

A PROJECT REPORT ON

“SECURE E WALLET ARCHITECTURE

USING BCT FEATURES”

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY,
PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING
(INFORMATION TECHNOLOGY)

BY

| | |
|---------------------------|------------------|
| <i>Jay Pardeshi</i> | <i>71834755C</i> |
| <i>Jay Sharma</i> | <i>71834756M</i> |
| <i>Sunil Bhadu</i> | <i>71917204H</i> |
| <i>Atharva Tribhuwane</i> | <i>71835034M</i> |

UNDER THE GUIDANCE OF

Prof. Yugashree Bhadane



DEPARTMENT OF INFORMATION TECHNOLOGY

Dhole Patil College Of Engineering
WAGHOLI, PUNE 412207

Savitribai Phule Pune University, PUNE
2021-2022



CERTIFICATE

This is to certify that the project report entitled
“SECURE E WALLET ARCHITECTURE USING BCT FEATURES”

Submitted By

| | |
|--------------------|-----------|
| Jay Pardeshi | 71834755C |
| Jay Sharma | 71834756M |
| Sunil Bhadu | 71917204H |
| Atharva Tribhuwane | 71835034M |

are bonafide student of this institute and the work has been carried out by him under the supervision of **Prof. Yugashree Bhadane** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Information Technology Engineering).

Date:-

Place:- DPCOE

Prof. Yugashree Bhadane
Internal Guide
Dept. IT Engineering

Prof. Rahul Ghode
Head Of Department
Dept. IT Engineering

External Examiner

Dr. Nihar Walimbe
Principal DPCOE, Pune

Acknowledgments

It gives us great pleasure in presenting the project report on ‘**SECURE E WALLET ARCHITECTURE USING BCT FEATURES**’.

We would like to take this opportunity to thank my internal guide **Prof. Yugashree Bhadane** for giving me all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful.

We are also grateful to **Prof. Rahul Ghode**, Head of IT Engineering Department, **Dhole Patil College Of Engineering, PUNE** for his indispensable support and suggestions.

In the end our special thanks to **all staff of IT department** for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our project

Jay Pardeshi
Jay Sharma
Sunil Bhadu
Atharva Tribhuwane
(B.E. Information Technology Engg.)

Abstract

In 2016, the Indian government, led by Prime Minister Narendra Singh Modi, announced that the nation's two highest-denomination bank notes would cease to be legal tenders. At the time, the two denominations accounted for roughly 86 percent of cash in circulation in India. People who possessed the banknotes were to deposit them in the bank. With the move, the Indian government aimed to punish tax evaders in retrospect. The logic was that people with hoards of “black money” would have to answer questions if they attempted to deposit the demonetized banknotes. Banking and technology are very closely associated and innovations have changed banking drastically over the period of time. The digital innovations in the banking sector started with the introduction of money that replaced the barter system and then the gradual replacement of wax seals with digital signatures. One such disruptive innovation which is changing the banking sector globally is Blockchain Technology (BCT). Blockchain is a shared distributed ledger which stores business transactions to a permanent unbreakable chain which can be viewed by the parties in a transaction. Blockchain technology has the potential to disrupt the financial business applications as it provides permanent and tamper proof recording of transactions in a distributed network

Index Terms:- cashless economy, security, distributed database, visual cryptography, hash algorithm,etc.

TABLE OF CONTENTS

| | |
|------------------------|------|
| Acknowledgment | II |
| Abstract | III |
| List of figures | VIII |
| List of table | IX |

| SR. NO. | TITLE | PAGE NO. |
|----------|---|-----------|
| 1 | INTRODUCTION | 10 |
| 1.1 | Overview | 10 |
| 1.2 | Motivation | 11 |
| 1.3 | Problem Definition and Objectives: | 12 |
| 1.3.1 | Objectives: | 12 |
| 1.4 | Project Scope and Limitations:: | 13 |
| 1.5 | Methodologies of Problem solving: | 13 |
| 2 | LITERATURE SURVEY | 14 |
| 3 | SOFTWARE REQUIREMENT SPECIFICATION | 17 |
| 3.1 | Project Scope | 17 |
| 3.2 | Assumptions and Dependencies | 17 |
| 3.2.1 | User Classes and Characteristics. | 17 |
| 3.3 | Functional Requirements | 18 |
| 3.4 | External Interface Requirements | 18 |
| 3.4.1 | User Interfaces | 18 |
| 3.4.2 | Hardware Interfaces | 18 |
| 3.4.3 | Software Interfaces | 18 |

| | | |
|----------|---|-----------|
| 3.5 | Non Functional Requirements | 19 |
| 3.5.1 | Performance Requirements | 19 |
| 3.5.2 | Safety Requirements | 19 |
| 3.5.3 | Security Requirements | 19 |
| 3.5.4 | Software Quality Attributes | 19 |
| 3.6 | System Requirements | 20 |
| 3.6.1 | Database Requirements | 20 |
| 3.6.2 | Software Requirements | 20 |
| 3.6.3 | Hardware Requirements | 21 |
| 3.7 | Analysis Models: SDLC Model to be applied | 21 |
| 4 | SYSTEM DESIGN | 23 |
| 4.1 | System Architecture | 23 |
| 4.2 | MATHEMATICAL MODEL | 24 |
| 4.3 | Data Flow Diagrams | 25 |
| 4.3.1 | Level 0 Data Flow Diagram | 25 |
| 4.3.2 | Level 1 Data Flow Diagram | 25 |
| 4.4 | Entity Relationship Diagrams | 26 |
| 4.5 | UML Diagrams | 26 |
| 4.5.1 | Class Diagram | 26 |
| 4.5.2 | Use Case Diagram | 27 |
| 4.5.3 | Sequence Diagram | 28 |
| 5 | PROJECT PLAN | 29 |
| 5.0.1 | Reconciled Estimates | 29 |
| 5.1 | Project Estimate | 30 |
| 5.1.1 | COCOMO Model | 31 |

| | | |
|----------|--|-----------|
| 5.1.2 | Reconciled Estimates | 35 |
| 5.1.3 | Project Resources | 35 |
| 5.2 | Risk Management | 35 |
| 5.3 | Project Schedule | 37 |
| 5.3.1 | Project task set | 37 |
| 5.3.2 | Timeline Chart | 38 |
| 5.4 | Team Organization | 39 |
| 5.4.1 | Team Structure | 39 |
| 6 | PROJECT IMENTATION | 41 |
| 6.1 | Overview of Project Modules | 41 |
| 6.2 | Tools and Technologies Used: | 41 |
| 6.3 | Algorithm Details: | 43 |
| 6.3.1 | AES | 43 |
| 6.3.2 | MD5:Hash Function | 44 |
| 7 | SOFTWARE TESTING | 45 |
| 7.1 | Types of Testing | 45 |
| 7.1.1 | Unit Testing | 45 |
| 7.1.2 | Alpha Testing | 45 |
| 7.1.3 | Acceptance Testing | 45 |
| 7.1.4 | Beta Testing | 46 |
| 7.1.5 | Performance Testing | 46 |
| 7.1.6 | Security Testing | 46 |
| 7.1.7 | White Box Testing | 47 |
| 7.1.8 | Black Box Testing | 47 |
| 7.1.9 | Regression Testing | 47 |

| | | |
|-----------|-------------------------------|-----------|
| 7.1.10 | System Testing | 48 |
| 7.1.11 | Smoke Testing | 48 |
| 7.1.12 | Integration Testing | 48 |
| 7.2 | Test Cases | 49 |
| 8 | RESULT | 51 |
| 8.1 | Screenshots | 51 |
| 9 | CONCLUSION | 56 |
| 9.1 | Future Scope | 56 |
| 9.2 | Applications | 56 |
| 10 | REFERENCES | 57 |
| | Annexure A | 59 |

List of Figures

| | | |
|-----|--|----|
| 3.1 | Waterfall Model | 21 |
| 4.1 | System Architecture | 22 |
| 4.2 | Level 0 Data Flow Diagram | 25 |
| 4.3 | Level 1 Data Flow Diagram | 25 |
| 4.4 | Entity Relationship Diagrams | 26 |
| 4.5 | Class Diagram | 27 |
| 4.6 | Use Case Diagram | 27 |
| 4.7 | Sequence Diagram | 28 |
| 5.1 | Timeline Chart | 38 |
| 8.1 | User Login | 51 |
| 8.2 | User Home | 52 |
| 8.3 | User Account | 53 |
| 8.4 | User Add Beneficiary | 54 |
| 8.5 | User Transactions | 55 |
| A.1 | NP Problem | 59 |
| A.2 | NP Problem Complete | 60 |

List of Tables

| | | |
|-----|-----------------------------|----|
| 2.1 | Literature Survey | 14 |
| 5.1 | Project Estimate | 30 |
| 5.2 | Time line Chart | 37 |
| 7.1 | Test Cases | 49 |

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

India is currently the seventh-largest economy in the world. It currently has an estimated population of about 1.34 bln people, or about 18 percent of the world's population, according to the World Economic Forum. Despite its GDP dropping by roughly 5.7 percent in the quarter that ended June of this year, India remains the fastest growing large economy in the world — other than China. If estimates are any thing to go by, India will have overtaken China as the world's most populous country by 2024, which would help solidify its position as the nation with the world's largest youth population. The World Economic Forum also projects that India's economy will be the second-largest economy in the world by 2050, with China occupying the first position. Poor as the policy might have been for average Indians, though, there were bright spots for proponents of a cashless economy. The World Economic Forum reported that the number of digital transactions in India increased following the demonetization policy — a plus for the government, who would now have increased ability to track the flow of money within the economy. The growth in digital transactions in India is, in turn, a big plus for Blockchain and cryptocurrency. Just about 0.5 percent of the people in India are into Bitcoin, the cryptocurrency that popularized the Blockchain technology. By inference, if such few people in India know about Bitcoin, it's safe to say that only about 0.5 percent of India's population is conversant with Blockchain technology. However, on the national level, a lot of work is going on to integrate Blockchain technology into various sectors of the economy including the financial and health sectors. In 2016, the Indian bank,

ICICI Bank, announced that it had completed a cross-border transaction executed on a Blockchain. In September of this year, the Institute for Development and Research in Banking Technology, or IDRBT, founded by the Reserve Bank of India, announced plans to launch a new Blockchain platform. The Reserve Bank of India is India's central bank. The announcement followed a report published by the IDRBT in January of this year, that India could use Blockchain to digitize its national currency, the rupee. Given the positives — increased tax payments, for instance — that the demonetization policy in India has yielded through increased digital transactions, it's plausible that the Indian government will double down on its drive to grow a cashless economy. There are some challenges, but it seems promising. If, as in any place in the world, the Indian government wants to boost its cashless economy, it needs to find lasting solutions to the challenges confronting the propagation of a cashless economy. Some of those challenges include financial inclusion, high setup and transaction costs and transaction times.

1.2 MOTIVATION

The digital innovations in the banking sector started with the introduction of money that replaced the barter system and then the gradual replacement of wax seals with digital signatures. One such disruptive innovation which is changing the banking sector globally is Blockchain Technology (BCT). Blockchain is a shared distributed ledger which stores business transactions to a permanent unbreakable chain which can be viewed by the parties in a transaction. Blockchain technology has the potential to disrupt the financial business applications as it provides permanent and tamper proof recording of transactions in a distributed network.

1.3 PROBLEM DEFINITION AND OBJECTIVES

1. If, as in any place in the world, the Indian government wants to boost its cashless economy, it needs to find lasting solutions to the challenges confronting the propagation of a cashless economy. Some of those challenges include financial inclusion, high setup and transaction costs and transaction times
2. Due to a considerable segment of the Indian economy remaining informal, there's still a huge part of the population that doesn't rely on traditional financial institutions for financial services. Based on the cashless technologies employed today, most people would need a bank account in order to live in a cashless economy — an uphill battle. In essence, for you to run a cashless economy, you'll need an alternative to traditional financial services. .

1.3.1 Objectives:

1. To implement a java based web application.
2. To implement AES.
3. To implement visual cryptography.
4. To implement a block chain.
5. To implement a distributed database system using WLAN.

1.4 PROJECT SCOPE AND LIMITATIONS

Project will be developed as a prototype model using JSP and servlet technology. It will run as a local host. System will be communicated through a wireless local area network. System communication will be limited in the wireless local area network, but in the future if we host the project using WAN , it can communicate world wide.

1.5 METHODOLOGIES OF PROBLEM SOLVING

BCT The Blockchain technology almost entirely eliminates the need to belong in the traditional financial system, in order to be financially included. It costs a merchant between Rs 4,000 and Rs 8,000 to set up a card-swiping terminal in India. That is definitely not a problem for big-ticket merchants, but smaller merchants who collectively constitute a big part of the economy might not be happy to pay that much in addition to the subsequent transaction fees. For instance, The Hindu reported in May that Indian consumers are moving back to cash-based transactions, because of demonetization and high digital transaction costs. This makes a case for a cheaper way of conducting digital transactions. Again, Blockchain fits the bill. If a cashless economy is ever going to be the order of the day, it needs to have a real time feature to it. Today's technologies have done a great job in reducing the wait times between when a transaction is completed and when the funds become accessible. But, it's not yet at the level where the entire populace will be motivated to go digital. And this is another problem that the Blockchain technology solves brilliantly.

CHAPTER 2

LITERATURE SURVEY

Table 2.1: Literature Survey

| Sr. No. | Title | Authors / Year | Advantage | Description |
|---------|---|---|-------------------|---|
| 1. | The Implementation of E-money in Mobile Phone: A Case Study at PT Bank KEB Hana | Didik Haryadi ; Harissa ; Victory Haris Kusumawardhana ; Harco Leslie Hendric Spits Warnars2018 | | This study aims to analyze the design of e-money, as well as provide some development ideas that must be done related to the implementation of e-money. Here the system uses e payment using QR code and encryption technology. |
| 2. | A Landscape of Cryptocurrencies | Zhaofang Li ; Qinghua Lu ; Shiping Chen ; | Used Cryptography | This study aims to analyze the design of e-money, as well as provide some development ideas that must be done related to the implementation of e-money. |

| | | | | |
|--|--|-------------------------------|--|--|
| | | Yue Liu ; Xiwei Xu 2019 | | Here the system uses e payment using QR code and encryption technology. . |
|--|--|-------------------------------|--|--|

| | | | | |
|----|--|---|-----|--|
| | | | | |
| 3. | Security Manage ment and Visuali za tion in a Blockch ain based Collabo rative Defense | Christian Killer ; Bruno Rodrigues ; Burkhard Stiller 2019 | BCT | This work presents the design of a security management dashboard for BloSS, designed for interactive use by cyber security analysts. This work is about DDos attacks in defense systems. |
| 4. | On the Effectiv eness of Multi Token Economie s | Sean Kang ; Kideok Cho ; Kyle Park 2019 | BCT | This paper addresses the token classification, the reason for adopting multi-token economies and the effectiveness of them. We analyze the Steemit as a representative example of |

| | | | | |
|---|---|--|-----|---|
| | | | | <p>multi token economies.</p> <p>We describe how the multi-token economy has been working and show the distinctive features of multi token economies.</p> <p>We also propose the evaluation criteria for multi-token economies.</p> |
| 5 | Digitizing Invoice and Managing VAT Payment Using Blockchain Smart Contract | Van-Cam NGUYEN ; Hoai-Luan PHAM ; Thi-Hong TRAN ; Huu-Thuan HUYNH ; Yasuhiko NAKASHIMA2019 | BCT | <p>This paper proposed implementation of the VAT system as an on line system using BCT. A distributed database system is used in online systems. System can be prevented from hacking using BCT.</p> |

CHAPTER 3

SOFTWARE REQUIREMENT

SPECIFICATION

3.1 PROJECT SCOPE

- To develop a prototype model for cashless india using BCT.
- This model will be run at a local host using Glassfish server.
- BCT features such as decentralization, cryptography and hash codes will be implemented.

3.2 ASSUMPTIONS AND DEPENDENCIES

This document will provide a general description of the project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project such as interface, functional requirements and performance requirements.

3.2.1 User Classes and Characteristics

Find the different user classes that you anticipate will use this product. User classes can be differentiated based on use frequency or product functions subset used or technical expertise or privilege levels or educational level and experience. It also describes the pertinent behavior or characteristics of each user class. Few

requirements may be limited only to specific user classes. Differentiate the very most important or useful user classes for this item or product from those who are less significant to satisfy.

3.3 FUNCTIONAL REQUIREMENTS

Functional user requirements are nothing but very high-level statements about what the system should and also it should describe clearly an overview of system services in detail.

3.4 EXTERNAL INTERFACE REQUIREMENTS

3.4.1 User Interfaces

The user interface or UI for the software should be compatible to be used by any standard browser such as IE, Mozilla or Google chrome. Using this UI user can have access to the system. The UI or user interface can be developed by using many tools or software packages like JFrame.

3.4.2 Hardware Interfaces

A hardware interface is needed to run the software. Java (JDK) and NetBeans compatible hardware is required which is a minimal requirement.

3.4.3 Software Interfaces

It uses Java as the front end programming tool. MySQL has been used as a back end application tool. Latest version of java anything higher than 7.0 can be used.

3.5 NON FUNCTIONAL REQUIREMENTS

3.5.1 Performance Requirements

- System can work optimal or faster on 8 GB or more of RAM.
- The system is targeted to be available all time. Once there is a fatal error or system down, the system will provide understandable feedback to the user.

3.5.2 Safety Requirements

- The system is designed in modules where errors can be detected.

3.5.3 Security Requirements

- The system is designed in modules where errors can be detected and fixed easily.

3.5.4 Software Quality Attributes

- Usability:

This relates to how easily people can use apps/websites. A measure of us ability could be the time it takes for end users to become familiar with my app/website functions, without training or help.

- Reliability:

This can be defined as the available time or UP time of software.

- Performance:

This is essentially how fast an app/website works. A performance requirement for the app/website could be started in less than 20 seconds.

- Security :

Say that app/website saves all the previous code and lets you reuse a saved code.

3.6 SYSTEM REQUIREMENTS

3.6.1 Database Requirements

MySQL Database

MySQL is an open source database which is mainly a RDBMS i.e. relational database management system. As a database server, the primary function of this software is to store and retrieve data as requested by others

from end software applications like java which may Or may not run either on the same computer or on different computers. This can be across the network either in the internet or intranet.

3.6.2 Software Requirements

1. Operating System: Microsoft Windows 7 and Above
2. Programming Language: Java
3. IDE: Netbeans, Android Studio

3.6.3 Hardware Requirements

1. Processor: Intel Core I3 or Higher
2. RAM: 4 GB or Higher
3. Hard Disk: 100 GB (min)

3.7 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

SDLC model to be applied

Waterfall Model:

The Waterfall Model is among the very first and oldest models of software development life cycle. It is also called a linear-sequential life cycle model. This is very simple in nature and easy to understand or use. This is a step by step method so the next step can only be begun once earlier has been completed. This is mainly used for small scale projects. Constant or fixed requirements should be there for this type of model.

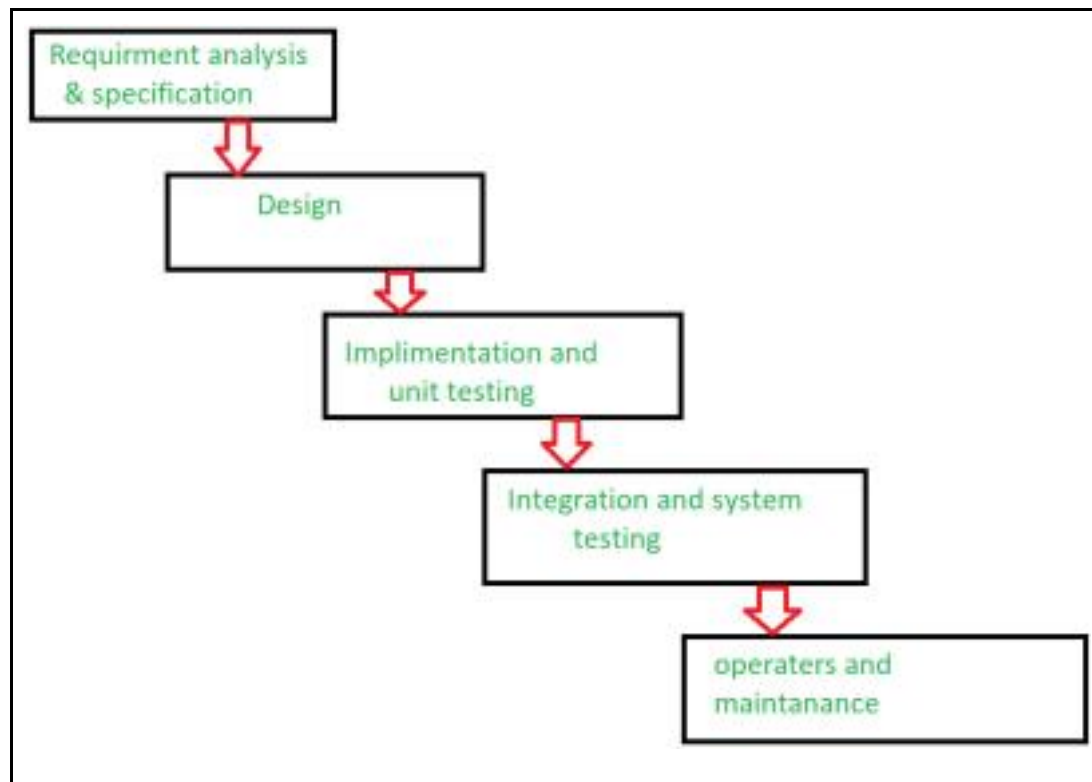
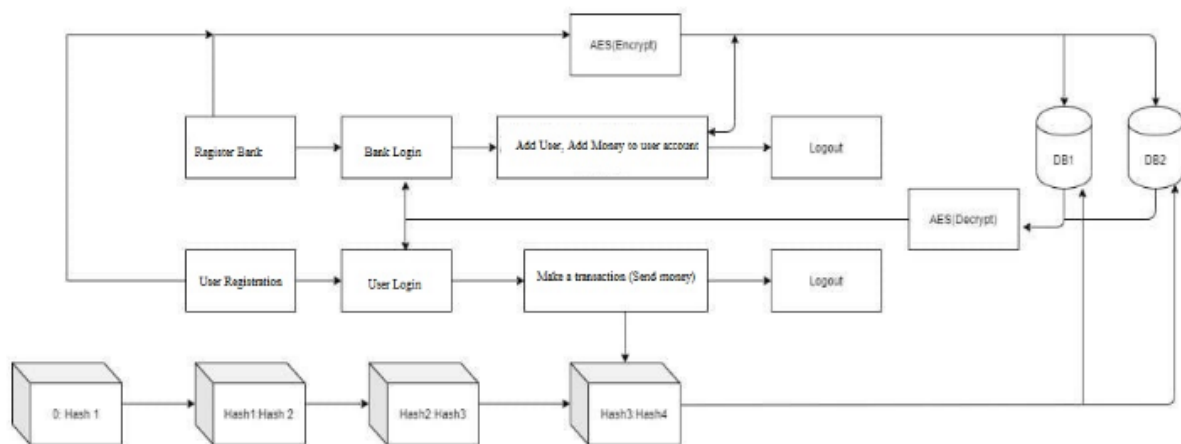


Figure 3.1: Waterfall Model

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE



Whenever any transaction will occur in the system , the record of that transaction is maintained in the form of hash value in a block. Each next block will get attached to the previous block and in this way a virtual block chain will occur. The hash value of a current block is generated using the data of a current block and the hash of the previous block. In this way if any of the block is tempered the subsequent all the block’s hash must be changed . Such multiple copies are maintained at different servers , which will assure the data security and confidentiality. As everything through the application interface, it will maintain the transparency in transaction.

Let

S be Closed system defined as, $S = Ip, Op, Ss, Su, Fi, A$

To select the input from the system and perform various actions from the set of actions A so that Su state can be attained.

$S = Ip, Op, Ss, Su, Fi, A$

Where,

$IP1 = \text{Username, Password, image}$

Set of actions = $A = F1, F2, F3, F4$

Where

$F1 = \text{Send Mail}$

$F2 = \text{Merge Images}$

$F3 = \text{Encrypt Database}$

$F4 = \text{Generate Hash}$

$S = \text{Set of users}$

$Ss = \text{rest state, registration state, login state}$

Su - success state is successful analysis

Fi - failure state

Objects:

1) Input1: $Ip1 = \text{Username, Password}$

2) Input2 : $Ip2 = \text{image from mail}$

1) Output1 : $Op1 = \text{Transaction Record}$

2) Output2 : $Op2 = \text{Encrypted Database}$

3) Output3 : $Op3 = \text{Hash Codes.}$

4.3 DATA FLOW DIAGRAMS

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing.

4.3.1 Level 0 Data Flow Diagram

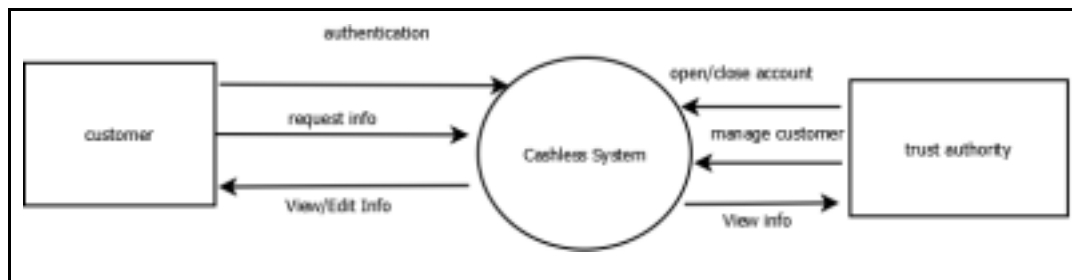


Figure 4.2: Level 0 Data Flow Diagram

4.3.2 Level 1 Data Flow Diagram

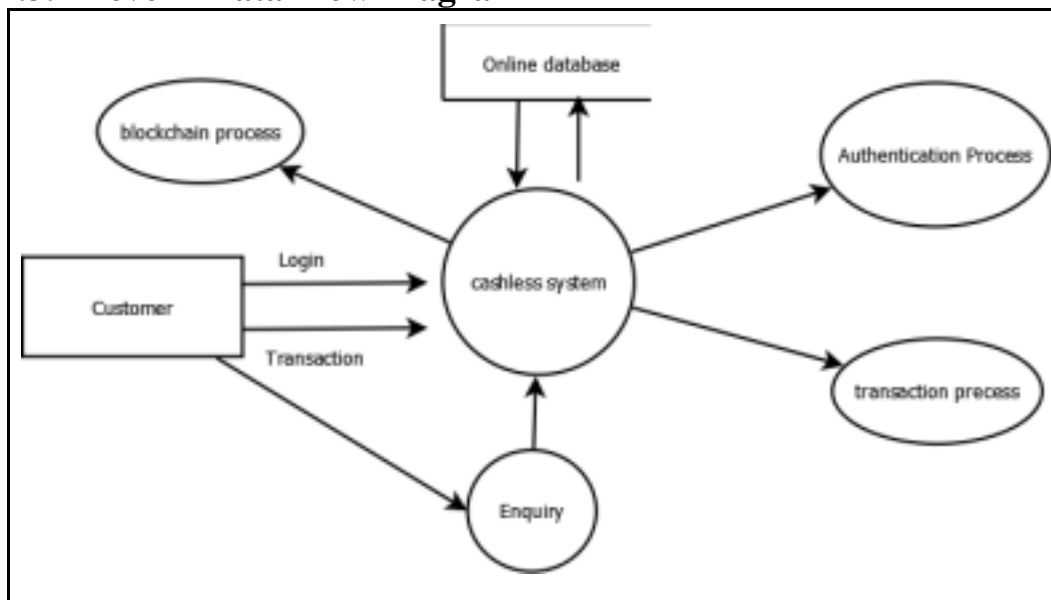


Figure 4.3: Level 1 Data Flow Diagram

4.4 ENTITY RELATIONSHIP DIAGRAMS

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

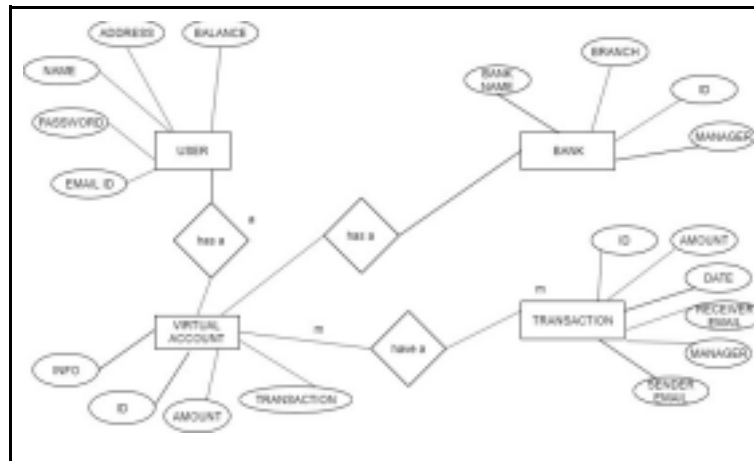


Figure 4.4: Entity Relationship Diagrams

4.5 UML DIAGRAMS

4.5.1 Class Diagram

A class diagram in the world of Unified Modeling Language or UML can be defined as a type of static structure diagram which mainly defines the structure of a system. It works by showing the system's classes and their attributes and operations or methods also the relationships among objects.



```

    usecaseDiagram
        actor customer
        actor trustAuthority as trust authority
        usecase login
        usecase addBeneficiary as add beneficiary
        usecase transaction
        usecase generateBlock as generate block
        usecase updateAccount as update account
        usecase custId as cust_id
        usecase password
        usecase invalidDetails as invalid details
        usecase accountNo as account no
        usecase name
        usecase amount
        usecase insufficientFund as insufficient fund
        usecase blockchain as Blockchain
        usecase customerId as customer_id

        login ..> custId : <<include>>
        login ..> password : <<include>>
        login ..> invalidDetails : <<exclude>>
        addBeneficiary ..> accountNo : <<include>>
        addBeneficiary ..> name : <<include>>
        addBeneficiary ..> invalidDetails : <<exclude>>
        transaction ..> amount : <<include>>
        transaction ..> insufficientFund : <<exclude>>
        transaction ..> generateBlock : <<extend>>
        generateBlock ..> blockchain : <<include>>
        updateAccount ..> customerId : <<include>>
        customer -- login
        customer -- addBeneficiary
        customer -- transaction
        trustAuthority -- transaction
        trustAuthority -- updateAccount
    
```

Figure 4.6: Use Case Diagram

4.5.3 Sequence Diagram

Sequence diagrams can be used to provide a graphical representation of object interactions or object coordination over the time. These basically display an actor or user, and the objects and components they interact with in the execution of a use case. The sequence diagrams display the own of messages from one object to another object, and as such correspond to the methods and events supported by a class/object.

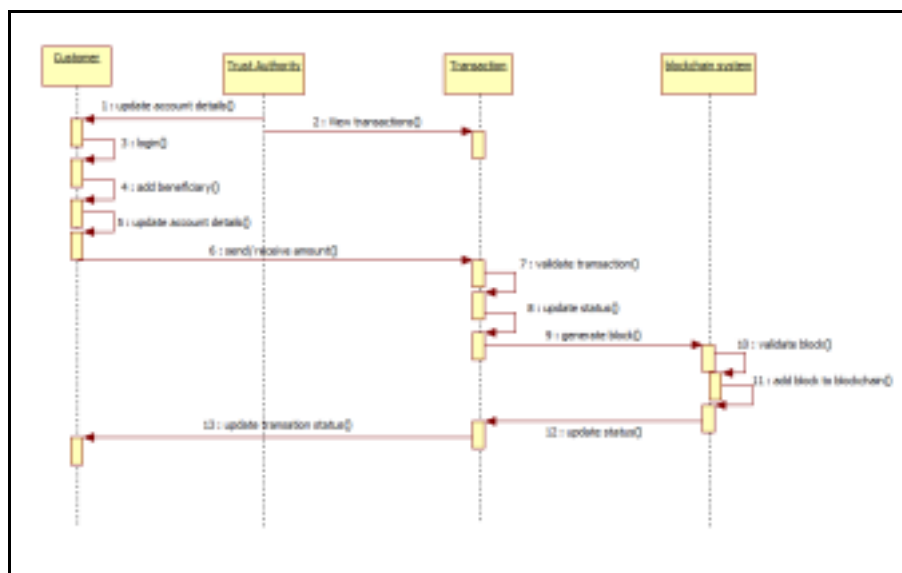


Figure 4.7: Sequence Diagram

CHAPTER 5

PROJECT PLAN

| Phase | Task | Description |
|---------|----------------|--|
| Phase 1 | Analysis | Analyze the information related to Project Topic |
| Phase 2 | System Design | Assign the module and design the process flow Control |
| Phase 3 | Implementation | Implement the code for all the modules and integrate all the modules |
| Phase 4 | Testing | Test the code and overall process whether the process works properly Test the code and over all process weather the process works properly |
| Phase 5 | Maintenance | Modification of a software product after delivery to improve performance or maintainability. |

5.0.1 Reconciled Estimates

5.1 PROJECT ESTIMATE

| Sr.No. | Milestone Name | Milestone Description |
|--------|-----------------------------|---|
| 1. | Requirement Analysis | Complete specification of system |
| 2. | High level design | Identify the modules and the different entities and their relationships |
| 3. | Detailed design | GUI design, program specification etc |
| 4. | Build | Writing code for different modules |
| 5. | Testing | Test the different modules together |
| 6. | Final Review and Deployment | Checking all the requirements are fulfilled |

Table 5.1: Project Estimate

5.1.1 COCOMO Model

Cocoma (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models. The key parameters which define the quality of any software products, which are also an outcome of the Cocoma are primarily Effort & Schedule:

- Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.
- Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of Cocoma have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Boehm's definition of organic, semi detached, and embedded systems:

1. Organic – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has

been solved in the past and also the team members have a nominal experience regarding the problem.

2. Semi-detached – A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environments lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.

3. Embedded – A software project requiring the highest level of complexity, creativity, and experience requirement falls under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

All the above system types utilize different values of the constants used in Effort Calculations. Types of Models: COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. Any of the three forms can be adopted according to our requirements. These are types of COCOMO model:

1. Basic COCOMO Model The first level, Basic COCOMO can be used for quick and slightly rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations.

The basic COCOMO model estimates the software development effort using only lines of code.

$$E = a(KLOC)^b$$

newline Where,

E is the efforts applied by person in months, $a = 3.0$ and $b = 1.12$, then

$$KLOC=2.25$$

Hence Efforts = $3.0 (1.8)1.12$,

$$E = 5.79 \text{ Person-month}$$

$$E = 6 \text{ Person-month}$$

Total of 6 Person-Months are required to complete the project successfully.

Department of IT Engineering 2021-22 24

$$D = cb(E)^{db}$$

Where,

D = Development time in chronological months, $cb = 2.5$ and $db = 0.35$, and $E = 6 \text{ Person-Month}$

Hence, Development Time= $2.5 (1.8)0.35$

$$D = 3.07 \text{ months}$$

The approximate duration of the project is 3 months.

$$P = E/D$$

Where,

P = Number of persons to accomplish a project.

Hence, Number of Persons required completing the project

$$P = 6/3$$

$$P = 2 \text{ persons}$$

Therefore 2 persons are required to successfully complete the project on schedule.

2. Intermediate COCOMO Model Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMO additionally accounts for the influence of individual project phases, i.e in case of Detailed it accounts for both these cost drivers and also calculations are performed phase wise henceforth producing a more accurate result. These two models are further discussed below.
3. Detailed COCOMO Model In detailed cocomo, the whole software is divided into different modules and then apply COCOMO in different modules to estimate effort and then sum the effort.

Project Cost

The model followed is the Constructive Cost Model(COCOMO) for estimating the effort required in completing the project. Like all the estimation models, the COCOMO Model requires sizing information. This information can be specified in the form of:

- Object Point
- Function Point
- Lines of source Code (KLOC) for our project, This work uses the sizing information in the form Lines of Source Code.
- Total lines of code for our project, KLOC =1.8K (approx.).
 “SECURE E WALLET ARCHITECTURE USING BCT FEATURES”
- Cost of each person per month, $C_p = \text{Rs.}11,000/-$ (Per person-month) So, $C = 3 \times C_p = 3 \times 11000 = 33,000-$

Therefore, the cost of the project is 33,000+10000(cost of camera approx) = 43,000/- (approx).

5.1.2 Reconciled Estimates

The part of the project will be hardware, which need to implement in our system on besides, also need to estimates the cost of the application which are designing keeping in mind the following factor:

- Its market demand, what it has got to offer to the customer
- Its relevance in the current world.
- The extent to which it can adhere to its objective of secured data transmission.

5.1.3 Project Resources

1. Designer: To design systems and perform requirement gathering.
2. Developer: To develop system and provide to tester for testing

5.2 RISK MANAGEMENT :

Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to the questionnaire revealed some risks. Each risk is categorized as per the categories mentioned in [?].You can refer to the following risk identification questionnaire.

1. Have top software and customer managers formally committed to support the project? Answer : Yes
2. Are end-users enthusiastically committed to the project and the system/product to be built? Answer : Yes
3. Are requirements fully understood by the software engineering team and its customers? Answer : Yes
4. Have customers been involved fully in the definition of requirements? Answer : Yes
5. Do end-users have realistic expectations? Answer : Yes
6. Does the software engineering team have the right mix of skills? Answer : Yes
7. Are project requirements stable? Answer : Yes
8. Is the number of people on the project team adequate to do the job? Answer : Yes
9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built? Answer : Yes

5.2.2 NP Hard

A problem is NP-hard if solving it in polynomial time would make it possible to solve all problems in class NP in polynomial time. Some NP-hard problems are also in NP (these are called "NP-complete"), some are not. If you could reduce an NP problem to an NP-hard problem and then solve it in polynomial time, you could solve all NP problems. Also, there are decision problems in NP-hard but are not NP-complete, such as the infamous halting problem

5.2.3 Risk Analysis

- **Technical Risk:** The probability of loss incurred through the execution of a technical process in which the outcome is uncertain. Untested engineering, technological or manufacturing procedures entail some

level of technical risk that can result in the loss of time, resources, and possibly harm to individuals and facilities. Like mobile phone battery off, network error in user and server, multiple requests at time. • Operational Risk: Operational risk is the prospect of loss resulting from inadequate or failed procedures, systems or policies. Employee errors. System failures. Fraud or other criminal activity. Any event that disrupts business processes. Like user registration, login, send request to service provider.

- Schedule Risk: Schedule risk is the risk that the project takes longer than scheduled. It can lead to cost risks, as longer projects always cost more, and to performance risk, if the project is completed too late to perform its intended tasks fully.

- Business Risk: Business risk is the possibility that a company will have lower than anticipated profits or experience a loss rather than taking a profit. Business risk is influenced by numerous factors, including sales volume, per-unit price, input costs, competition, and the overall economic climate and government regulations.

5.3 PROJECT SCHEDULE :

5.3.1 Project task set

Major Tasks in the Project stages are:

| Priority (High to low) | Risks | Back-up plan |
|------------------------------|----------|--------------|
| 1 | Schedule | Overtime |

| | | |
|---|-------------|------------|
| 2 | Operational | Validation |
| 3 | Business | Marketing |
| 4 | Technical | - |

- Task 1: Requirement Gathering
- Task 2: Literature Survey
- Task 3: System Design
- Task 4: Functionality Implementation
- Task 5: Testing

5.3.2 Timeline Chart

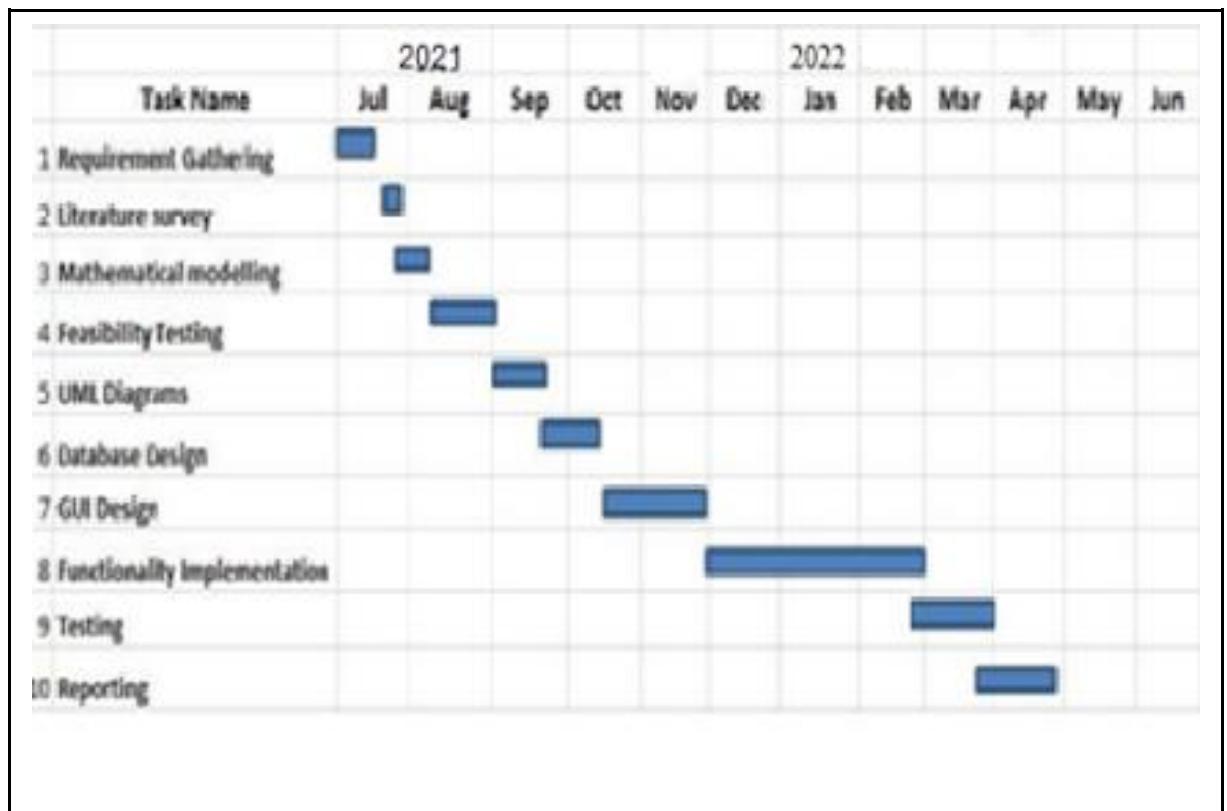


Figure 5.1: Timeline Chart

5.4 TEAM ORGANIZATION

5.4.1 Team Structure

Whatever activities are done related to the project that we all show all details log to our guide. All the reports are noted to the guide.

| Work Task | Description | Duration |
|---------------------|---|----------|
| Literature Search | Related work done for conceptual data similarity | 6 weeks |
| System analysis | Critical analysis and comparison of technologies studied and results achieved in research | 4 weeks |
| Design and Planning | Modeling and design and dataset searching or creation | 8 weeks |
| Implementation | Divided into phases | |
| Phase A | Implementation module 1 | 2 weeks |
| Phase B | Implementation module 2 | 2 weeks |
| Phase C | Implementation module 3 | 2 weeks |
| System Testing | Test system quality, fix errors if any and improve if needed. Test system for | 3 weeks |

| | | |
|----------------|-----------------------------------|---------|
| | different data sets | |
| Final Report | Prepare and upload Initial Report | 2 weeks |
| Initial Report | Prepare and upload Initial Report | 2 weeks |

Table 5.2: Timeline Chart

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

BCT: First and foremost, blockchain is a public electronic ledger built around a P2P system that can be openly shared among disparate users to create an unchangeable record of transactions, each time-stamped and linked to the previous one. Every time a set of transactions is added, that data becomes another block in the chain (hence, the name). Blockchain can only be updated by consensus between participants in the system, and once new data is entered it can never be erased. It is a write-once, append-many technology, making it a verifiable and auditable record of each and every transaction. Farmer will transfer the products to the agent through the application interface, agent in turn will transfer any product to another agent through application interface only. Also the record of each and every transaction will be maintained at different places which will maintain transparency also the database is secured through AES. System login is secured through visual cryptography.

6.2 TOOLS AND TECHNOLOGIES USED:

JDK 1.8 Installation 1. Double click jdk-8-ea-bin-b32-windows-i586 to run the installation program. JDK License dialog displayed. Accept the license in order to install JDK.

2. The JRE Custom setup dialog enables you to choose a custom directory for JRE Files.
3. The complete dialog indicates a successful installation. NetBeans IDE 7.

3.1 Installation To install the software:

1. After the download completes, run the installer. For Windows, the installer executable file has the .exe extension. Double-click the installer file to run it.
2. If you downloaded the All bundle, you can customize your “SECURE installation. Perform the following steps at the Welcome page of the installation wizard:
 - a. Click Customize.
 - b. In the Customize Installation dialog box, make your selections.
 - At the Welcome page of the installation wizard, click Next. At the License agreement page, review the license agreement, click the acceptance check box, and click Next. At the JUnit License Agreement page, decide if you want to install JUnit and click the appropriate option, click Next. At the NetBeans IDE installation page, do the following:
 - Accept the default installation directory for the NetBeans IDE or specify another directory. Note: The installation directory must be empty and the user profile you are using to run the installer must have read/ write permissions for this directory.
 - Accept the default JDK installation to use with the NetBeans IDE or select a different installation from the drop-down list. If the installation wizard did not find a compatible JDK installation to use with the NetBeans IDE, your JDK is not installed in the default location. In this case, specify the path to an installed JDK and click Next, or cancel the current installation. After installing the required JDK version you can restart the installation.
 - If you are installing Apache Tomcat, on its installation page, accept the default installation directory or specify another installation location. Click Next.
 - At the Summary page, verify that the list of components to

be installed is correct and that you have adequate space on your system for the installation. • Click Install to begin the installation. • At the Setup Complete page, provide anonymous usage data if desired, and click Finish.

MySQL Database

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

6.3 ALGORITHM DETAILS:

Algorithm:

6.3.1 AES

AES is used to encrypt the database.

The encryption process uses a set of specially derived keys called round keys. These are applied, along with other operations, on an array of data that holds exactly one block of data, the data to be encrypted. This array we call the state array.

STEPS:

- Derive the set of round keys from the cipher key.
- Initialize the state array with the block data (plaintext).

- Add the initial round key to the starting state array.
- Perform nine rounds of state manipulation.
- Perform the tenth and final round of state manipulation
- Copy the final state array out as the encrypted data (ciphertext).

6.3.2 MD5:Hash Function

Step 1. Append Padding Bits. The message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. ...

Step 2. Append Length. ...

Step 3. Initialize MD Buffer. ...

Step 4. Process Message in 16-Word Blocks. ...

Step 5. Output.

In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value.

As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. An MD5 hash is typically expressed as a 32 digit hexadecimal number.

CHAPTER 7

SOFTWARE TESTING

7.1 TYPES OF TESTING

7.1.1 Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

7.1.2 Alpha Testing

It is the most common type of testing used in the Software industry. The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user. Alpha testing is carried out at the end of the software development phase but before the Beta Testing. Still, minor design changes may be made as a result of such testing. Alpha testing is conducted at the developer's site. In-house virtual user environments can be created for this type of testing.

7.1.3 Acceptance Testing

An acceptance test is performed by the client and verifies whether the end to end flow of the system is as per the business requirements or not and if it is as per the needs of the end user. Client accepts the software

only when all the features and functionalities work as expected. It is the last phase of the testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

7.1.4 Beta Testing

Beta Testing is a formal type of software testing which is carried out by the customer. It is performed in the Real Environment before releasing the product to the market for the actual end users. Beta testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta testing is successful when the customer accepts the software. Usually, this testing is typically done by end-users or others. It is the final testing done before releasing an application for commercial purpose. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area. So the end user actually uses the software and shares the feedback to the company. Company then takes necessary action before releasing the software to the worldwide.

7.1.5 Performance Testing

This term is often used interchangeably with ‘stress’ and ‘load’ testing. Performance Testing is done to check whether the system meets the performance requirements. Different performance and load tools are used to do this testing.

7.1.6 Security Testing

It is a type of testing performed by a special team of testers. A system can be penetrated by any hacking way. Security Testing is done to check how the software or application or website is secure from internal and external threats. This testing includes how much software

is secure from the malicious program, viruses and how secure and strong the authorization and authentication processes are. It also checks how software behaves for any hackers attack and malicious programs and how software is maintained for data security after such a hacker attack.

7.1.7 White Box Testing

White Box testing is based on the knowledge about the internal logic of an application's code. It is also known as Glass box Testing. Internal software and code working should be known for performing this type of testing. Under these tests are based on the coverage of code statements, branches, paths, conditions etc.

7.1.8 Black Box Testing

Black Box testing, also known as Behavioral testing, is a software testing method in which the internal structure or design or implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box, inside which one cannot see. This method attempts to find errors like incorrect or missing functions, interface error, behavior or performance error etc.

7.1.9 Regression Testing

Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing. It is difficult to cover all the systems in Regression Testing, so typically automation testing tools are used for these types of testing.

7.1.10 System Testing

Under System Testing technique, the entire system is tested as per the requirements. It is a Black-box type testing that is based on overall requirement specifications and covers all the combined parts of a system.

7.1.11 Smoke Testing

Whenever a new build is provided by the development team then the software testing team validates the build and ensures that no major issue exists. The testing team ensures that the build is stable and a detailed level

of testing is carried out further. Smoke Testing checks that no showstopper defect exists in the build which will prevent the testing team from testing the application in detail. If testers find that the major critical functionality is broken down at the initial stage itself then the testing team can reject the build and inform the development team. Smoke Testing is carried out to a detailed level of any functional or regression testing.

7.1.12 Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing. It has two tests under it. These are- (a) Top to bottom. (b) Bottom to top. (a) Top to bottom: In this, the system is divided into different modules. Each and every module is tested from

top to bottom. (b) Bottom to top: In this type of testing, every module is tested individually and at the end all modules are integrated.

7.2 TEST CASES :

| N o. | Test Case | User Input | Expected Result | Actual Result | Stat us |
|------|---------------|--|--|--|---------|
| 1. | Register User | User gives all the credentials asked by the system | Registration successful | Registration successful | Pass |
| 1. | Register User | If User miss any information to enter | Registration Failed | Registration Unsuccessful please Try again | Pass |
| 2 . | Login | System takes the username and password | Login successful | Successful | Pass |
| 2 . | Login | If incorrect information is entered | Login failed | Login failed, please give correct password or username | Pass |
| 3. | Browse Image | Click on browse Button | Unless image is selected button should remain disabled | Button is disabled Unless proper image is selected | Pass |

| | | | | | |
|--------|--------------------------------|------------------------------|--|---|----------|
| 4 . | Browse Button | Click on browse Button | Browse File Dialog should open | Browse file dialog opened successful ly | Pas s |
| 5 . | Select non image file | Select image | Error message to be shown | Error message Is seen | Pas s |

Table 7.1: Test Cases

CHAPTER 8

RESULTS

8.1 SCREENSHOTS :

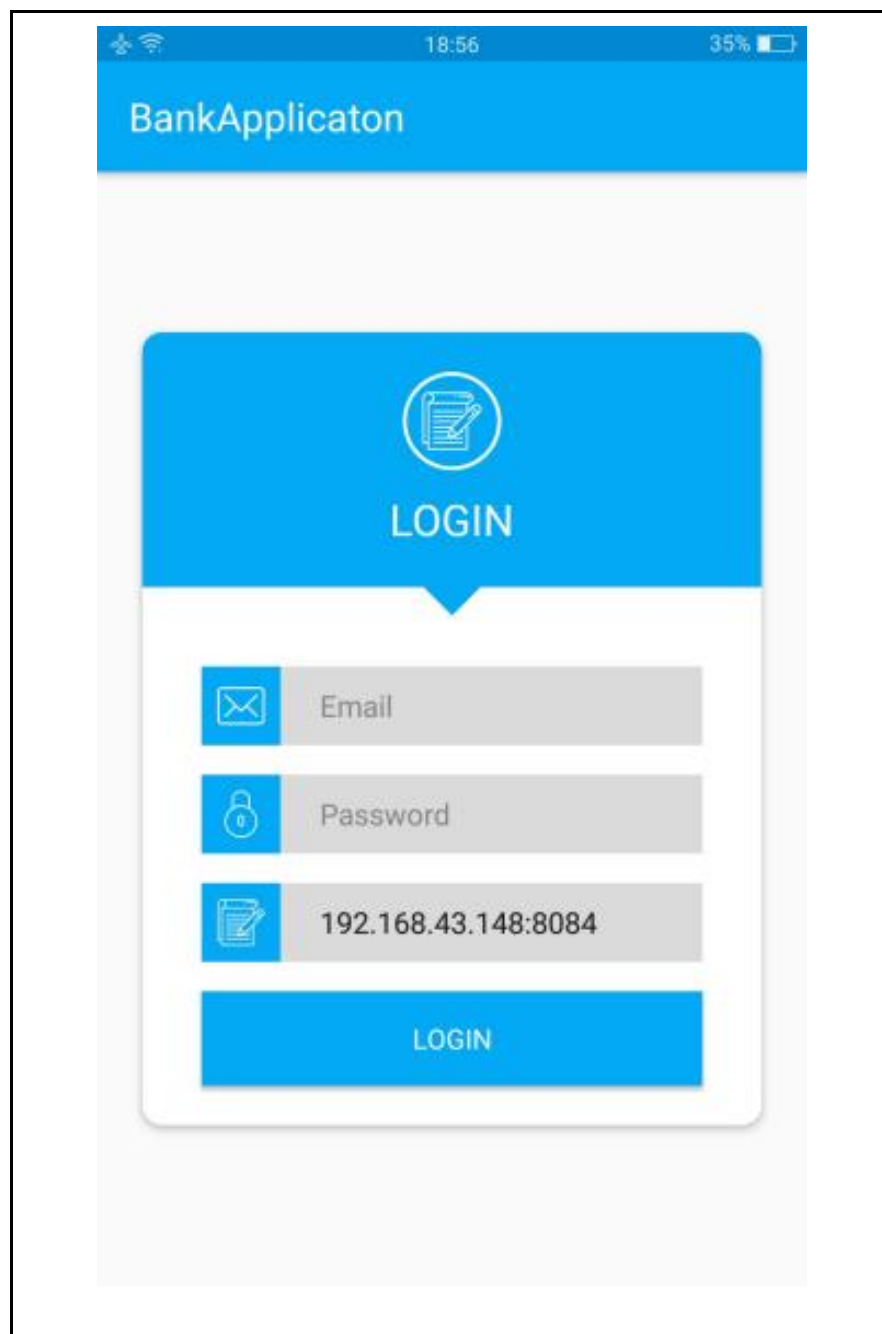


Figure 8.1: User Login

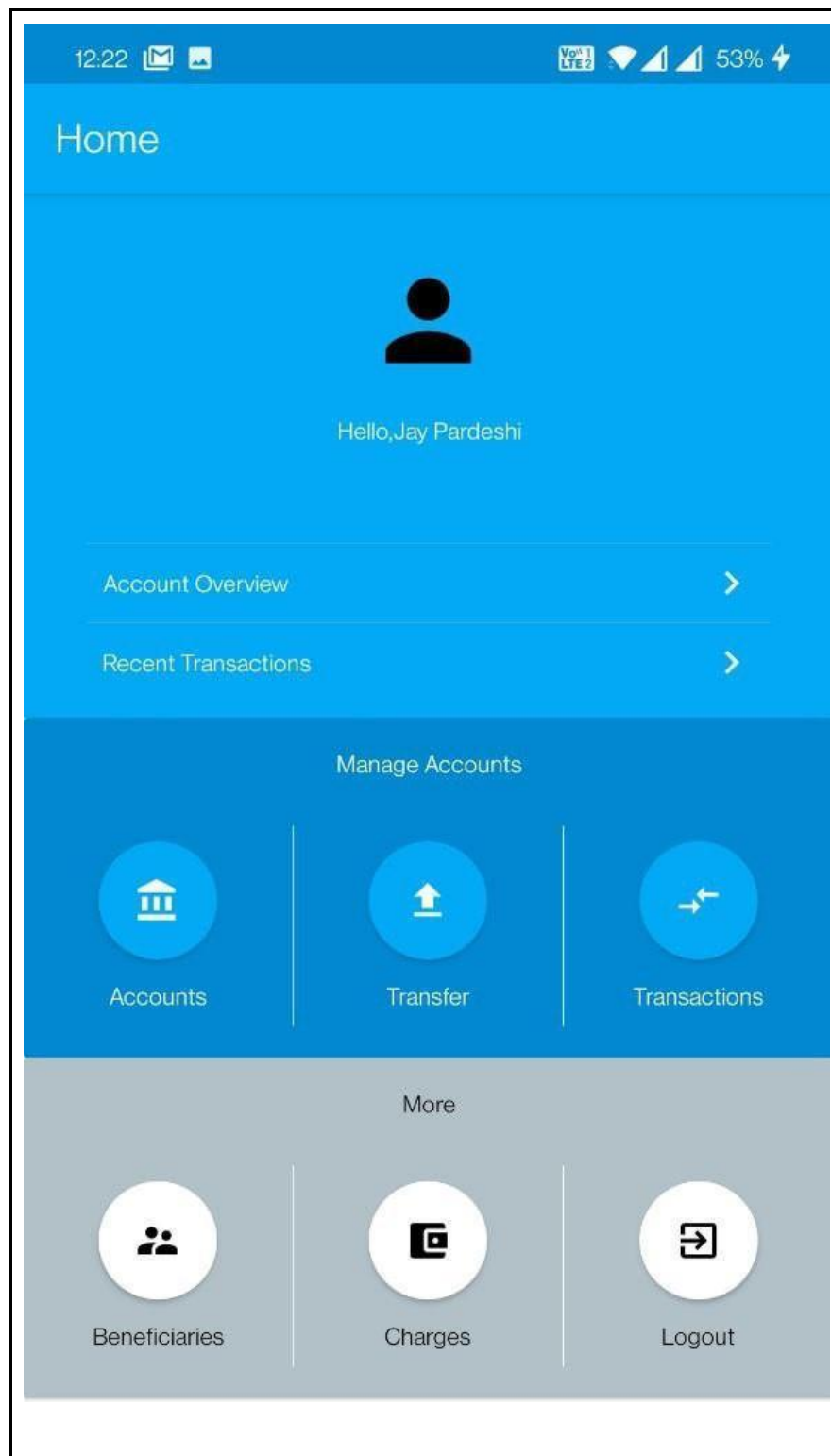


Figure 8.2: User Home

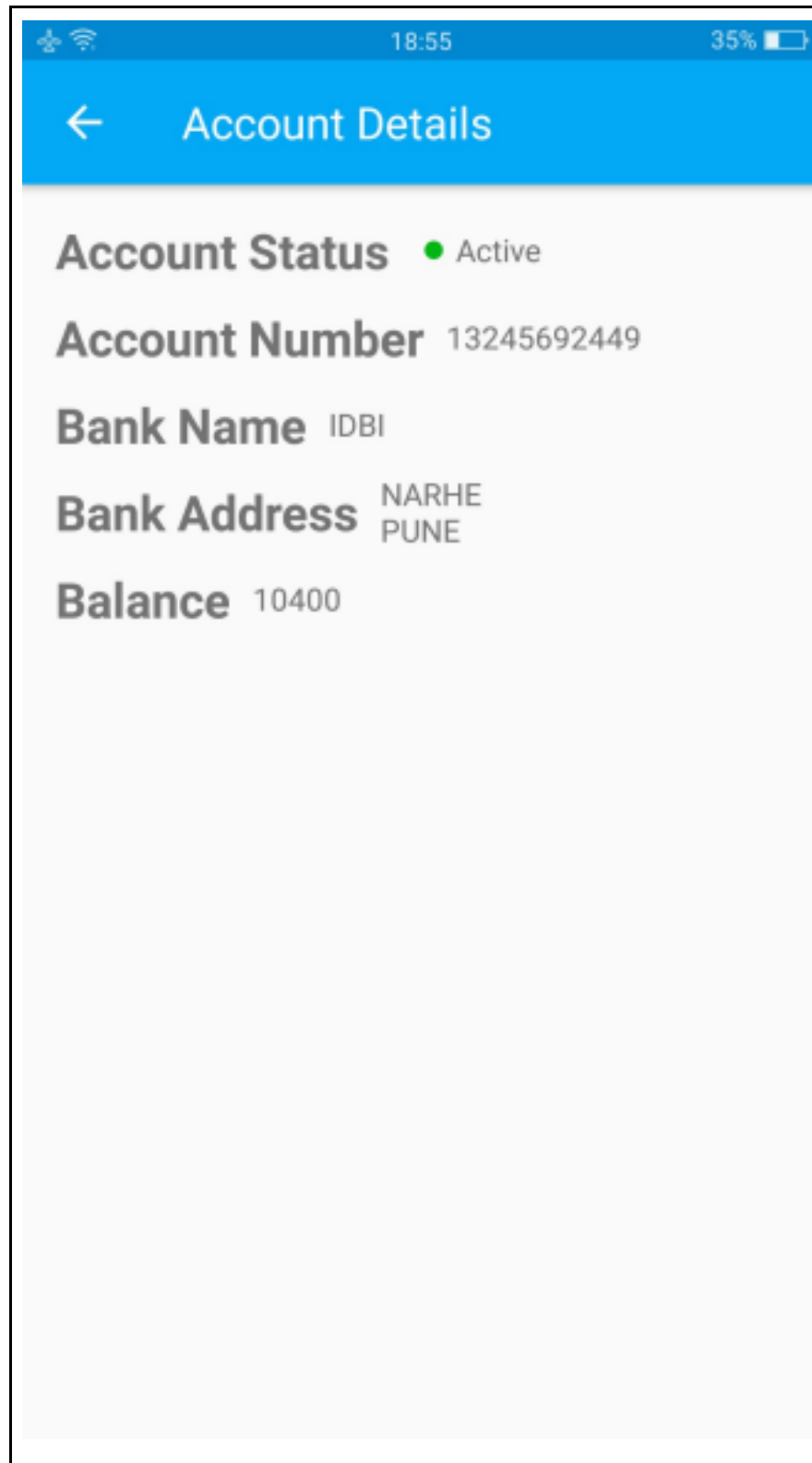


Figure 8.3: User Account

← Add Beneficiary

Select Account Type*

Savings ▼

Account Number

IFSC

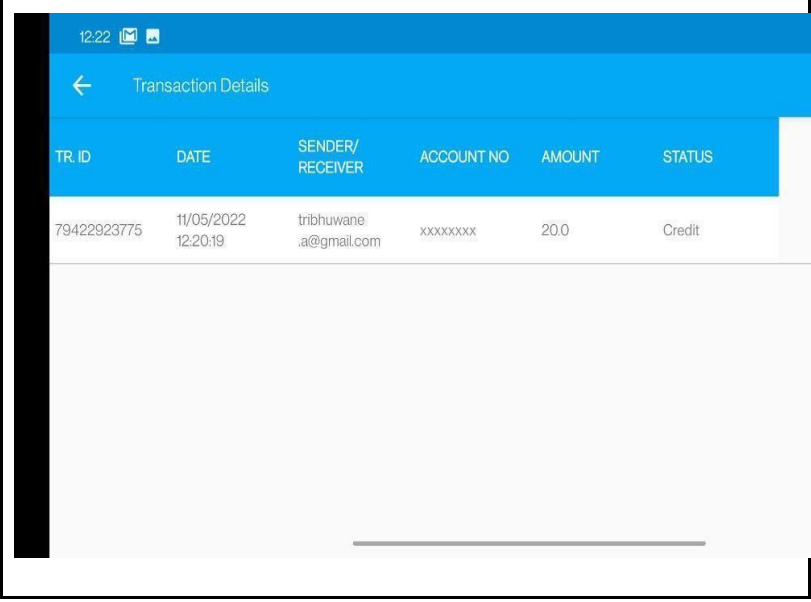
Beneficiary Name

Beneficiary Email

Beneficiary Address

SUBMIT BENEFICIARY

Figure 8.4: User Add Beneficiary

A screenshot of a mobile application interface titled "Transaction Details". The interface features a blue header bar with a back arrow and the title. Below the header is a table with six columns: TR. ID, DATE, SENDER/RECEIVER, ACCOUNT NO, AMOUNT, and STATUS. The table contains one data row. The status "Credit" is highlighted in green. The bottom of the screen shows a white home indicator bar.

| TR. ID | DATE | SENDER/ RECEIVER | ACCOUNT NO | AMOUNT | STATUS |
|-------------|------------------------|----------------------------|------------|--------|--------|
| 79422923775 | 11/05/2022 12:20:19 | tribhuwane .a@gmail.com | xxxxxxxx | 20.0 | Credit |

Figure 8.5: User Transactions

CHAPTER 9

CONCLUSION

Thus we have implemented a prototype web based software application in Java for application of BCT for a cashless economy . We have implemented block chain features such as: 1. Decentralization 2. Visual Cryptography 3. Hash Algorithm 4. Encrypted Database. Thus it is possible to track every transaction in a cashless system using BCT. Also the system can be transparent using BCT.

9.1 FUTURE SCOPE

In future we will try for sponsorship from the government and will implement a project on a large scale with some domain and hosting space online.

9.2 APPLICATIONS

1. Farmers
2. Government Organizations
3. Banking Sector.
4. Educational System

CHAPTER 10

REFERENCES

1. F. Lv and S. Chen, "Research on Establishing a Traceability System of Quality and Safety of Agricultural Products Based on Blockchain Technology," Rural Finance Research, vol. 12, pp. 22-26, 2016
2. Y. Yang and Z. Jia, "Application and Challenge of Blockchain Technology in the Field of Agricultural Internet of Things,— Information Technology, vol. 258, pp. 24-26, 2017.
3. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system,— Consulted, 2008.
4. Y. Yuan and F. Y. Wang, "Blockchain: The State of the Art and Future Trends," Acta Automatica Sinica, 2016.
5. Y. Yuan, T. Zhou, A. Y. Zhou, Y. C. Duan, and F. Y. Wang, "Blockchain Technology: From Data Intelligence to Knowledge Automation," Zidonghua Xue bao/acta Automatica Sinica, vol. 43, pp 1485-1490, 2017.
6. Y.-b. Zhang, "The New Ecosystem of Cross-border E-commerce between EU and China based on Blockchain," China Business And Market, vol. 32, pp. 66-72, 2018.
7. T. Hong, "Accelerating the Application of Blockchain in the Field of Agricultural Products E - commerce in China," Journal of Agricultural Information, pp. 18-20, 2016.

8. Y. Yuan and F.-Y. Wang, "Parallel Blockchain: Concept, Methods and Issues," IEEE Acta Automatica Sinica, vol. 43, pp. 1703-1712, 2017.
9. Andreas M A. Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly Media, 2014
10. Jerry B, Andrea C. Bitcoin: A Primer for Policymakers. Mercatus Center, George Mason University, 2013.

ANNEXURE A

- Definitions: P, NP, NP-Hard, NP-Complete Problems:
- P Class of problems: Solutions to P class of problems have deterministic algorithms running in polynomial.
- NP Class of problems: Solutions to NP class of problems have non-deterministic algorithms running in polynomials.
- NP-Hard class of problems: A problem is in NP-Hard class if an already proved NP-Hard problem reduces to it.
- NP-Complete class of problems: A problem is NP-Complete if it is NP-Hard and it is NP (i.e. there exists a non-deterministic algorithm running in polynomial time which solves it).

Therefore, our system is NP-Complete. Hence it Is Feasible.

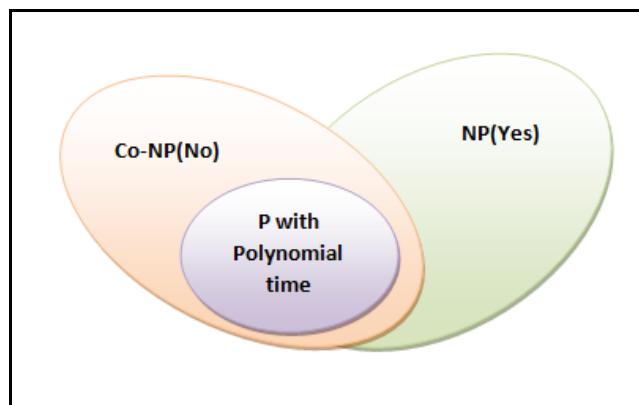


Figure A.1: NP Problem

- What is NP? • "NP" means "we can solve it in polynomial time if we can break the normal rules of step-by-step computing".

- What is NP-Complete? • Since this amazing "N" computer can also do any thing a normal computer can, we know that "P" problems are also in "NP". • So, the easy problems are in "P" (and "NP"), but the really hard ones are **only** in "NP", and they are called "NP-complete". • It is like saying there are things that People can do ("P"), there are things that Super People can do ("SP"), and there are things **only** Super People can do ("SP-complete").

- NP-Complete: We have use Bloom filtering for detection of packet drop attack whether it is drop by itself or by hacker. Hence the 'P' is NP-Complete in this case.

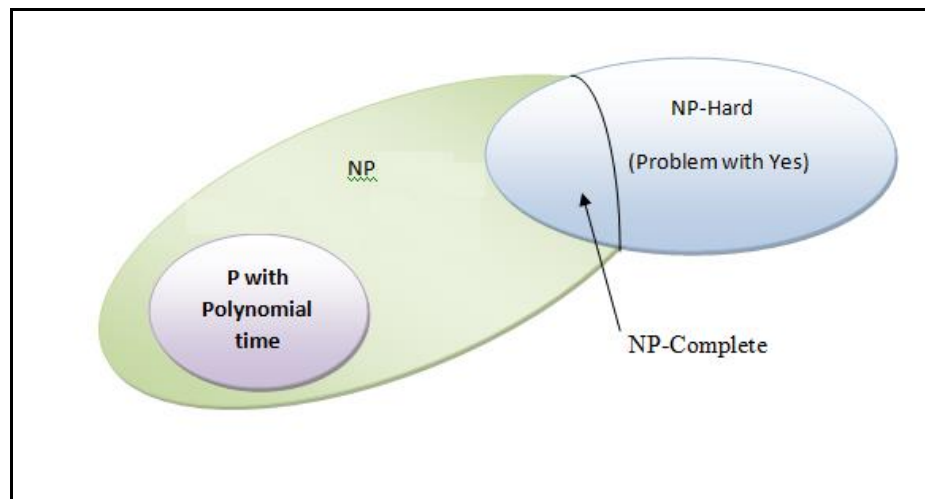


Figure A.2: NP Problem Complete