

EXPERTISE WITHOUT TOKENS TO PROTECT BANKING TRANSACTIONS AS GATEWAY

A PROJECT REPORT

Submitted by

**SAKTHI ARUMUGARAJ
SWARNAMALYA.N**

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

**BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING**



PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution, Affiliated To Anna University, Chennai

APRIL 2023

PANIMALAR ENGINEERING COLLEGE

An Autonomous Institution, Affiliated To Anna University, Chennai

BONAFIDE CERTIFICATE

Certified that this project report “**EXPERTISE WITHOUT TOKENS TO PROTECT BANKING TRANSACTIONS AS GATEWAY** ” is the bonafide work of “**SAKTHI ARUMUGARAJ (211419104227), SWARNAMALYA.N (211419104279)** ” who carried out the project work under my supervision.

SIGNATURE

**Dr.L.JABASHEELA,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT**

DEPARTMENT OF CSE,
ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

**Mrs.D. JENNIFER ,M.E.,(Ph.D.)
ASSISTANT PROFESSOR**

SUPERVISOR

DEPARTMENT OF CSE, PANIMALAR
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI, POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the End-Semester Project

Viva-Voce Examination held on 12/04/2023

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **SAKTHI ARUMUGARAJ (211419104227), SWARNAMALYA.N (211419104279)** hereby declare that this project report titled **“EXPERTISE WITHOUT TOKENS TO PROTECT BANKING TRANSACTIONS AS GATEWAY ”**, under the guidance of **Mrs. D. Jennifer M.E.,(Ph.D.)** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

SAKTHI ARUMUGARAJ

SWARNAMALYA.N

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **Mrs.D.Jennifer,M.E.,(Ph.D.)** and coordinator **Dr.K.VALARMATHI, M.E., Ph.D.** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

SAKTHI ARUMUGARAJ

SWARNAMALYA.N

ABSTRACT

The analysis and development of suggestions for the protection of information in geographically dispersed structures, characteristic of modern banks, based on cloud computing technology, is the primary goal of our proposed work. The processed information passes through the database of banks and payment systems, which potentially makes it available to the attacker. The proposed system proposes a solution to the problem of maintaining the uniqueness of information in them based on cloud technology without tokens and gives recommendations on the introduction of cloud computing technology into modern banking systems. In this project we store the information in the cloud based on security. The information processing used by analysis and manipulating the info and output in which the result cannot be altered and which is based on analyzing that information. The BVCS used in the model is that it maximizes the recovered data in $(2, n)$ -BVCS. In this project we proposed transparency, reducing transaction cost and decentralization.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF TABLES	iv
	LIST OF FIGURES	v
	LIST OF SYMBOLS, ABBREVIATIONS	vi
1.	INTRODUCTION	
	1.1 Overview	1
	1.2 Problem Definition	3
2.	LITERATURE SURVEY	4
3.	SYSTEM ANALYSIS	
	3.1 Existing System	9
	3.2 Proposed system	9
	3.3 Feasibility Study	10
	3.4 Hardware Environment	11
	3.5 Software Environment	12
4.	SYSTEM DESIGN	
	4.1. ER diagram	14
	4.2 Data dictionary	15
	4.3 Table Normalization	17
	4.4 Data Flow Diagram	19
	4.5 UML Diagrams	21

CHAPTER NO.	TITLE	PAGE NO.
5.	SYSTEM ARCHITECTURE	
	5.1 Module Design Specification	27
	5.2 Algorithms	28
6.	SYSTEM IMPLEMENTATION	30
7.	SYSTEM TESTING	69
	7.1 Unit Testing	69
	7.2 Integration Testing	72
	7.3 Test Cases & Reports	72
8.	CONCLUSION	
	8.1 Results & Discussion	74
	8.2 Conclusion and Future Enhancements	74
	APPENDICES	
	A.1 Sample Screens	76
	A.2 Publications	
	A.3 Plagiarism Report	
	REFERENCES	82

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NUMBER
4.2.1	Data dictionary-banking registration	15
4.2.2	Data dictionary-Viewing public user details	16
4.2.3	Data dictionary-viewing public user registration to the bank	16

LIST OF FIGURES

FIGURE NO	FIGURE TITLE	PAGE NUMBER
4.1.1	ER-Diagram	14
4.3.1	Table normalization-view_upload	17
4.3.2	Table normalization-user register	18
4.3.3	Table normalization-upload	18
4.3.4	Table normalization-secrete_file	19
4.4.1	Data flow-Level-0	20
4.4.2	Data flow-Level-1	20
4.4.3	Data flow-Level-2	20
4.4.4	Data flow-Overall diagram	21
4.5.1	Use case diagram	22
4.5.2	Activity diagram	23
4.5.3	Sequence diagram	24
4.5.4	Class diagram	25
5.1	Architecture diagram	26

LIST OF ABBREVIATIONS

ABBREVIATIONS

EXPANSIONS

AI

Artificial Intelligence

ML

Machine Learning

FCM

Fuzzy Cognitive Mapping

BVCS

Binocular Visual Cryptography Scheme

JVM

Java Virtual Machine

OTP

One Time Password

CHAPTER -1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

It is undeniably critical for service institutions to align their service resources with customer needs. And, as customer demands shift, resource allocation should change accordingly. Otherwise, a resource shortage or surplus will occur, resulting in a loss for the organization. The commercial bank, as an essential type of financial service institution, is no exception to this rule. In Practice, however, adjusting resource allocation effectively and efficiently is a difficult job for the bank manager. Many bank managers are troubled by the problem of human resource allocation (i.e., staffing) under demand variation, because human resource management is one of the most important but difficult pieces of work in the banking industry.

The seasonal demand variation is a frequent form of customer demand variation that commercial banks must consider when determining staffing policies. The economy and culture are the sources of this seasonality. Being a major economic node, the banking industry is impacted by a variety of economic variables, including seasonality. The authorization granted to representatives in current banking systems and business measures is managed and monitored by designed admittance control techniques, in which static job based models center on access to data and capacities. As a result, banks and organizations are looking for frameworks and applications to automate and improve their business processes and provide astute information to executives. In this context, the concept of incorporating AI (ML) processes into financial business metrics has emerged. Over the past ten years, mobile devices have largely taken the place of desktop computers in many daily tasks and are now the end user first option for Web-based activities.

For instance, in recent years, a number of banks have closed some of their locations as a result of the adoption of mobile banking. Without a question, it is essential for service organizations to align their service resources with customer needs. Additionally, the resource allocation needs to be modified as client demands change. Otherwise, the organization will suffer a loss due to a shortage or surplus of resources. The commercial bank, a significant class of financial service organization, is not an exception to this norm. But in reality, it can be difficult for the bank manager to modify resource allocation in a way that is both effective and efficient. Because human resource management is one of the most crucial but difficult tasks in the banking industry, many bank managers are particularly troubled by the problem of allocating human resources (i.e., hiring) under demand variation.

This paper focuses on the problem of developing staffing policies in the face of a frequent type of customer demand variation encountered by commercial banks seasonal demand variation. The economy and culture are the sources of this seasonality. As a key economic hub, the banking industry is influenced by a variety of economic factors, including seasonality. We have not performed the pure performance simulations and comparison between the solutions since they are redundant.

The centralized solution is of constant complexity while the decentralized solution is at least linear. As such, the performance of the centralized solution will eventually outperform the decentralized solution. However, this does not negate a need for a decentralized solution. While it could be difficult to justify a decentralized SG as a replacement for better performing centralized SG in developed countries with proper energy infrastructure and privacy and security laws; there are extreme environments where a decentralized solution might be preferred (e.g., undeveloped countries, conflict zone).

1.2 PROBLEM DEFINITION

We have not performed the pure performance simulations and comparison between the solutions since they are redundant. The centralized solution is of constant complexity while the decentralized solution is at least linear. As such, the performance of the centralized solution will eventually outperform the decentralized solution. However, this does not negate a need for a decentralized solution. While it could be difficult to justify a decentralized SG as a replacement for better performing centralized SG in developed countries with proper energy infrastructure and privacy and security laws; there are extreme environments where a decentralized solution might be preferred (e.g., undeveloped countries, conflict zone).

CHAPTER-2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

REFERENCE PAPER:1

Project title: The Role of Technology in Improving the Customer Experience in the Banking Sector: A Systematic Mapping Study

Author name: Roberto Vergallo and Luca Mainetti

Year of publish:2022

Information Technology (IT) has revolutionized the way we manage our money. The adoption of innovative technologies in banking scenarios allows to access old and new financial services but in a faster and more secure, comfortable, rewarding and engaging way. The number, the performances and the seamless integration of these innovations is a driver for banks to retain their customers and avoid costly change of hearts. This paper aims to overcome the following gaps: the lack of a comprehensive map of the research made in the field in the past decade; a discussion on the current research trends of top publications and journals is missing; the next research challenges are yet to be identified. To face these limitations, we designed and submitted 7 different queries to pull papers out of 4 popular scientific databases. A selection of the top 20% works allowed us to seek the most performant technologies as well as other promising ones that have not been experimented yet in the field. Main results prove that the combined study of technology and CX in the banking sector is not approached systematically and thus the development of a new specific research line is needed.

REFERENCE PAPER:2

Project Title: Banking on Blockchain: An Evaluation of Innovation Decision Making

Author name: Priya D. Dozier and Troy A. Montgomery

Year of publish:2022

Blockchain technology has received significant attention following the introduction of Bit coin in 2008. In particular, the financial services industry has sought to capitalize on the underlying technology, known as distributed ledger technology, to generate new products, services, and innovative business models. In this article, through an in-depth investigation of the technology innovation evaluation process, we utilize a grounded theory approach to study 12 financial service organizations. This article is unique in that it explores the technology evaluation process concurrently as decision makers reacted to the potential uses, as opposed to a retrospective view after a technology innovation had been adopted. Evidence suggests that, organizations applied a specific process to determine the value of blockchain that consisted of understand, organize, and test, which collectively helped create the proof-of-value model. Surprisingly, we find that financial service organizations tend to view blockchain innovation as a lower priority due to the lack of a clear path to value. Additionally, financial service organizations consistently leverage industry consortiums to link to external knowledge and help with the decision-making process. Our findings have direct implications to both innovation researchers as well as practitioners seeking to evaluate blockchain technology.

REFERENCE PAPER:3

Project Title: A Simple and Secure Reformation-Based Password Scheme

Author name: MUSHTAQ ALI , AMANULLAH BALOCH , ABDUL WAHEED

Year of publish:2021

The electronic applications of financial institutions like banks and insurance companies use either token-based, biometric-based, or knowledge-based password scheme to keep the confidential information of their customers safe from hackers. They also need an extra device like earphones during password entry causing to create a gap for information leakage. Moreover, most of the passwords store passwords' actual content on a server database that causes penetration in the financial institutions' database. In this article, a reformation-based password scheme involving no mental computation and using no extra device is proposed. The proposed scheme works on the password characters' indices, which change dynamically after each login process. It gets the password characters' indices from the end-user and obtains his password characters' indices from the database. Next, the textual passwords are formed from the user-provided indices and those obtained from the database. The textual passwords are then compared, and if found match, then login is succeeded, otherwise failed. Our proposed password scheme's experimental results on the password data set showed better security and usability compared to state-of-art password schemes.

REFERENCE PAPER:4

Project Title: Banking Digitalization: (Re)Thinking Strategies and Trends Using Problem Structuring Method

Author name: João F. C. Rodrigues, Fernando A. F. Ferreira.

Year of publish:2019

Digital transformation is becoming more common in an increasingly digital world, which has contributed to significant changes in business processes around the globe. This article sought to apply problem structuring methods, namely, fuzzy cognitive mapping and system dynamics, to develop a conceptual model that can be applied to analyses of banking industry digitalization. The results are based on two group sessions that were held with a panel of decision makers who deal with banking digitalization processes on a daily basis. The findings were validated by a senior staff member responsible for banking project assessment and development at Caixa Geral de Depósitos S.A.—a Portuguese state-owned banking corporation. The approach adopted in this article is unique because digitalization determinants were collected and analyzed based on the participants' know-how and experience. Static and dynamic analyses produced a deeper, broader understanding of the dynamic cause-and-effect relationships between the identified determinants of banking digitalization, giving rise to a realistic model that can be used to support relevant decision-making processes.

REFERENCE PAPER:5

Project Title: A New Fuzzy Cognitive Map Learning Algorithm for Speech Emotion Recognition

Author name: Wei Zhang, Xueying Zhang, and Ying Sun

Year of publish:2017

Selecting an appropriate recognition method is crucial in speech emotion recognition applications. However, the current methods do not consider the relationship between emotions. Thus, in this study, a speech emotion recognition system based on the

fuzzy cognitive map (FCM) approach is constructed. Moreover, a new FCM learning algorithm for speech emotion recognition is proposed. This algorithm includes the use of the pleasure-arousal-dominance emotion scale to calculate the weights between emotions and certain mathematical derivations to determine the network structure. The proposed algorithm can handle a large number of concepts, whereas a typical FCM can handle only relatively simple networks (maps). Different acoustic features, including fundamental speech features and a new spectral feature, are extracted to evaluate the performance of the proposed method. Three experiments are conducted in this paper, namely, single feature experiment, feature combination experiment, and comparison between the proposed algorithm and typical networks. All experiments are performed on TYUT2.0 and EMO-DB databases. Results of the feature combination experiments show that the recognition rates of the combination features are 10%–20% better than those of single features. The proposed FCM learning algorithm generates 5%–20% performance improvement compared with traditional classification networks

CHAPTER -3

SYSTEM ANALYSIS

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In our existing system, we implement an online banking system for online money transactions. At first Creates standard interface between the clients and all the banks, By using this portal any client who maintains accounts in various banks can directly log on to online Banking System Interface and make any kind of transactions.

3.1.1 DISADVANTAGES

- The banking Transaction log is easy to modified
- It is centralized
- It cannot be reduce the transaction cost.

3.2 PROPOSED SYSTEM

In our proposed system we can develop recommendations for the protection of information in geographically-distributed structures, typical of modern banks, based on the cloud. Each block stores information on addresses, transactions, account balances, and the structure itself is completely decentralized. After creating a new block, all transactions that claim to be included in it are checked by this protocol for legality, and the transactions that violate the balance conditions are discarded. The hash value of the header is called the hash value of the block. The hash value of transactions calculated by the algorithm BVCS, and this hash value is used for block

hashing. In addition it is used to check the integrity of the data and get a unique identifier. Each block contains the previous block hash value. This hash value is generated by using the BVCS Algorithm.

3.2.1 ADVANTAGES

- **Transparency:** all legitimate users have the ability to make entries, while it is impossible to change previously registered data.
- **Reducing transaction costs:** Allows you to independently conduct transactions to affiliates and other services of the structure.
- **Decentralization:** there is no central point of information gathering. Information about specific blocks scattered across all servers of the corporate structure.

3.3 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Economic Feasibility :

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be

justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility:

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

Social Feasibility:

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

3.4 HARDWARE ENVIRONMENT

Processor - Pentium –III

RAM - 4 GB

Hard Disk - 260 GB

Operating System - Windows95/98/2000/XP

3.5 SOFTWARE ENVIRONMENT

Front End	- Java, Jsp
Scripts	- JavaScript.
Server side Script	- Java Server Pages.
Database	- My sql

Front End:

Java:

Java is a set of several computer software and specifications developed by Sun Microsystems, later acquired by Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. While less common, Java applets run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

Writing in the Java programming language is the primary way to produce code that will be deployed as byte code in a Java Virtual Machine (JVM); byte code compilers are also available for other languages, including Ada, JavaScript, Python, and Ruby. In addition, several languages have been designed to run natively on the JVM, including Scala, Clojure and Groovy. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and Objective-C.[11] Java eschews certain low-level constructs such as pointers and has a very simple

memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.

Netbeans:

NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers. The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5. NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

NetBeans IDE 6.0 introduced support for developing IDE modules and rich client applications based on the NetBeans platform, a Java Swing GUI builder (formerly known as "Project Matisse"), improved CVS support, WebLogic 9 and JBoss 4 support, and many editor enhancements. NetBeans 6 is available in official repositories of major Linux distributions. NetBeans IDE 6.5, released in November 2008, extended the existing Java EE features. NetBeans IDE 6.8 is the first IDE to provide complete support of Java EE 6 and the GlassFish Enterprise Server v3.

Back End:

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site (<http://www.mysql.com/>) provides the latest information about MySQL software.

CHAPTER -4

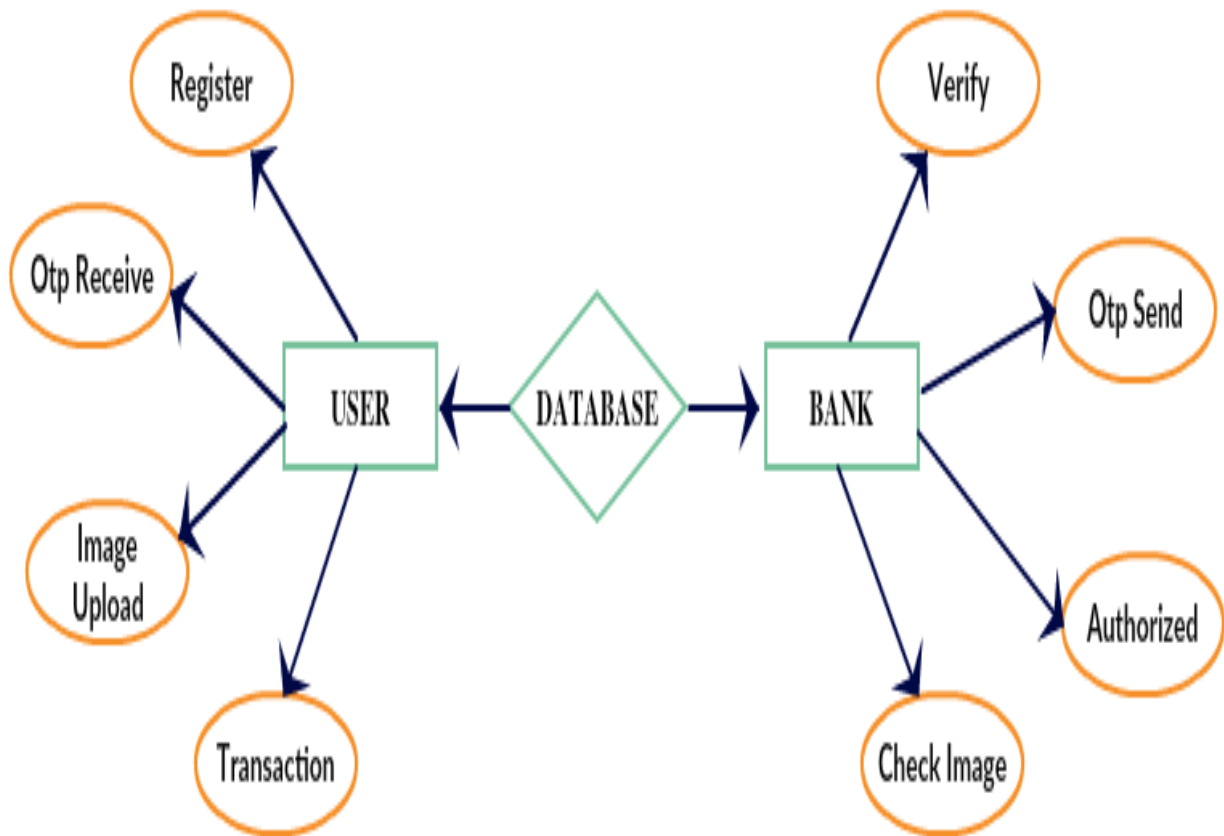
SYSTEM DESIGN

CHAPTER 4

SYSTEM DESIGN

4.1 ER DIAGRAM

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. The ER diagram for this project is the following



4.1.1 ER diagram

4.2 DATA DICTIONERY

A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project. Here are the data tables used in this project

S.no	username	email	Pass_word	con_pass
1	elangovan	elangovan1120308@gmail.com	elango	elango
2	Frank	frank@gmail.com	123	123
3	gopalbk	gopalbk@gmail.com	Gopal123	Gopal123
4	hdfc	hdfc@gmail.com	Hdfc*	Hdfc*

table 4.2.1 Banking Registration

sno	username	Id_d	email_id	Banking_sector
1	Kannan	ID117544	maitoasiva143@gmail.com	Hdfc
2	Yuvaraja	ID113068	yuvarajaprince8@gmail.com	Hdfc
3	Yuvaraja	ID113068	yuvarajaprince8@gmail.com	Hdfc
4	Yuvaraja	ID113068	yuvarajaprince8@gmail.com	Hdfc
5	Yuvaraja	ID113068	yuvarajaprince8@gmail.com	Hdfc

Table 4.2.2 Viewing public user details :

sno	username	email	pass_word	con_pass
1	elango	elangovan20001@gmail.com	elango	elango
2	mathan	rkumathan@gmail.com	elango	elango
3	ram	ram@gmail.com	ram	ram

Table 4.2.3 Viewing public user registration to the bank

4.3 TABLE NORMALIZATION

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

Table Name: Database: Comment:

Columns and Indices Table Options Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
sno	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	<input type="text" value="NULL"/>	
user_name	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
id_d	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
email_id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
BANKING_SECT...	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
add_ress	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
vehicle	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
key_1	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	

Indices Foreign Keys Column Details

PRIMARY

Index Settings

Index Name:

Index Kind:

Index Type:

Index Columns (Use Drag'n'Drop)

sno

Apply Changes Discard Changes Close

4.3.1 view_upload

MySQL Table Editor

Table Name: Database: Comment:

Columns and Indices Table Options Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
sno	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	<input type="text" value="NULL"/>	
username	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
email	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
pass_word	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
con_pass	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
date	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
mobile_no	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
city	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
address	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	

Indices Foreign Keys Column Details

PRIMARY

Index Settings

Index Name:

Index Kind:

Index Type:

Index Columns (Use Drag'n'Drop)

sno

Apply Changes Discard Changes Close

4.3.2 User register

MySQL Table Editor

Table Name: Database: Comment:

Columns and Indices Table Options Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
sno	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	<input type="text" value="NULL"/>	
id_no	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
name	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
email	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
file	LOBLOB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
subject	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
address	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
blood	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
sex	VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	

Indices Foreign Keys Column Details

PRIMARY

Index Settings

Index Name:

Index Kind:

Index Type:

Index Columns (Use Drag'n'Drop)

sno

Apply Changes Discard Changes Close

4.3.3 Upload

MySQL Table Editor

Table Name: Database: Comment:

Columns and Indices Table Options Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
sno	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	<input type="text" value="NULL"/>	
file_id	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
filename	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	
encrypt	LONGBLOB	<input checked="" type="checkbox"/>			<input type="text" value="NULL"/>	
secrete	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	<input type="text" value="NULL"/>	

Indices Foreign Keys Column Details

PRIMARY

Index Settings

Index Name:

Index Kind:

Index Type:

Index Columns (Use Drag'n'Drop)

sno

Apply Changes Discard Changes Close

4.3.4 secrete_file

4.4 DATA FLOW DIAGRAM

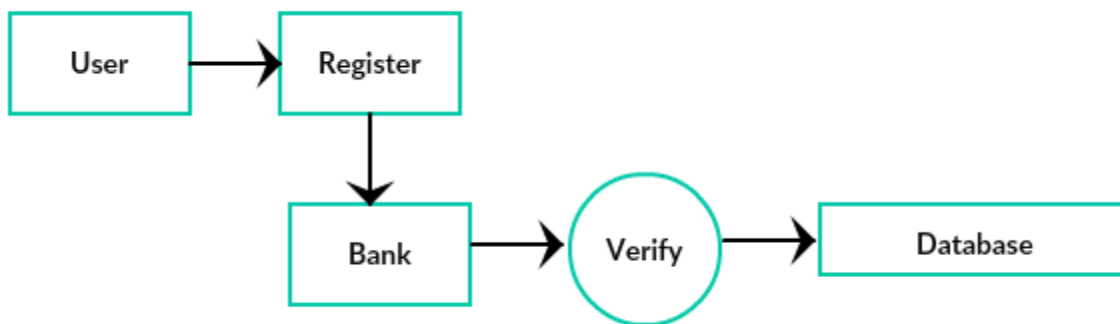
A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed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

Level-0



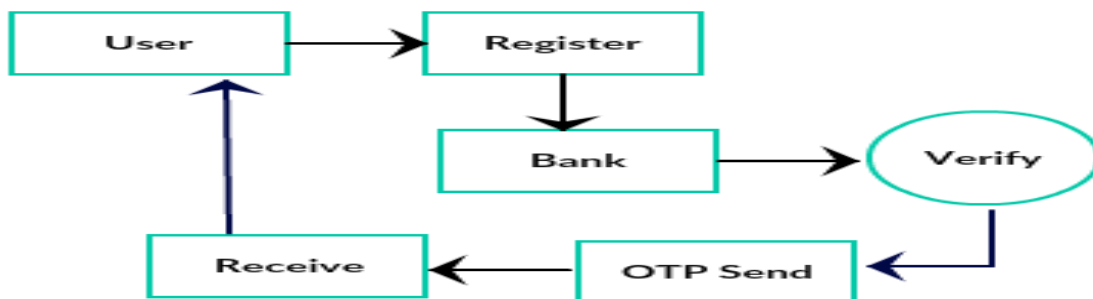
4.4.1 Level-0

Level-1



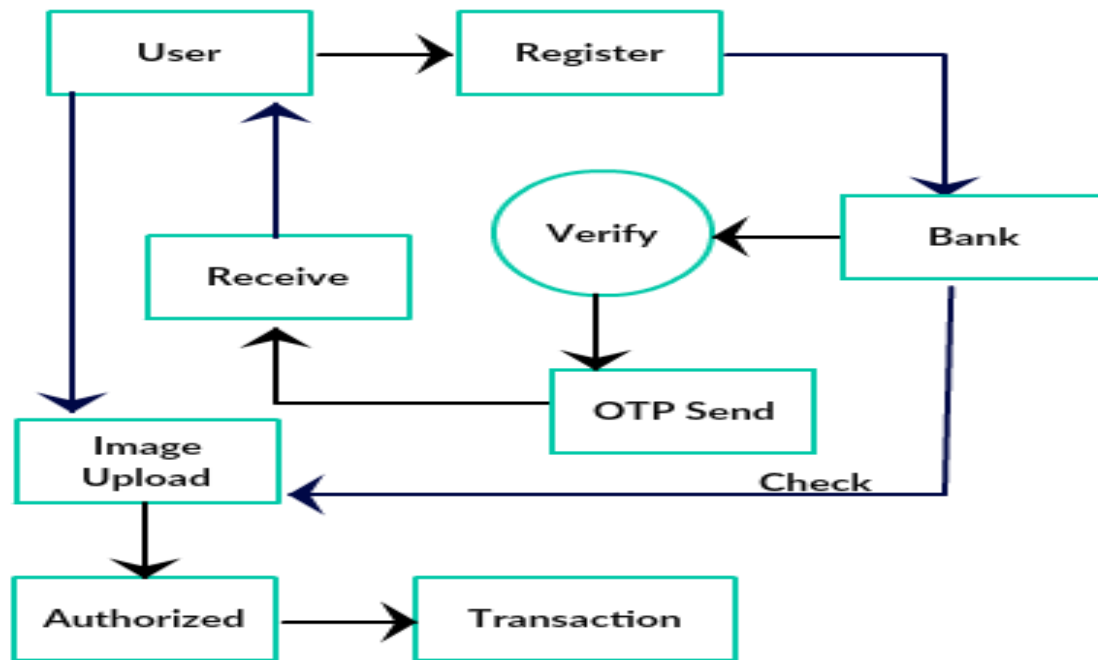
4.4.2 Level-1

Level-2



4.4.3 Level-2

Over All Diagram:

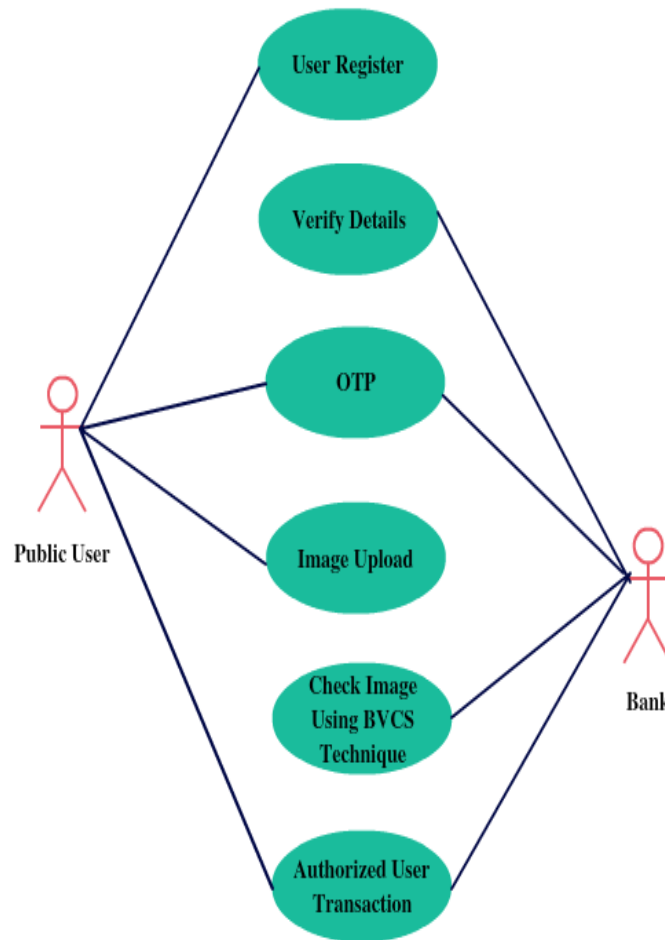


4.4.4 overall diagram

4.5 UML DIAGRAMS

4.5.1 USE CASE DIAGRAMS:

A use case diagram is a type of behavioral diagram created from a Use-case analysis. The purpose of the use case is to present overview of the functionality provided by the system in terms of actors, their goals and any dependencies between those use cases.

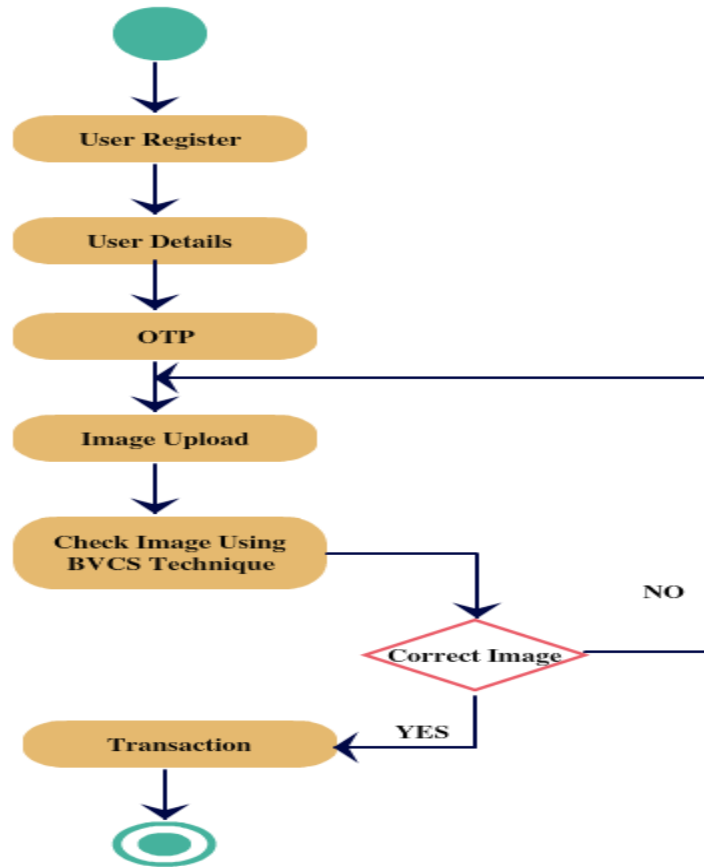


4.5.1 Use Case diagram

4.5.2 ACTIVITY DIAGRAMS

Activity diagrams are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structural development.

The following Activity diagram shows how the optimization of work flows in this project



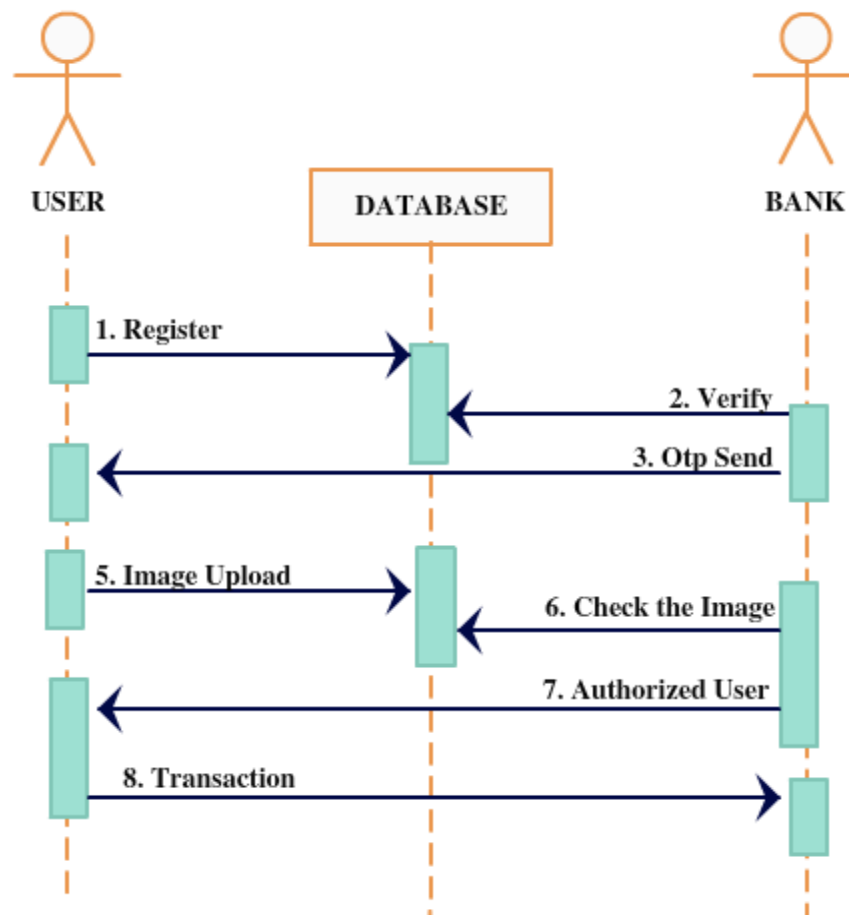
4.5.2 Activity diagram

4.5.3 SEQUENCE DIAGRAM:

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

It is a construct of a message sequence chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams.

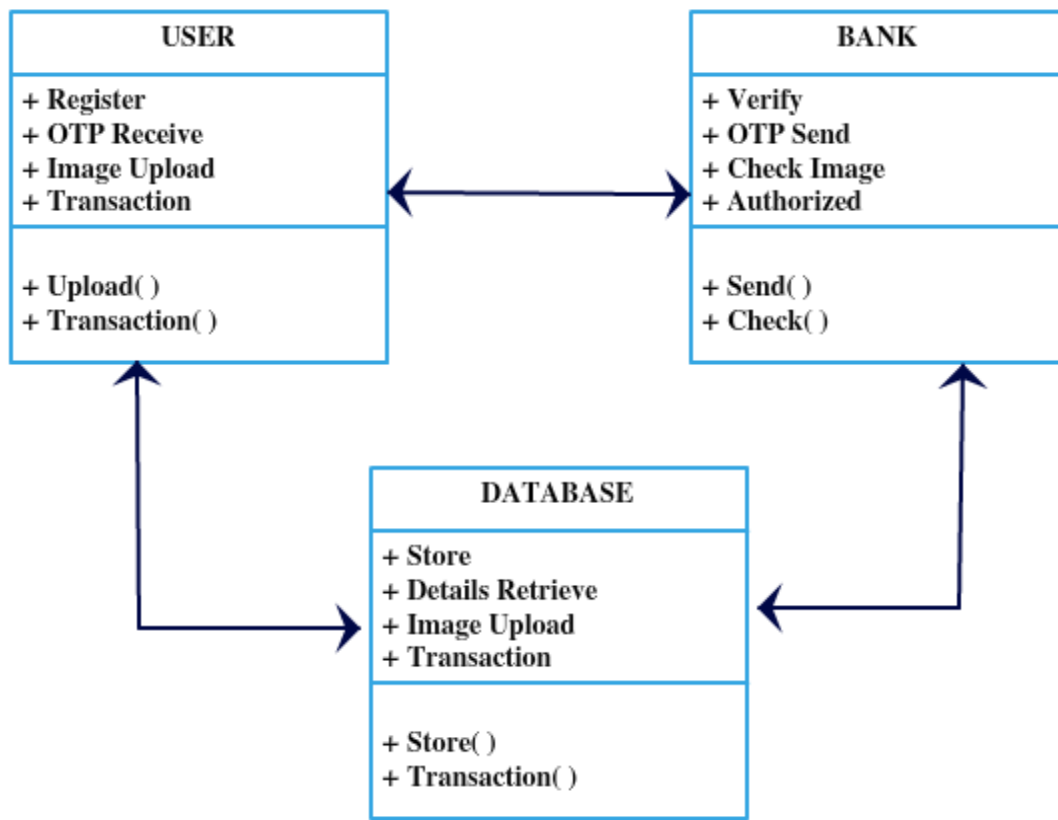
The below diagram shows the sequence flow shows how the optimization of work flows in this project



4.5.3 Sequence diagram

4.5.4 CLASS DIAGRAM

A class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition. Public visibility allows all other classes to view the marked information. Protected visibility allows child classes to access information they inherited from a parent class.



4.5.4 Class Diagram

CHAPTER-5

SYSTEM ARCHITECTURE

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW:

Here the public user creates an account by giving mail id and password. In Bank login page the details should be filled. Admin page has to be filled by the authenticated person who can only view about user details. The bank has to approve the user details to make transactions. When we enter the bank details it sends an image that should be uploaded and gets a key value. The received key value should be entered to make the transactions. After entering the key value and amount the amount gets successfully sent to the recipient.

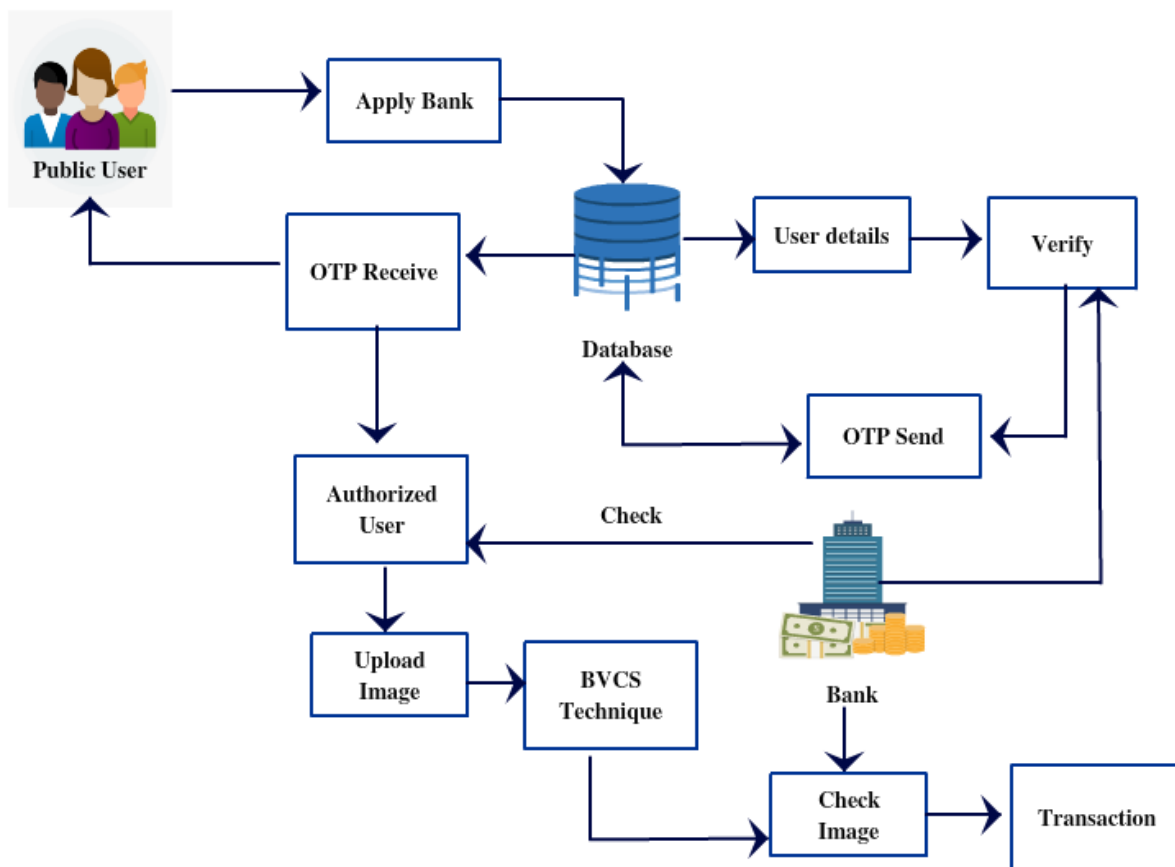


Fig 5.1 Architecture diagram

5.1 MODULE DESIGN SPECIFICATION

Modules:

- Public User registration
- Bank Verification
- BVSC
- Image authenticate (Matching and Validation)

Modules Description

Public User registration:

In this module the user has to register by giving the information about the user in the application, before login. Without a login user can't view the items with any details in the application. A public user can register to this bank application so they can access. A registered user may also refer to an individual who is signed up for a service.

Bank Verification:

In this module, the banker will verify the user details by viewing the user registration detail and the banker send a opt (one time password) to the user, then the user can access the web application on banking sector to transaction.

BVSC (Binocular VCS):

In this module the user will upload the image to authenticate The BVCS technique Will check the image.(BVSC)binocular vision. Vision that incorporates images from two eyes simultaneously. The slight differences between the two images-seen from slightly different positions-make it possible to perceive distances between objects in what is known as depth perception.

Image authenticates (Matching and Validation)

In this module the bank will authenticate the user image to check if the user is authorized or not Image authentication (also called forensic image authentication). if the image is correct then only the user will be able to do the transaction process. If the image is not the same image, then the user will be an unauthorized person so the user cannot do any transaction process.

5.2 ALGORITHMS

BCV-Visual Cryptography:

BCV-Visual Cryptography is a special encryption technique to **hide information in images**, which **divide secret image into multiple layers**. Each layer holds some information. The receiver aligns the layers and the secret information is revealed by human vision without any complex computation. Visual Cryptography is a new technique for securing the visual information like picture, text etc. The basic idea of Visual Cryptography is that images are divided into several parts called shares. These shares are distributed among concerned participants and to decrypt them arrange or stacked to get the image back. At first there are various measures on which performance of visual cryptography schemes depends, such as pixel expansion, visual quality, image quality, contrast, security, quality of shares, size, computational complexity. Visual Cryptography techniques was developed for binary images only but later on it was advanced for colour images also.

The image has been split into two component images. Each component image has a pair of pixels for every pixel in the original image. These pixel pairs are shaded black or white according to the following rule: if the original image pixel was black, the pixel pairs in the component images must be complementary; randomly shade

one ■□, and the other □■. When these complementary pairs are overlapped, they will appear dark gray. On the other hand, if the original image pixel was white, the pixel pairs in the component images must match: both ■□ or both □■. When these matching pairs are overlapped, they will appear light gray.

So, when the two component images are superimposed, the original image appears. However, without the other component, a component image reveals no information about the original image; it is indistinguishable from a random pattern of ■□ / □■ pairs. Moreover, if you have one component image, you can use the shading rules above to produce a counterfeit component image that combines with it to produce any image at all.

HONEY ENCRYPTION:

Honey encryption is a type of data encryption that "produces a ciphertext, which, when decrypted with an incorrect key as guessed by the attacker, presents a plausible-looking yet incorrect plaintext password or encryption key." The two main things in this construct are the implementation of the message space where all the probable values of passwords are placed. The second thing is the Distribution-transforming Encoder that encodes or decodes the message space using the specified functions.

CHAPTER -6

SYSTEM

IMPLEMENTATION

CHAPTER 6

SYSTEM IMPLEMENTATION

Image Encryption.java:

```
<% @page import="javax.swing.JOptionPane"%>
<% @page import="java.text.SimpleDateFormat"%>
<% @page import="java.util.Calendar"%>
<% @page import="java.sql.ResultSet"%>
<% @page import="java.sql.Statement"%>
<% @page import="java.sql.Connection"%>
<% @page import="java.sql.DriverManager"%>
<% @page import="java.io.IOException"%>
<% @page import="javax.imageio.ImageIO"%>
<% @page import="java.io.File"%>
<% @page import="java.awt.Color"%>
<% @page import="java.awt.RenderingHints"%>
<% @page import="java.awt.FontMetrics"%>
<% @page import="java.awt.Font"%>
<% @page import="java.awt.Graphics2D"%>
<% @page import="java.awt.image.BufferedImage"%>
<% @page import="Connection.Vino_Encryption" %>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
```

```

<%
    String nameq=(String)session.getAttribute("name");
    session.setAttribute("name",nameq);
    String names=(String)session.getAttribute("name");
    session.setAttribute("name1",names);
    //String en=(String)session.getAttribute("name1");
    System.out.println(names);
    final String ALPHA_NUM =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    final String nums="1234567890";
    String text = "";
    StringBuffer sb = new StringBuffer(5);
    for (int i = 0; i < 5; i++) {
    int ndx = (int) (Math.random() * ALPHA_NUM.length());
    sb.append(ALPHA_NUM.charAt(ndx));
    }
    text= sb.toString();
    String temp=text;
    StringBuffer sb1 = new StringBuffer(2);
    for (int i = 0; i < 1; i++) {
    int ndx = (int) (Math.random() * nums.length());
    sb1.append(ALPHA_NUM.charAt(ndx));
    }
    String name="image"+sb1.toString();
    //System.out.println(grs.getAlphaNumeric(20));

```

```

    BufferedImage img = new BufferedImage(1, 1,
    BufferedImage.TYPE_INT_ARGB);

    Graphics2D g2d = img.createGraphics();

    Font font = new Font("Arial", Font.PLAIN, 48);

    g2d.setFont(font);

    FontMetrics fm = g2d.getFontMetrics();

    int width = fm.stringWidth(text);

    int height = fm.getHeight();

    g2d.dispose();

    img = new BufferedImage(width, height,
    BufferedImage.TYPE_INT_ARGB);

    g2d = img.createGraphics();

    g2d.setRenderingHint(RenderingHints.KEY_ALPHA_INTERPOLATION,
    RenderingHints.VALUE_ALPHA_INTERPOLATION_QUALITY);

    g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
    RenderingHints.VALUE_ANTIALIAS_ON);

    g2d.setRenderingHint(RenderingHints.KEY_COLOR_RENDERING,
    RenderingHints.VALUE_COLOR_RENDER_QUALITY);

    g2d.setRenderingHint(RenderingHints.KEY_DITHERING,
    RenderingHints.VALUE_DITHER_ENABLE);

    g2d.setRenderingHint(RenderingHints.KEY_FRACTIONALMETRICS,
    RenderingHints.VALUE_FRACTIONALMETRICS_ON);

    g2d.setRenderingHint(RenderingHints.KEY_INTERPOLATION,
    RenderingHints.VALUE_INTERPOLATION_BILINEAR);

    g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
    RenderingHints.VALUE_RENDER_QUALITY);

    g2d.setRenderingHint(RenderingHints.KEY_STROKE_CONTROL,
    RenderingHints.VALUE_STROKE_PURE);

    g2d.setFont(font);

```



```

fm = g2d.getFontMetrics();
g2d.setColor(Color.BLACK);
g2d.drawString(text, 0, fm.getAscent());
g2d.dispose();
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/voting","root","a
dmin");

    Statement st=con.createStatement();
    ResultSet Rs;
    Statement st1=con.createStatement();
    ResultSet Rs1;
    String name1=names+".png";
    String timeStamp = new
SimpleDateFormat("MM/dd/yyyy").format(Calendar.getInstance().getTime());
    System.out.println(timeStamp);

    Rs1=st1.executeQuery("Select * from eledetails where
edate='"+timeStamp+"'");
    if(Rs1.next())
    {
        Rs=st.executeQuery("Select * from voting where username='"+name1+"'");
        if(Rs.next()!=true)
        {
            String path = request.getSession().getServletContext().getRealPath("/");
            // String patt=path.replace("\\build", "");
            String path2=path+"serverImage\\";
            String strPath = path+"serverImage\\"+names+".png";

```

```

        System.out.println(strPath);
        ImageIO.write(img, "png", new File(strPath));

session.setAttribute("fname",name1);
session.setAttribute("keyd", temp);

System.out.println(names);
        st.executeUpdate("insert into filekey values '"+temp+"','"+name1+"'");
        try
        {

        Vino_Encryption nn=new Vino_Encryption(name1,strPath,path2);
response.sendRedirect("keymail.jsp?nam="+names);
        }
        catch(Exception e)
        {
            out.println(e);
        }
        }
        else
        {
            //String sdd1=;

            String msg11="You Already Voting";
                session.setAttribute("msg11",msg11);
                response.sendRedirect("employepage.jsp");

```

```

    }
    }
    else
    {
        // String sdd1="Today Not Election Date";

        String msg11="Today Not Election Date";
        session.setAttribute("msg11",msg11);
        response.sendRedirect("employepage.jsp?alr1=");
    }

}
catch (IOException ex) {
    ex.printStackTrace();
}
%>

```

</head>

<body>

</body></html>

Decryptframe.java:

```
package Connection;
```

```
import java.awt.Component;
```

```
import java.awt.Container;
```

```
import java.awt.Dimension;
```

```
import java.awt.FocusTraversalPolicy;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.filechooser.FileFilter;
```

```
public class DecryptFrame extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JPanel pnlAll = new JPanel();
    private JPanel pnlKeyFile = new JPanel();
    private JPanel pnlEncFile = new JPanel();
```

```

private JPanel pnlScrolls = new JPanel();
private JPanel pnlScrollOverlay = new JPanel();
private JPanel pnlScrollClean = new JPanel();

private JLabel lblDescr = new JLabel("<html>Enter a key file and an
encrypted image below to decrypt it. You could also decrypt it by printing the key
and the encrypted image on transparent paper and overlaying them
manually.</html>");

private JLabel lblOverlay = new JLabel(new ImageIcon(),
JLabel.CENTER);

private JLabel lblClean = new JLabel(new ImageIcon(), JLabel.CENTER);
private JTextField tfKey = new JTextField();
private JTextField tfEncr = new JTextField();
private JButton btnSelectKey = new JButton("Select keyfile");
private JButton btnSelectEncr = new JButton("Select encrypted image");
private JButton btnDecrypt = new JButton("Decrypt");
private JButton btnSaveOverlay = new JButton("Save overlayed image to
file");
private JButton btnSaveClean = new JButton("Save decrypted image to
file");

private JScrollPane scrOverlay = new JScrollPane(lblOverlay);
private JScrollPane scrClean = new JScrollPane(lblClean);

private JFileChooser fileChooser = new JFileChooser();
private BufferedImage imgOverlay = null;
private BufferedImage imgClean = null;
File fKeyFile = null;
File fEncrFile = null;

```

```

public DecryptFrame(JFrame parent) {
    // size
    tfKey.setMaximumSize(new
Dimension(tfKey.getMaximumSize().width, tfKey.getPreferredSize().height));
    tfEncr.setMaximumSize(new
Dimension(tfEncr.getMaximumSize().width, tfEncr.getPreferredSize().height));
    int iButMaxWidth = (btnSelectKey.getPreferredSize().width >
btnSelectEncr.getPreferredSize().width) ?

    btnSelectKey.getPreferredSize().width :
btnSelectEncr.getPreferredSize().width;
    btnSelectKey.setPreferredSize(new Dimension(iButMaxWidth,
btnSelectKey.getPreferredSize().height));
    btnSelectEncr.setPreferredSize(new Dimension(iButMaxWidth,
btnSelectEncr.getPreferredSize().height));


    // orientation
    lblDescr.setAlignmentX(LEFT_ALIGNMENT);
    pnlKeyFile.setAlignmentX(LEFT_ALIGNMENT);
    pnlEncFile.setAlignmentX(LEFT_ALIGNMENT);
    pnlScrolls.setAlignmentX(LEFT_ALIGNMENT);


    // action listener
    btnSelectKey.addActionListener(this);
    btnSelectEncr.addActionListener(this);
    btnDecrypt.addActionListener(this);
    btnSaveOverlay.addActionListener(this);
    btnSaveClean.addActionListener(this);

```

```

tfKey.setEditable(false);
tfEncr.setEditable(false);
btnSaveOverlay.setEnabled(false);
btnSaveClean.setEnabled(false);

fileChooser.setFileFilter(new FileFilter() {
    public boolean accept(File arg0) {
        if (arg0.isDirectory()) return true;
        if (arg0.getName().endsWith(".png")) return true;
        return false;
    }

    public String getDescription() {
        return "Image (*.png)";
    }
});

```

```

pnlKeyFile.setLayout(new BoxLayout(pnlKeyFile,
BoxLayout.X_AXIS));
pnlKeyFile.add(tfKey);
pnlKeyFile.add(Box.createRigidArea(new Dimension(10, 0)));
pnlKeyFile.add(btnSelectKey);

```

```

pnlEncFile.setLayout(new BoxLayout(pnlEncFile,
BoxLayout.X_AXIS));
pnlEncFile.add(tfEncr);
pnlEncFile.add(Box.createRigidArea(new Dimension(10, 0)));

```

```

        pnlEncFile.add(btnSelectEncr);

        pnlScrollOverlay.setLayout(new BorderLayout(pnlScrollOverlay,
BoxLayout.Y_AXIS));
        pnlScrollOverlay.add(scrOverlay);
        pnlScrollOverlay.add(Box.createRigidArea(new Dimension(0, 10)));
        pnlScrollOverlay.add(btnSaveOverlay);

        pnlScrollClean.setLayout(new BorderLayout(pnlScrollClean,
BoxLayout.Y_AXIS));
        pnlScrollClean.add(scrClean);
        pnlScrollClean.add(Box.createRigidArea(new Dimension(0, 10)));
        pnlScrollClean.add(btnSaveClean);

        pnlScrolls.setLayout(new BorderLayout(pnlScrolls,
BoxLayout.X_AXIS));
        pnlScrolls.add(pnlScrollOverlay);
        pnlScrolls.add(Box.createRigidArea(new Dimension(10, 0)));
        pnlScrolls.add(pnlScrollClean);

        pnlAll.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
        pnlAll.setLayout(new BorderLayout(pnlAll, BorderLayout.Y_AXIS));
        pnlAll.add(lblDescr);
        pnlAll.add(pnlKeyFile);
        pnlAll.add(pnlEncFile);
        pnlAll.add(btnDecrypt);
        pnlAll.add(Box.createVerticalStrut(10));
        pnlAll.add(pnlScrolls);

```



```

        setFocusTraversalPolicy(new MyFocusTraversalPolicy());

        add(pnlAll);
        setSize(500, 500);
        setMinimumSize(new Dimension(384, 253));
        setLocationRelativeTo(parent);
        setTitle("Visual Cryptography - Decrypt Image");
        setVisible(true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals(btnDecrypt.getText())) {
            if (fKeyFile == null || !fKeyFile.exists() || fEncrFile == null ||
!fEncrFile.exists()) {
                JOptionPane.showMessageDialog(this, "File not found",
"ERROR", JOptionPane.ERROR_MESSAGE);
                return;
            }

            BufferedImage imgKey =
Crypting.loadAndCheckEncrFile(fKeyFile);
            if (imgKey == null) {
                JOptionPane.showMessageDialog(this,
fKeyFile.getName() + " is not a valid key file", "ERROR",
JOptionPane.ERROR_MESSAGE);
            }
        }
    }

```

```

        return;
    }

    BufferedImage imgEnc =
Crypting.loadAndCheckEncrFile(fEncrFile);

    if (imgEnc == null) {

        JOptionPane.showMessageDialog(this,
fEncrFile.getName() + " is not an encrypted image", "ERROR",
JOptionPane.ERROR_MESSAGE);

        return;
    }

    imgOverlay = Crypting.overlayImages(imgKey, imgEnc);
    if (imgOverlay == null) {

        JOptionPane.showMessageDialog(this, "Decryption
failed - key and encrypted image not the same size?", "ERROR",
JOptionPane.ERROR_MESSAGE);

        return;
    }

    imgClean = Crypting.decryptImage(imgOverlay);
    if (imgClean == null) {

        JOptionPane.showMessageDialog(this, "Decryption
failed - key and encrypted image not the same size?", "ERROR",
JOptionPane.ERROR_MESSAGE);

        return;
    }

    lblOverlay.setIcon(new ImageIcon(imgOverlay));
    lblClean.setIcon(new ImageIcon(imgClean));

```

```

        btnSaveOverlay.setEnabled(true);
        btnSaveClean.setEnabled(true);
    } else if (e.getActionCommand().equals(btnSaveOverlay.getText())) {
        if (imgOverlay == null) return;
        fileChooser.setSelectedFile(new File(""));
        fileChooser.setDialogTitle("Save overlay as..");
        if (fileChooser.showSaveDialog(this) ==
JFileChooser.APPROVE_OPTION) {
            File f = fileChooser.getSelectedFile();
            if (!f.toString().endsWith(".png")) {
                f = new File(f.toString() + ".png");
            }
            try {
                ImageIO.write(imgOverlay, "png", f);
            } catch (IOException e1) {
                JOptionPane.showMessageDialog(this, "Could not
Save file because: " + e1.getLocalizedMessage(), "ERROR",
JOptionPane.ERROR_MESSAGE);
            }
        }
    } else if (e.getActionCommand().equals(btnSaveClean.getText())) {
        if (imgClean == null) return;
        fileChooser.setSelectedFile(new File(""));
        fileChooser.setDialogTitle("Save decrypted image as..");
        if (fileChooser.showSaveDialog(this) ==
JFileChooser.APPROVE_OPTION) {
            File f = fileChooser.getSelectedFile();

```

```

        if (!f.toString().endsWith(".png")) {
            f = new File(f.toString() + ".png");
        }
        try {
            ImageIO.write(imgClean, "png", f);
        } catch (IOException e1) {
            JOptionPane.showMessageDialog(this, "Could not
Save file because: " + e1.getLocalizedMessage(), "ERROR",
JOptionPane.ERROR_MESSAGE);
        }
    }
} else if (e.getActionCommand().equals(btnSelectKey.getText())) {
    fileChooser.setDialogTitle("Open keyfile..");
    if (fileChooser.showOpenDialog(this) ==
JFileChooser.APPROVE_OPTION) {
        if (!fileChooser.getSelectedFile().exists()) return;
        if (!fileChooser.getSelectedFile().getName().endsWith(".png"))
return;

        fKeyFile = fileChooser.getSelectedFile();
        tfKey.setText(fKeyFile.toString());
    }
} else if (e.getActionCommand().equals(btnSelectEncr.getText())) {
    fileChooser.setDialogTitle("Open source image..");
    if (fileChooser.showOpenDialog(this) ==
JFileChooser.APPROVE_OPTION) {
        if (!fileChooser.getSelectedFile().exists()) return;
        if (!fileChooser.getSelectedFile().getName().endsWith(".png"))
return;

        fEncrFile = fileChooser.getSelectedFile();

```

```

        tfEncr.setText(fEncrFile.toString());
    }
}
}

```

```

class MyFocusTraversalPolicy extends FocusTraversalPolicy {
    public Component getComponentAfter(Container focusCycleRoot,
Component aComponent) {
        if(aComponent.equals(btnSelectKey)) return btnSelectEncr;
        else if(aComponent.equals(btnSelectEncr)) return btnDecrypt;
        else if(aComponent.equals(btnDecrypt)) {
            if (btnSaveOverlay.isEnabled()) return btnSaveOverlay;
            if (btnSaveClean.isEnabled()) return btnSaveClean;
            return btnSelectKey;
        }
        else if(aComponent.equals(btnSaveOverlay) &&
btnSaveClean.isEnabled()) return btnSaveClean;
        return btnSelectKey;
    }
}

```

```

    public Component getComponentBefore(Container focusCycleRoot,
Component aComponent) {
        if(aComponent.equals(btnSelectKey)) {
            if (btnSaveClean.isEnabled()) return btnSaveClean;
            if (btnSaveOverlay.isEnabled()) return btnSaveOverlay;
            return btnDecrypt;
        }
        else if(aComponent.equals(btnSelectEncr)) return btnSelectKey;
    }
}

```

```

        else if(aComponent.equals(btnDecrypt)) return btnSelectEncr;
        else if(aComponent.equals(btnSaveOverlay)) return btnDecrypt;
        else if(aComponent.equals(btnSaveClean) &&
btnSaveOverlay.isEnabled()) return btnSaveOverlay;
        return btnDecrypt;
    }

    public Component getDefaultComponent(Container focusCycleRoot) {
        return btnSelectKey;
    }

    public Component getFirstComponent(Container focusCycleRoot) {
        return btnSelectKey;
    }

    public Component getLastComponent(Container focusCycleRoot) {
        return btnSaveClean;
    }
}

```

EncryptFram.java:

```

package Connection;

import java.awt.Component;
import java.awt.Container;

```

```
import java.awt.Dimension;
import java.awt.FocusTraversalPolicy;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.filechooser.FileFilter;
```

```
public class EncryptFrame extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JPanel pnlAll = new JPanel();
    private JPanel pnlKeyFile = new JPanel();
```

```

private JPanel pnlImgFile = new JPanel();

private JLabel lblDescr = new JLabel("<html>Add a valid key file and a
valid source image (png, jpg or gif, will be converted to b/w, not larger than half
the keyfile) below to encrypt the source image.</html>");

private JLabel lblImg = new JLabel(new ImageIcon(), JLabel.CENTER);
private JTextField tfKey = new JTextField();
private JTextField tfImage = new JTextField();
private JButton btnSelectKey = new JButton("Select keyfile");
private JButton btnSelectImage = new JButton("Select image");
private JButton btnEncrypt = new JButton("Encrypt");
private JButton btnSave = new JButton("Save encrypted image to file");
private JScrollPane scrImage = new JScrollPane(lblImg);

private JFileChooser fileChooser = new JFileChooser();
private BufferedImage imgEncr = null;
File fKeyFile = null;
File fSrcFile = null;

public EncryptFrame(JFrame parent) {
    // size
    tfKey.setMaximumSize(new
Dimension(tfKey.getMaximumSize().width, tfKey.getPreferredSize().height));
    tfImage.setMaximumSize(new
Dimension(tfImage.getMaximumSize().width, tfImage.getPreferredSize().height));
    int iButMaxWidth = (btnSelectKey.getPreferredSize().width >
btnSelectImage.getPreferredSize().width) ?

```



```
        btnSelectKey.getPreferredSize().width :
btnSelectImage.getPreferredSize().width;

        btnSelectKey.setPreferredSize(new Dimension(iButMaxWidth,
btnSelectKey.getPreferredSize().height));

        btnSelectImage.setPreferredSize(new Dimension(iButMaxWidth,
btnSelectImage.getPreferredSize().height));
```

```
// orientation
```

```
lblDescr.setAlignmentX(LEFT_ALIGNMENT);
pnlKeyFile.setAlignmentX(LEFT_ALIGNMENT);
pnlImgFile.setAlignmentX(LEFT_ALIGNMENT);
scrImage.setAlignmentX(LEFT_ALIGNMENT);
btnSave.setAlignmentX(LEFT_ALIGNMENT);
```

```
// action listener
```

```
btnSelectKey.addActionListener(this);
btnSelectImage.addActionListener(this);
btnEncrypt.addActionListener(this);
btnSave.addActionListener(this);
```

```
tfKey.setEditable(false);
tfImage.setEditable(false);
btnSave.setEnabled(false);
```

```
fileChooser.setFileFilter(new FileFilter() {
    public boolean accept(File arg0) {
```

```

        if (arg0.isDirectory()) return true;
        if (arg0.getName().toLowerCase().endsWith(".png"))
return true;
        if (arg0.getName().toLowerCase().endsWith(".jpg"))
return true;
        if (arg0.getName().toLowerCase().endsWith(".gif"))
return true;
        return false;
    }

    public String getDescription() {
        return "Image";
    }
});

```

```

        pnlKeyFile.setLayout(new BoxLayout(pnlKeyFile,
BoxLayout.X_AXIS));
        pnlKeyFile.add(tfKey);
        pnlKeyFile.add(Box.createRigidArea(new Dimension(10, 0)));
        pnlKeyFile.add(btnSelectKey);

```

```

        pnlImgFile.setLayout(new BoxLayout(pnlImgFile,
BoxLayout.X_AXIS));
        pnlImgFile.add(tfImage);
        pnlImgFile.add(Box.createRigidArea(new Dimension(10, 0)));
        pnlImgFile.add(btnSelectImage);

```

```

        pnlAll.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
        pnlAll.setLayout(new BoxLayout(pnlAll, BoxLayout.Y_AXIS));

```

```

        pnlAll.add(lblDescr);
        pnlAll.add(pnlKeyFile);
        pnlAll.add(pnlImgFile);
        pnlAll.add(btnEncrypt);
        pnlAll.add(Box.createVerticalStrut(10));
        pnlAll.add(scrImage);
        pnlAll.add(btnSave);

        setFocusTraversalPolicy(new MyFocusTraversalPolicy());

        add(pnlAll);
        setSize(500, 500);
        setMinimumSize(new Dimension(384, 253));
        setLocationRelativeTo(parent);
        setTitle("Visual Cryptography - Encrypt Image");
        setVisible(true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals(btnEncrypt.getText())) {
            if (fKeyFile == null || !fKeyFile.exists() || fSrcFile == null ||
!fSrcFile.exists()) {
                JOptionPane.showMessageDialog(this, "File not found",
"ERROR", JOptionPane.ERROR_MESSAGE);
                return;
            }
        }
    }

```

```

        }

        BufferedImage imgKey =
Crypting.loadAndCheckEncrFile(fKeyFile);

        if (imgKey == null) {

            JOptionPane.showMessageDialog(this,
fKeyFile.getName() + " is not a valid key file", "ERROR",
JOptionPane.ERROR_MESSAGE);

            return;

        }

        BufferedImage imgSrc =
Crypting.loadAndCheckSource(fSrcFile, imgKey.getWidth() / 2,
imgKey.getHeight() / 2, true);

        if (imgSrc == null) {

            JOptionPane.showMessageDialog(this,
fSrcFile.getName() + " is not fit for encryption", "ERROR",
JOptionPane.ERROR_MESSAGE);

            return;

        }

        imgEncr = Crypting.encryptImage(imgKey, imgSrc);

        if (imgSrc == null) {

            JOptionPane.showMessageDialog(this, "Could not
encrypt file. You should never see this :", "ERROR",
JOptionPane.ERROR_MESSAGE);

            return;

        }

        lblImg.setIcon(new ImageIcon(imgEncr));

        btnSave.setEnabled(true);

    } else if (e.getActionCommand().equals(btnSave.getText())) {

```

```

        if (imgEncr == null) return;
        fileChooser.setSelectedFile(new File(""));
        fileChooser.setDialogTitle("Save as..");
        if (fileChooser.showSaveDialog(this) ==
JFileChooser.APPROVE_OPTION) {
            File f = fileChooser.getSelectedFile();
            if (!f.toString().endsWith(".png")) {
                f = new File(f.toString() + ".png");
            }
            try {
                ImageIO.write(imgEncr, "png", f);
            } catch (IOException e1) {
                JOptionPane.showMessageDialog(this, "Could not
Save file because: " + e1.getLocalizedMessage(), "ERROR",
JOptionPane.ERROR_MESSAGE);
            }
        }
    } else if (e.getActionCommand().equals(btnSelectKey.getText())) {
        fileChooser.setDialogTitle("Open keyfile..");
        if (fileChooser.showOpenDialog(this) ==
JFileChooser.APPROVE_OPTION) {
            if (!fileChooser.getSelectedFile().exists()) return;
            if (!fileChooser.getSelectedFile().getName().endsWith(".png"))
return;

            fKeyFile = fileChooser.getSelectedFile();
            tfKey.setText(fKeyFile.toString());
        }
    } else if (e.getActionCommand().equals(btnSelectImage.getText())) {

```

```

        fileChooser.setDialogTitle("Open source image..");
        if (fileChooser.showOpenDialog(this) ==
JFileChooser.APPROVE_OPTION) {
            if (!fileChooser.getSelectedFile().exists()) return;
            //if
(!fileChooser.getSelectedFile().getName().endsWith(".png")) return;
            fSrcFile = fileChooser.getSelectedFile();
            tfImage.setText(fSrcFile.toString());
        }
    }
}

```

```

class MyFocusTraversalPolicy extends FocusTraversalPolicy {
    public Component getComponentAfter(Container focusCycleRoot,
Component aComponent) {
        if(aComponent.equals(btnSelectKey)) return btnSelectImage;
        else if(aComponent.equals(btnSelectImage)) return btnEncrypt;
        else if(aComponent.equals(btnEncrypt) && btnSave.isEnabled())
return btnSave;
        return btnSelectKey;
    }
}

```

```

    public Component getComponentBefore(Container focusCycleRoot,
Component aComponent) {
        if(aComponent.equals(btnSelectKey) && btnSave.isEnabled()) return
btnSave;
        else if(aComponent.equals(btnSelectImage)) return btnSelectKey;
        else if(aComponent.equals(btnEncrypt)) return btnSelectImage;
        return btnEncrypt;
    }
}

```

```

    }

    public Component getDefaultComponent(Container focusCycleRoot) {
        return btnSelectKey;
    }

    public Component getFirstComponent(Container focusCycleRoot) {
        return btnSelectKey;
    }

    public Component getLastComponent(Container focusCycleRoot) {
        return btnSave;}}}}

```

Key Generation.java:

```

package Connection;

import java.awt.Component;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.FocusTraversalPolicy;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.text.NumberFormat;
import javax.imageio.ImageIO;
import javax.swing.BorderFactory;

```

```

import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFormattedTextField;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.filechooser.FileFilter;

```

```

public class KeyGenFrame extends JFrame implements ActionListener {
    private static final long serialVersionUID = 1L;
    private JPanel pnlAll = new JPanel();
    private JPanel pnlRes = new JPanel();

    private JLabel lblDescr = new JLabel("<html>The entered resolution below  
is the largest resolution an image to be encrypted with this key can have " +
        "(The generated key will be twice as large).</html>");
    private JLabel lblWidth = new JLabel("Width:");
    private JLabel lblHeight = new JLabel("Height:");
    private JLabel lblImg = new JLabel(new ImageIcon(), JLabel.CENTER);
    private JFormattedTextField tfWidth = new
JFormattedTextField(NumberFormat.getIntegerInstance());
    private JFormattedTextField tfHeight = new
JFormattedTextField(NumberFormat.getIntegerInstance());

```



```

private JButton btnGenerate = new JButton("Generate Key");
private JButton btnSave = new JButton("Save key to file");
private JScrollPane scrImage = new JScrollPane(lblImg);

private BufferedImage imgKey = null;
JFileChooser fileChooser = new JFileChooser();

public KeyGenFrame(JFrame parent) {
    // size
    tfWidth.setMaximumSize(new
Dimension(tfWidth.getMaximumSize().width, tfWidth.getPreferredSize().height));
    tfHeight.setMaximumSize(new
Dimension(tfWidth.getMaximumSize().width, tfWidth.getPreferredSize().height));

    // orientation
    lblDescr.setAlignmentX(LEFT_ALIGNMENT);
    pnlRes.setAlignmentX(LEFT_ALIGNMENT);
    scrImage.setAlignmentX(LEFT_ALIGNMENT);
    btnSave.setAlignmentX(LEFT_ALIGNMENT);

    // action listener
    btnGenerate.addActionListener(this);
    btnSave.addActionListener(this);
    tfWidth.addActionListener(this);
    tfHeight.addActionListener(this);

    // default value
    tfWidth.setText("200");

```

```

tfHeight.setText("200");
btnSave.setEnabled(false);

fileChooser.setDialogTitle("Save as..");
fileChooser.setFileFilter(new FileFilter() {
    public boolean accept(File arg0) {
        if (arg0.isDirectory()) return true;
        if (arg0.getName().endsWith(".png")) return true;
        return false;
    }

    public String getDescription() {
        return "Image (*.png)";
    }
});

pnlRes.setLayout(new BoxLayout(pnlRes, BoxLayout.X_AXIS));
pnlRes.add(lblWidth);
pnlRes.add(Box.createRigidArea(new Dimension(10, 0)));
pnlRes.add(tfWidth);
pnlRes.add(Box.createRigidArea(new Dimension(10, 0)));
pnlRes.add(lblHeight);
pnlRes.add(Box.createRigidArea(new Dimension(10, 0)));
pnlRes.add(tfHeight);
pnlRes.add(Box.createRigidArea(new Dimension(10, 0)));
pnlRes.add(btnGenerate);

```

```

        pnlAll.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
        pnlAll.setLayout(new BoxLayout(pnlAll, BoxLayout.Y_AXIS));
        pnlAll.add(lblDescr);
        pnlAll.add(pnlRes);
        pnlAll.add(Box.createVerticalStrut(10));
        pnlAll.add(scrImage);
        pnlAll.add(btnSave);

        setFocusTraversalPolicy(new MyFocusTraversalPolicy());

        add(pnlAll);
        setSize(500, 500);
        setMinimumSize(new Dimension(384, 253));
        setLocationRelativeTo(parent);
        setTitle("Visual Cryptography - Generate Key");
        setVisible(true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    }

```

```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals(btnGenerate.getText())) {
        imgKey =
Crypting.generateKey(Integer.parseInt(tfWidth.getText()),
Integer.parseInt(tfHeight.getText()));
        lblImg.setIcon(new ImageIcon(imgKey));
        btnSave.setEnabled(true);
    }
}

```

```

    } else if (e.getActionCommand().equals(btnSave.getText())) {
        if (imgKey == null) return;
        fileChooser.setSelectedFile(new File(""));
        if (fileChooser.showSaveDialog(this) ==
JFileChooser.APPROVE_OPTION) {
            File f = fileChooser.getSelectedFile();
            if (!f.toString().endsWith(".png")) {
                f = new File(f.toString() + ".png");
            }
            try {
                ImageIO.write(imgKey, "png", f);
            } catch (IOException e1) {
                JOptionPane.showMessageDialog(this, "Could not
Save file because: " + e1.getLocalizedMessage(), "ERROR",
JOptionPane.ERROR_MESSAGE);
            }
        }
    } else {
        // tfWidth or tfHeight
        btnGenerate.doClick();
    }
}

```

```

class MyFocusTraversalPolicy extends FocusTraversalPolicy {
    public Component getComponentAfter(Container focusCycleRoot,
Component aComponent) {
        if(aComponent.equals(tfWidth)) return tfHeight;
        else if(aComponent.equals(tfHeight)) return btnGenerate;
    }
}

```

```

        else if(aComponent.equals(btnGenerate) && btnSave.isEnabled())
return btnSave;
        return tfWidth;
    }

```

```

    public Component getComponentBefore(Container focusCycleRoot,
Component aComponent) {
        if(aComponent.equals(tfWidth) && btnSave.isEnabled()) return
btnSave;
        else if(aComponent.equals(tfHeight)) return tfWidth;
        else if(aComponent.equals(btnGenerate)) return tfHeight;
        return btnGenerate;
    }

```

```

    public Component getDefaultComponent(Container focusCycleRoot) {
        return tfWidth;
    }

```

```

    public Component getFirstComponent(Container focusCycleRoot) {
        return tfWidth;
    }

```

```

    public Component getLastComponent(Container focusCycleRoot) {
        return btnSave;
    }
}

```

```

}

```

Encryption.java:

```
package Connection;

import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.net.URL;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.JOptionPane;

/**
 *
 *
 */

public class Vino_Encryption {
    // private JFileChooser fileChooser = new JFileChooser();
    private BufferedImage imgKey = null;
    private BufferedImage imgEnc = null;
    File fSrcFile = null;

    public Vino_Encryption(String name,String path,String opath) throws
    IOException {

        // URL location =
        Vino_Encryption.class.getProtectionDomain().getCodeSource().getLocation();
        //System.out.println(location.getFile());
    }
}
```

```

//String path = request.getSession().getServletContext().getRealPath("/");
// String patt=path.replace("\\build", "");

// String strPath = path+"\\serverImage\\"+names+".png";
//name= this.getClass().getResource(name).toString();
System.out.println(path);
fSrcFile = new File(path);

System.out.println("D://" + name);
//tfImage.setText(fSrcFile.toString());
if (fSrcFile == null || !fSrcFile.exists()) {
    System.out.println( "File not found ERROR");

}

BufferedImage imgSrc =
Crypting.loadAndCheckSource(fSrcFile, 0, 0, false);
if (imgSrc == null) {
    System.out.println( fSrcFile.getName() + " is not fit for
encryption");

}

imgKey = Crypting.generateKey(imgSrc.getWidth(),
imgSrc.getHeight());
imgEnc = Crypting.encryptImage(imgKey, imgSrc);

if (imgKey == null || imgEnc == null) {
    System.out.println( "Error while encrypting ");
}

```

```

}

    if (imgKey == null) return;
    File f = new File(opath+"\\k"+name);
    System.out.println(opath+"\\k"+name);
    if (!f.toString().endsWith(".png")) {
        f = new File(f.toString() + ".png");
    }
    try {
        ImageIO.write(imgKey, "png", f);
    } catch (IOException e1) {
        System.out.println( "Could not Save file because: " +
e1.getLocalizedMessage());
    }
    File f1 = new File(opath+"\\s"+name);
    System.out.println(opath+"\\s"+name);
    try {
        ImageIO.write(imgEnc, "png", f1);
        System.out.println( "Successfully Save");

    } catch (IOException e1) {
        System.out.println( "Could not Save file because: "
+ e1.getLocalizedMessage());
    }
} }

```


Decryption.Java:

```
package Connection;

import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;
import javax.swing.JOptionPane;

*/

public class vino_decryption {
    File fKeyFile=null;
    File fEncrFile=null;
    private BufferedImage imgOverlay = null;
    private BufferedImage imgClean = null;
    private final Object request=null;
    public int vino_decryption1 (String kname,String sname,String names,String
path2)
    {
        try
        {
            System.out.println("Key File"+kname);
            fKeyFile=new File(kname);
            System.out.println(kname);
            System.out.println("Source File"+sname);
            fEncrFile=new File(sname);
            System.out.println(sname);
            if (fKeyFile == null || !fKeyFile.exists() || fEncrFile == null ||
!fEncrFile.exists()) {
```

```

        System.out.println( "File not found ERROR");
        return 1;

    }

    BufferedImage imgKey =
Crypting.loadAndCheckEncrFile(fKeyFile);
    if (imgKey == null) {
        System.out.println( fKeyFile.getName() + " is not a valid
key file");
        return 1;

    }

    BufferedImage imgEnc =
Crypting.loadAndCheckEncrFile(fEncrFile);
    if (imgEnc == null) {
        System.out.println( fEncrFile.getName() + " is not an
encrypted image");
        return 1;

    }

    imgOverlay = Crypting.overlayImages(imgKey, imgEnc);
    if (imgOverlay == null) {
        System.out.println("Decryption failed - key and
encrypted image not the same size?ERROR");
        return 1;
    }

```

```

    }
    imgClean = Crypting.decryptImage(imgOverlay);
    if (imgClean == null) {
        System.out.println("Decryption failed - key and encrypted
image not the same size?ERROR");
        return 1;

    }

    /*String path =
request.getSession().getServletContext().getRealPath("/");
    // String patt=path.replace("\\build", "");

    String strPath = path+"\\ "+saveFile;*/
    File f = new File(path2+names+"1.png");

    if (!f.toString().endsWith(".png")) {
        f = new File(f.toString() + ".png");
    }
    try {
        ImageIO.write(imgOverlay, "png", f);
    } catch (IOException e1) {
        System.out.println( "Could not Save file because: "
+ e1.getLocalizedMessage());
        return 1;
    }
    File f1 = new File(path2+names+"2.png");
    if (!f1.toString().endsWith(".png")) {

```

```

        f1 = new File(f1.toString() + ".png");
    }
    try {
        ImageIO.write(imgClean, "png", f1);
    } catch (IOException e1) {
        System.out.println("Could not Save file because: "
+ e1.getLocalizedMessage());
        return 1;
    }
}
catch(Exception e)
{
    System.out.println(e);
    return 1;
}
return 0;

}

/* public static void main(String args[])
    vino_decryption vv=new vino_decryption();

}*/
}

```

CHAPTER-7

SYSTEM TESTING

CHAPTER 7

SYSTEM TESTING

INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

7.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

7.1.1 WHITE BOX TESTING:

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides

- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .We have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

7.1.2. BASIC PATH TESTING:

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw a correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

$$V(G) = E - N + 2 \text{ or}$$

$$V(G) = P + 1 \text{ or}$$

$$V(G) = \text{Number of Regions}$$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis set of linearly independent paths.

7.1.3. CONDITIONAL TESTING:

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on a particular condition is traced to uncover any possible errors.

7.1.4. DATA FLOW TESTING:

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variables were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

7.1.5. LOOP TESTING:

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

All the loops were tested at their limits, just above them and just below them.

All the loops were skipped at least once.

For nested loops, test the innermost loop first and then work outwards.

For concatenated loops the values of dependent loops were set with the help of connected loops.

Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the inputs have been validated.

7.2 INTEGRATION TESTING

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets is conducted.

The objective is to take unit tested modules and combine them to test it as a whole. Thus, in the integration-testing step all the errors uncovered are corrected for the next testing steps.

7.3 TESTCASES AND REPORTS

TEST CASE ID	TESTCASE/ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	User enter mail id and password	Display Login page	Display Login page	Pass
2	Enter the Bank details in bank login	Display Bank page	Display Bank page	Pass
3	Enter the Admin page	Display Admin page and admin enter mail id and password	Display Admin page and admin enter mail id and password	Pass

4	Bank Login	Check the entered details and accept	Check the entered details and accept	Pass
5	Public user Login	Check the entered details and accept	Check the entered details and accept	Pass
6	Image Verification page	Upload the image from mail received	Upload the image for mail received	Pass
7	Key value	Key value appears	Key value appears	Pass
8	Transaction Page	After entering the key value and enter the amount	After entering the key value and enter the amount	Pass

CHAPTER-8

CONCLUSION

CHAPTER 8

CONCLUSION

8.1 RESULTS AND DECLARATION

The results obtained and analyses carried out were validated both by the expert panel and a senior management expert who is responsible for project assessment and development state-owned Banking Corporation. Collectively and individually, these participants considered the framework developed advancement over current practices in analyses of banking digitalization determinants.

The objective of this article was not, as mentioned previously, to obtain the ideal model but instead to foster the development of new approaches and the use of FCM-SD methods to improve banking digitalization analysis.

8.2 CONCLUSION

The development of information technology and electronic business every day has an increasingly significant impact on all spheres of modern life. This technology is designed to change the traditional perception of how people interact through a network. The main advantage of the technology is the complete synchronization of processes, integrity and uniqueness of all processed information, regardless of mining and tokens. Cloud based technology helps to improve distributed databases in terms of storage, synchronization, loss and integrity of data. Thus, the tool which at implementation in the Banking system without mining and tokens will considerably simplify processes of maintenance of integrity and uniqueness of information on bank transactions, and its implementation in the processes of smart contracts will reduce the number of participants at commission of some transactions.

8.2.1 FUTURE ENHANCEMENT

The proposed approach covers a wide range of possible criteria used in banking digitalization analyses. While the determinant of technological factors is clearly quite important in the evolution of banking digitalization, the cause-and-effect relationships between a variety of determinants need to be examined, especially those that are interconnected with human factors. Otherwise, as shown by the proposed model, cause-and-effect relationships between socioeconomic and human factors, clients, and/or profitability, as well as the negative influence of risks and security, may have an unexpected impact on banking digitalization evolution over time.

APPENDICES

APPENDICES

A.1 SAMPLE SCREENS

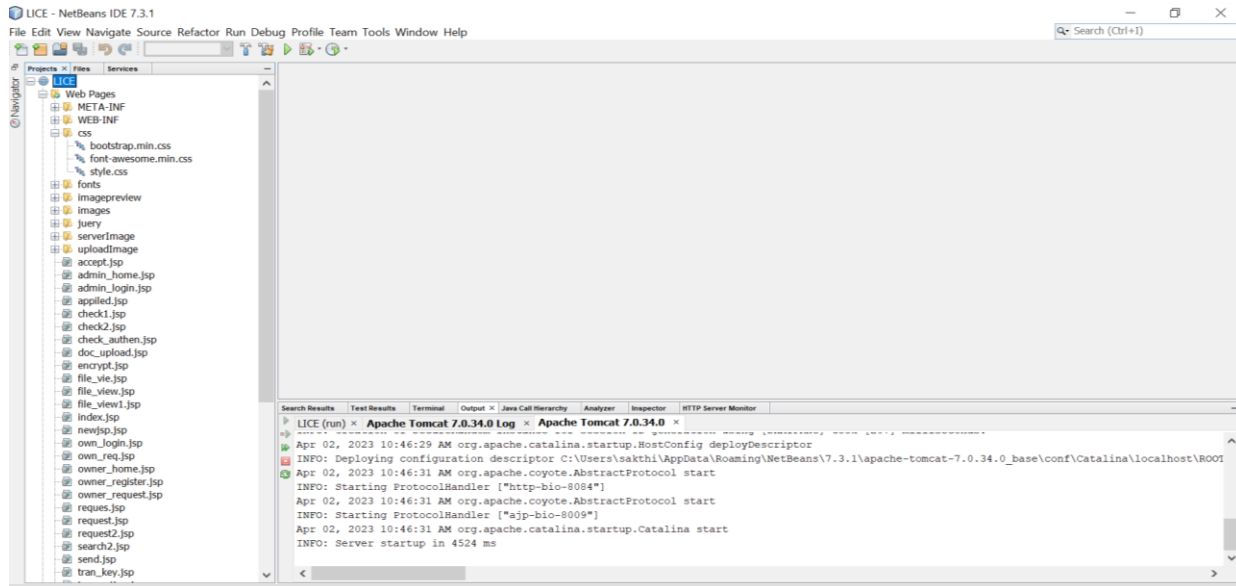


Fig A.1.1 NetBeans IDE

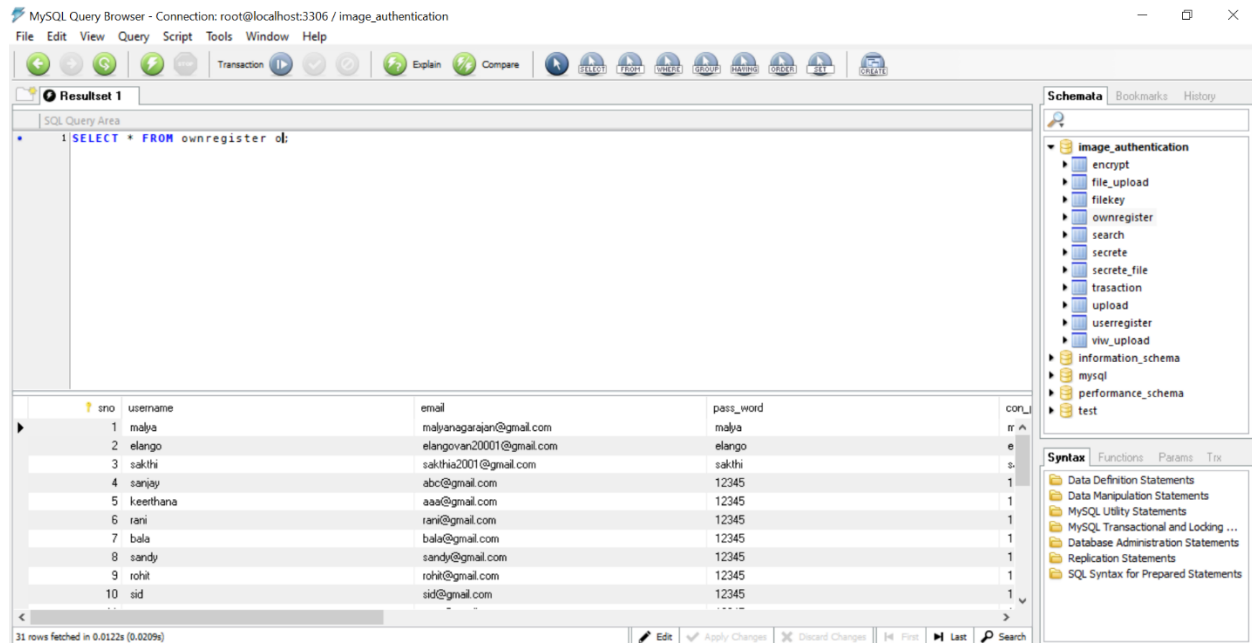


Fig A.1.2 MySQL Query browser

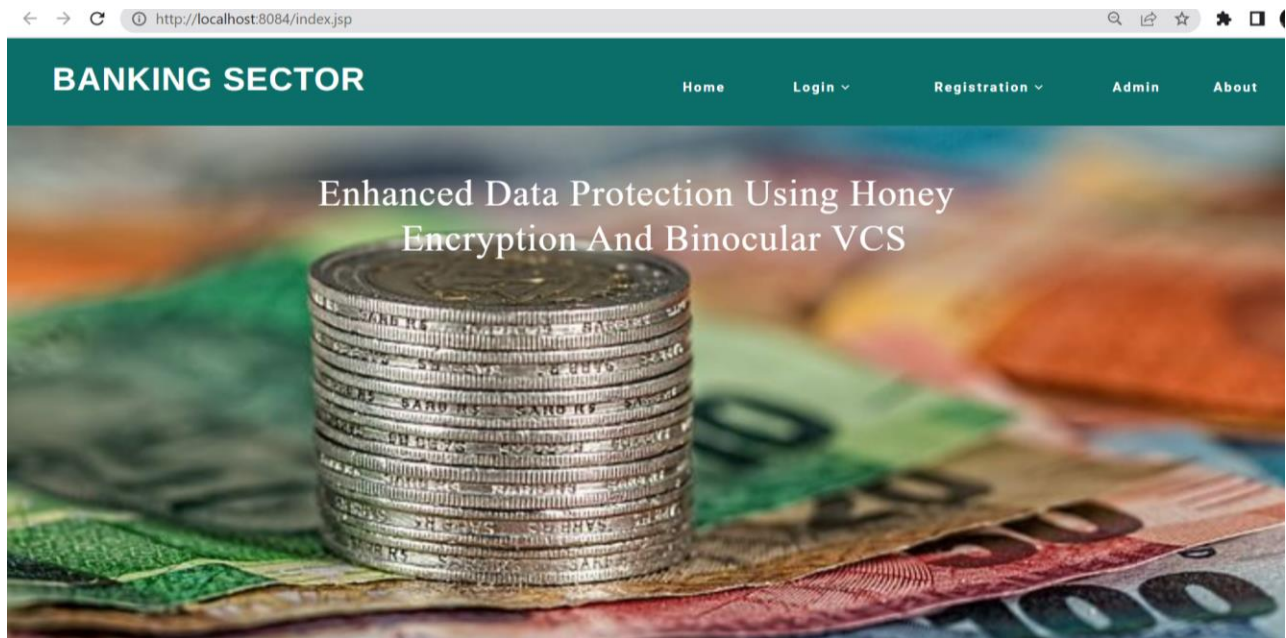


Fig A.1.3 Home page

PUBLIC USER REGISTRATION

USERNAME :

EMAIL ID :

PASSWORD :

CONFIRM PASSWORD :

MOBILE NUMBER :

CITY NAME :

ADDRESS :

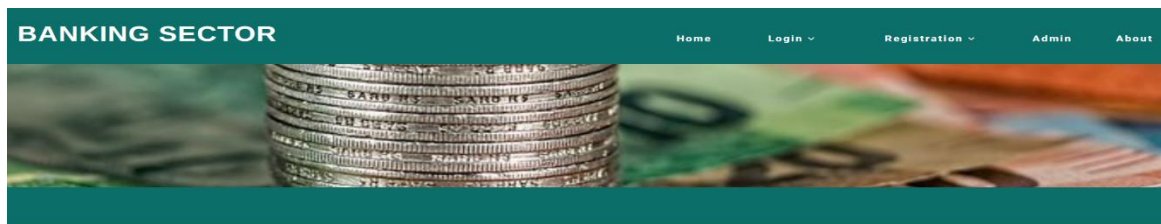
Fig A.1.4 Public User Registration

http://localhost:8084/user_register.jsp

BANKING REGISTRATION

USERNAME :
 EMAIL ID :
 PASSWORD :
 CONFIRM PASSWORD :
 MOBILE NUMBER :
 CITY NAME :
 ADDRESS :

Fig A.1.5 Bank Registration



ADMIN LOGIN

ADMIN :
 PASSWORD :

Fig A.1.6 Admin Login

BANKING SECTOR

Home Verification Public Access Logout

PUBLIC USER DETAILS

USERNAME	EMAIL ID	MOBILE NO	STATUS	ACCEPT
malaya	malayanagarajan@gmail.com	8967984329	YES	Accept
elango	elangovan20001@gmail.com	8765432134	YES	Accept
sakthi	sakthia2001@gmail.com	8237678897	YES	Accept
sanjay	abc@gmail.com	8723465782	YES	Accept
keerthana	aas@gmail.com	8765677778	YES	Accept
rani	rani@gmail.com	8762663722	YES	Accept
bala	bala@gmail.com	8877665544	YES	Accept
sandy	sandy@gmail.com	8888888887	YES	Accept
rohit	rohit@gmail.com	8877777777	YES	Accept
sid	sid@gmail.com	8773563636	YES	Accept
guru	guru@gmail.com	8777777777	YES	Accept
ranji	ranji@gmail.com	8777777722	YES	Accept
ramm	ram@gmail.com	8777777777	YES	Accept
dan	dan@gmail.com	8666666666	YES	Accept
maia	maia@gmail.com	8655555555	NO	Accept
jack	jack@gmail.com	8333333333	YES	Accept

Fig A.1.7 Admin views details of User , Bank and verifies(accept/reject)



BANKING SECTOR

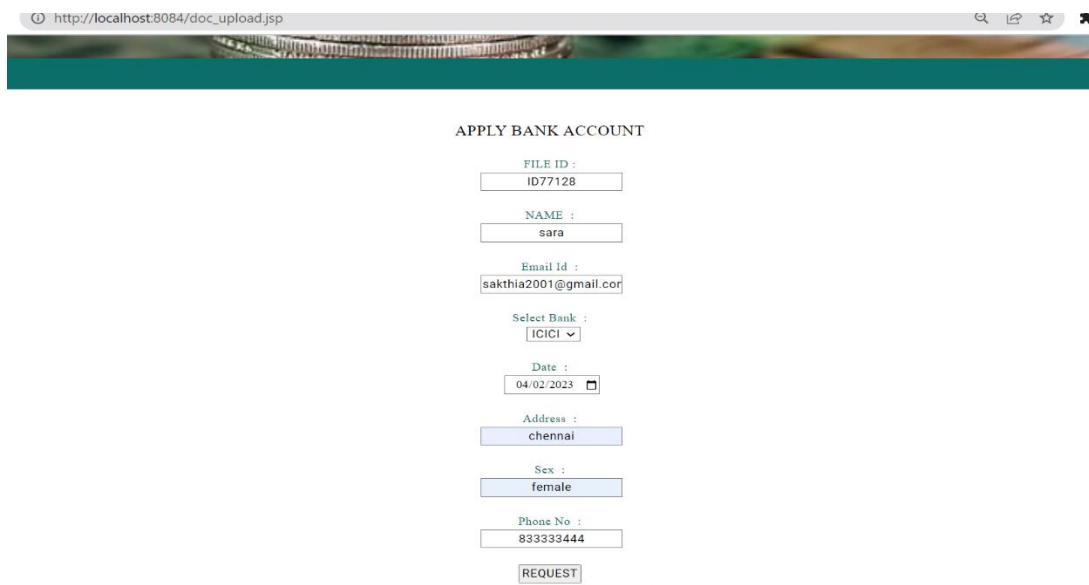
Home Login Registration Admin About

Public User LOGIN

EMAIL ID :

PASSWORD :

A.1.8 Public user Login page



http://localhost:8084/doc_upload.jsp

APPLY BANK ACCOUNT

FILE ID :

NAME :

Email Id :

Select Bank :

Date :

Address :

Sex :

Phone No :

Fig A.1.9 Public User enters user account details

sss	ID43559	sakthia2001@gmail.com	HDFC	chennai	female	EMP100084444444	Accepted	
ddd	ID21919	sakthia2001@gmail.com	HDFC	d	male	EMP1000822222222	Accepted	
apple	ID72288	sakthia2001@gmail.com	HDFC	c	male	EMP1000833333333	Accepted	
apple	ID32199	sakthia2001@gmail.com	HDFC	sd	male	EMP1000855555555	Accepted	
up	ID37686	sakthia2001@gmail.com	HDFC	u	male	EMP1000866666666	Accepted	
up	ID727	up@gmail.com	HDFC	r	male	EMP1000877777777	Accepted	
black	ID7424	black@gmail.com	HDFC	e	male	EMP100081112333333	Accepted	
alice	ID40744	alice@gmail.com	HDFC	d	female	EMP10008122233445	Accepted	
alice	ID94249	sakthia2001@gmail.com	HDFC	g	female	EMP100081111111111	Accepted	
alice	ID117797	alice@gmail.com	HDFC	f	female	EMP1000844444444	Accepted	
alice	ID15385	sakthia2001@gmail.com	HDFC	f	female	EMP1000877777777	Accepted	
alice2	ID111875	alice2@gmail.com	HDFC	w	female	EMP1000877777777	Accepted	
alice3	ID99167	sakthia2001@gmail.com	HDFC	d	female	EMP100081111111111	Accepted	
alice4	ID29207	sakthia2001@gmail.com	HDFC	f	female	EMP10008122233445	Accepted	
alice5	ID9837	sakthia2001@gmail.com	HDFC	f	female	EMP1000877777777	Accepted	
alice5	ID82719	sakthia2001@gmail.com	HDFC	k	female	EMP100081112333333	Accepted	
alice5	ID82719	sakthia2001@gmail.com	HDFC	k	female	EMP100081112333333	Accepted	
alice2	ID102767	sakthia2001@gmail.com	HDFC	k	male	EMP1000877777777	Accepted	
dog	ID7353	sakthia2001@gmail.com	ICICI	h	female	EMP10008122233445	Accepted	
akash	ID6355	sakthia2001@gmail.com	ICICI	chennai	male	EMP1000877777777	Accepted	
sathya	ID96855	sakthia2001@gmail.com	HDFC	chennai	female	EMP1000877777777	Accepted	
sathya	ID72078	sakthia2001@gmail.com	HDFC	chennai	male	EMP1000866666666	Accepted	
sathya	ID82266	sakthia2001@gmail.com	HDFC	chennai	female	EMP1000866666666	Accepted	
sathya	ID115197	sakthia2001@gmail.com	HDFC	chennai	female	EMP10008122233445	Accepted	
sathya	ID1799	sakthia2001@gmail.com	HDFC	chennai	female	EMP100081111111111	Accepted	
sathya	ID63426	sakthia2001@gmail.com	HDFC	chennai	male	EMP1000811133333	Accepted	
eee	ID8206	sakthia2001@gmail.com	HDFC	chennai	male	EMP10008333333	Accepted	
abi	ID2870	sakthia2001@gmail.com	HDFC	chennai	female	EMP10008333333	Accepted	
sara	ID77128	sakthia2001@gmail.com	ICICI	chennai	female	EMP1000833333444	Request	Accept

Fig A.1.10 Bank approves or reject after checking user account details

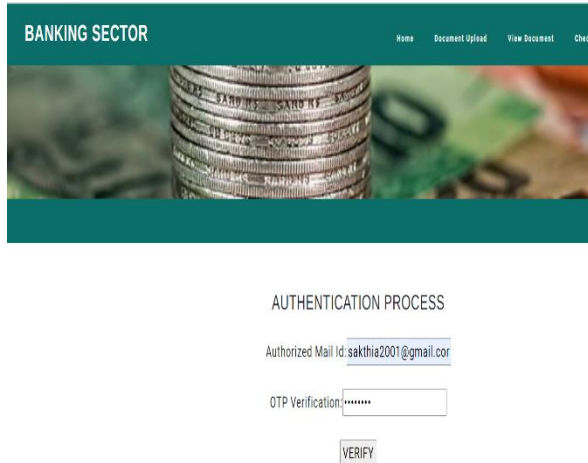


Fig A.1.11 Authentication page

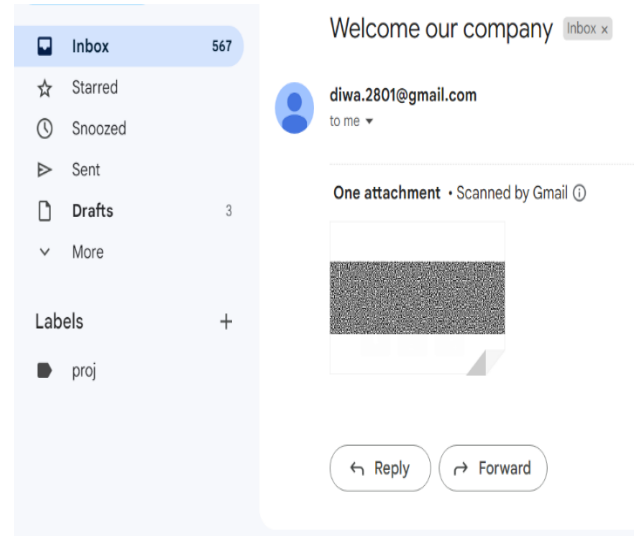


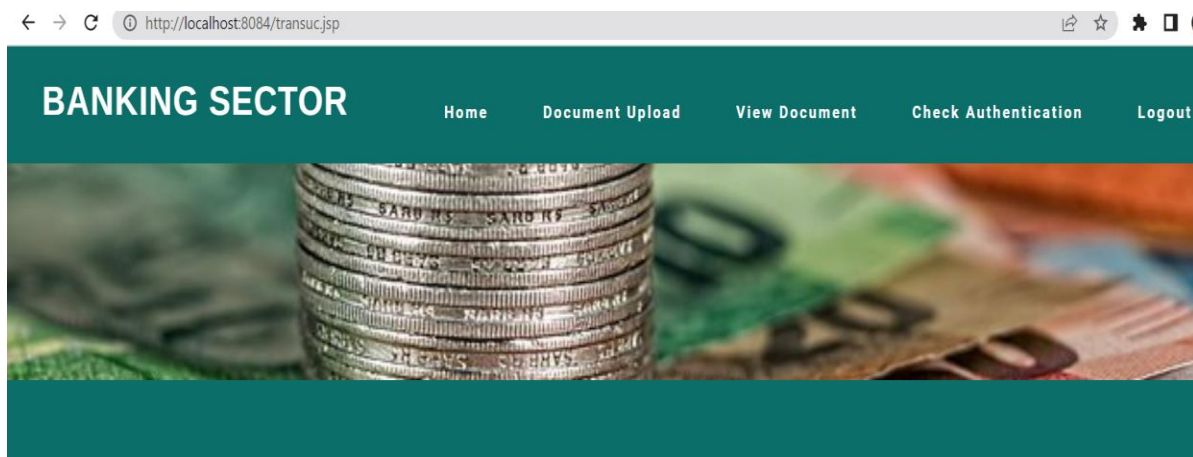
Fig A.1.12 User's key in Gmail



Fig A.1.13 Image Verification process



Fig A.1.14 Transaction page



Transaction Succesfull...

Fig A.1.15 Final page

REFERENCES

REFERENCES

- [1] Roberto Vergallo and Luca Mainetti , "The Role of Technology in Improving the Customer Experience in the Banking Sector: A Systematic Mapping Study" , 2022
<https://sci-hub.ru/10.1109/tsp.2018.2883021>
- [2] Priya D. Dozier and Troy A. Montgomery, "Banking on Blockchain: An Evaluation of Innovation Decision Making" , 2022
<http://dx.doi.org/10.1109/TEM.2019.2948142><http://dx.doi.org/10.1109/TEM.2020.2993171>
- [3] Andreas Krause and Simone Giansante, " Network-Based Computational Techniques to Determine the Risk Drivers of Bank Failures During a Systemic Banking Crisis", 2018
<http://dx.doi.org/10.1109/TETCI.2018.280531>
- [4] Wei Zhang, Xueying Zhang, and Ying Sun, "A New Fuzzy Cognitive Map Learning Algorithm for Speech Emotion Recognition", 2017
<https://sci-hub.se/10.1155/2017/4127401>
- [5] Yuanqiao chen , Xiaoqiang Zhang, Bomin Bian, and Haohua L , "Optimal Staffing Policy in Commercial Banks Under Seasonal Demand Variation", 2019
<https://sci-hub.se/10.1109/access.2019.2937687>
- [6] João F. C. Rodrigues, Fernando A. F. Ferreira, "Banking Digitalization: (Re)Thinking Strategies and Trends Using Problem Structuring Method " , 2019
<http://dx.doi.org/10.1109/TEM.2020.2993171>

[7] Ismail Turk, Pelin Angin, and Ahmet Cosar , "RONFC: A Novel Enabler-Independent NFC Protocol for Mobile Transactions" ,2019

<https://doi.org/10.1109/access.2019.2929011>

[8] Mushtaq Ali , Amanullah Baloch , Abdul Waheed, "A Simple and Secure Reformation-Based Password Scheme", 2021

<http://dx.doi.org/10.1109/ACCESS.2020.304905>

[9] Cegarra-Navarro, J-G., Jiménez-Jiménez, D. & Garcia-Perez, "An integrative view of knowledge processes and a learning culture for ambidexterity: Towards improved organisational performance in the banking sector".

<http://dx.doi.org/10.1109/TEM.2019.2917430>

[10] Luke Pfister, Yoram Bresler, “Learning Filter Bank Sparsifying Transforms”,2018

<https://sci-hub.ru/10.1109/tsp.2018.2883021>