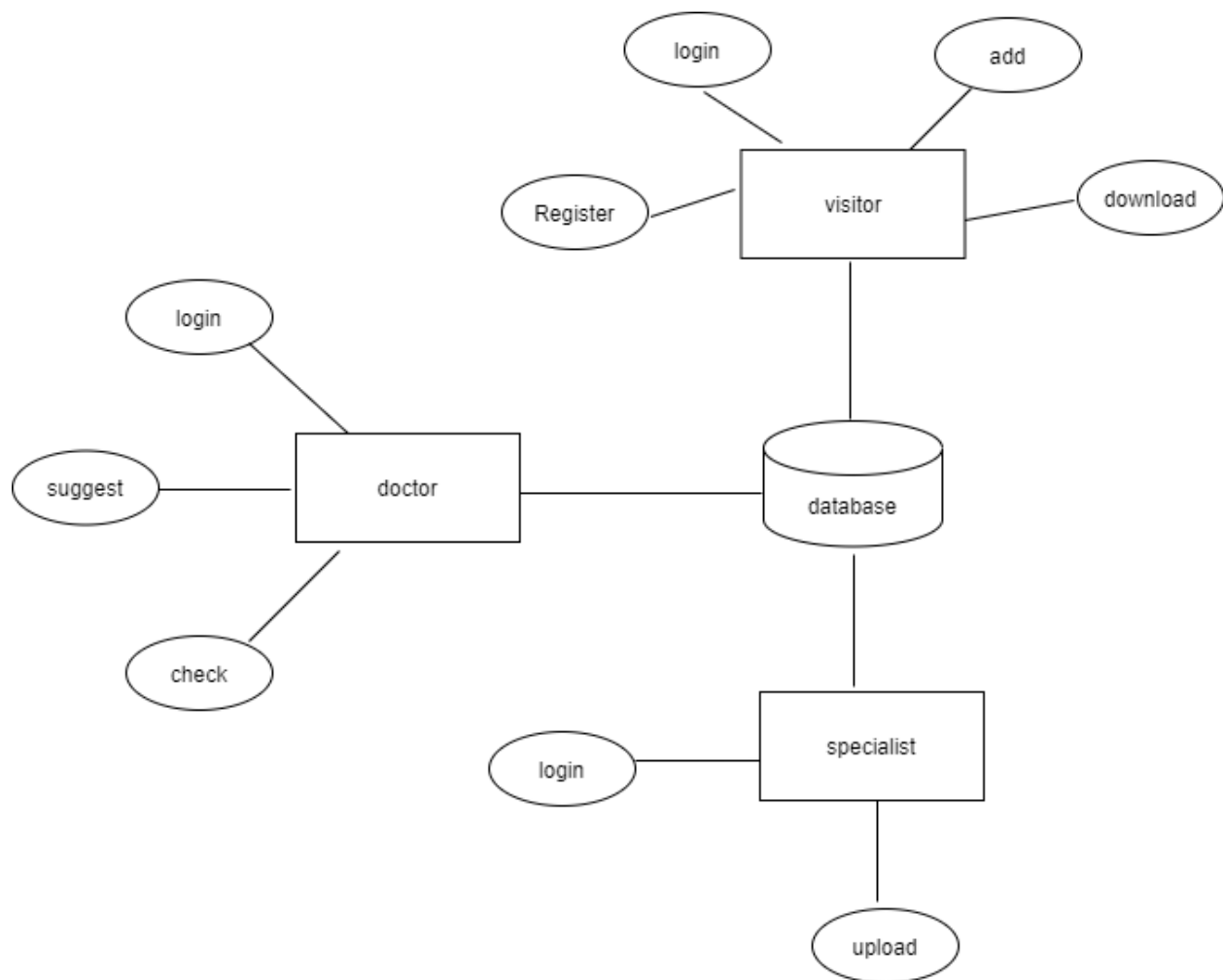
Level 3:



EXPLANATION:

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.

4.1.7 ER-DIAGRAM:

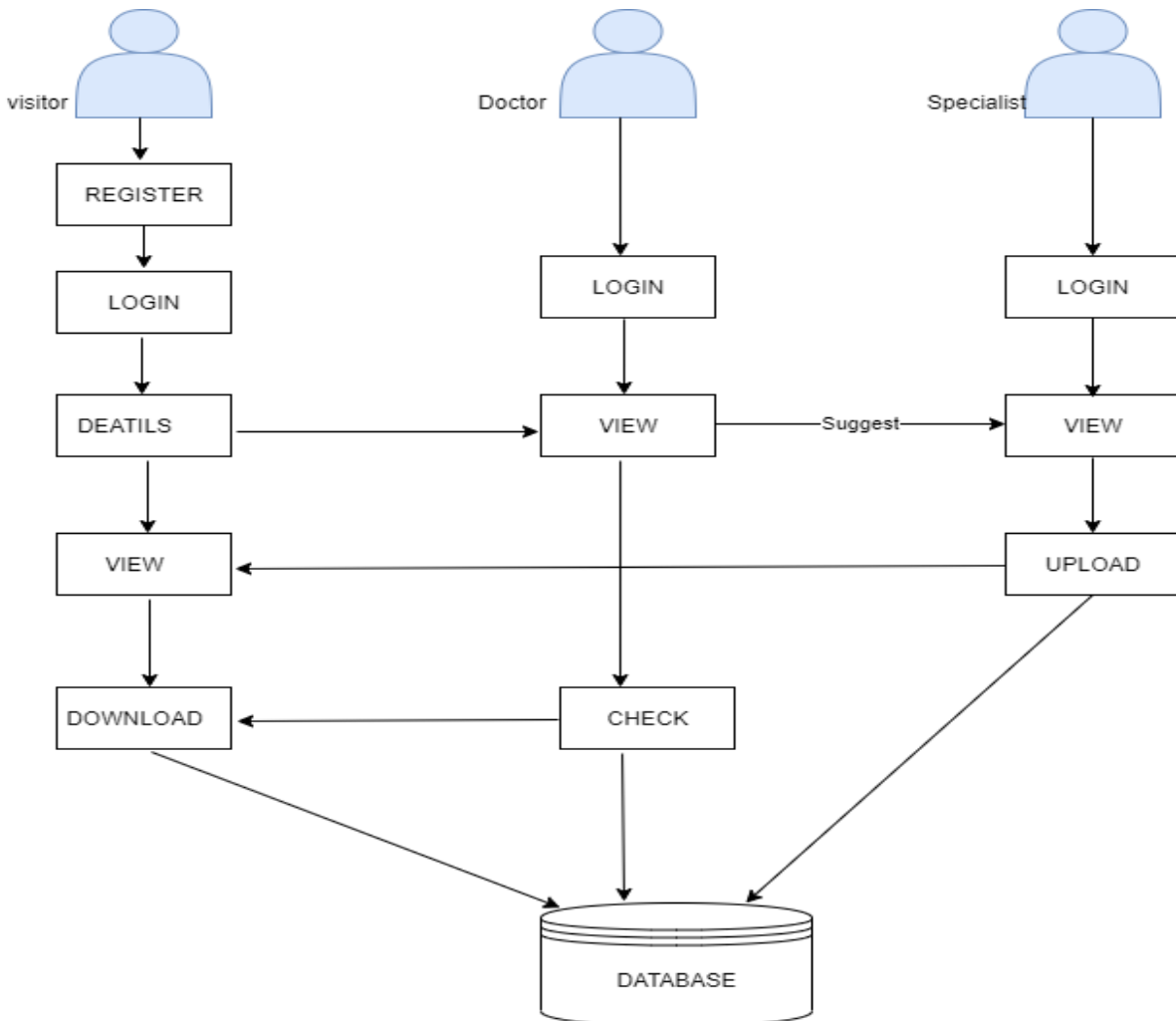


EXPLANATION:

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the

interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

4.1.8 SYSTEM ARCHITECTURE:



EXPLANATION:

The systems architect establishes the basic structure of the system, we propose a Hash code Solomon algorithm and a we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion

stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme

CHAPTER 5

DEVELOPMENT TOOLS

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 Sun 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 byte code 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 additional features as needed.
2. Encapsulation: It is the mechanism of combining the information and providing the abstraction.
3. Polymorphism: As the name suggests 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 provides excellent server-side scripting support for creating database driven web applications. JSP enables the developers to directly insert Java code into JSP files, this makes the development process very simple and its maintenance also becomes very easy.

JSP pages are efficient, it loads into the web server's 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 cc easy to develop database driven web application.

Java is known for its characteristic of "write once, run anywhere." JSP pages are plat 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 servlet 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, and 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 separate the roles of web application/HTML designer from a software developer. The JSP technology is blessed with a few 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 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 Web Logic and IBM Web Sphere.

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 display 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 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 the implementation of searched based application. It deals with the source code for main viewpoint for Anonymous Database Management.

INDEX.JSP.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

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

```
<title>Insert title here</title>
```

```
</head>
```

```
<style>
```

```
body{
```

```
background-image:url("image/c3.gif");
```

```
background-size:1375px 650px;
```

```
}
```

```
ul li{
```

```
display:inline;
```

```
text-decoration: none;
```

```
background:RAL 6027;
```

```
margin-left:10px;
```

```
padding:15px;
```

```
font-size:25px;
```

```
}
```

```
a{
```

```
text-decoration:none;
```

```
font-size:25px;
```

```
font-family:Bernard MT Condensed;
```

```
text-shadow: 2px 2px red;
```

```
}
```

```
</style>
```

```
<body>
```

```
<ul>
```

```
<li>Home</li>
```

```
<li><a href="visitorlog.jsp">visitor</a></li>
```

```
<li><a href="doctorlog.jsp">Doctor</a></li>
```

```
<li><a href="speciallist.jsp">Specialist</a></li>
```

```
</ul>
```

</body>

</html>

visitorlogging.jsp;

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

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

<title>Insert title here</title>

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

<script type="text/javascript" src="js/jquery.min.js"></script>

<script type="text/javascript" src="js/bootstrap.min.js"></script>

</head>

<style>

```
body{  
  
background-image:url("image/c7.jpeg");  
  
background-size:cover;  
  
}
```

```
input[type=submit]:hover {  
  
background-color: #45a049;}
```

```
a{  
  
text-decoration:none;  
  
color:white;  
  
}
```

```
h4{  
  
font-family:Castellar;  
  
color:white;  
  
text-shadow: 1px 2px black;  
  
}
```

```
</style>
```

```
<body>
```

```
<center><br><br><br><br><br><br><br><br>
```

```
<form action="visitorlog" method="post">
```

```
<div class="box">
```

```
<h4> ENTER EMAIL ID:</h4>
```

```
<input type="email" name="email" placeholder="EMAIL ID" required  
style="width:240px;height:30px;border-radius: 5px;text-align:center;">
```

```
<h4>ENTER Password:</h4>
```

```
<input type="password" name="pass" placeholder="PASSWORD" required  
style="width:240px;height:30px;border-radius: 5px;text-align:center;"><br><br>
```

```
<input type="submit" value="Login" style="width:100px;height:30px;border-  
radius: 5px;text-align:center;">
```

```
<a href="visiterreg.jsp"> NEW USER SINGUP</a>
```

```
</div>
```

```
</form>
```

```
</center>
```

```
</body>
```

```
</html>
```

visitorform.jsp;

```
<html lang="en">
```

```
<head>
```

```
<script type="text/javascript"
```

```
src="https://github.com/rubyeffect/easy_fill/tree/master/lib/easy_fill-  
0.0.1.min.js"></script>
```

```
<title>Reach Me</title>
```

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

```
<link rel="stylesheet" href="css/bootstrap.min.css">
```

```
<script type="text/javascript" src="js/jquery.min.js"></script>
```

```
<script type="text/javascript" src="js/bootstrap.min.js"></script>
```

```
</head>
```

```
<style>
```

```
body
```

```
{
```

```
    background-image:url("image/c8.jpg");
```

```
    background-repeat:no-repeat;
```

```
    background-position:left center;
```

```
        background-size: 500px 300px;

    }
```

```
#container{

padding:1px 20px;

}
```

```
#body_header

{
```

```
        width:auto;

        margin:0px auto;

        text-align:center;

        font-size:25px;

    }
```

```
form {

    max-width: 300px;

    margin: 10px auto;

    padding: 10px 20px;
```

```
background: #f4f7f8;  
  
border-radius: 8px;  
  
background-color: transparent;  
  
}
```

```
h1 {  
  
margin: 0 0 30px 0;  
  
text-align: center;  
  
}
```

```
input[type="text"],  
  
input[type="password"],  
  
input[type="date"],  
  
input[type="datetime"],  
  
input[type="email"],  
  
input[type="number"],  
  
input[type="search"],
```

```
input[type="tel"],  
  
input[type="time"],  
  
input[type="url"],  
  
textarea,  
  
select {  
  
    background: rgba(255,255,255,0.1);  
  
    border: none;  
  
    font-size: 16px;  
  
    height: auto;  
  
    margin: auto;  
  
    outline: 0;  
  
    padding: 8px;  
  
    width: 100%;  
  
    background-color: #e8eeef;  
  
    color: black;  
  
    box-shadow: 0 1px 0 rgba(0,0,0,0.03) inset;  
  
    margin-bottom: 30px;
```

```
}
```

```
input[type="radio"],
```

```
input[type="checkbox"]
```

```
{
```

```
margin: 0 4px 8px 0;
```

```
}
```

```
select {
```

```
padding: 6px;
```

```
height: 32px;
```

```
border-radius: 2px;
```

```
}
```

```
button {
```

```
padding: 19px 39px 18px 39px;
```

color: #FFF;

background-color: #4bc970;

font-size: 18px;

text-align: center;

font-style: normal;

border-radius: 5px;

width: 100%;

border: 1px solid #3ac162;

border-width: 1px 1px 3px;

box-shadow: 0 -1px 0 rgba(255,255,255,0.1) inset;

margin-bottom: 10px;

}

fieldset {

margin-bottom: 30px;

border: none;

}

```
legend {  
  
    font-size: 1.4em;  
  
    margin-bottom: 10px;  
  
}
```

```
label {  
  
    display: block;  
  
    margin-bottom: 8px;  
  
}
```

```
label.light {  
  
    font-weight: 300;  
  
    display: inline;  
  
}
```

```
.number {
```

```
background-color: #5fcf80;

color: #fff;

height: 30px;

width: 30px;

display: inline-block;

font-size: 0.8em;

margin-right: 4px;

line-height: 30px;

text-align: center;

text-shadow: 0 1px 0 rgba(255,255,255,0.2);

border-radius: 100%;

}
```

```
form {
```

```
    max-width: 480px;
```

}

</style>

<body>

<% String email=session.getAttribute("vmail").toString(); %>

<div id="container">

<!--This is a division tag for body container-->

<div id="body_header">

<!--This is a division tag for body header-->

<h1> Register Form</h1>

<p>Make your appointments more easier</p>

</div>

<form action="vistorform" method="post">

<fieldset>

<label for="name"> Name</label>

<input type="text" id="name" name="name" placeholder="ENTER NAME"
required pattern="[a-zA-Z0-9]+">

<label for="state">Age</label>

<input type="number" id="skype_name" name="age" placeholder="Age"
pattern="">

<label for="mail">Email</label>

<input type="email" id="mail" value="<%=email %>" name="email"
placeholder="" required>

<label for="state">state</label>

<input type="text" id="" name="state" placeholder="" >

<label for="city">city</label>

<input type="text" id="" name="city" placeholder="">

<label for="appointment_for">Consult Doctor</label>

<select id="appointment_for" name="Doctor" required>

<option value="PSYCHOLOGY">PSYCHOLOGY</option>

<option value="PHYSIOTHERAPY">PHYSIOTHERAPY</option>

<option value="NUTRITIONIST">NUTRITIONIST</option>

</select>

<label for="appointment_description">Appointment Description:</label>

<textarea id="appointment_description" name="description"
placeholder="Description"></textarea>

<label for="date">Date*:</label>

<input type="date" name="date" value="" required></input>

</fieldset>

<button type="submit">SUBMIT</button>

</form>

</div>

</body>

</html>

dbcon.java;

```
package dbcon;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
public class dbcon {
```

```
    static Connection con;
```

```
    public static Connection create()
```

```
    {
```

```
        try
```

```
{  
  
    Class.forName("com.mysql.jdbc.Driver");  
  
    con=DriverManager.getConnection("jdbc:mysql://localhost:3306/specialist",  
"root","root");  
  
    }catch(Exception e)  
  
    {  
  
        e.printStackTrace();  
  
    }  
  
    return con;  
  
    }  
  
}
```

Doctorlogin.java;

```
package servlet;
```

```
import java.io.IOException;
```

```
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.swing.JDialog;

import javax.swing.JOptionPane;


/**
 * Servlet implementation class CloudServer
 */

@WebServlet("/doctorslog")

public class doctorslog extends HttpServlet {

    private static final long serialVersionUID = 1L;


    /**
     * @see HttpServlet#HttpServlet()

```

```
*/
```

```
public doctorslog() {
```

```
    super();
```

```
    // TODO Auto-generated constructor stub
```

```
}
```

```
/**
```

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

```
*/
```

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

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

```
}
```

```
/**
```

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

```
*/
```

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

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

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

```
    System.out.println("Enter your admin Name: "+cloud);
```

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

```
    System.out.println("Enter your password: "+pass);
```

```
    if(cloud.equals("Doctor@gmail.com")&& pass.equals("Doctor"))
```

```
    {
```

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

```
    }
```

```
    else
```

```
    {
```

```
        /*JOptionPane jpane=new JOptionPane("Invalid Log In");
```

```
        JDialog jdialog=jpane.createDialog("Alert");

        jdialog.setAlwaysOnTop(true);*/

        response.sendRedirect("error.jsp");

    }

}
```

CHAPTER 8

SOFTWARE TESTING:

8.1. FEASIBILITY STUDY

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of the existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success.

In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility study should provide a historical background of the business or project, description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

They are 3 types of Feasibility

- Economical feasibility
- Technical feasibility
- Operational feasibility

8.1.1. ECONOMICAL FEASIBILITY

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not.

8.1.2. 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.

8.1.3. OPERATIONAL 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.

8.2. SYSTEM TESTING

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases makes it shine.

8.2.1. VARIOUS LEVELS OF TESTING

1. White Box Testing
2. Black Box Testing
3. Unit Testing
4. Functional Testing
5. Performance Testing
6. Integration Testing
7. Validation Testing
8. System Testing
9. Output Testing
10. User Acceptance Testing

8.2.1.1. WHITE BOX TESTING

White-box testing, sometimes called glass-box, is a test case design method that uses the control structure of the procedural design to derive test cases. Using White Box testing methods, we can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.

8.2.1.2. 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. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

8.2.1.3. UNIT TESTING

Unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Unit testing is software verification and validation method in which the individual units of source code are tested fit for use. A unit is the smallest testable part of an application. In this testing, each class is tested to be working satisfactorily.

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. 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.

8.2.1.4. FUNCTIONAL TESTING

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does. Functional testing differs from system testing in that functional testing "verifies a program by checking it against ... design document(s) or specification(s)", while system testing "validate a program by checking it against the published user or system requirements" (Kane, Falk, Nguyen 1999, p. 52). Functional testing typically involves five steps. The identification of functions that the software is expected to perform

1. The creation of input data based on the function's specifications
2. The determination of output based on the function's specifications
3. The execution of the test case
4. The comparison of actual and expected outputs.

8.2.1.5. PERFORMANCE TESTING

In general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate, measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

Performance testing is a subset of performance engineering, an emerging computer science practice which strives to build performance into the implementation, design and architecture of a system.

8.2.1.6. INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when put together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready. All the errors found in the system are corrected for the next phase.

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly for example across procedure calls or process activations, and this is done after testing individual modules, i.e. unit testing.

8.2.1.7. VALIDATION TESTING

Verification and Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it full fills its intended purpose. These are critical components of a quality management system such as ISO 9000. The words "verification" and "validation" are sometimes preceded with "Independent" (or IV&V), indicating that the verification and validation is to be performed by a disinterested third party.

It is sometimes said that validation can be expressed by the query "Are you building the right thing?" and verification by "Are you building it right?". In practice, the usage of these terms varies. Sometimes they are even used interchangeably.

8.2.1.8. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

8.2.1.9. OUTPUT TESTING

After performing the validation testing, next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in to two ways. One is on screen, and another is printed format. The output comes as the specified requirements by the user. Hence output testing does not result in any correction in the system.

8.2.1.10. USER ACCEPTANCE TESTING

User acceptance of a system is the factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

- Input screen design.
- Output screen design.
- Online message to guide user.
- Format of the ad-hoc reports and other outputs.

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using the test data. While testing the system by using test data errors are again uncovered and correct.

CHAPTER 9

APPLICATION

9.1 GENERAL:

In this paper we propose a secured Anonymous database management system. We have presented two secure protocols for privately checking whether a k-anonymous database retain its anonymity once a new tuple is being inserted to it. Since the proposed protocols ensure the updated database remain sk-anonymous, the results returned from a user's (or a doctors) query are also k-anonymous. Thus, the visitors or the data provider's privacy cannot be violated from any query.

9.2 APPLICATION

Anonymous database Management system is used to maintain the visitor's database in the anonymous format. It is also very easy to retrieve the accurate data from an anonymous database, here all the information about the visitors are maintained securely and also here we achieve the confidentiality for the data's stored in the database. Concerning the actual execution of the database update, once the system has verified that the user's tuple can be safely inserted to the database without compromising k-anonymity, the user is required to send to the

Private Updater the non-anonymous attributes' values to be stored in the k-anonymous database as well.

9.3 FUTURE ENHANCEMENTS:

1. Implementing a real-world anonymous database system.
2. Improving the efficiency of protocols, in terms of number of messages exchanged and in terms of their sizes, as well.
3. Implement using two or more algorithms.

CONCLUSION:

We have proposed in this system clinical decision support user to specialist. In these technologies and an increase in our graying population. Ensuring the privacy of patient data in such systems is of paramount importance. Technologies, and practices are rapidly moving into the home. The factors driving this migration include the costs of health care, the growing numbers of older adults, the increasing prevalence of chronic conditions and diseases and improved survival rates for people with those conditions and diseases, and a wide range of technological innovations. The clinical specialist care that results vary considerably in its safety, effectiveness, and efficiency, as well as in its quality and cost.

REFERENCE OR BIBLIOGRAPHY:

- [1] G. Bieber, M. Haescher, and M. Vahl, “Sensor requirements for activity recognition on smart watches,” in Proc. 6th Int. Conf. Pervasive Technol. Related Assistive Environ., 2013, Art. no. 67.

- [2] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, “An evaluation of multi-resolution storage for sensor networks,” in Proc. 1st Int. Conf. Embedded Netw. Sens. Syst., 2003, pp. 89–102.

- [3] Cisco, “Cisco global cloud index: Forecast and methodology, 20142019 white paper.” [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/globalcloud-index-gci/Cloud_Index_White_Paper.html.

- [4] X. Liu, K. R. Choo, R. H. Deng, R. Lu, and J. Weng, “Efficient and privacy-preserving outsourced calculation of rational numbers,” IEEE Trans. Depend. Secure Comput., 2016, doi: 10.1109/ TDSC.2016.2536601. TABLE 3 A Comparative Summary Function/Algorithm

- [5] J. H. Cheon, M. Kim, and M. Kim, “Optimized search-and-compute circuits and their application to query evaluation on encrypted data,” IEEE Trans. Inf. Forensics Secur., vol. 11, no. 1, pp. 188–199, Jan. 2016.

- [6] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-preserving patient-centric clinical decision support system on naïve Bayesian classification," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 2, pp. 655–668, Mar. 2016
- [7] X. Liu, R. H. Deng, K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 11, pp. 2401–2414, Nov. 2016.
- [8] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Des. Codes Cryptography*, vol. 71, no. 1, pp. 57–81, 2014.
- [9] H. R. Warner, A. F. Toronto, L. G. Veasey, and R. Stephenson, "A mathematical approach to medical diagnosis: Application to congenital heart disease," *J. Amer. Med. Assoc.*, vol. 177, no. 3, pp. 177–183, 1961.
- [10] C. Schurink, P. Lucas, I. Hoepelman, and M. Bonten, "Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care units," *Lancet Infectious Diseases*, vol. 5, no. 5, pp. 305–312, 2005