**TABLE OF CONTENTS**

**1. INTRODUCTION**…………………………………………………………1

1.1.Background…………………………………………………………….1

1.2.Problem Definition……………………………………….………………………..2

[1.3.Existing solutions……………………………………...……………….3](#_Toc497666917)

1.4.Proposed System………………………………………........................3

1.5.Scope…………...……………………………………………………....4

[1.6.Advantages………………………………………………...…………...3](#_Toc497666920)

1.7.Technology Overview………………...……………………………......5

**[2. LITERATURE REVIEW](#_Toc497666923)**[……………………………………………….12](#_Toc497666923)

**3. METHODOLOGY**………………………………………………………16

3.1.Introduction…………………………………………………………...16

3.2. Types of Techniques…………………………………………………17

3.2.1. Secure and Random Distribution of Exam Questions………...17

3.2.2. Preventing the “Unattended Exam” Issue .…………………....18

3.2.3. Preventing Students from Exchanging Mobile/Tablet Device..18

[3.3.Input/output design…………………………………………………...21](#_Toc497666926)

3.4. System Study…………………………………………………….……21

**4. DESIGN**…………………..………………………………………………..25

[4.1.Identifying Design goals](#_Toc497666937)……………………………………………...25

[4.2.UML Diagrams](#_Toc497666937)……….……………………………………………...25

4.2.1.Use Case Diagram…..………………...…..................................28

4.2.2.ClassDiagram…………………………………………….…....29

4.2.3.SequenceDiagram………………………………………..........30

4.2.4.Activity Diagra………………………………………………...31

**5. DEVELOPMENT AND IMPLEMENTATION**……………………...33

5.1.MODULAR APPROACH…………………………………………..33

5.2 IMPLEMENTATION……………………………………………….35

5.2.1. System Requirements Specifications......……………………35

5.2.1.Functional Requirements…..……………………....………..35

5.2.2.Hardware Requirements……………….……………………35

5.2.3.Software Requirements….……………….…………………35

5.3.Input/Output - Screenshots…………………………………………35

5.4. Sample code…………….…………………………………………35

6. TESTING………………………………………………………………..36

6.1. Types Testing…….……..……….……...…………………..............36

6.2 Test cases…..……………………………………………………......38

**7. CONCLUSION**……………………………………………………………....68

**REFERENCES & BIBLOGRAPHY**..............................................................69

**ABSTRACT**

This project implements a purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. **INTRODUCTION**
   1. **BACKGROUND**

Bitcoin is a cryptocurrency that has recently emerged as a popular medium of exchange, with a rich and extensive ecosystem. Bitcoin is a consensus network that enables a new payment system as completely digital money. Bitcoin uses peer-to-peer (p2p) technology, and it operates without any trusted third-party authority that may appear as a bank, a chartered accountant (ca), a notary, or any other centralized service. In particular, an owner has full control over its bitcoins, and she could spend them anytime and anywhere without involving any centralized authority. Bitcoin design is open-source, and nobody owns or controls it. The concepts of Bitcoin were conceived in the month of January, 2009 by a researcher going by the name ‘Satoshi Nakamoto’ pseudonymously. The open source project known as Bitcoin was created on the proof-of-work principle that transactions can be securely processed on a decentralized peer to peer network without the need for a central clearinghouse. Bitcoins are controlled by the user of the currency around the world. Bitcoin operates as a p2p file sharing protocol and it is based on SHA-256 algorithm. Bitcoin coin (BTC) is essentially a hashed chain of digital signatures based upon asymmetric or public key cryptography. Each participating Bitcoin address in the P2P network is associated with a matching public key and private key wherein a message signed by private key can be verified by others using the matching public key. The public transaction history also allows network analysis to cluster some groups of public Bitcoin pseudonyms into individual users. Real time analysis of the Bitcoin peer‑to‑peer network then provides enough data to identify a portion of users with specific IP addresses. Existing security solutions for Bitcoin lacks the required measures that could ensure an adequate level of security for its users. We believe that security solutions should cover all the major protocols running critical functions in Bitcoin, such as blockchain, consensus, key management, and networking protocols. Blockchain was initially put forward as an underlying technical framework of Bitcoin. Although due to the excessive value fluctuation and supervisory management reasons, Bitcoin was once forbidden or restricted to a variable extent in China, Russia, Europe and some other countries. The confidentiality, security and reliability of blockchain technology are realized by the public. Blockchain is considered a bran-new data storage, transmission and management mechanism because it realizes reliable transfer of data and value in a decentralized way, without the need of any trusted third-party organization. Bitcoin derives its reliability from a public record of valid past transactions. Users check against this history to verify new transactions. Once it has been verified, it is then added to the block chain. Because this history is just a record of past events, disagreements over transactions are possible. In this paper, we will have a quick study about what is blockchain then we'll discuss different application in blockchain and what service do they offer at the end, we shall talk about the security issues and those challenges we need to overcome.

## 1.2. PROBLEM DEFINITION

Bitcoin can be considered as an area for further research. The technology is still under development and can be substantially improved upon. Technical problems such as hardware crash of a node can result into a lost wallet and orphan bit coins. This is important considering the maximum number of bit coins are fixed to value of millions. Also certain issues such as deflation of currency, value fluctuations are considered as major impedance in making the currency mainstream spent the same bit coins in multiple transactions; send two conflicting transactions in rapid succession. Combination of the double spending and the Finney attack. improves on adverse effects of selfish mining and block with holding attack.

* Sizes
* Densities
* Brute force attack
* Double spending or Race attack
* Finney attacks.
* Required to specify the number of blocks , in advance
* Unable to handle erroneous data

## EXISTING SOLUTIONS

Bitcoin’s blockchain uses High energy consumption to achieve distributed consensus in the network. Although, the use of High energy consumption makes the mining process more resistant to various security threats such as sybil and double spending, it consumes a ridiculous amount of energy and computing resources. Wallets can be lost Since there is no trusted third party if a uses lost the private key associated with her wallet due to a hard drive crash or a virus corrupts data or lost the device carrying the key, all the bitcoins in the wallet has been considered lost for forever. There is nothing that can be done to recover the bitcoins, and these will be forever orphaned in the system. This can bankrupt a wealthy Bitcoin investor within seconds. (Facilitate) Criminal activity The considerable amount of anonymity provided by the Bitcoin system helps the would-be cyber criminals to perform various illicit activities such as ransomware, tax evasion, underground market, and money laundering

## 

## PROPOSED SYSTEM

Bitcoin and block chain technology, implementation of block chain and storing block information like ip address, mac address. Storing information about timestamp for the block formation. All information hashing. Studying to strengthen security using these characteristics of blockchain. The most important part of the blockchain is security related to the personal key used in encryption and there are studies on how to protect the personal key. To solve this problem, studies on applying both hardware and software securities for approving transactions are ongoing. Blockchain technologies solved the trust Anonymity problem between node to node, so data transfer or even transaction can be anonymous, only need to know the person's blockchain address. Decentralized. The basic feature of blockchain, means that blockchain doesn't have to rely on centralized node anymore, the data can be record, store and update distributed. Transparent. The data's record by blockchain system is transparent to each node, it also transparent on update the data that is why blockchain can be trusted.

## SCOPE

## The scope of the project is Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

## ADVANTAGES

* **TRANSPARENCY**

One of the prime reasons blockchain is intriguing to businesses is that this technology is almost always open source. That means other users or developers have the opportunity to modify it as they see fit. But what's most important about it being open source is that it makes altering logged data within a blockchain incredibly difficult. After all, if there are countless eyes on the network, someone is probably going to see that logged data has been altered. This makes blockchain a particularly secure technology.

* **REDUCED TRANSACTION COSTS**

As noted, blockchain allows peer-to-peer and business-to-business transactions to be completed without the need for a third party, which is often a bank. Since there's no middleman involvement tied to blockchain transactions, it means they can actually reduce costs to the user or businesses over time.

* **FASTER TRANSACTION SETTLEMENTS**

When it comes to traditional banks, it's not uncommon for transactions to take days to completely settle. This is due to protocols in bank transferring software, as well as the fact that financial institutions are only open during normal business hours, five days a week. You also have financial institutions located in various time zones around the world, which can delay processing times. Comparatively, blockchain technology is working 24 hours a day, seven days a week, meaning blockchain-based transactions process considerably more quickly.

* **DECENTRALIZATION**

Another central reason blockchain is so exciting is its lack of a central data hub. Instead of running a massive data center and verifying transactions through that hub, blockchain actually allows individual transactions to have their own proof of validity and the authorization to enforce those constraints. With information on a particular blockchain piecemealed throughout the world on individual servers, it ensures that if this information fell into unwanted hands (e.g., a cyber-criminal), only a small amount of data, and not the entire network, would be compromised.

* **USER-CONTROLLED NETWORKS**

Lastly, cryptocurrency investors are tend to be really encouraged by the control aspect of blockchain. Rather than having a third party run the show, users and developers are the ones who get to call the shots. For instance, an inability to reach an 80% consensus on an upgrade tied to bitcoin's blockchain is what necessitated a fork into two separate currencies (bitcoin and bitcoin cash) more than four months ago. Having a say goes a long way with investors and developers.

## TECHNOLOGY OVERVIEW

Java is a simple and yet powerful object oriented programming language and it is in many respects similar to C++. Java originated at Sun Microsystems, Inc. in 1991. It was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. It was developed to provide a platform-independent programming language.

**NetBeans**

The NetBeans IDE is open source and is written in the Java programming language. It provides the services common to creating desktop applications such as window and menu management, settings storage and is also the first IDE to fully support JDK 5.0 features. The NetBeans platform and IDE are free for commercial and non-commercial use, and they are supported by Sun Microsystems.

**Database Support**

* Database schema browsing to see the tables, views, and stored procedures defined in a database
* Database schema editing using wizards
* Data view to see data stored in tables
* SQL and DDL command execution to help you write and execute more complicated SQL or DDL commands
* Migration of table definitions across databases from different vendors
* Works with databases, such as MySQL, PostgreSQL, Oracle, IBM DB2, Microsoft SQL Server, PointBase, Sybase, Informix, Cloudscape, Derby, and more

The NetBeans IDE also provides full-featured refactoring tools, which allow you to rename and move classes, fields, and methods, as well as change method parameters. In addition, you get a debugger and an Ant-based project system.

2. LITERATURE REVIEW

# 

**G. Varriale, “Bitcoin: how to regulate a virtual currency,” International Financial Law Review, 2013, 32(6), pp: 43-45.** The purpose of the paper is to determine the future role of virtual currencies. This paper indicates their pros and cons as alternatives to “real” money and explains their appearance as the reflection of the present trends. It also presents the possible scenarios of their development. The paper is based on the former foresight research results and literature review. It highlights the main trends in contemporary economy and their impact on financial services. The Bitcoin case is the starting point for the virtual currencies’ market analysis and construction of possible market changes scenarios.

**N. Satoshi, “Bitcoin: A peer-to-peer electronic cash system,” Consulted, 2008, pp:1-9.** A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.

**M. Swan, “Blockchain: Blueprint for a New Economy,” O'Reilly Media, Inc.; 2015.**Bitcoin is starting to come into its own as a digital currency, but the blockchain technology behind it could prove to be much more significant. Concepts, features, and functionality of Bitcoin and the blockchain Using the blockchain for automated tracking of all digital endeavors Enabling censorship resistant organizational models Creating a decentralized digital repository to verify identity Possibility of cheaper, more efficient services traditionally provided by nations Blockchain for science: making better use of the data-mining network Personal health record storage, including access to one’s own genomic data Open access academic publishing on the blockchain

**G. Paul, P. Sarkar, S. Mukherjee, “Towards a More Democratic Mining in Bitcoins” In: Prakash A, Shyamasundar R, editors. Information Systems Security. vol. 8880 of Lecture Notes in Computer Science. Springer International Publishing, 2014. pp: 185-203.** Bitcoin is a peer-to-peer electronic cash system that uses a decentralized architecture. It has enjoyed superiority compared to other cyptocurrencies but it has also attracted attackers to take advantage of the possible operational insecurity. All the Bitcoin miners independently try to find the winning block by finding a hash lower than a particular target.The proposed approach does not rely on finding a hash value lower than the target, rather it awards the miner involved in generating the minimum hash value across the entire distributed network. Fraudulent activities easily get caught due to effective use of the Timestamp. The new scheme thus introduces fair competition among the miners. Moreover, it facilitates the generation of Bitcoins at a fixed rate. Finally, we calculate and show how the new scheme can lead to an energy-efficient Bitcoin.

**L. Wang, Y. Liu, “Exploring Miner Evolution in Bitcoin Network,” In: Mirkovic J, Liu Y, editors. Passive and Active Measurement. vol. 8995 of Lecture Notes in Computer Science. Springer International Publishing, 2015, pp: 290-302.** Bitcoin, a peer-to-peer network based crypto digital currency, has attracted a lot of attentions from the media, the academia, and the general public. A user in Bitcoin network can create Bitcoins by packing and verifying new transactions in the network using their computation power. Driven by the price surge of Bitcoin, users are increasingly investing on expensive specialized hardware for Bitcoin mining. To obtain steady payouts, users also pool their computation resources to conduct pool mining.

**3. METHODOLOGY**

**3.1. INTRODUCTION**

E-learning or electronic learning is learning through electronic technologies such as Online Forum, Video Conference, Web or Learning Management System (LMS) to access educational course, degree or program outside the traditional classroom. Several benefits of e-learning are its cost effective and saves time, learning 24/7 anywhere. M-learning or mobile learning is education using personal mobile devices which are further connected to Internet to obtain learning materials through mobile apps, social interactions and online educational hubs. It is flexible, allowing students to access education anywhere, anytime. Mobile learning helps educational institutes to deliver knowledge and educational contents to students on any platform and at any time. Students use mobile apps and tools to upload assignments, download course instruction and interact through online social groups to complete tasks. Examination is an assessment to measure knowledge or skills in a given area. An examination is conducted by schools, colleges, universities to assess the understanding capability of the student. Examinations are necessary as it constraints students to learn not only the subject of interest but also the subjects that are very important in the modern world. Since a very long time, pen project based method is used to conduct examination. Students’ mobile/tablet devices or computer/ laptop are connected to the school’s Wi-Fi or LAN network through which they may illegally exchange information during an exam. Applying simple policies, such as turning the network down during exams to cut off any possible communication between students, is not a practical solution as students in

Different classes may not take their exams at the same time. Moreover, the network has to be up during exams in order to be able to submit students’ answers to the Exam Server. A dynamic network access policy has to be generated and applied on each student’s device according to predefined conditions. Employing an identity based firewall with dynamic access policy seems to be a good solution to be adopted in such a scenario

The Quiz Engine embedded in Moodle is not built based on Service Oriented Architecture. It is implemented as a bulk of PHP code which has to be accessed through standard web browsers that are a bit slow on mobile devices and cannot address the exam security issues that exist in m learning environment. Moodbile services extension to Moodle does not touch the Moodle’s Quiz Engine. Thus, we need to develop a new Quiz Engine that can be deployed as a service oriented application, so that its services can be consumed by a mobile application designed to cater to m-learning specific security requirements. As well, it should be Integra table with Moodle/Moodbile in order to have a complete LMS which suites the m-learning environment and addresses all of its security issues. The core services of the proposed Exam Engine are discussed below.

**Blockchain Formation:**

For each transactions blocks will be formed. Every block stores 3 types of informations. Main data. Depending on what service is this blockchain applicate, for example: transaction records, bank clearing records, contract records or IOT data record.

**Hash:**

When a transaction executed, it had been hash to a code and then broadcast to each node. Because it could be contained thousands of transaction records in each node's block, blockchain used Merkle tree function to generate a nal hash value, which is also Merkle tree root. This nal hash value will be record in block header (hash of current block), by using Merkle tree function, data transmission and computing resources can be drastically reduced.

**Timestamp:** Time of block generated

**Tamper-proofing:**The advantage of tamper-proofing is achieved by the unique date structure and data writing mechanism of blockchain. Once a record, which is known as a transaction is being created in the chained data structure of blockchain, a new timestamp will be recorded at the same time.

**Disaster Recovery:**

Blockchain performs data recording and storing synchronously at all users’ side by constructing open source sharing protocols. Unlike the traditional centralized database which stores data in one or several centers. In an application built on blockchain, every user has the right to generate data and keep a full copy of data. This mechanism may cause redundancy to some extent, but reliability and fault tolerance capability of the network is improved.

**Privacy Protection:**

Blockchain adopts asymmetric encryption mechanism to enable users to encrypt data with their own private key. Moreover, the hash value of a user’s public key is calculated and perform as the ID indicator of the user. On one hand, the hash value has no relation with the real identity of user, thus keeping user’s personal privacy information safe. On the other hand, the process of calculating hash value is invertible, which means an adversary can’t figure out a user’s public key from the public user address, and calculating the private key from the public key is impossible.

**3.2. SYSTEM STUDY**

**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.

Three key considerations involved in the feasibility analysis are

**ECONOMICAL 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.

**4. DESIGN**

**4.1. Identifying Design Goals**

There are several reasons to identify the design goals of any system. These goals will help to design the system in an efficient manner. There are several criteria to identify these goals. Some of the criteria were explained below:

1. **Performance criteria**
2. **Response time:** The response time of the method is very low because the system simple design developed on the high performance system.
3. **Throughput:** The throughput of the system is high.
4. **Memory:**Memory used by the system is very low.
5. **Dependability criteria:**
6. **Robustness:** the system should be designed to work efficiently on images of any type of formats without any problem.
7. **Availability:** the system should be ready to accept command from user at any point of time.
8. **Fault Tolerance:** the system should not allow the user to work with fault input. It displays error messages foe every specific fault occurred.
9. **Maintenance criteria :**
10. **Portability:**The system should work on all the platforms like linux, windows, android .
11. **Readability:** the code generated should be able to understand the purpose of the project, so as to make the user to make the modifications easily.
12. **Traceability:** the code generated should be easy to map with the functions with the operations selected by the user.
13. **End-user criteria:**
14. **Utility:** the system should be made to operate on al inputs of end-user under any kind of circumstances. It should complete all the commands or instructions given by user without any interruptions.
15. **Usability:** the interface of the user is to be defined with all options which make the work of the end-user easier.

**4.2. UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creatingmodels of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**4.2.1. USECASE DIAGRAM:**

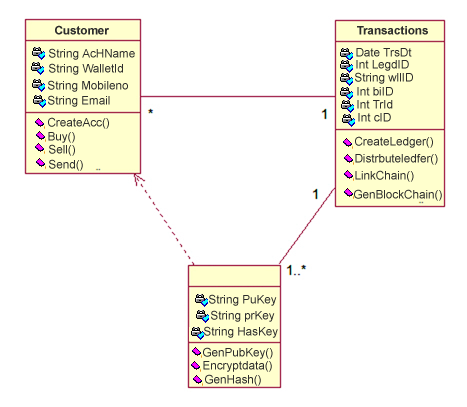
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

User

User

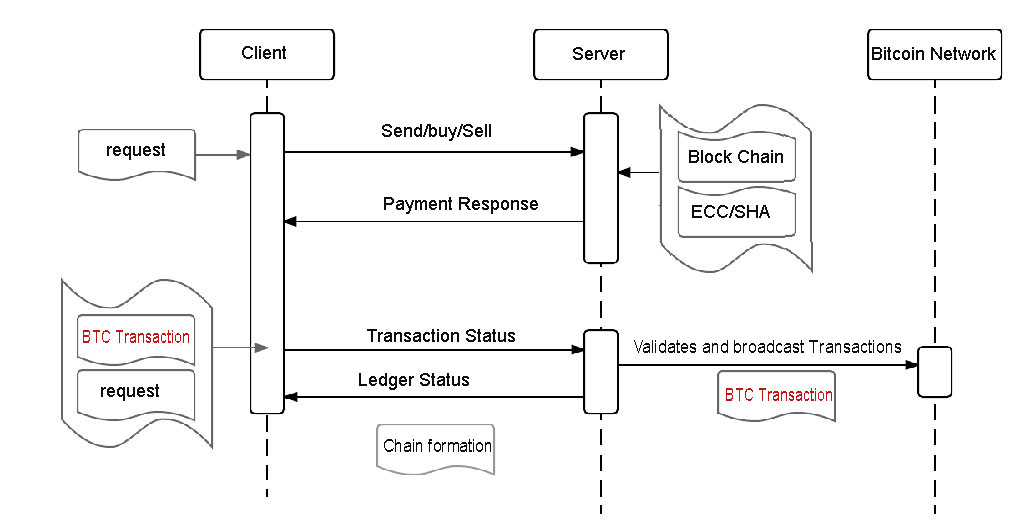
**4.2.2. CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**4.2.3. SEQUENCE DIAGRAM**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**4.2.4. ACTIVITY DIAGRAM:**

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

Registration

Login

Login check

Select Transaction

Block Generate

Transaction Complete

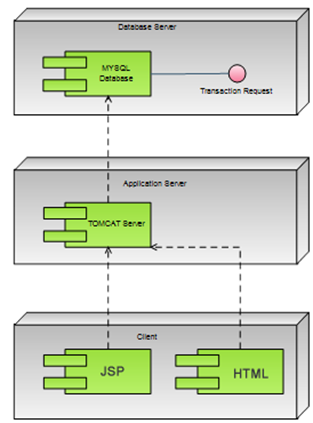
Logout

Distribute Ledger

Link to ledger

**4.2.5. DEPLOYMENT DIAGRAM:**

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

****

1. **DEVELOPMENT AND IMPLEMENTATION**

**5.1 MODULAR APPROACH**

**BLOCKCHAIN FORMATION**

For each transaction blocks will be formed. Every block stores 3 types of informations. Main data. Depending on what service is this blockchain applicate, for example: transaction records, bank clearing records, contract records or IOT data record?

**Hash.**

When a transaction executed, it had been hash to a code and then broadcast to each node. Because it could be contained thousands of transaction records in each node's block, blockchain used Merkle tree function to generate a nal hash value, which is also Merkle tree root. This nal hash value will be record in block header (hash of current block), by using Merkle tree function, data transmission and computing resources can be drastically reduced.

**Timestamp:** Time of block generated.

**Tamper-proofing:**The advantage of tamper-proofing is achieved by the unique date structure and data writing mechanism of blockchain. Once a record, which is known as a transaction is being created in the chained data structure of blockchain, a new timestamp will be recorded at the same time.

**Disaster Recovery:**

Blockchain performs data recording and storing synchronously at all users’ side by constructing open source sharing protocols. Unlike the traditional centralized database which stores data in one or several centers. In an application built on blockchain, every user has the right to generate data and keep a full copy of data. This mechanism may cause redundancy to some extent, but reliability and fault tolerance capability of the network is improved.

**Privacy Protection:**

Blockchain adopts asymmetric encryption mechanism to enable users to encrypt data with their own private key. Moreover, the hash value of a user’s public key is calculated and performs as the ID indicator of the user. On one hand, the hash value has no relation with the real identity of user, thus keeping user’s personal privacy information safe. On the other hand, the process of calculating hash value is invertible, which means an adversary can’t figure out a user’s public key from the public user address, and calculating the private key from the public key is impossible.

**5.1.1. DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

****

**5.1.2 BLOCK DIAGRAM OF BLOCKCHAIN**



**5.2 SYSTEM REQUIREMENTS SPECIFICATIONS**

Following are the hardware and software requirements used.

**5.2.1 FUNCTIONAL REQUIREMENTS:**

**Functional requirements**

In software engineering a functional requirement defines a software system or its component. A function is defined as a set of inputs the behavior and outputs. Functional requirement may be calculations, technical details, data manipulation and processing and also specify what a system is supposed to be accomplished. Here register all the types of users but all the users should not be registered with same user name.

INPUT: No. of bitcons, Sell, buy, Send, username, password

OUTPUT: BTC Network details, Ledger details and Transactions details

PROCESS: Block chain formation, Hasing, encryption ledger distribution

STORAGE: users, transaction details, ledger details and evaluation reports

**Non-Functional Requirements**

A non- functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design.

The major non-functional Requirements of the system are as follows.

**Usability**

The system is designed with completely automated process hence there is no or less user intervention.

**Reliability**

The system is more reliable because of the qualities that are inherited from the Chosen platform. The code built by using java is more reliable.

**Performance**

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

**Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

**5.2.2 HARDWARE REQUIREMENTS:**

The selection of hardware is very important in the existence and proper working of any software

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

**5.2.3. SOFTWARE REQUIREMENTS:**

One of the most difficult tasks is that, the selection of the software once system requirement is known is determining whether a particular software package fits the requirements. Here are some of the software used in this project.

* Operating system : Windows 7.
* Coding Language : JAVA,jsp
* Database : Mysql
* IDE : NetBeans
* Server : Apache

**5.3 INPUT/OUTPUT- SCREENS**

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**5.4 SAMPLE CODE**

SAMPLE CODE

<title></title>

<%@ page import="java.sql.\*"%>

<%@ page import="java.util.\*" %>

<%

Connection connection = null;

try {

Class.forName("com.mysql.jdbc.Driver");

connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/crosspi","root","root");

String sql="";

}

catch(Exception e)

{

System.out.println(e);

}

%>

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\*

\* @author RenownTechnologies

\*/

public class FruiUMP extends HttpServlet {

/\*\*

\* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

\* methods.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

public void frui(int smf,int smt,int smr, int sms,double m)

{

}

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet FruiUMP</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Servlet FruiUMP at " + request.getContextPath() + "</h1>");

out.println("</body>");

out.println("</html>");

}

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

/\*\*

\* Handles the HTTP <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

**6. TESTING**

# 6.1. TYPES OF TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS**

**Unit testing**

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

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

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

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Testing**

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

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

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

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

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**6.2 Test Case Reports**

|  |  |
| --- | --- |
| Test Case# : 1 | Priority(H,L): High |
| Test Objective: To let the user enter the details for users. | |
| Test Description: details for users is tested against standard formats. | |
| Requirements Verified: details for users is checked | |
| Test Environment: JSP, Servlets | |
| Test setup or Pre-conditions: User should enter some thing | |
| Actions | Expected Results |
| details for users  Correct Id | A message “Check your details for users” is displayed.  Just check and leave. |
| Pass: Yes Conditional Pass: Fail: | |
| PrP Problems or issues: Nil | |

Test case 1 to check whether the data entered in a predefined format

|  |  |
| --- | --- |
| Test Case# : 2 | Priority(H,L): High |
| Test Objective: details of registered user phone no in number format | |
| Test Description: details of registered user phone no in number format is checked | |
| Requirements Verified: details of registered user phone no in number format is checked in the database | |
| Test Environment: JSP, Servlets | |
| Test setup or Pre-conditions: User initiates any control mechanism like Submit button. | |
| Actions | Expected Results |
| If the details of registered user phone no in number format  If the username is valid | A message “details of registered user phone no in number format.  Check and his account is created if all the required fields are entered correctly. |
| Pass: Yes Conditional Pass: Fail: | |
| PrP Problems or issues: Nil | |

Test case 2 which verifies the format of the data that user enters and checks whether

it is valid or not

|  |  |
| --- | --- |
| Test Case# : 3 | Priority(H,L): High |
| Test Objective: To let the user browse file the required | |
| Test Description: browse files is checked | |
| Requirements Verified: All the necessary datasets should be browsed | |
| Test Environment: JSP, Servlets | |
| Test setup or Pre-conditions: User initiates any control mechanism like Submit or Go buttons | |
| Actions | Expected Results |
| browse dataset Incomplete Necessary fields  Completion of all the necessary fields | All the necessary datasets should be browsed submitted until that has been done  Just check and go to the next activity. |
| Pass: Yes Conditional Pass: Fail: | |
| PrP Problems or issues: Nil | |

1. **CONCLUSION**

Bitcoin is becoming widely used and widely trusted as a valid currency. Many users employ Bitcoin for the sake of anonymity for a variety of reasons. Some just want more privacy in their lives, while others need anonymity in order to accomplish their goals due to legal reasons. There's no doubt that blockchain is a hot issue in recent years, although it has some topics we need to notice, some problems has already been improved along with new technique's developing on application side, getting more and more mature and stable. The government have to make corresponding laws for this technology, and enterprise should ready for embrace blockchain technologies, preventing it brings too much impact to current system. When we enjoy in the advantage of blockchain technologies bring to us, in the same time, we still have to stay cautious on its inuence and security issues that it could be have.

**REFERENCES & BIBLOGRAPHY**

[1] I. Bentov, A. Gabizon, and A. Mizrahi, “Cryptocurrencies without proof of work," CoRR, vol. abs/1406.5694, 2014.

[2] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, Sok: “Research perspectives and challenges for bitcoin and cryptocurrencies," in IEEE Symposium on Security and Privacy, pp. 104{121, May 2015.

[3] N. T. Courtois and L. Bahack, “On subversive miner strategies and block withholding attack in bitcoin digital currency," CoRR, vol. abs/1402.1718, 2014.

[4] I. Eyal and E. G. Sirer, “Majority is not enough: Bitcoin mining is vulnerable," CoRR, vol. abs/1311.0243, 2013.

[5] J. Garay, A. Kiayias, and N. Leonardos, The Bit-coin Backbone Protocol: Analysis and Applications,pp. 281{310, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[6] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun, “Is bitcoin a decentralized currency?," IEEE Security Privacy, vol. 12, pp. 54{60, May 2014.

[7] A. Gervais, G. O. Karame, K. Wust, V. Glykantzis, H. Ritzdorf, and S. Capkun, “On the security and performance of proof of work blockchains," in Pro- ceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS'16), pp. 3{16, New York, NY, USA, 2016.

[8] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, “Tampering with the delivery of blocks and transactions in bitcoin," in Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Com- munications Security (CCS'15), pp. 692{705, New York, NY, USA, 2015.

[9] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, \Eclipse attacks on bitcoin's peer-to-peer network," in 24th USENIX Security Symposium, pp. 129{144, Washington, D.C., 2015.

[10] G. Karame, “On the security and scalability of bitcoin's blockchain," in Proceedings of ACM SIGSAC Conference on Computer and Communications Secu- rity (CCS'16), pp. 1861{1862, New York, NY, USA, 2016.

[11] G. O. Karame, “Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin," in Proceedings of Conference on Computer and Communication Security, pp. 1{17, 2012.

[12] S. King and S. Nadal, Ppcoin: Peer-to-peer Crypto-Currency with Proof-of-Stake, 2012. (<https://archive.org/stream/PPCoinPaper/> ppcoin-paper\_djvu.txt)

[13] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in 2016 IEEE Symposium on Security and Privacy (SP'16), pp. 839{858, May 2016.

[14] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S.Gilbert, and P. Saxena, “A secure sharding protocol for open blockchains," in Proceedings of ACM SIGSAC Conference on Computer and Communica- tions Security (CCS'16), pp. 17{30, New York, NY,

USA, 2016.

[15] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, Feb. 24, 2013. (<http://bitcoin.org/> bitcoin.pdf)

[16] E. U. Opara, O. A. Soluade, “Straddling the next cyber frontier: The empirical analysis on network security, exploits, and vulnerabilities," International Journal of Electronics and Information Engineering, vol. 3, no. 1, pp. 10{18, 2015.

[17] M. Rosenfeld, “Analysis of hashrate-based double spending," CoRR, vol. abs/1402.2009, 2014.

[18] J. Singh, “Cyber-attacks in cloud computing: A case study," International Journal of Electronics and In- formation Engineering, vol. 1, no. 2, pp. 78{87, 2014.

[19] Y. Sompolinsky and A. Zohar, Secure High-Rate Transaction Processing in Bitcoin, pp. 507{527, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[20] W. T. Tsai, R. Blower, Y. Zhu, and L. Yu, “A system view of nancial blockchains," in IEEE Symposium on Service-Oriented System Engineering (SOSE'16), pp. 450{457, Mar. 2016.

[21] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, “Blockchain contract: Securing a blockchain applied to smart contracts," in IEEE International Conference on Consumer Electronics (ICCE'16), pp. 467{468, Jan. 2016.