A Mini Project Report

On

**VOICE BASED ENCRYPTION**

*Submitted to JNTU HYDERABAD*

*In Partial Fulfillment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted

By

**Thani Chandu (198R1A05N1)**

Under the Esteemed guidance of

**Mr.Y. Shyam Sundar**

Assistant Professor, Department of CSE



**Department of Computer Science & Engineering**

**CMR ENGINEERING COLLEGE**

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad) Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

**2022-2023**

**CMR ENGINEERING COLLEGE**

*(Accredited by NBA,Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad) Kandlakoya, Medchal Road, Hyderabad-501 401*

**Department of Computer Science & Engineering**



**CERTIFICATE**

This is to certify that the project entitled **“VOICE BASED ENCRYPTION”** is a bonafide work carried out by

**Thani Chandu (198R1A05N1)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF**

**TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from CMR

Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

|  |  |  |  |
| --- | --- | --- | --- |
| Internal Guide | Mini Project Coordinator | Head of the Department | External Examiner |
| **Mr.Y. Shyam sundar** | **Mr. S. Kiran Kumar** | **Dr. Sheo Kumar** |  |
| Assistant Professor | Assistant Professor | Professor & H.O.D |  |
| CSE, CMREC | CSE, CMREC | CSE, CMREC |  |

**DECLARATION**

This is to certify that the work reported in the present project entitled "**VOICE BASED**

**ENCRYPTION”** is a record of bonafide work done by us in the Department of Computer Science and Engineering, CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source.We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or

Institute for the award of any degree or diploma to the best of our knowledge and belief

**Thani Chandu (198R1A05N1)**

##### ACKNOWLEDGEMENT

I am extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr.Sheo Kumar**, HOD, **Department of CSE, CMR Engineering College** for their constant support**.**

I am extremely thankful to **Mr.Y. Shyam Sundar,** Assistant Professor, Internal Guide, Department of CSE, for his/ her constant guidance, encouragement and moral support throughout the project.

I will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

I thank **S Kiran Kumar** Mini Project Coordinator for his constant support in carrying out the project activities and reviews.

I express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, I am very much thankful to my parents who guided me for every step.

**Thani Chandu (198R1A05N1)**

**VOICE BASED ENCRYPTION**

**CONTENTS PAGE NO**

**ABSTRACT**

**LIST OF FIGURES**

1. **INTRODUCTION 1**

Introduction to project

Existing system

Proposed system\

1. **LITERATURE SURVEY 3**
2. **ANALYSIS 4**

Functional Requirements

Non-Functional Requirements

1. **SOFTWARE & HARDWARE REQUIREMENTS 6**

Software requirements

Hardware requirements

Modules

1. **STUDY OF THE SYSTEM 7**

Feasibility study

Technical feasibility

Operational feasibility

Economic feasibility

1. **ARCHITECTURES 8**

Technical architecture

System architecture

1. **SOFTWARE REQUIREMENT SPECIFICATION 10**

Purpose

Life cycle models and methodologies

Advantages

Disadvantages

1. **SOFTWARE DEVELOPMENT ENVIRONMENT 12**

About Servlet, HTML, eclipse, tomcat, my SQL

database and JDBC drivers

1. **SYSTEM DESIGN 15**

Software design

Input/output design

1. **SEQUENCE DIAGRAM 23**

Use case diagram

Class diagram ER diagram

1. **SOFTWARE TESTING 24**

Testing strategies

1. **CODING AND IMPLEMENTATION 34**

Sample code

Data Dictionary

1. **OUTPUT SCREEN 47**

1. **CONCLUSION 50**

1. **FUTURE ENHANCEMENTS 51**

1. **BIBLIOGRAPHY AND REFERENCES 52**

**ABSTRACT**

In several distributed systems a user should only be able to access data if a user posses a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper we present a system for realizing complex access control on encrypted data that we call Cipher text-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential even if the storage server is un trusted; moreover, our methods are secure against collusion attacks. Previous Attribute Based Encryption systems used attributes to describe the encrypted data and built policies into user’s keys; while in our system attributes are used to describe a user’s credentials, and a party encrypting data determines a policy for who can decrypt. Thus, our methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). In addition, we provide an implementation of our system and give performance measurements. Our methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). In addition, we provide an implementation of our system and give performance measurements.

## LIST OF FIGURES

**S.NO DESCRIPTION PAGE NO**

Fig.1.1 Technical architecture 6

Fig.1.2 System architecture 7

Fig 2 Waterfall model 9

Fig 3 JDBC Drivers 11

Fig 9.1 Data flow diagram 13

Fig 9.2 Sequence diagram 14

Fig 9.3 Use case diagram 15

Fig 9.4 Class diagram 16

Fig 9.5 ER diagram 17

Fig 9.6 Component diagram 18

Fig 12.01 Output screen 1 33

Fig.12.02 Output screen 2 34

Fig.12.03 Output screen 3 35

Fig.12.04 Output screen 4 36

Fig.12.05 Output screen 5 37

Fig.12.06 Output screen 6 38

Fig.12.07 Output screen 7 39

Fig.12.08 Output screen 8 40

Fig.12.09 Output screen 9 41

Fig.12.10 Output screen 10 42

ii

#### 1.INTRODUCTION

**1.1. INTRODUCTION TO PROJECT:**

The purpose of voice communication is to realize the transmission of information in human daily life, and people’s requirements for voice calls are increasing continuously and developing rapidly. Voice communication changes people’s communication methods; because of its immediacy, convenience and other characteristics, it is widely used, and it is the main method in modern communication systems. With the development of communication technology, the security of traditional voice communication has been greatly challenged, and a series of voice signal eavesdropping cases have occurred. Party and government organizations in various countries attach great importance to the confidentiality of voice communications. Enterprises hope that voice communications will not compromise their confidentiality, and individuals hope to protect their privacy. Encrypting the important speech at the sending end and sending the encrypted signal may prevent unauthenticated access to the signal. Strength of the algorithm determines the security of the signal. Cryptographic algorithms can be classified into four major categories: frequency domain scrambling, time domain scrambling, amplitude domain scrambling and combinational domain scrambling. Analogue algorithms can also be used for enhancing secure voice signal transmission. Transformations such as Wavelet transform, Fast Fourier transform, discrete cosine transform, discrete wavelet transform are used by analogue algorithms to ensure security. Digital encryption is more secure than analog but it needs complex implementations and large bandwidth for signal transmission.

**1.2. EXISTING SYSTEM:**

Traditionally, this type of expressive access control is enforced by employing a trusted server to store data locally. The server is entrusted as a reference monitor that checks that a user presents proper certification before allowing him to access records or files. However, services are increasingly storing data in a distributed fashion across many servers. Confidentiality of secret information has to be maintained. Speech signals that carry very sensitive information need to be secured from unauthorized acces

**1.3. THE PROPOSED SYSTEM**:

In this work, we provide the first construction of a cipher text-policy attribute-based encryption (CP-ABE) to address this problem, and give the first construction of such a scheme.

In our system, a user’s private key will be associated with an arbitrary number of attributes expressed as strings. On the other hand, when a party encrypts a message in our system, they specify an associated access structure over attributes. Decryption is the counter process for encryption in which the actual data is retrieved from cipher data. There are various encryption and decryption algorithms are devised in course of time for preventing unauthorized access and maintaining the confidentiality of the information. Key space analysis plays a vital role in performance analysis of cryptographic system. A good cryptographic algorithm should ensure a large key space, and should be sensitive to the key value.

#### 2. LITERATURE SURVEY

Hous Jianhua Hao et.al , described the algorithm based on scrambling technique in frequency domain. In which the encryption algorithm specified, scrambles the sample speech signal in the limited frequency domain range to guarantee that the encrypted speech signal bandwidth does not be extended. Encryption based on chaotic maps is well-defined by E. Mosa et al in 2009[3]. It has been given that the technique used will provide extra security level to the encryption system without much increase in algorithm complexity. It is achieved by using different secret block size and changing secret key every time. The user is free to choose the required block size and the sub- keys that form the secret key. This way of generating the secret key ensures security. It is achieved by using different secret block size and changing secret key every time. The user is free to choose the required block size and the sub- keys that form the secret key. This way of generating the secret key ensures security. Also a brief description about logistic maps, henon map and baker encryption are specified and the effectiveness of the above mentioned are compared. Jianhua Hao et.al , described the algorithm based on scrambling technique in frequency domain. In which the encryption algorithm specified, scrambles the sample speech signal in the limited frequency domain range to guarantee thatthe encrypted speech signal bandwidth does not be extended. Encryption based on chaotic maps is well-defined by E. Mosa et al in 2009.

#### 3.ANALYSIS

The minimum requirements of the project are listed below:

* Examine the tools and methodologies required to gain an overview of the system requirements for the proposed database.
* Examine suitable database management systems that can be used to implement the proposed database.
* Evaluate appropriate website authoring and web graphic creation tools that can be used to develop web-based forms for the proposed database
* Produce and apply suitable criteria for evaluating the solution

**3.1. FUNCTIONAL REQUIREMENTS:**

Functional requirement should include function performed by a specific screen outline workflows performed by the system and other business or compliance requirement the system must meet. The voice based encryption functional and design guidelines apply to device makers planning to implement the general voice based encryption APIs and SDKs. If your voice based encryption implementation is more specialized.

### 3.2. NON-FUNCTIONAL REQUIREMENTS:

Non-Functional requirements include quantitative constraints, such as response time (i.e.

how fast the system reacts to user commands.) or accuracy (.e. how precise are the systems numerical answers.). **Non-functional requirements or NFRs** are a set of specifications that describe the system’s operation capabilities and constraints and attempt to improve its functionality. These are basically the requirements that outline how well it will operate including things like speed, security, reliability, [data integrity,](https://www.altexsoft.com/blog/data-integrity/) etc.

**Administrative user interface:** The ‘administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection.

**The operational or generic user interface:** The ‘operational or generic user interface’ helps the end users of the system in transactions through the existing data and required services.

#### 4. SOFTWARE & HARDWARE REQUIREMENTS

**4.1. SOFTWARE REQUIREMENTS:**

 **Operating System Server :** Windows XP or later

|  |  |  |
| --- | --- | --- |
|  | **Database Server** | **:** Oracle or MYSQL |
|  | **Client** | **:** Microsoft Internet Explorer |
|  | **Tools and environment** | **:** Apache Tomcat 1.6, jdk1.6 |
|  | **User Interface** | **:** Html , Jsp |
|  | **Code Behind** | **:** Java, Jdbc, SQL. |

**4.2. HARDWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
|  | Processor | : Intel Pentium IV or later |
|  | Ram | : Minimum 512 MB |
|  | HDD | : 80 GB |

**4.3 MODULES**

In this implementation we have

1. Cloud Storage Module
2. Data Owner Module
3. Data User Module
4. Authority Module

**MODULE** **DESCRIPTION:**

**Cloud Storage:**

Cloud storage is a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a [hosting](http://en.wikipedia.org/wiki/Internet_hosting_service) company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running..

**Data Owner:**

The data owner encrypts his message under access policy, then computes the complement circuit, which outputs the opposite bit of the output of f, and encrypts a random element R of the same length to under the policy

**Data User:**

The users can outsource their complex access control policy decision and part process of decryption to the cloud. Such extended encryption ensures that the users can obtain either the message M or the random element R, which avoids the scenario when the cloud server deceives the users that they are not satisfied to the access policy, however, they meet the access policy actually.

**Authority:**

Authority generates private keys for the data owner and user.

#### 5. STUDY OF THE SYSTEM

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI’S at the top level have been categorized as:

**5.1. FEASIBILITY STUDY:**

The next step in analysis is to verify the feasibility of the proposed system. “All projects are feasible given unlimited resources and infinite time. Feasibility has applied to Maintenance of Elementary School Data pertains to the following areas:

* Technical feasibility
* Operational feasibility
* Economic feasibility

* 1. **TECHNICAL FEASIBILITY:**

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system.

* 1. **OPERATIONAL FEASIBILITY:**

To determine the operational feasibility of the system we should take into consideration the awareness level of the users.

* 1. **ECONOMIC FEASIBILITY:**

To decide whether a project is economically feasible, we must consider various factors

as:

* Cost benefit analysis
* Long-term returns
* Maintenance costs

The proposed Maintenance of Elementary School Data is computer based. It requires average computing capabilities and access to internet, which are very basic requirements and can be afforded by any organization hence it doesn’t incur additional economic overheads, which renders the system economically feasible.

### 6. ARCHITECTURE

**6.1. TECHNICAL ARCHITECTURE**:

Technical Architecture (TA) is a form of IT architecture that is used to

design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system- relevant requirements are met.

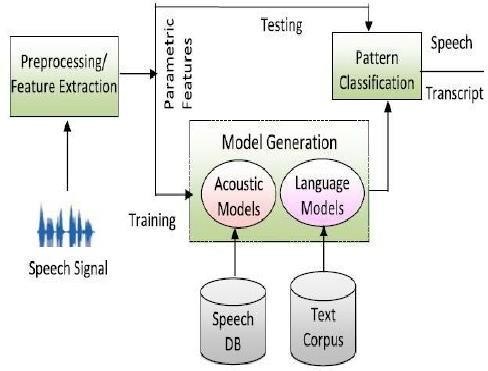
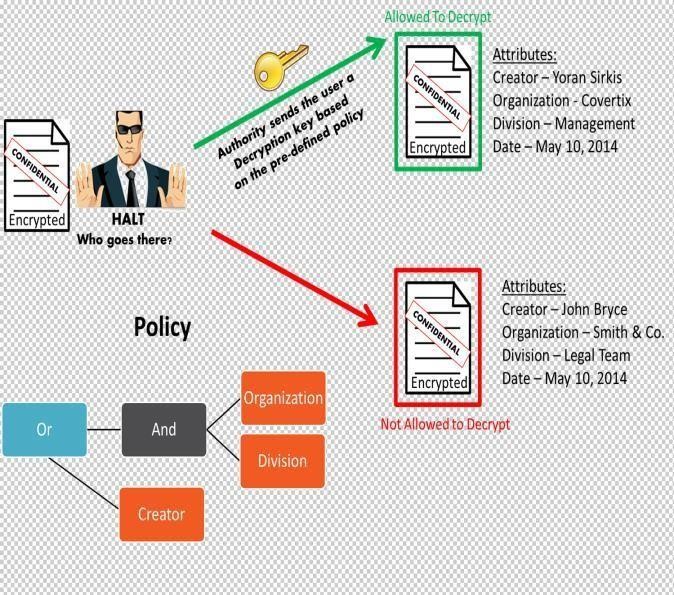


Fig.6.1: Technical architecture

**6.2. SYSTEM ARCHITECTURE:**

System Architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system. It also focuses on high-level structure in systems and system elements. It addresses the architectural principles, concepts, properties, and characteristics of the system-of-interest.



**Fig.1.2**

Fig.6.2: System architecture

#### 7. SOFTWARE REQUIREMENT SPECIFICATION

Software Requirements Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for [software u](https://www.techtarget.com/searchapparchitecture/definition/software)nder development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations.

The SRS phase consists of two basic activities:

**Problem/Requirement Analysis:** The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

**Requirement Specification:** Here, the focus is on specifying what has been found giving analysis such as representation, Specification languages and tools, and checking the specifications are addressed during this activity.

**7.1. PURPOSE:**

The purpose of this document is to describe all external requirements for the E-learning System. It also describes the interfaces for the system.

**Proposed System Architecture:** The proposed system is built around conventional three-tier architecture.

**7.2. ADVANTAGES:**

* Easy to explain to the users.
* Structures approach.
* Stages and activities are well defined.
* Helps to plan and schedule the project.
* Verification at each stage ensures early detection of errors/misunderstanding.  Each phase has specific deliverables.

**7.3. DISADVANTAGES:**

* Assumes that the requirements of a system can be frozen.
* Very difficult to go back to any stage after it finished.
* A little flexibility and adjusting scope is difficult and expensive.
* Costly and required more time, in addition to the detailed plan.

#### 8. SOFTWARE DEVELOPMENT ENVIRONMENT

**History of web Application:** 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.

**Introduction to Web Server:** A Web Server is a computer on the World Wide Web that stores Html documents that can be retrieved via a Web browser. It is a computer that delivers web pages.

**About JSP:**

JSP pages are platform independent. A Java Server Page is a simple text file consisting of HTML or XML content along with JSP elements. When a client requests a JSP page of the web server and it has not been run before, the page is first passed to a JSP engine which compiles the page to a servlet, runs it and returns the resulting content to the client. Thereafter, the web server's servlet engine will run the compiled page.

**8.1. About SERVLETS:**

Servlets are the Java platform technology of choice for extending and enhancing Web servers. Servlets provide a component based platform independent method for building Web- based applications. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection.

**8.1.2. About HTML:**

HTML (hyper text markup language) is a language used to create hyper text documents that have hyper links embedded in them. It consists of tags embedded in the text of a document with HTML. We can build web pages or web documents. it is basically a formatting language and not a programming language.

**8.1.3. About ECLIPSE:**

An eclipse is an astronomical event that occurs when an astronomical object is temporarily obscured, either by passing into the shadow of another body or by having another body pass between it and the viewer. The term eclipse is most often used to describe either a solar eclipse, when the Moon's shadow crosses the Earth's surface, or a lunar eclipse, when the Moon moves into the Earth's shadow.

**8.1.4. About TOMCAT:**

Tomcat is an application server from the Apache Software Foundation that executes Java servlets and renders Web pages that include Java Server Page coding. Described as a "reference implementation" of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open collaboration of developers and is available from the Apache Web site in both binary and source versions.

**8.1.5. About MYSQL DATABASE:**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses.

MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company.

MySQL is becoming so popular because of many good reasons .

**8.2.JDBC DRIVERS:**

The JDBC API only defines interfaces for objects used for performing various database- related tasks like opening and closing connections, executing SQL commands.

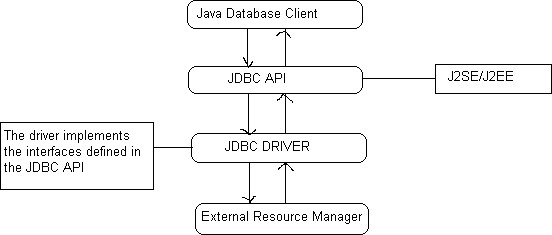


Fig.8.6. JDBC driver

Depending on the mechanism of implementation, JDBC drivers are broadly classified into four types.

**TYPE1:** Type1 JDBC drivers implement the JDBC API on top of a lower level API like ODBC.

These drivers are not generally portable because of the independency on native libraries.

**TYPE2:** Type2 drivers are written in mixture of java and native code. Type2 drivers use vendors specific native APIs for accessing the data source.

**TYPE3:** Type3 drivers use an intermediate middleware server for accessing the external data sources. The calls to the middleware server are database independent.

**TYPE4:** Type4 drivers are written in pure java and implement the JDBC interfaces and translate the JDBC specific calls to vendor specific access calls.

#### 9. SYSTEM DESIGN

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps.

**9.1. SOFTWARE DESIGN:**

In designing the software following principles are followed:

1. **Modularity and partitioning**: software is designed such that, each system should consist of hierarchy of modules and serve to partition into separate function.
2. **Coupling:** modules should have little dependence on other modules of a system.
3. **Cohesion:** modules should carry out in a single processing function.
4. **Shared use:** avoid duplication by allowing a single module be called by other that need the function it provide.

**9.2. INPUT/OUTPUT DESIGN:**

**Input design:** considering the requirements, procedures to collect the necessary input data in most efficiently designed.

**Output design:** All the screens of the system are designed with a view to provide the user with easy operations in simpler and efficient way, minimum keystrokes possible.

**10. SEQUENCE DIAGRAM:**

**10.1. SEQUENCE DIAGRAM:**

A sequence diagram, in the context of UML, represents object collaboration and is used to define event sequences between objects for a certain outcome. A sequence diagram is an essential component used in processes related to analysis, design and documentation.

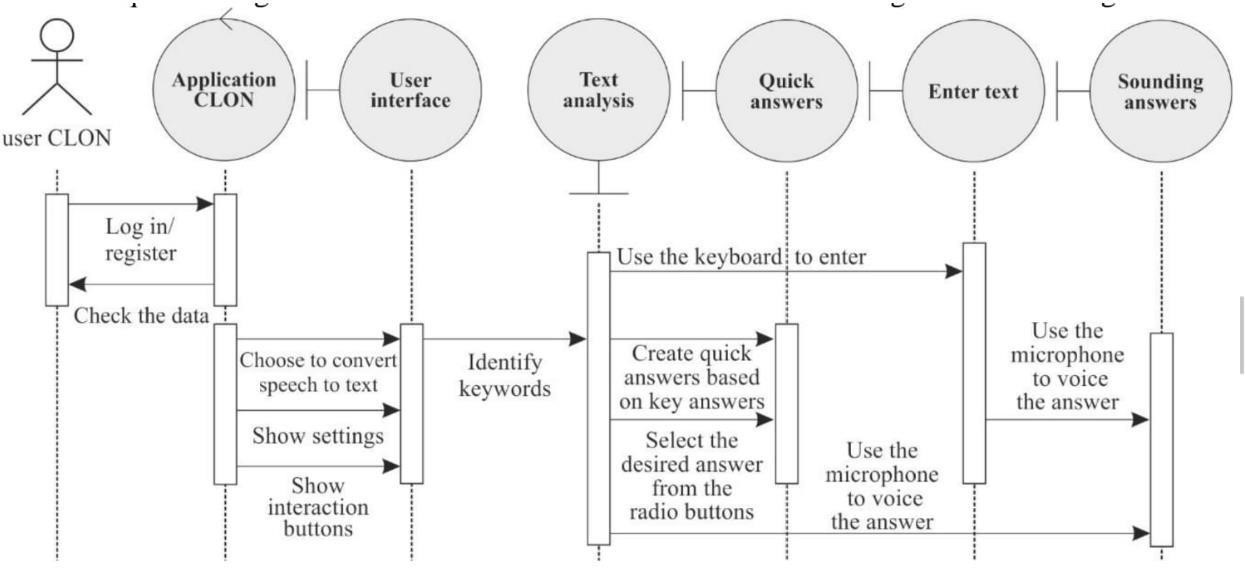


Fig.10.1.1. Sequence Diagram

**10.2. DATA FLOW DIAGRAM:**

A data-flow diagram is a way of representing a flow of data through a process or a system. The

DFD also provides information about the outputs and inputs of each entity and the process itself.

It is a graphical or visual representation using a standardized set of symbols and notations.

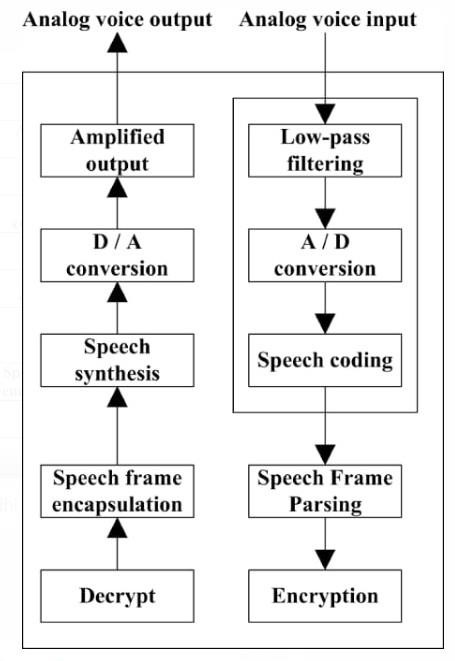


Fig.10.2.1. Data Flow Diagram

**10.3.USE CASE DIAGRAM:**

A use case diagram is a graph of actors set of use cases enclosed by a system boundary, communication associations between the actors and users and generalization among use cases.

The use case model defines the outside(actors) and inside (use case) of the system’s behavior.

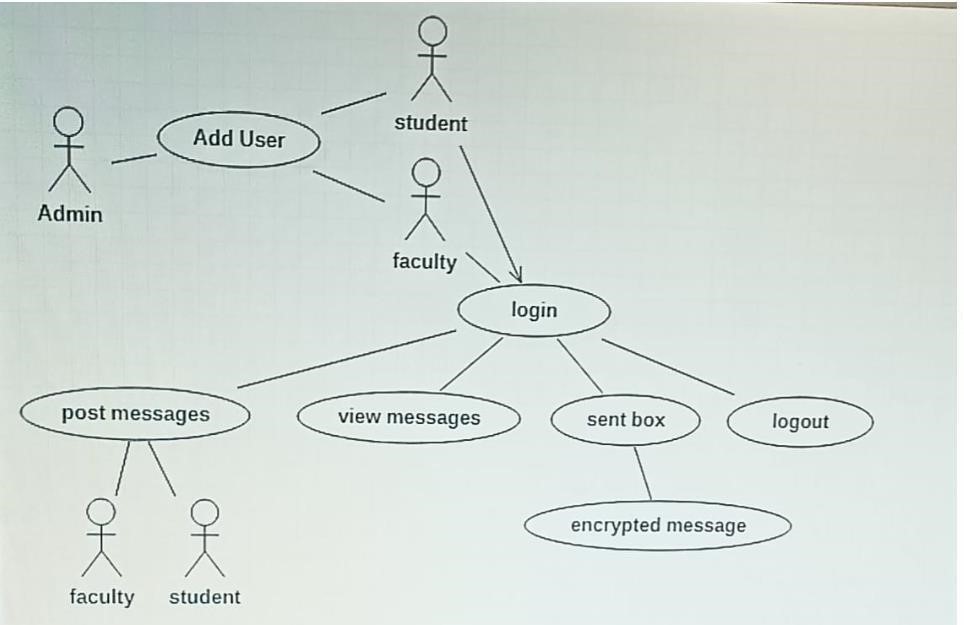


Fig.10.3.1. Use Case Diagram

**10.4. CLASS DIAGRAM:**

A class is a representation of an object and, in many ways; it is simply a template from which objects are created. Classes form the main building blocks of an object-oriented application.

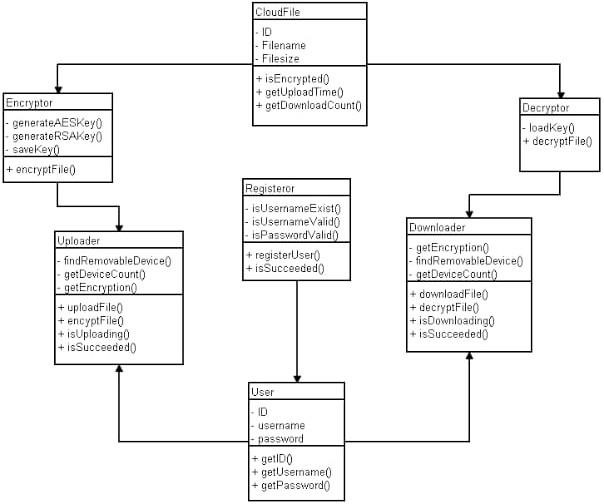


Fig.10.4.1. Class Diagram

**10.5. ER Diagram:**

Data models are tools used in analysis to describe the data requirements and assumptions in the system from a top-down perspective. They also set the stage for the design of databases later on in the SDLC.

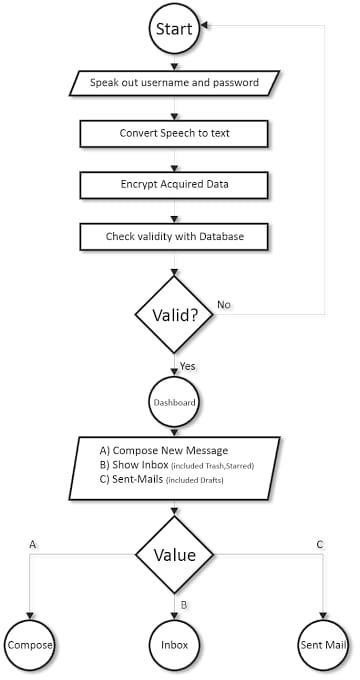


Fig.10.5.1. ER Diagram

**10.6. State Diagram:**

state diagram is a type of used in computer science and related fields to describe the behavior of systems. 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

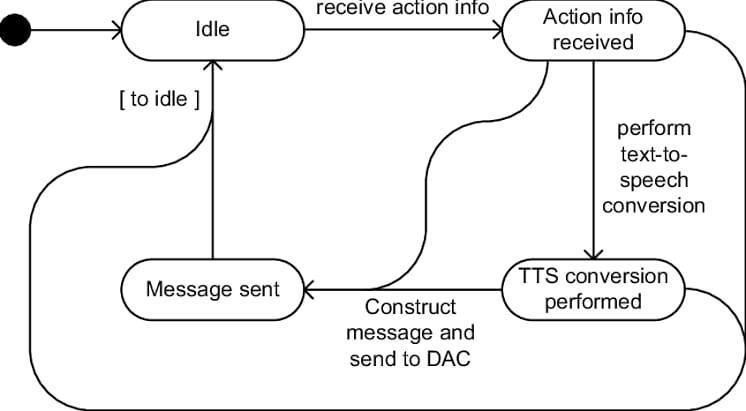


Fig.10.6.1. State Diagram

**10.7. Deployment Diagram:**

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.[ To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected

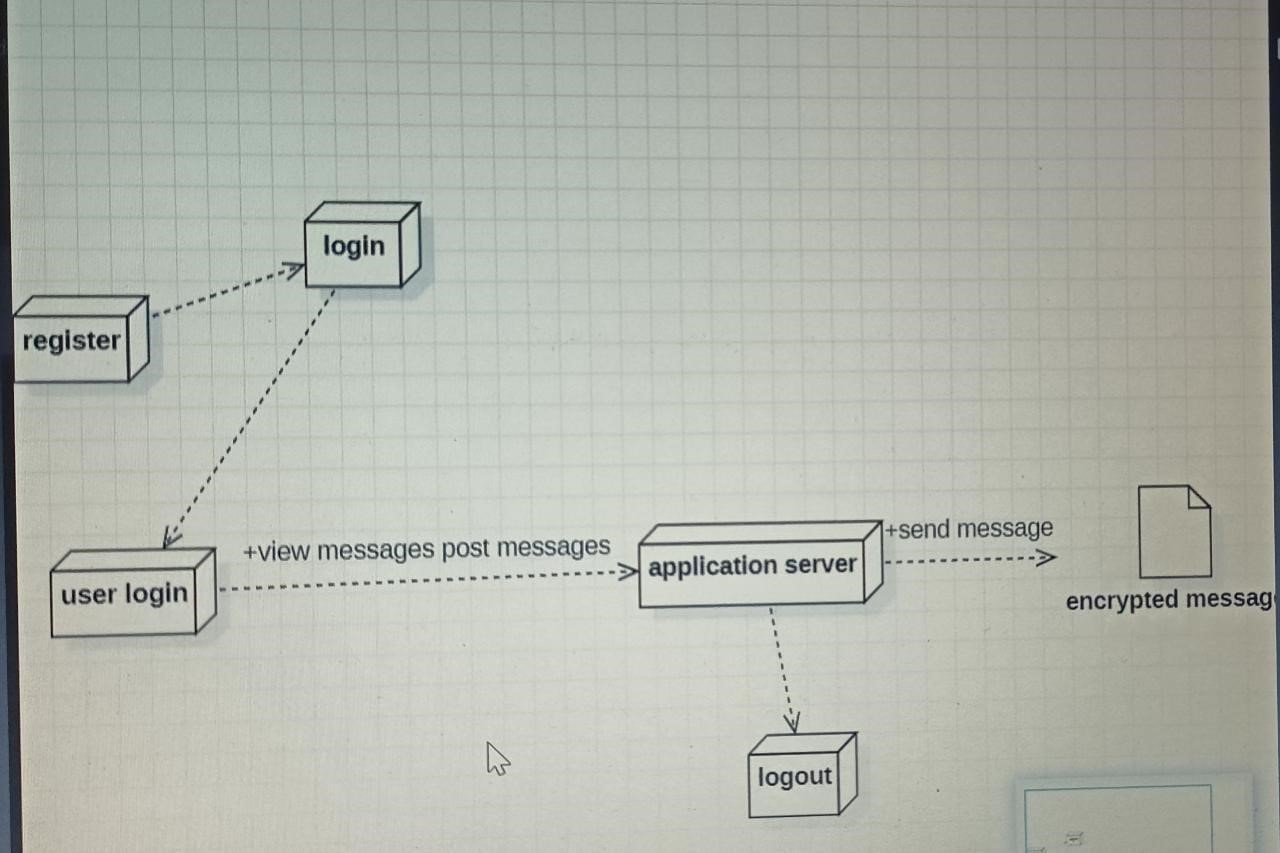


Fig.10.7.1. Deployment Diagram

**10.8. Flow Chart Diagram:**

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

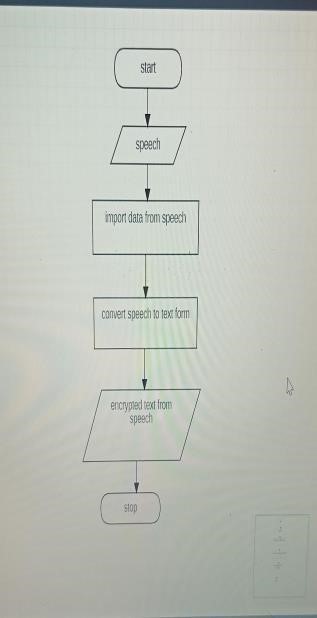


Fig.10.8.1. Flow Chart Diagram

# 11. SOFTWARE TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation**.**The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error

**11.1. TESTING STRATEGIES:**

**White box testing:** White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases.

**Black Box Testing:** Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error.

**Unit Testing:** Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program.

**Integration Testing:** All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules.

**Validation Testing:** It concentrates on confirming that the software is error-free in all respects.

All the specified validations are verified and the software is subjected to hard-core testing.

 System Testing: This testing is a series of different tests whose primary is to fully exercise the computer-based system.

##### 

##### 12. CODING AND IMPLEMENTATION

**12.1. SAMPLE CODE:**

**Index.Jsp:**

<!DOCTYPE HTML>

<html>

<head>

<title>Voidmain</title>

<meta name="description" content="website description" />

<meta name="keywords" content="website keywords, website keywords" />

<meta http-equiv="content-type" content="text/html; charset=windows-1252" />

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

<body>

<div id="main">

<div id="header">

<div id="logo">

<div id="logo\_text">

<!-- class="logo\_colour", allows you to change the colour of the text -->

<h1><a href="index.html">Robust Algorithm <span class="logo\_colour">forSecure</span></a></h1>

<h2>Voice Communication</h2>

</div>

</div>

</div>

<div id="content\_header"></div>

<div id="site\_content">

<div id="content">

<!-- insert the page content here -->

<% String status=request.getParameter("status"); if(status!=null) { %>

<h1><%=status%></h1>

<%}else{%>

<h1>Login Here</h1>

<%}%>

<form action="LoginServlet">

<div class="form\_settings">

<p>

<span>User Name :</span><input class="contact" type="text" name="username" value="" /> </p>

<p>

<span>Password :</span><input class="contact" type="password" name="password" value=""/> </p>

<p style="padding-top: 15px">

<span>&nbsp;</span><input class="submit" type="submit" name="contact\_submitted"value="Login" /> </p>

<form action="LoginServlet">

<div class="form\_settings">

<p>

<span>User Name :</span><input class="contact" type="text" name="username" value="" />

</p>

<p>

<span>Password :</span><input class="contact" type="password" name="password" value=""/> </p>

<p style="padding-top: 15px">

<span>&nbsp;</span><input class="submit" type="submit" name="contact\_submitted" value="Login" /> </p>

</div>

</form>

</div>

</div>

<div id="content\_footer"></div>

</div>

<%@page import="java.util.Date"%>

<%@page import="com.voidmain.dao.HibernateDAO"%>

<%@page import="com.voidmain.dao.HibernateTemplate"%>

<!DOCTYPE HTML>

<html>

<head>

<title>Voidmain</title>

<meta name="description" content="website description" />

<meta name="keywords" content="website keywords, website keywords" />

<meta http-equiv="content-type"

content="text/html; charset=windows-1252" />

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

<link rel="stylesheet" href="speech-input.css">

</head>

<body>

<form action="LoginServlet">

<div class="form\_settings">

<p>

<span>User Name :</span><input class="contact" type="text" name="username" value="" /> </p>

<p>

<span>Password :</span><input class="contact" type="password" name="password" value=""/> </p>

<p style="padding-top: 15px">

<span>&nbsp;</span><input class="submit" type="submit" name="contact\_submitted" value="Login" /> </p>

</div>

</form>

</div>

</div>

<div id="content\_footer"></div>

</div>

</body>

</html>

**Add message.Java:**

<%String role = (String) request.getSession().getAttribute("role"); String username = (String) request.getSession().getAttribute("username"); if (role.equals("admin")) {%>

<li><a href="adduser.jsp">Add User</a></li>

<li><a href="viewusers.jsp">View User</a></li>

<%

}

%>

<%

if (role.equals("user")) {

%>

<li><a href="addmessage.jsp">Post Message</a></li>

<li><a href="viewmessages.jsp">View Messages</a></li>

<li><a href="sentbox.jsp">Sent Box</a></li>

<%@page import="java.util.Date"%>

<%@page import="com.voidmain.dao.HibernateDAO"%>

<%@page import="com.voidmain.dao.HibernateTemplate"%>

<!DOCTYPE HTML>

<html>

<head>

<title>Voidmain</title>

<meta name="description" content="website description" />

<meta name="keywords" content="website keywords, website keywords" />

<meta http-equiv="content-type"

content="text/html; charset=windows-1252" />

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

<link rel="stylesheet" href="speech-input.css">

</head>

<body>

<div id="main">

<div id="header">

<div id="logo">

<div id="logo\_text">

<!-- class="logo\_colour", allows you to change the colour of the text -->

<h1>

<a href="#">Personnel <span class="logo\_colour">Onus</span></a>

</h1>

</div>

</div>

<div id="menubar">

<ul id="menu">

<%String role = (String) request.getSession().getAttribute("role"); String username = (String) request.getSession().getAttribute("username"); if (role.equals("admin")) {%>

<li><a href="adduser.jsp">Add User</a></li>

<li><a href="viewusers.jsp">View User</a></li>

<%

}

%>

<%

if (role.equals("user")) {

%>

<li><a href="addmessage.jsp">Post Message</a></li>

<li><a href="viewmessages.jsp">View Messages</a></li>

<li><a href="sentbox.jsp">Sent Box</a></li>

<%

%>

<li><a href="logout.jsp">Logout</a></li>

</ul>

</div>

</div>

<div id="content\_header"></div>

<div id="site\_content">

<div id="content">

<!-- insert the page content here -->

<%

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

if (status != null) {

%>

<h1><%=status%></h1>

<%

} else {

%>

<h1>Post Message</h1>

<%

}

%>

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

<input type="hidden" name="type" value="com.voidmain.pojo.Message">

<input type="hidden" name="operation" value="add"> <input type="hidden" name="redirect" value="addmessage.jsp">

<div>

<table align="center">

<tr>

<td><span>Message</span></td>

<td><textarea rows="5" cols="20" name="message" class="speech-input"></textarea></td> </tr>

<tr>

<td><span>Select Gender</span></td>

<td>

<select name="gender">

<option value="male">Male</option>

<option value="female">Female</option> </select>

</td>

</tr>

<tr>

<td><span>Select User Type</span></td>

<td>

<select name="userType">

<option value="">--select--</option>

<option value="student">Student</option>

<option value="faculty">Faculty</option>

</select>

</td>

</tr>

<tr>

<td><span>Age</span></td>

<td><input class="contact" type="text" name="age" value="" /></td> </tr>

<tr>

<td colspan="2" align="center"><input class="submit" type="submit" value="Post Message"

/></td>

</tr>

</table>

</div>

</form>

</div>

</div>

</div>

<div id="content\_footer"></div>

<div id="footer">

Copyright &copy; 2019 <a href=["http://www.voidmaintechnologies.com">](http://www.voidmaintechnologies.com/)Voidmain

Technologies</a>

</div>

<script src="speech-input.js"></script>

<script async defer src="https://buttons.github.io/buttons.js" style="width: 10"></script>

</body> </html> Adduser.jsp:

<%@page import="com.voidmain.dao.HibernateDAO"%>

<%@include file="header.jsp"%>

<div id="content">

<!-- insert the page content here -->

<%

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

if (status != null) {

%>

<h1><%=status%></h1>

<%

} else {

%>

<h1>Add User</h1>

<%

}

%>

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

<input type="hidden" name="type" value="com.voidmain.pojo.User"> <input type="hidden" name="operation" value="add"> <input type="hidden" name="redirect" value="adduser.jsp">

<div class="form\_settings">

<p>

<span>Name</span><input class="contact" type="text" name="name" value="" />

</p>

<p>

<span>User Id</span><input class="contact" type="text" name="userId" value="" />

</p>

<p>

<span>Password</span><input class="contact" type="password" name="password" value="" />

</p>

<p>

<span>Select Gender</span> <select name="gender">

<option value="male">Male</option>

<option value="female">Female</option>

</select>

</p>

<p>

<span>Select User Type</span> <select name="userType">

<option value="">--select--</option>

<option value="student">Student</option>

<option value="faculty">Faculty</option>

</select>

</p>

<p>

<span>Age</span><input class="contact" type="text" name="age" value="" />

</p>

<p style="padding-top: 15px">

<span>&nbsp;</span><input class="submit" type="submit" name="contact\_submitted" value="Add User" />

</p>

</div>

</form>

</div>

<%@include file="footer.jsp"%> Sentbox.Java:

<%@page import="com.voidmain.service.AES"%>

<%@page import="com.voidmain.pojo.User"%>

<%@page import="com.voidmain.pojo.Message"%>

<%@page import="com.voidmain.dao.HibernateTemplate"%>

<%@page import="com.voidmain.dao.HibernateDAO"%>

<%@page import="java.util.List"%>

<%@include file="header.jsp"%>

<div id="content">

<!-- insert the page content here -->

<%

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

if (status != null) {

%>

<h1><%=status%></h1>

<%

} else {

%>

<h1>View Messages</h1>

<%

}

%>

<table style="width: 100%; border-spacing: 0;">

<tr>

<th>Sender</th>

<th>Date</th>

<th>Message</th>

<th>Delete</th>

<%

User user = HibernateDAO.getUserById(username); for (Message message : HibernateDAO.getMessage()) {

if(message.getSender().equals(username))

{

%>

<tr>

<td><%=message.getSender()%></td>

<td><%=message.getMessageDate()%></td>

<td><%=message.getMessage()%></td>

<td><a href="sentbox.jsp?id=<%=message.getMessageId()%>">delete</a></td> </tr>

<%

}

}

%>

</table>

<%

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

if (id != null) {

if (HibernateTemplate.deleteObject(Message.class, Integer.parseInt(id)) == 1) {

response.sendRedirect("sentbox.jsp?status=success");

} else {

response.sendRedirect("sentbox.jsp?status=failed");

}

}

%>

</div>

<%@include file="footer.jsp"%> View message.jsp:

<%@page import="com.voidmain.service.AES"%>

<%@page import="com.voidmain.util.VoiceUtil"%>

<%@page import="com.voidmain.pojo.User"%>

<%@page import="com.voidmain.pojo.Message"%>

<%@page import="com.voidmain.dao.HibernateTemplate"%>

<%@page import="com.voidmain.dao.HibernateDAO"%>

<%@page import="java.util.List"%>

<%@include file="header.jsp"%> <div id="content">

<!-- insert the page content here -->

<%

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

if (status != null) {

%>

<h1><%=status%></h1>

<%

} else {

%>

<h1>View Messages</h1>

<%

}

%>

<table style="width: 100%; border-spacing: 0;">

<tr>

<th>Sender</th>

<th>Date</th>

<th>Message</th>

<th>Read</th>

<th>Delete</th>

<%

User user = HibernateDAO.getUserById(username);

String attributeKey=user.getGender().charAt(0)+user.getAge()+user.getUserType().charAt(0); for (Message message : HibernateDAO.getMessage()) {

String finalKey=message.getSeckey()+attributeKey;

AES aes=new AES(); aes.setKey(finalKey);

String mess=aes.decrypt(message.getMessage());

if(mess!=null)

{

%>

<tr>

<td><%=message.getSender()%></td>

<td><%=message.getMessageDate()%></td>

<td><%=mess%></td>

<td><a href="viewmessages.jsp?message=<%=message.getMessage()%>">read</a>

<%

if (message.getSender().equals(username)) {

%>

<td><a href="viewmessages.jsp?id=<%=message.getMessageId()%>">delete</a> </td>

<%

} else {

%>

<td>No Access</td>

<%

}

%>

</tr>

<%

}

}

%>

</table><%String message=request.getParameter("message");

if(message!=null)

{

VoiceUtil.getVoice().speak(message);

}

%>

<%

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

if (id != null) {

if (HibernateTemplate.deleteObject(Message.class, Integer.parseInt(id)) == 1) {

response.sendRedirect("viewmessages.jsp?status=success");

} else {

response.sendRedirect("viewmessages.jsp?status=failed");

}

}

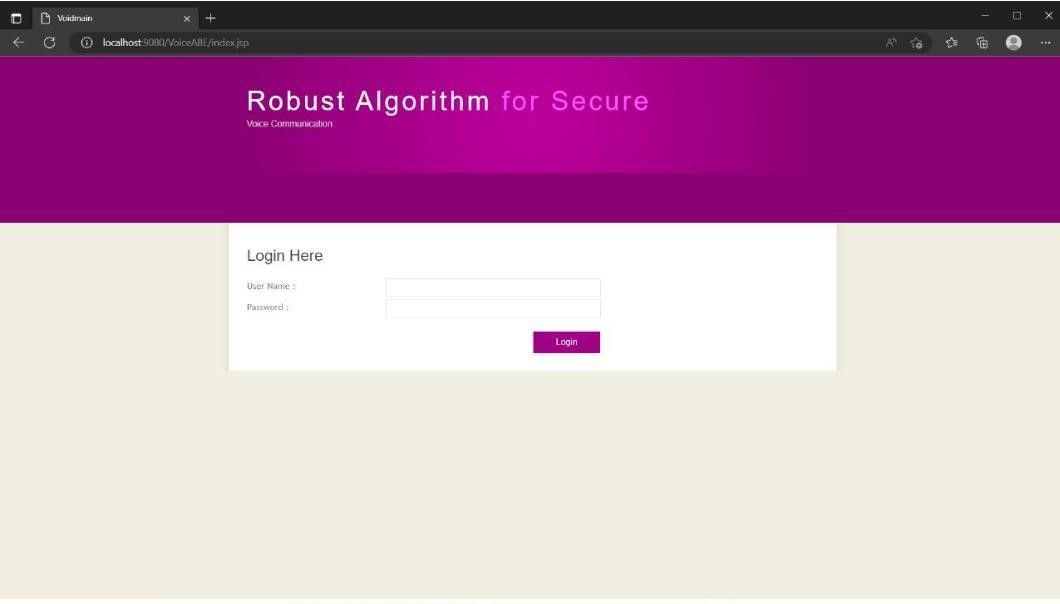
%>

</div>

<%@include file="footer.jsp"%>

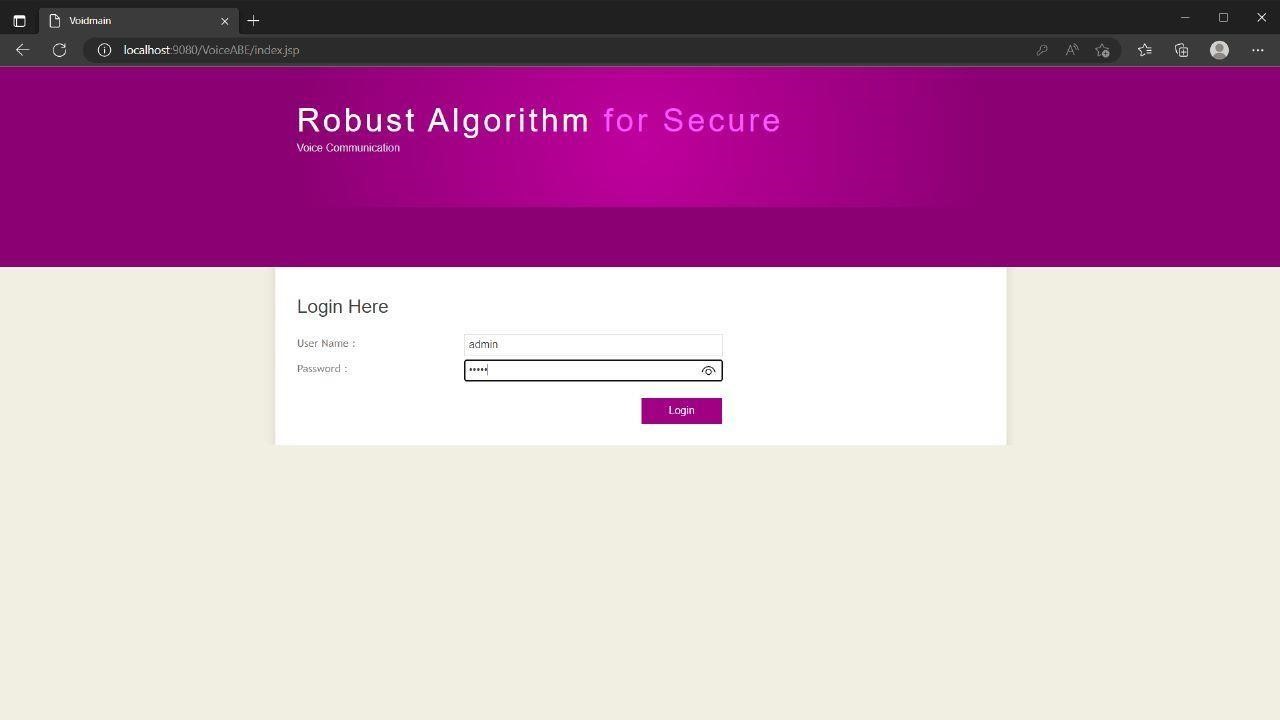
##### 

##### 13.OUTPUT SCREENS



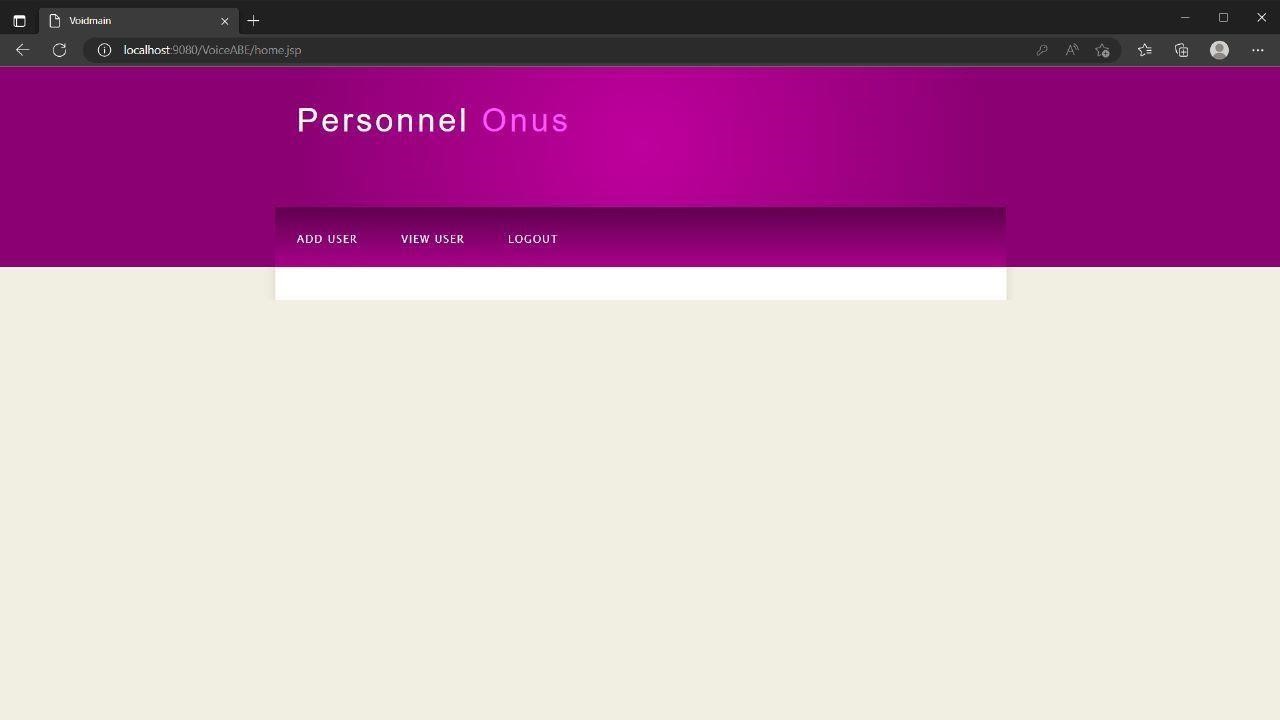
###### Fig.13.0.1

In this page, it is mandatory to enter the details of a user. This is named to be **Admin login page** as follows…….



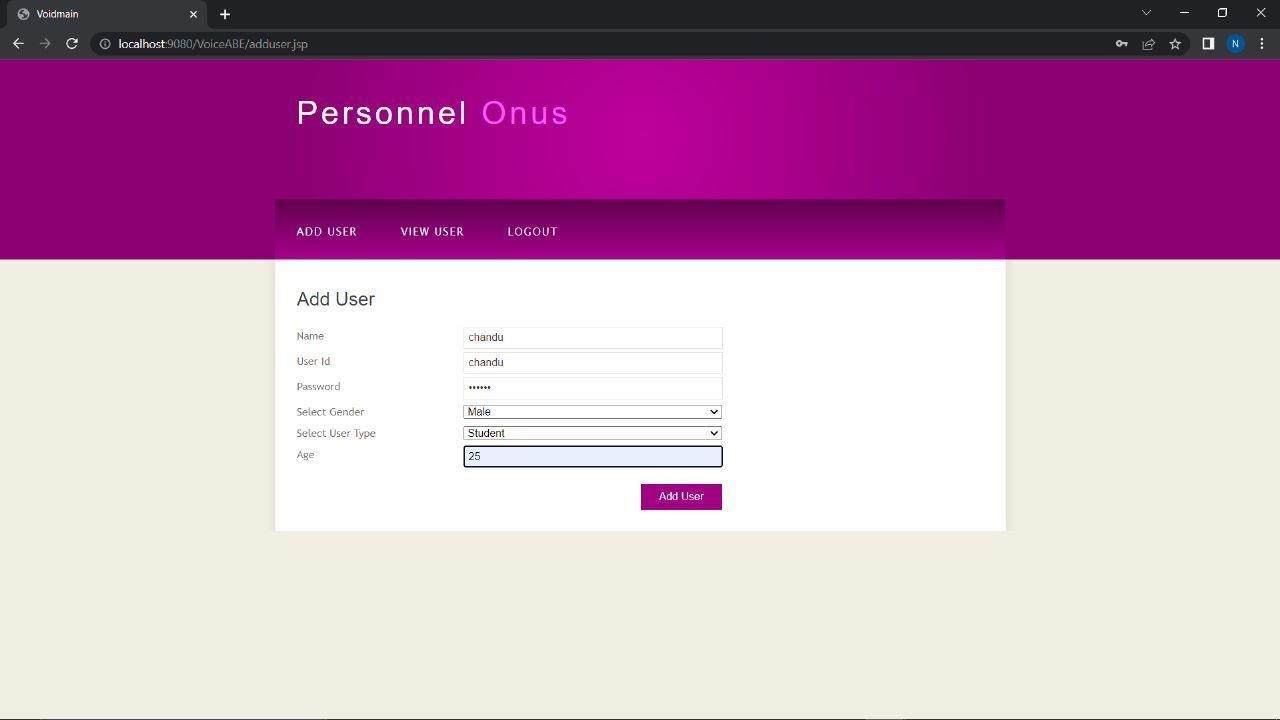
**Fig.13.0.2.**

After entering the details in login page, this visual will be opened. In this there are various kind of tabs to enter the user details.



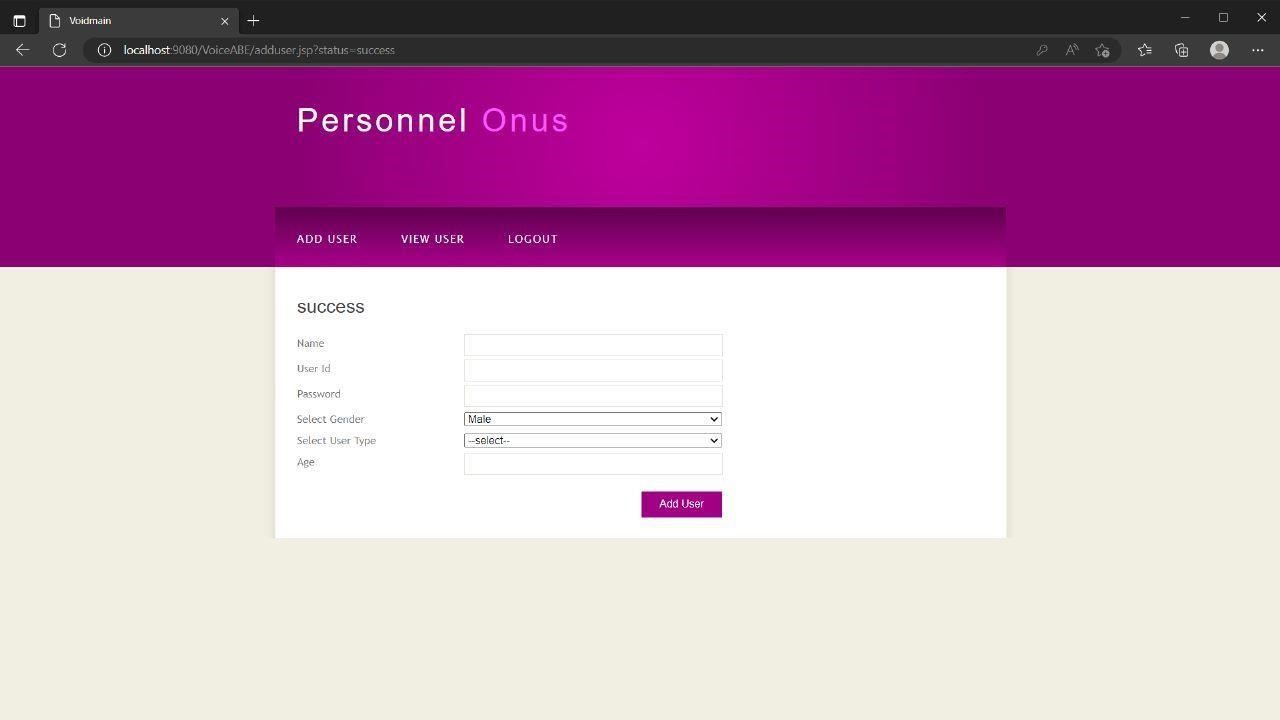
###### Fig.13.0.3

This is an add user page, where a user can fill their details.



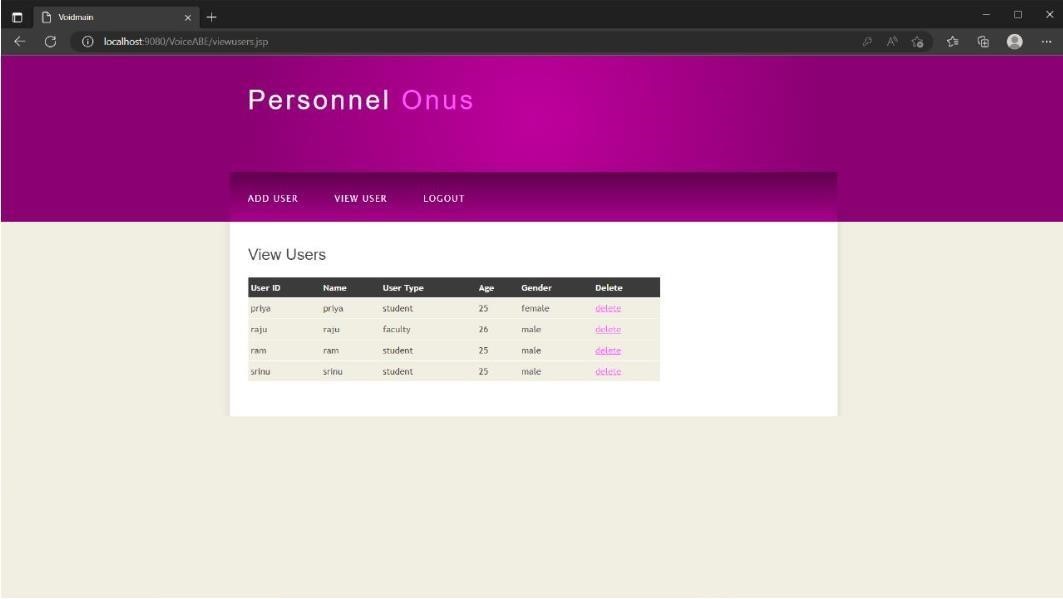
###### Fig.13.0.4

Here in this we are adding the user details which is shown on the admin home page.



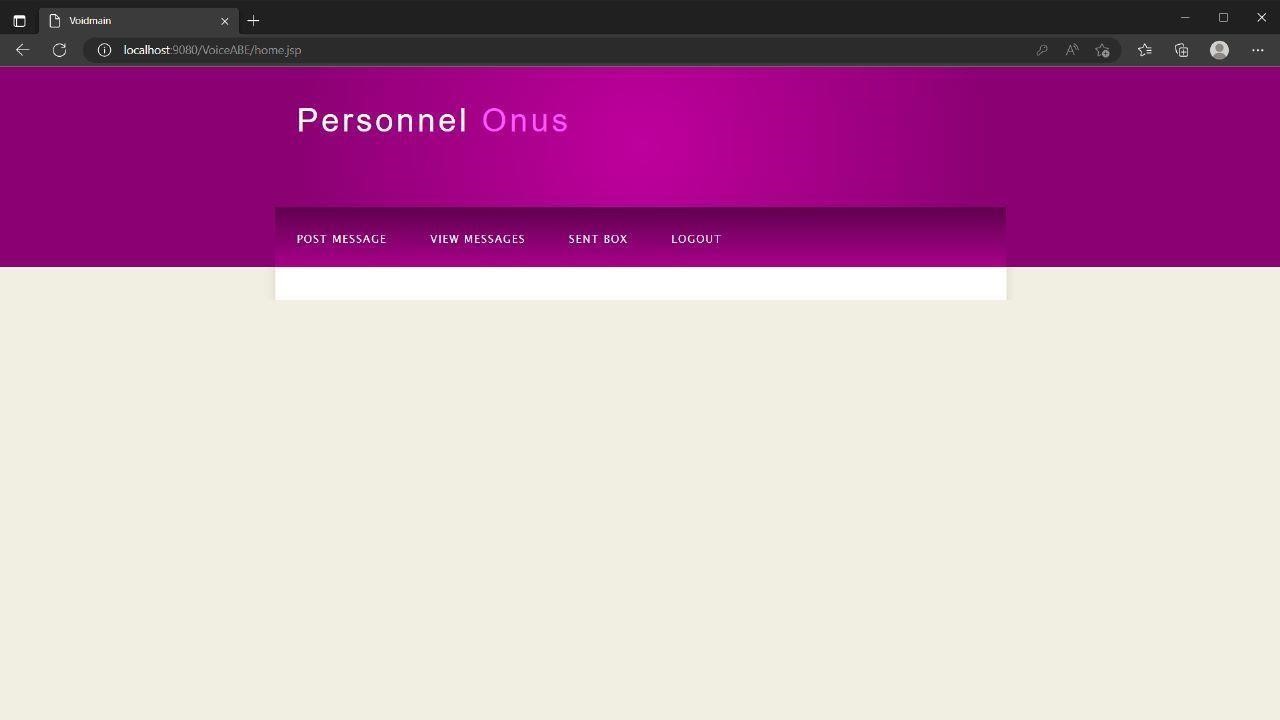
###### Fig.13.0.5

Through this page we can confirm that the user added successfully……….

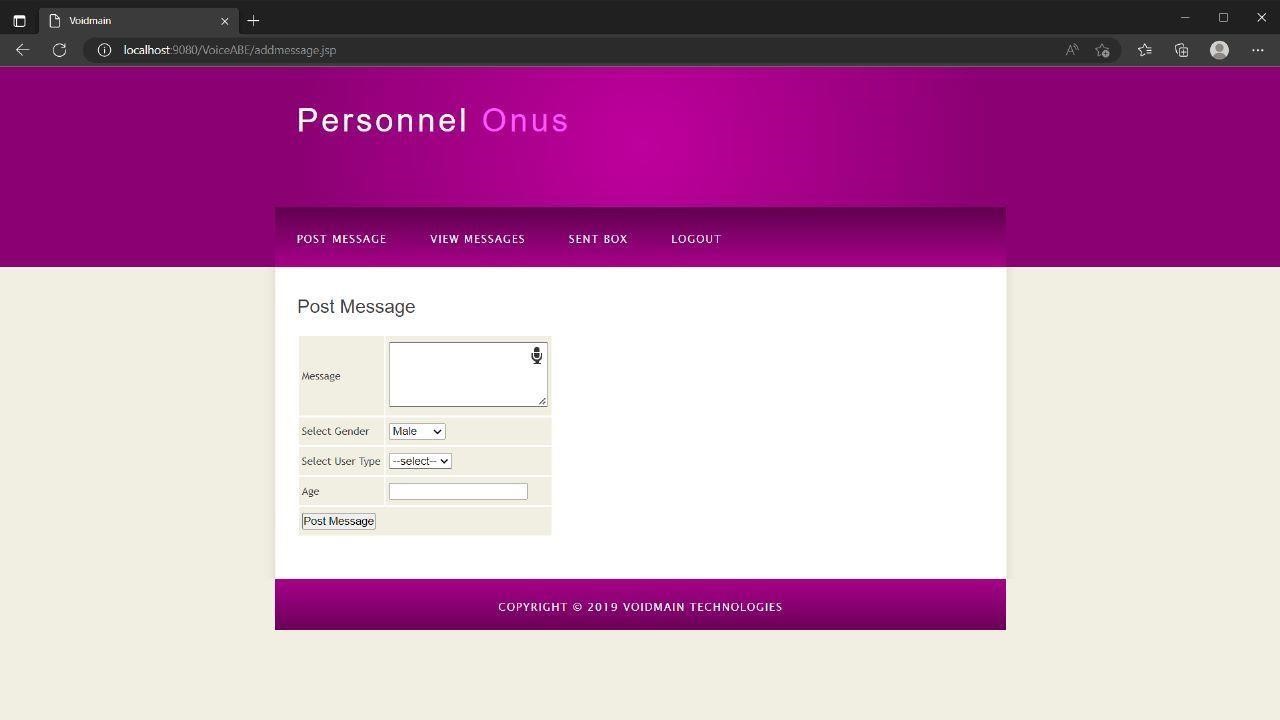


###### Fig.13.0.6

After submitting the details of the user success page will be shown as below.

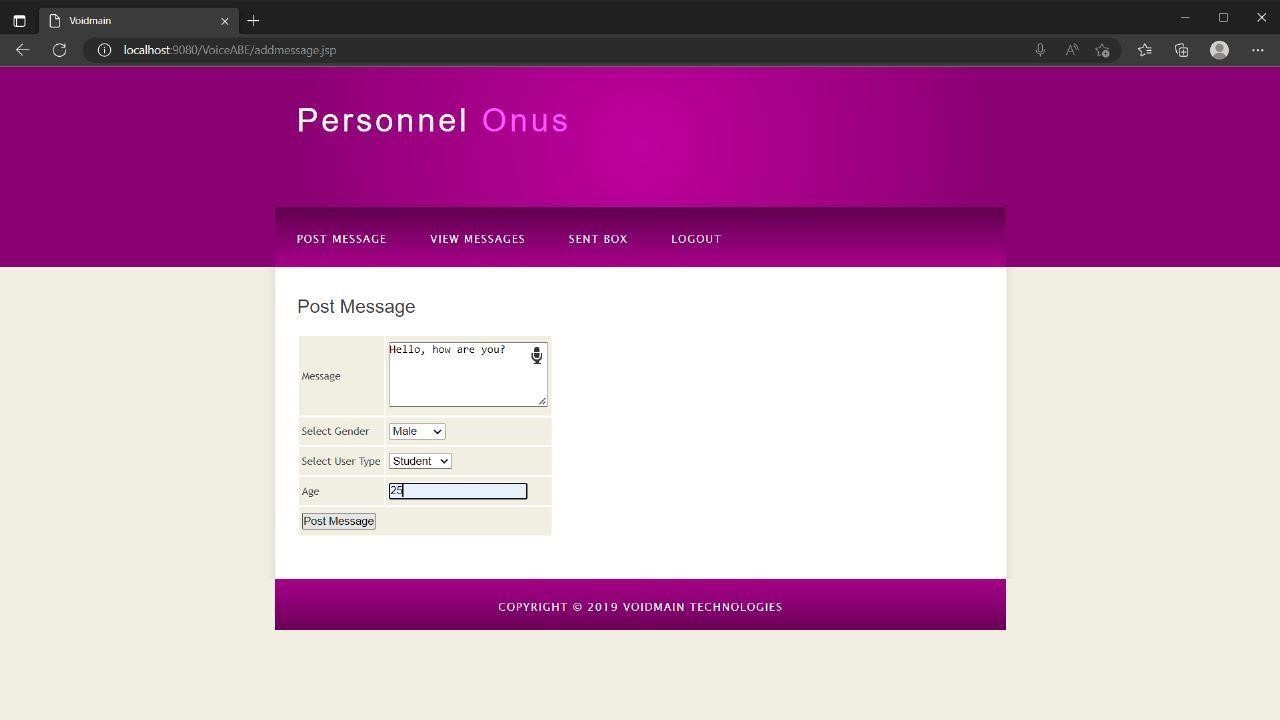


**Fig.13.0.7 This is a user home page as follows………….**



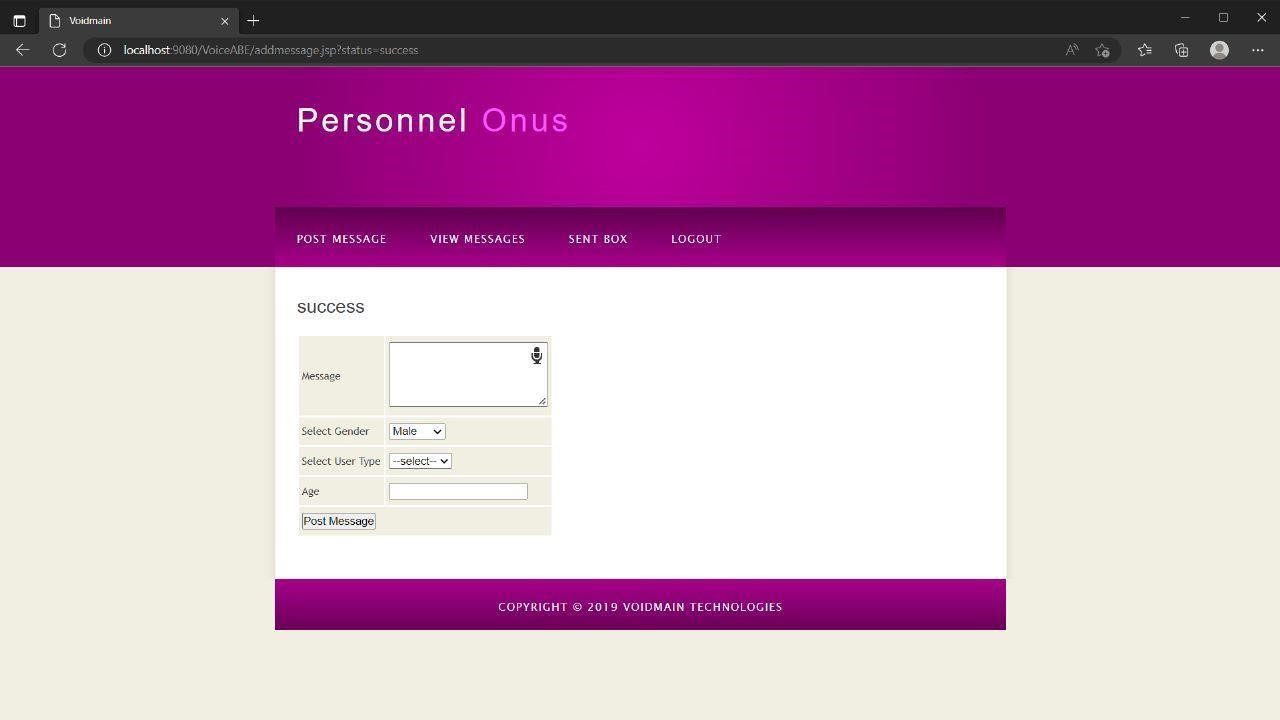
###### Fig.13.0.8

A user can post a message through voice or text



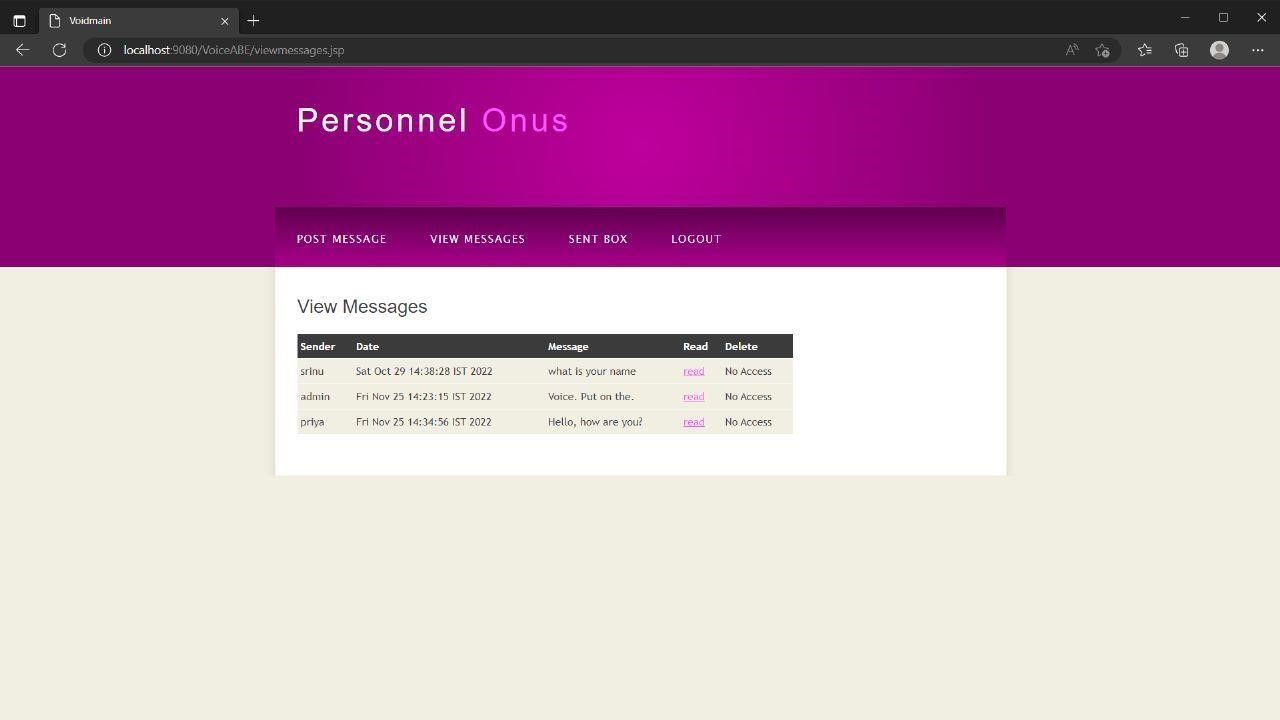
###### Fig.13.0.9

User can post a message to student or faculty by selecting user type……



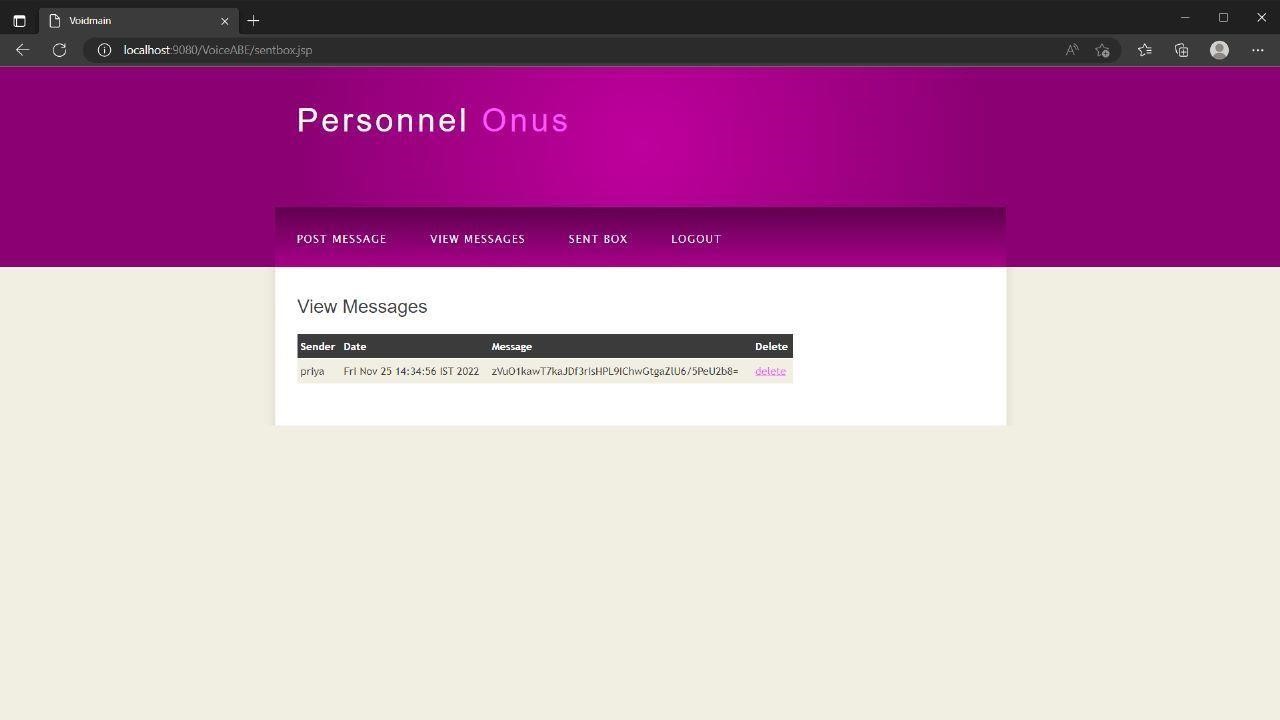
###### Fig.13.1.1

Message successfully sent page



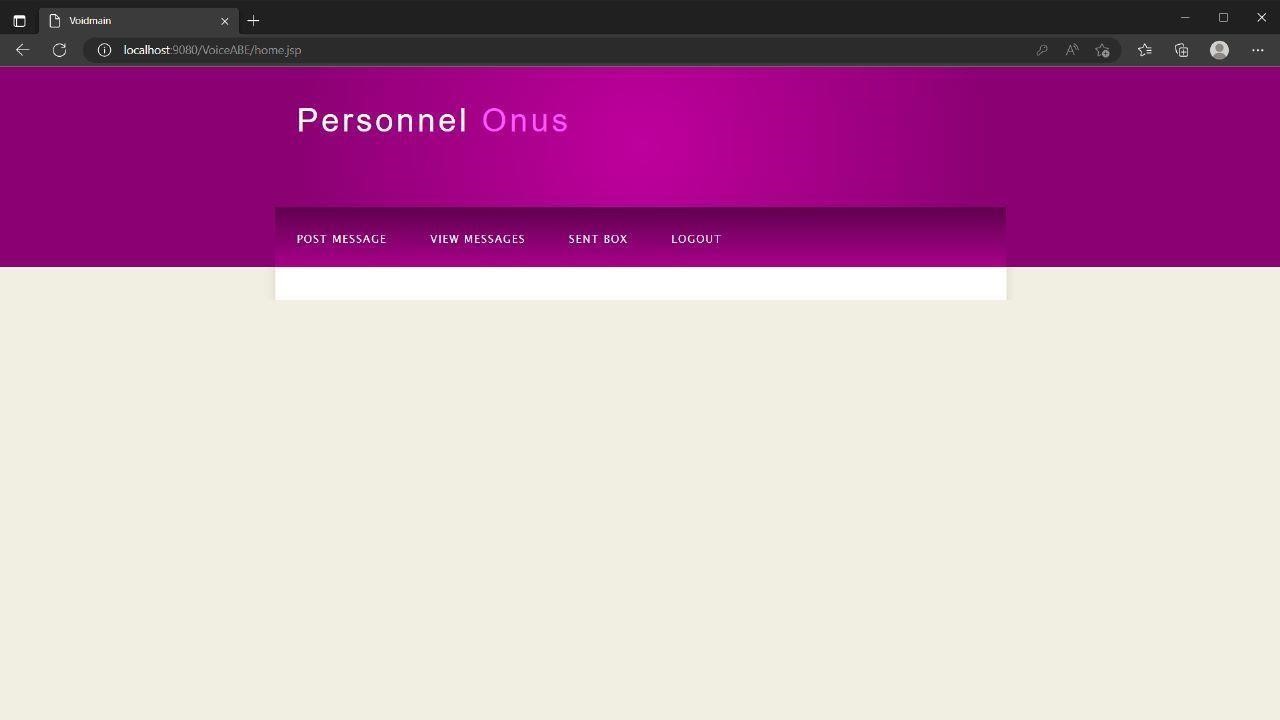
**Fig.13.1.2**

These are the messages sent to a particular user………



**Fig.13.1.3**

This is a message sent by the user…………….



**Fig.13.1.4**

User can log out by clicking a tab named logout

##### 

##### 14.CONCLUSION

The main objective of this paper was to analyze the primary factors that affect farmers’ awareness of different ecosystem services. Through the questionnaire survey on Jinqiao Village, we obtained certain research results and prospects, and further analysis concluded as follows: the farmers’ personal characteristics (such as age, gender, educational level and immigration), family characteristics (such as the proportion of agricultural income and cultivated area) and subjective attitude (such as returning farmland to forest and protecting the environment), certainly influence farmers’ understanding of ecological services. Ecosystem services and human well-being are difficult matters; however, we should pay more attention to this subject and provide more effective recommendations.

##### 15. FUTURE ENHANCEMENTS

**Project review:**

At the beginning of this project several minimum requirements were submitted. It seems rather obvious to state that the success of the project depends on the extent to which it met these minimum requirements.

The minimum requirements were specified These were:

* Examine the tools and methodologies required to gain an overview of the system requirements for the proposed database.
* Examine suitable database management systems that can be used to implement the proposed database.
* Evaluate appropriate website authoring and web graphic creation tools that can be used to develop web based forms for the proposed database
* Produce and apply suitable criteria for evaluating the solution

To meet these requirements research was conducting to find the various software development methodologies. After analyzing each technique, the chosen methodology was adapted to meet the needs of this project, and then followed as accurately as possible. Therefore, this project went through the appropriate development processes thus meeting this requirement

The main aim of this project was to produce an appropriate solution which would meet the requirements of the user this project not only met all the functional requirements of the application, but exceeded them by implementing several of the non-essential requirements listed in section.

##### 

##### 16. BIBLIOGRAPHY

##### 

**Textbooks Referred:** afsa A, Alimi N, Sghaier A, et al. (2017). A hardware-software co-designed AES-ECC cryptosystem[C]//2017 international conference on advanced systems and electric technologies (IC\_ASET). IEEE, 50-54 Liu S, Wang S, Liu X, et al (2020) Fuzzy detection aided real-time and robust visual tracking under complex environments. IEEE Trans Fuzzy Syst.

[https://doi.org/10.1109/TFUZZ.2020.3006520 S](https://doi.org/10.1109/TFUZZ.2020.3006520)ultan A, Yang X, Hajomer A A E, et al. (2018). Chaotic constellation mapping for physical-layer data encryption in OFDM-PON. IEEE photonics technology letters, PP:1-1 Vaidyanathan S, Sambas A, Mamat M, WS MS (2017) Analysis, synchronisation and circuit implementation of a novel jerk chaotic system and its application for voice encryption. Int J Model Identif Control 28(2):153–166

**Websites:**

[http://www.java.sun.com](http://www.java.sun.com/) [http://www.javaworld.com](http://www.javaworld.com/) [http://www.coreservlets.com](http://www.coreservlets.com/) [http://www.theserverside.com](http://www.theserverside.com/) [http://www.w3schools.com](http://www.w3schools.com/) [http://www.google.com](http://www.google.com/)