CHAPTER 1: INTRODUCTION

1.1 GENERAL CONCEPTS

Now a day, education has become essential part of life, still we need to maintain reputation and trust in Product. Everyone must show his/her Document and QR Code to any other person for some purpose/job. After seeing the document 3rd person cannot validate the originality of the QR Code.

Blockchain - A Revolution Bigger Than the Internet

The Internet is entering the second era that's based on Blockchain [**2**] [**3**]- the Internet of Value, a new platform to change the world of business. It's a novel solution to the age-old human problem of trust. It provides architecture for so-called trust less trust. It allows the user to trust the outputs of the system without trusting any actor within it.

The pace with which this technology is evolving, it's making it difficult for different sectors/domains to keep, without the changes. The world is increasingly getting connected with the amalgamation of connected devices and solutions. So how do we fit in-For truly digitization process in Fintech / Banking and other sectors as well got to be seamless.

“Blockchain technology" can be seen as a group of technologies, like a bag of bricks. From the bag, we can take out bricks and put them together in different ways to create different results.

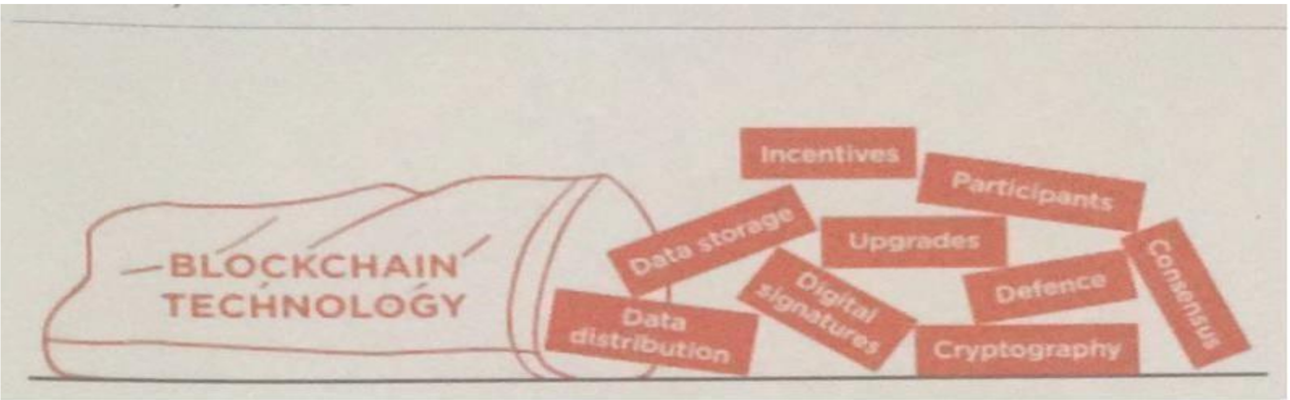


Figure 1.1: Blockchain Technology

1.2 MOTIVATION

In previous years, it has been come into the light and come in our daily routine life, that we got to know, below cases,

- Some company has fired xyz employee due to fraud Product document.
- Someone is selling the same Product to a number of peoples.
- The same driving license number is issued to the number of people.
- Same Voter ID is issued to many people. • A doctor has a fake degree, and he is practicing.

Many people paste the other people’s photograph on some other ID proof and use the scan copy/ Photocopy as an Identity proof.

From above we see that above incident happens as we have no channel to check the authenticity. If someone has a fake document, we have no options to verify the authenticity.

But when we see the cryptocurrency mechanism/ Properties, we found that it uses blockchain as a base, and it is secure by nature and has the following properties.

- Decentralized
- Digital cash system
- Digital money is created from code.
- Monitored by a peer-to-peer internet protocol. • An encrypted string of data or a hash encoded to signify one unit of currency.

We can build the trust for education QR Code by using blockchain technology. By using this technology, there is no need for a central authority to validate documents. Your college won't have to send you a copy of your transcript and prove to anyone you have your degree. We are building a platform that will be open, accessible and one piece of software at a time and Customer can get Blockchain-based Product Products. Blockchain-based Product Products are the digital QR Code and registered on Ethereum.

Blockchain that will be cryptographically signed and tamper proof.

(Ethereum blockchain is on 2nd number after Bitcoin blockchain). Another person can view the QR Code online, and no 3rd party validation is required for these digital QR Codes.

1.3 RELATED WORK

As of now, mainly Blockchain is used in cryptocurrency. When Santoshi Nakamoto (Bitcoin Developer) saw problems in centralized currency, he tried to build a digital cash system without a central entity, and it would be like a Peer-to-Peer network, this became the birth of cryptocurrency.

Cryptocurrency is a method/way in the Blockchain using encryption technique to control the creation of monetary units and to verify the transfer of funds. The transaction is known instantly by the whole network. But minors take some time to confirm this transaction. This is a minor's job in a cryptocurrency-network, and they get rewarded with a token (some amount) of the cryptocurrency.

In a decentralized network, we don't need a central server which keeps the record of the transaction/balances. Every node in the system has a copy of all transactions to check if current transactions are valid or not.

Top 5 Cryptocurrency (2018/03/15-as per Market Capitalization=Price*Circulating Supply)

1.3.1. Bitcoin BTC

1.3.2. Ripple XRP

1.3.3. Ethereum ETH

1.3.4. Bitcoin Cash BCH

1.3.5. Cardano ADA.

1.3.1 Bitcoin BTC:

Satoshi Nakamoto is the unknown inventor of Bitcoin. It was released in 2009, and its symbol is BTC.

"A new electronic cash system that uses a peer-to-peer network to prevent double-spending. It is completely decentralized with no central authority or server" – Satoshi Nakamoto, 09 January 2009, announcing Bitcoin on SourceForge [**4**]. It is a digital currency system based on peer-to-peer virtual data [**5**]**.** It uses peer-to-peer technology or network to operate with no central authority or banks; managing transactions and the issuing of Bitcoin is carried out collectively by the system.

Bitcoin is the 1st cryptocurrency that usages Cryptography to control its creation and transactions, rather than a central authority. It provides a new payment system that is digital in nature and no central authority/mediators are involved. It can be considered as "Cash for Internet".

- Market Cap: $222,014,656,865
- Price: $13,238.0000
- Available Supply: 16,771,012

1.3.2 Ripple XRP

Ripple was developed by Arthur Britto, David Schwartz & Ryan Fugger. It was released in 2013, and its symbol is XRP.

It is a real-time payment network that immediately offers certain and low-cost international payments. It "enables banks to settle cross-border payments in real time, with end-to-end transparency, and at lower costs." It is based around a shared, public database which uses a consensus process that allows for payments, exchanges, and remittance in a distributed process. Its Ledger does not require mining that is the major difference from Bitcoin and other cryptocurrency that uses mining concept. That's why it does not require more computing power.

- Market Cap: $88,309,754,593
- Price: $2.2796
- Available Supply: 38,739,144,847

1.3.3 Ethereum ETH

Ethereum was developed by the Ethereum Foundation (a Swiss non-profit foundation). It was released in 2015, and its symbol is ETH.

It is a distributed SW platform that uses Smart contract to interact with the blockchain. Application based on Ethereum runs without any fraud and 3rd party validation.

- Market Cap: $66,287,547,582
- Price: $686.4400
- Available Supply: 96,567140

1.3.4 Bitcoin Cash BCH:

Bitcoin Cash was developed by Bitmain group. It was released in 2017, and its symbol is BCH. It is the continuation of the Bitcoin project as peer-to-peer digital cash. It is a fork of the Bitcoin blockchain ledger, with upgraded consensus rules that allow it to grow and scale. Its block size limit to eight megabytes. The rule change increasing the Bitcoin block size limit of one megabyte to eight megabytes is classified as a hard fork.

- Market Cap: $39,092,477,988
- Price: $2,315.4250
- Available Supply: 16,883,500

1.3.5 Cardano ADA

Cardano was developed by Aggelos Kiayias, and it was released in 2017, and its symbol is ADA.

- Market Cap: $13,290,216,358
- Price: $0.5126
- Available Supply: 25,927,070,538

1.4 PROBLEM STATEMENT

As education becomes more diversified, decentralized, and democratized, we still need to maintain reputation, trust in Product, and proof of learning. Nowadays everyone must show his/her Document and

QR Code to any other person for some purpose/job. After seeing the document 3rd person cannot validate the originality of the QR Code. Major problem:

- Authenticity
- Trust
- Accessibility

Possible Solution:

- Database with no update feature
- Digital Signature • Blockchain.

We are building a platform that will be open, accessible and one piece of software at a time and Customer can get Blockchain-based Product Products. Blockchain-based Product Products are the digital QR Code and registered on the Ethereum Blockchain that will be cryptographically signed and tamper proof. Other people can view the QR Code online, and no 3rd party validation is required for these digital QR Codes.

We are going to build a web-based platform for Customer where they can enroll and select a course, which will have two major parts,

- After choosing the course, they have to give Verifys and result will be saved on blockchain server.
  - At admin/university/college side, they can manage courses and Company profiles.

1.5 SCOPE OF THIS THESIS

Previous work in the field of the blockchain, which is mainly focused on cryptocurrency and its mining. In 2017, the blockchain rose to a high level, most of the attention has been on cryptocurrencies such as Bitcoin and Ethereum as investors try to catch the next wave. Now it is going to different sector-Education, Product registry, Banking Share marking….

In this Project, I have investigated the possibilities of use of blockchain technology in the education sector. I have worked on Product generation by using.

1.5 SCOPE OF THIS THESIS

Previous work in the field of the blockchain, which is mainly focused on cryptocurrency and its mining. In 2017, the blockchain rose to a high level, most of the attention has been on cryptocurrencies such as Bitcoin and Ethereum as investors try to catch the next wave. Now it is going to different sector- Education, Product registry, Banking Share marking….

In this Project, I have investigated the possibilities of use of blockchain technology in the education sector. I have worked on Product generation by using this technology, in which candidate will enroll for a course and have to give the online Verify. After completion of the Verify, if a candidate is Pass result will be saved on blockchain ledger, and if a candidate fails, the result will not be kept on blockchain, and user have to reattempt the Verify.

The purpose of this report/thesis is to analyze the use of new emerging technology (blockchain) in the field of education so that candidate gets the benefit and employer has the transparency. That will reduce the fraud cases as data cannot be erased/ Rewrite on blockchain server.

CHAPTER 2: LITERATURE REVIEW

2.1 BLOCKCHAIN:

Since its 2008 appearance as a cornerstone of the cryptocurrency Bitcoin, blockchain technology gained widespread attention as a modality to securely validate and store information without a trusted third party [**6**]. Blockchain is a decentralized transaction and data management technology developed first for Bitcoin cryptocurrency [**7**]. Blockchain features a decentralized and incorruptible database that has high potential for a diverse range of uses [**8**].

A blockchain, originally block chain, is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Each block typically contains a cryptographic hash of the previous block, a timestamp and transaction data. By design, a blockchain is inherently resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way".

Blockchain is a decentralized ledger used to securely exchange digital currency, perform deals and transactions [**8**] and managed by peer-to-peer networks. All nodes follow the same protocol for inter-node communication and validating new blocks. Once data is validated in any block it cannot be altered by any block. To alter block data all subsequent block data should be altered that will result in collusion of the network and that transaction will be rejected by all nodes.

In 2008, Satoshi Nakamoto invented the blockchain for the use of cryptocurrency and Bitcoin was its 1st implementation. Bitcoin was the 1st public transaction ledger. The invention of this currency solved the double-spending problem without the need for a 3rd party. After that other cryptocurrency were invented on same concept.

In short, a blockchain is a distributed database that contains a list of records (data). Distributed means that instead of being stored on a central device somewhere, the entire database is actively synced and stored on a bunch of other devices. This is called a peer-to-peer network, much like how Napster was a peer-to-peer network for sharing music files.

The main advantage this technology provides is its ability to exchange transactions without relying on trusted third-party entities of any means. It can also provide data integrity, in-built authenticity and user transparency [**9**].

2.1.1 Blocks

A block contains a set of valid transactions that are in hash form and make a Merkle Tree. Each block typically contains a hash pointer as a link to a previous block, a timestamp and transaction data. By design, blockchains are inherently resistant to modification of the data [**11**]. These linking forms a block of chain. This process is iterative and confirms that previous block is reliable and correct. In this way we can go back to genesis block.

2.1.2 Block time

In blockchain block time refers to the time when network can create 1 more block in the chain. Its time vary from blockchain to blockchain some blockchain allows new block as frequently as every five seconds. This time also includes the time in which data becomes verifiable. In cryptocurrency term shorter block time means faster transaction. In Ethereum Blockchain Block time is approximate 14~15 seconds, while for Bitcoin is approx 10 minutes.

2.1.3 Decentralization

Blocks are stored in different locations (nodes) so blockchain eliminates several risks which come if data is in single location/storage. In which we don't has no central point of failure. Data stored on the blockchain is generally considered incorruptible, while centralized data is more easily controlled, information and data manipulation are possible.

2.2 BLCOKCHAIN WORKING:

Blockchain can be considered as the "Internet of value". On the Internet, anyone can write data and others can read it. In terms of cryptocurrency Keys fills the role of recording the transfer, which is traditionally carried out by banks. It also fills a second role, establishing trust and identity, because no one can edit a blockchain. The major functions carried out by banks - verifying identities to prevent fraud and then recording legitimate transactions -can be carried out by a blockchain more quickly and accurately.

Block orders in a blockchain

Blockchain can be considered as a book where, Blocks in a chain = pages in a book

A book has number of pages, and each page contains:

- **The text:** the information/data.
- **Information about itself:** Chapter number, Title or Page number which tells where we are in the book Similarly, in a blockchain block, each block has:

- T**he contents** of the block, for Verifyple in Bitcoin are the Bitcoin transactions and the miner incentive reward.
- **Headers** which contain the data about the block. It includes some technical information about the block, a reference to the previous block, and a fingerprint (hash) of the data contained in this block.

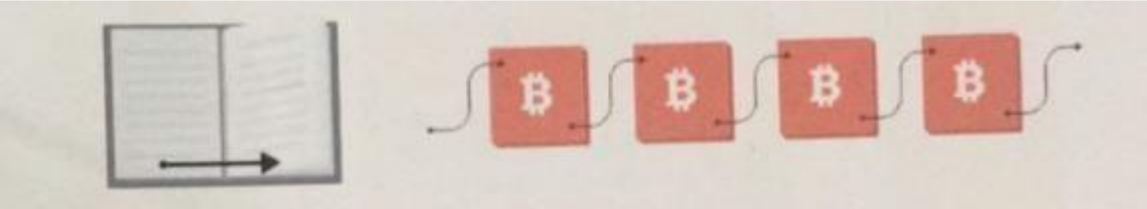


Figure 2.1: Blockchain vs. Book

Page by page: In book, Pages have page number in order. If some pages are missed and shuffled then it is easy to put them back into correct order so that information can be provided in proper way.

Block by block: In Blockchain, each block have a previous block address and previous block have its previous block address till genesis block.

BOOK ORDERING	**BLOCK ORDERING**
Page 1,2,3,4,5	Block n58ufO built on 84n855, Block 90fk5n built on n58ufO, Block 8n6d71 built on 90fk5n.
Implicit that the page builds on the page whose number is one less. e.g. Page 5 builds on page 4 (5 minus 1)	84n855, n58ufO, 90fk5n,8n6d71 represent fingerprints or hashes of the blocks.

Table 2.1: Book and Block Ordering

2.3 PUBLIC AND PRIVATE BLOCKCHAIN:

Blockchains can be divided into 2 major categories (Public and Private). Another way of describing public/private might be Permissionless vs. Permissioned or pseudonymous vs. identified participants.

2.3.1 Public Blockchains

It has below 2 basic properties:

- Anyone, without permission granted by another authority, can write data.
- Anyone, without permission granted by another authority, can read data.

1st blockchain, Bitcoin is designed as an ‘anyone-can-write' blockchain, where participants can add to the ledger without needing approval (**there is no 'boss' to decide).** Some of the largest, most known public blockchains are Bitcoin and Ethereum.

2.3.2 Private Blockchain

Private Blockchain provides a network where participants are known and trusted in which many rules/protocols aren't needed (or rather they are replaced with legal contracts) as participants will behave properly because he has signed this piece of paper. They do not rely on anonymous nodes to validate transactions.

2.4 CRYPTOCURRENCY:

Cryptocurrency is a medium of created, stored and exchanged electronically in the Blockchain using encryption technique to control the creation on monetary units and to verify the transfer of funds.

The transaction is known instantly by the whole network. But minors take some time to confirm this transaction. This is minor's job in a cryptocurrency-network, and in return they get cryptocurrency token.

In a decentralized network, we don’t need a central server who keeps records about the transactions. Every peer in the network needs to have a list of all transactions to check if current transactions are valid or an attempt to double spend.

2.4.1 Cryptocurrency Mining

Because of the random nature of hashing, achieving an acceptable block is never a guarantee. Thus, Bitcoin mining is a competitive venture, where miners are awarded new Bitcoin for each block successfully hashed and accepted in the blockchain [**5**]**.**

Bitcoin mining is a process of creating new Bitcoin by verifying the transactions in the Bitcoin network. Every transaction is kept in a public ledger, and that ledger is verified and maintained by all the computers participating in the Bitcoin network. This "chain" of transactions is known as the blockchain, and each transaction is essentially a public timestamp that can contain data [**12**].

Bitcoin miners donate their computer's processing power to run complex calculations. Whoever resolves the problem gets new cryptocurrency token as fees.

Miners, a decentralized network of users, validate and confirm transactions and they have setup of dedicated hardware to perform calculations, called „hashes‟. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them [**4**]**.** These strings of records (hashes) that keep track of every Bitcoin transaction and replicated on every system in the Bitcoin network.

The electricity power used to "mine" the cryptocurrency is a crucial factor as its prices are skyrocketing. According to the Bitcoin analysis blog Digiconomist, people mining cryptocurrency across the globe are using more than 30 terawatts-hours of energy. This is higher than the individual energy usage of at least 159 countries like Hungary, Oman, Ire Product, and Lebanon [**11**].

Ethereum is the world's second largest Blockchain network after Bitcoin and uses one- third the energy of Bitcoin. Approx 11 terawatt-hours a year, Ethereum use electricity which is the electricity consumption of Zambia. As Cryptocurrency mining is increasingly popular, its algorithm gets more and more difficult over time.

“More energy efficient algorithms, like proof-of-stage, have been in development over recent years. Bitcoin and mostly other cryptocurrency use proof-of-work methodology that required more energy consumption as compared to proof-of-stake algorithms. For Bitcoin mining operation setup, you need a place where energy costs are low. That's why an estimated 58 percent of global Bitcoin mining takes place in China.

2.4 EXISTING SYSTEM

- Generate the electronic file of a paper QR Code
- And calculate the hash value for it and store the hash value into the block ▪ The system create a QR-code string code to affix to the paper QR Code.
- Used to verify the authenticity of the paper QR Code through Mobile phone scanning.

Disadvantages of existing system:

- QR-code must be scanned with smartphone and internet connection is also required.
- Hyperledger cannot use public blockchain because of privacy and low scalability.
- Hyperledger preferred platform only for B2B business.

CHAPTER 3: PROPOSED WORK

3.1 DIGITIAL QR CODE GENERATION:

If Customer have an option to give Verify on web base portal, after completion of Verify, results/QR Code is saved on Blockchain. In this case other people can view the QR Code online and no 3rd party validation is required for these digital QR Codes.

We are proposing a web base portal for university/college/institution and Customer that will provide option to Company to get QR Code on blockchain and minimize the option of fraud and duplicate education QR Code.

Blockchain-based Product are registered on the Ethereum Blockchain that will be secure and tamper proof as data cannot be erased/ Rewrite on blockchain server. Since a blockchain is a permanent record of transactions that are distributed, every transaction can irrefutably be traced back to exactly when and where it happened. In addition, past transaction cannot be changed, while the present can‟t be hacked, because every transaction is verified by every single node in the network.

In this web-based portal, Company, and admin (university/Institution) will have login access and other than Company and admin can view Verify details and verify QR Code. It will have below two major parts,

- Company can select course, give Verifies and after successful completion can get QR Code on blockchain.
- Admin can manage Company, courses papers and question bank and can generate QR Code on blockchain.

3.2 PROPOSED MODEL:

In below figure, proposed model is shown.

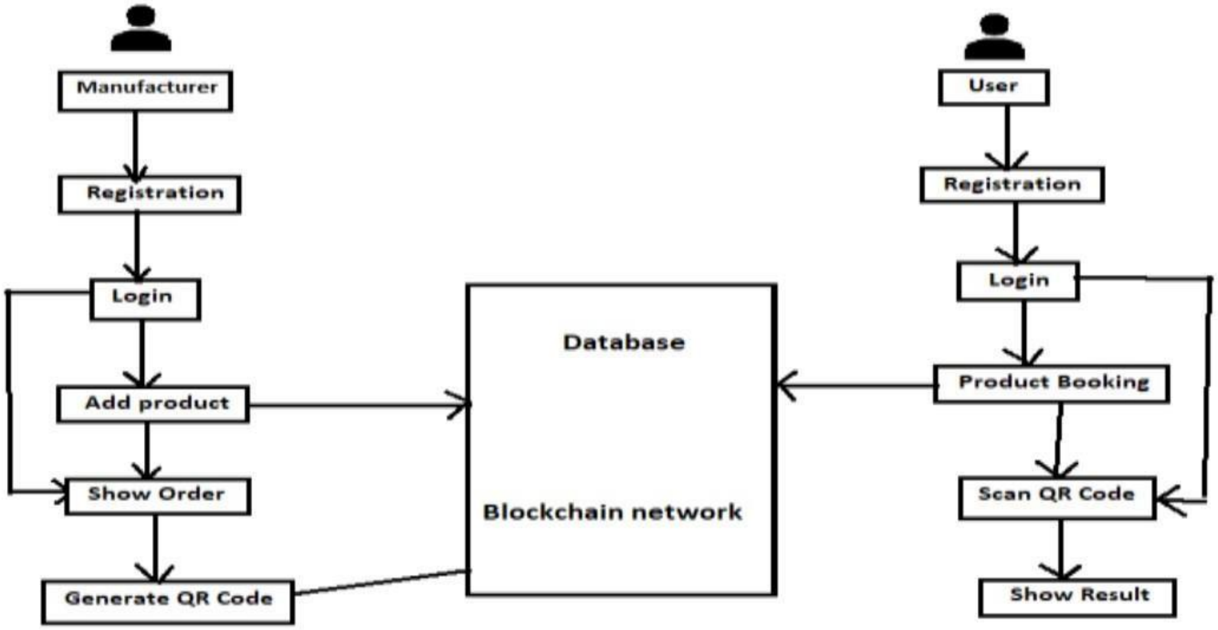


Figure 3.1: Proposed model for Digital QR Code

CHAPTER4. METHODOLOGY

We will use Ethereum blockchain to save Company data/QR Code. For that we need to write Smart Contract that is an interface to connect on blockchain.

4.1 SMART CONTRACTS:

Solidity is a language used for smart contracts on the Ethereum blockchain [**14**] and it is a set of code and data that have permanent address on the Ethereum blockchain. In Object Oriented Programming language, it is like a class where it includes state variables & functions. Smart Contracts and blockchain are the basis of all Decentralized Applications. Contracts and blockchain have immutable and distributed features as common features. If they are on blockchain then it will be painful to upgrade contracts.

Our contract will include:

4.1.1 State Variables-variables that hold values that are permanently stored on the Blockchain. We will use state variables to hold Company name, Course detail, QR Code number and validity date.

4.1.2 Functions-Functions are the executables of smart contracts. They are what we will call interacting with the Blockchain, and they have different levels of visibility, internally and externally. Keep in mind that whenever you want to change the value/state of a variable, a transaction must occur-costing Ether.

4.1.3 Events-Whenever an event is called, the value passed into the event will be logged in the transaction‟s log. This allows JavaScript callback functions or resolved promises to view the certain value you wanted to pass back after a transaction. This is because every time you make a transaction, a transaction log will be returned. We will use an event to log the ID of the newly created Candidate, which we‟ll display.

Ethereum Virtual Machines is implemented in C++, Go, Haskell, Java, JavaScript, and Python. It is the runtime environment for smart contracts in Ethereum. It handles the internal state and computation of the entire Ethereum Network.

4.3 GAS:

Gas is the internal pricing that we must pay for running a transaction or contract in Ethereum blockchain**.** A certain number of gases occurred whenever there is an operation performed by transaction or contract on the Ethereum platform.

Any computer code (complex or short) can be run inside EVM, A short code can result in more computation work as compared to complex code. It means that short code ode does not guarantee less computation work. Gas depends upon the calculation done inside the EVM; our focus should be on less computation work that will result in less amount of Gas. The payment is charged as a certain number of ether. The transaction fee is Transaction fee is combination of total gas used multiplied by gas price.

We will also use below tools:

Web3.js is a JavaScript API and with the help of this API We can interact with the Blockchain - making transactions and calls to smart contracts. Developers can focus on the content of their application as this API abstracts the communication with Ethereum Clients.

Truffle is a testing development framework for Ethereum. It includes a development blockchain, compilation and migration scripts to deploy your contract to the Blockchain, contract testing, and so on. It makes development easier.

4.4 SYSTEM REQUIREMENT SPECIFICATIONS SOFTWARE SPECIFICATIONS

Front End: Anaconda IDE

Backend: SQL

Language: Python

OPERATING SYSTEMS

- Windows 8,10 (32- or 64-bit)

HARDWARE SPECIFICATIOS

Hard disk	-	500 GB
Processor	-	Pentium IV 2.4 GHz
Ram	-	8 GB

CHAPTER 5

5. SOFTWARE REQUIREMENT SPECIFICATION

INTRODUCTION

Purpose: The main purpose of preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and to determine the operating characteristics of the system.

Scope: This Document plays a vital role in the development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during the testing phase. Any changes made to the requirements in the future will have to go through a formal change approval process.

DEVELOPERS RESPONSIBILITIES OVERVIEW:

The developer is responsible for:

- Developing the system which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

5.1 FUNCTIONAL REQUIREMENTS OUTPUT

DESIGN:

Outputs from computer systems are required primarily to communicate the results of processing for users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

OUTPUT DEFINITION:

The outputs should be defined in terms of the following points:

Type of the output, Content of the output, Format of the output, Location of the output, Frequency of the output, Volume of the output, Sequence of the output. Output is the primary purpose of this system. These guidelines apply for the most part to both project and screen outputs. Output design is often discussed before other aspects of design because, from the client's point of view, the output is the system. The output is what the client is buying when he or she pays for a development project. Inputs, databases, and processes exist to provide output.

- Problems often associated with business information output are information delay, information (data) overload, project domination, excessive distribution, and no tailoring.

- Mainframe printers: high volume, high speed, located in the data center Remote site printers: medium speed, close to end user.

- COM is Computer Output Microfilm. It is more compact than traditional output and may be produced as fast as non-impact printer output.

- Turnaround documents reduce the cost of internal information processing by reducing both data entry and associated errors.

- Periodic reports have set frequencies such as daily or weekly; ad hoc reports are produced at irregular intervals.

- Detail and summary reports differ in the former support day-to-day operation of the business while the latter includes statistics and ratios used by managers to assess the health of operations.

- Page breaks and control breaks allow for summary totals on key fields.

- Report requirements documents contain general report information and field specifications; print layout sheets present a picture of what the report will actually look like.

- Page decoupling is the separation of pages into cohesive groups.

- Two ways to design output for strategic purposes are (1) make it compatible with processes outside the immediate scope of the system, and (2) turn action documents into turnaround documents.

- People often receive reports they do not need because the number of reports received is perceived as a measure of power.

- Fields on a report should be selected carefully to provide uncluttered reports, facilitate 80-column remote printing, and reduce information (data) overload.

- The types of fields which should be considered for business output are: key fields for access to information, fields for control breaks, fields that change, and exception fields.

- Output may be designed to aid future change by stressing unstructured reports, defining field size for future growth, making field constants into variables, and leaving room on summary reports for added ratios and statistics.

- Output can now be more easily tailored to the needs of individual users because inquirybased systems allow users themselves to create ad hoc reports.

- An output intermediary can restrict access to key information and prevent unauthorized access.

- An information clearinghouse (or information center) is a service center that provides consultation, assistance, and documentation to encourage end-user development and use of applications.

- The specifications needed to describe the output of a system are: data flow diagrams, data flow specifications, data structure specifications, and data element specifications.

Output Documents

- External Reports: for use or distribution outside the organization; often on pre-printed forms.
- Internal Reports: for use within the organization.
- Periodic Reports: produced with a set frequency (daily, monthly, etc.) ○ Ad-Hoc (On Demand) Reports: irregular interval; produced upon user demand.
- Detail Reports: one line per transaction.

- Summary Reports: an overview.

- Exception Reports: only show errors, problems, out-of-range values, or unexpected conditions or events.

Output Design Objectives

- Assure Purposeful Output ○ Make Meaningful to User
- Provide Appropriate Quantity
- Appropriate Distribution
- Assure Timeliness
- Choose Effective Output Method.

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design as given below:

Input States: The main input stages can be listed as below:

Data recording, Data transcription, Data conversion, Data verification, Data control, Data transmission, Data validation, Data correction, Input **Media:**

At this stage a choice has to be made about the input media. To conclude about the input media consideration must be given to:

Type of Input, Flexibility of Format, Speed, Accuracy, Verification methods, Rejection rates, Ease of correction, Storage and handling requirements, Security, Easy to use, Portability.

- A source document differs from a turnaround document in that the former contains data that changes the status of a resource while the latter is a machine readable document.
- Transaction throughput is the number of error-free transactions entered during a specified time.
- A document should be concise because longer documents contain more data and so take longer to enter and have a greater chance of data entry errors.
- Numeric coding substitutes numbers for character data (e.g., 1=male, 2=female); mnemonic coding represents data in a form that is easier for the user to understand and remember. (E.g., M=male, F=female).

- The more quickly an error is detected, the closer the error is to the person who generated it and so the error is more easily corrected.
- An Verifyple of an illogical combination in a payroll system would be an option to eliminate federal tax withholding.
- By "multiple levels" of messages, means allowing the user to obtain more detailed explanations of an error by using a help option, but not forcing a lengthy message to a user who does not want it.
- An error suspense record would include the following fields: data entry operator identification, transaction entry date, transaction entry time, transaction type, transaction image, fields in error, error codes, date transaction reentered successfully.
- A data input specification is a detailed description of the individual fields (data elements) on an input document together with their characteristics (i.e., type and length).

Error Messages to be displayed for the end user.

Be specific and precise, not general, ambiguous, or vague. (BAD: Syntax error, Invalid entry, General Failure). Don't JUST say what's wrong ----Be constructive; suggest what needs to be done to correct the error condition.

5.2. PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application.

Requirement specification plays an important part in the analysis of a system. Only when the required specifications are properly given, it is possible to design a system which will fit into required environment. It rests largely in the part of the users of the existing system to give the required specifications because they are the people who finally use the system. This is because the requirements must be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system which does not cater to the requirements of the user is of no use.

- The system should be able to interface with the existing system.
- The system should be accurate.
- The system should be better than the existing system.

The existing system is completely dependent on the user to perform all the duties.

6. ABOUT THE SOFTWARE

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.

- **Python is Interactive** − You can sit at a Python prompt and interact with the interpreter directly to write your programs.

- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** − Python is a great language for beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

6.1 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

6.2 Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

6.3 Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

6.4 First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help","copyright","credits"or"license"for more information.

>>>
```

Type the following text at the Python prompt and press the Enter −

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result −

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file −

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows −

```
$ python test.py
```

This produces the following result −

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

Sr.No	Methods & Description
1	**GET** Sends data in unencrypted form to the server. Most common method.
2	**HEAD** Same as GET, but without response body
3	**POST** Used to send HTML form data to server. Data received by POST method is not cached by server.
4	**PUT** Replaces all current representations of the target resource with the uploaded content.
5	**DELETE** Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

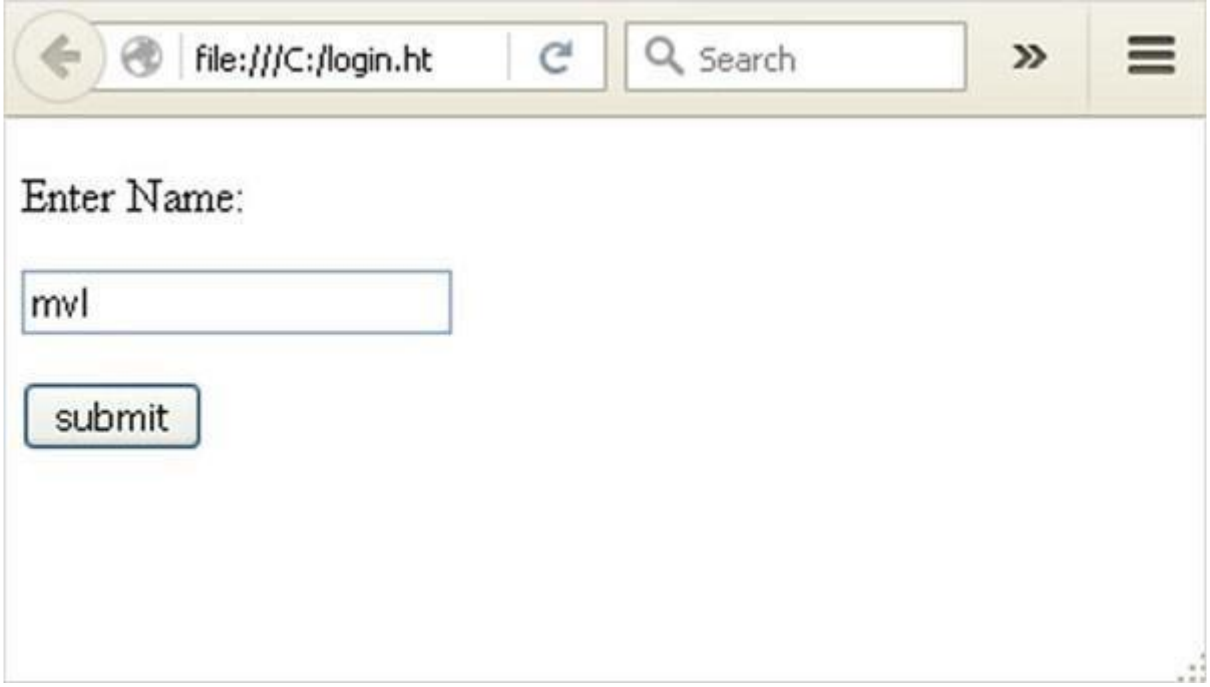
Save the following script as login.html

```
<html>

<body>

<formaction="http://localhost:5000/login"method="post">

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="submit"value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask importFlask, redirect,url_for, request app=Flask(__name__)

@app.route('/success/<name>') def

success(name):

return'welcome %s'% name

@app.route('/login',methods=['POST','GET'])

def login(): ifrequest.method=='POST':

user=request.form['nm'] return

redirect(url_for('success',name= user)) else:

user=request.args.get('nm') return

redirect(url_for('success',name= user)) if

__name__=='__main__':

app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.

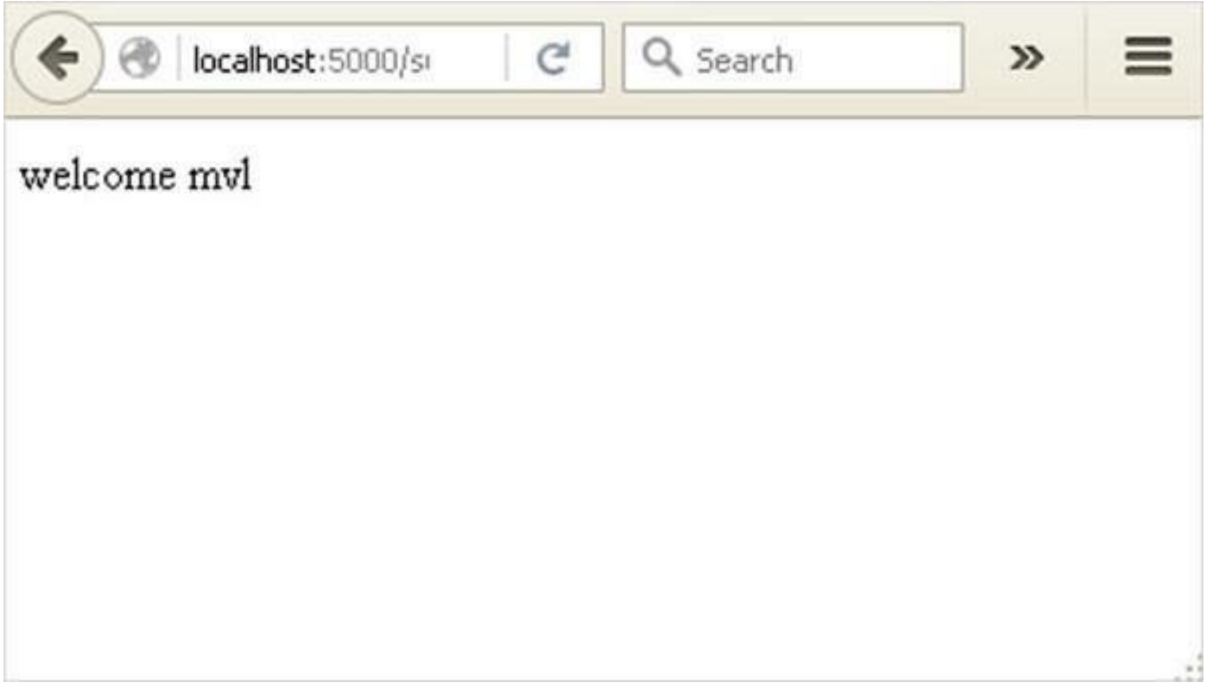


Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of ‘nm’ parameter obtained from the form data is obtained by −

```
user = request.form['nm']
```

It is passed to **‘/success’** URL as variable part. The browser displays a **welcome** message in the window.



welcome mvl

Change the method parameter to **'GET'** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by −

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to '/success' URL as before.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax like the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way, or a functional way. Good to know.
- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Anaconda or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages.

- Python was designed to for readability and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope, such as the scope of loops, functions and classes. Other programming languages often use curly brackets for this purpose.

6.5 Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: https://www.python.org/

Python QuickStart

Python is an interpreted programming language; this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

helloworld.py

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information. >>>
print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!") Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

C:\Users\Your Name>python myfile.py

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important. Python uses indentation to indicate a block of code. Example if 5 > 2:

  print ("Five is greater than two!")

Python will give you an error if you skip the indentation:

Example if 5 > 2:

```
print ("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment. print ("Hello, World!")
```

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

```
"""This is a multiline docstring.""" print("Hello, World!")
```

7. SYSTEM DESIGN

7.1. INTRODUCTION

Software design sits in the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement has been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess quality. Design is the only way that we can accurately translate an employee's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed, reviewed, and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

7.2 MODULES USED

User Interfaces

User interface design which we use to this project is Anaconda and Python studio.

For server communication we develop an IDE using Anaconda.

Using Python studio, we develop a Python application to share and scan the QR code.

Testrpc is a Node.js based Ethereum client for testing and development.

It uses Ethereum’s to simulate full client behavior and make developing Ethereum applications much faster.

Block Creation

A block is a container data structure. The average size of a block seems to be 1MB (source).

Here every QR Codes number will be created as a block.

For every block and hash code will generate for security.

Python based Block chain code generation:

In this module, based on QR Code numbers Block code will generate.

While creating Blockchain code users can increase the count based on their needs.

The major advantage of this module is that the user can share the Block chain code to another person in case of necessity.

When the user scan the QR Code an OTP will be sent to the registered mobile for verification.

Verification

In this module the user will upload the QR Codes like Products and so on.

Before upload, those QR Codes will be verified by the corresponding sector, if we upload school QR Code, the QR Code number will check with corresponds school database server if that QR Code is verified after that it will stored on server otherwise it will discard.

7.2 NORMALIZATION

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e., repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updating, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relations.

Insertion anomaly: Inability to add data to the database due to absence of other data.

Deletionanomaly: Unintended loss of data due to deletion of other data.

Updateanomaly: Data inconsistency resulting from data redundancy and partial update **Normal Forms**: These are the rules for structuring relations that eliminate anomalies.

FIRST NORMAL FORM:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

SECOND NORMAL FORM:

A relation is said to be in the second Normal form is it is in first normal form, and it should satisfy any one of the following rules.

- The Primary key is a not a composite primary key.
- No, non-key attributes are present.
- Every non key attribute is fully functionally dependent on a full set of primary keys.

THIRD NORMAL FORM:

A relation is said to be in third normal form if there exits no transitive dependencies. **Transitive Dependency**: If two non-key attributes depend on each other as well as on the primary key, then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data into multiple tables thereby making the data maintained in a consistent state.

7.3 E-R Diagrams

- The relation upon the system is structured through a conceptual ER-Diagram, which not only specifics the existing entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
- The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity. The attributes of each data object noted is the ERD can be described resign a data object description.
- The set of primary components that are identified by the ERD are
- Data object
- Relationships
- Attributes
- Various types of indicators.

7.4 DATA FLOW DIAGRAMS

A data flow diagram is a graphical tool used to describe and analyze the movement of data through a system. These are the central tools and the basis from which the other components are developed. The transformation of data from input to output, through processing, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments, and workstations. A full description of a system consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gone and Samson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. The Process is further identified with a number that will be used for identification purposes. The development of DFD'S is done on several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD at the next level. The lop-level diagram is often called context diagram. It consists of a single process bit, which plays a vital role in studying the current system. The process in the context level diagram is exploded into another process at the first level DFD.

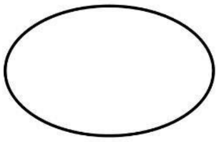
The idea behind the explosion of a process into more process is that understanding at one level of detail explodes into greater detail at the next level. This is done until further explosion is necessary, and an adequate amount of detail is described for analysts to understand the process. Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this led to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programmed in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

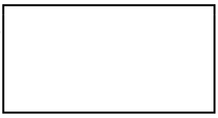
DFD SYMBOLS:

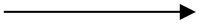
In the DFD, there are four symbols.

1. A square defines a source(originating) or destination of system data.
2. An arrow identifies data flow. It is the pipeline through which the information flows.
3. A circle or a bubble represents a process that transforms the incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data.

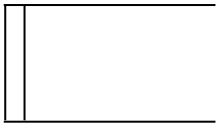


A Process that transforms the data flow.

Source  or Destination of data



Data flow



Data Store

CONSTRUCTING A DFD:

Several rules of thumb are used in drawing DFD'S:

1. The Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of the flow is from top to bottom and from left to right. Data traditionally flows from source to the destination although they may flow back to the source. One way to indicate this is to draw the long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower-level details, they are numbered.
4. The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized.

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces, redundancies and the like is then accounted for often through interviews.

SAILENT FEATURES OF DFD'S

1. The DFD shows the flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
2. The DFD does not indicate the time factor involved in any process, whether the dataflow takes place daily, weekly, monthly, or yearly.
3. The sequence of events is not brought out on the DFD.

TYPES OF DATA FLOW DIAGRAMS

1. Current Physical

2. Current Logical
3. New Logical
4. New Physical

CURRENT PHYSICAL:

In Current Physical DFD process label includes the name of the people or their positions or the names of computer systems that might provide some of the overall system-processing label includes an identification of the technology used to process the data. Similarly, data flows and data stores are often labelled with the names of the actual physical media on which data are stored such as file folders, computer files, business forms or computer tapes.

CURRENT LOGICAL:

The physical aspects in the system are removed as much as possible so that the current system is reduced to its essence to the data and the processors that transform them regardless of actual physical form.

NEW LOGICAL:

This is exactly like a current logical model if the user were completely happy with the user were completely happy with the functionality of the current system but had problems with how it was implemented typically through the new logical model will differ from the current logical model while having additional functions, absolute function removal and inefficient flows recognized.

NEW PHYSICAL:

The new physical represents only the physical implementation of the new system.

RULES GOVERNING THE DFD'S

PROCESS

1) No process can have only outputs.
2) No process can have only inputs. If an object has only inputs, then it must be a sin.
3) A process has a verb phrase label.

DATA STORE

1) Data cannot move directly from one data store to another data store, a process must move data.
2) Data cannot move directly from an outside source to a data store, a process, which receives, must move data from the source and place the data into the data store 3) A data store has a noun phrase label.

SOURCE OR SINK

The origin and /or destination of data.

1) Data cannot move direly from a source to sink it must be moved from a process.
2) A source and /or sink have a noun phrase Product.

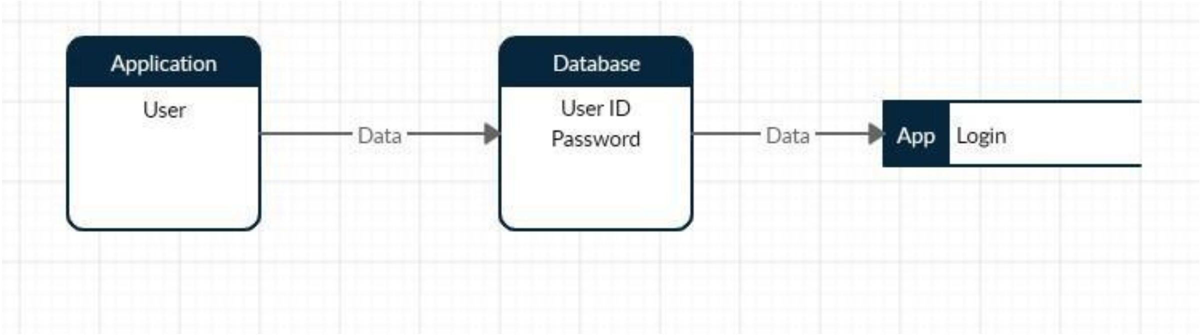
DATA FLOW

1) A Data Flow has only one direction of flow between symbols. It may flow in both directions between a process and a data store to show a read before an update. The latter is usually indicated, however, by two separate arrows since these happen at different types.
2) A join in DFD means that the same data comes from any of two or more different processes data store or sink to a common location.
3) A data flow cannot go directly back to the same process it leads. There must be at least one other process that handles the data flow produces some other data flow returns the original data in the beginning process.
4) A Data flow to a data store means update (delete or change).
5) A data Flow from a data store means retrieve or use.

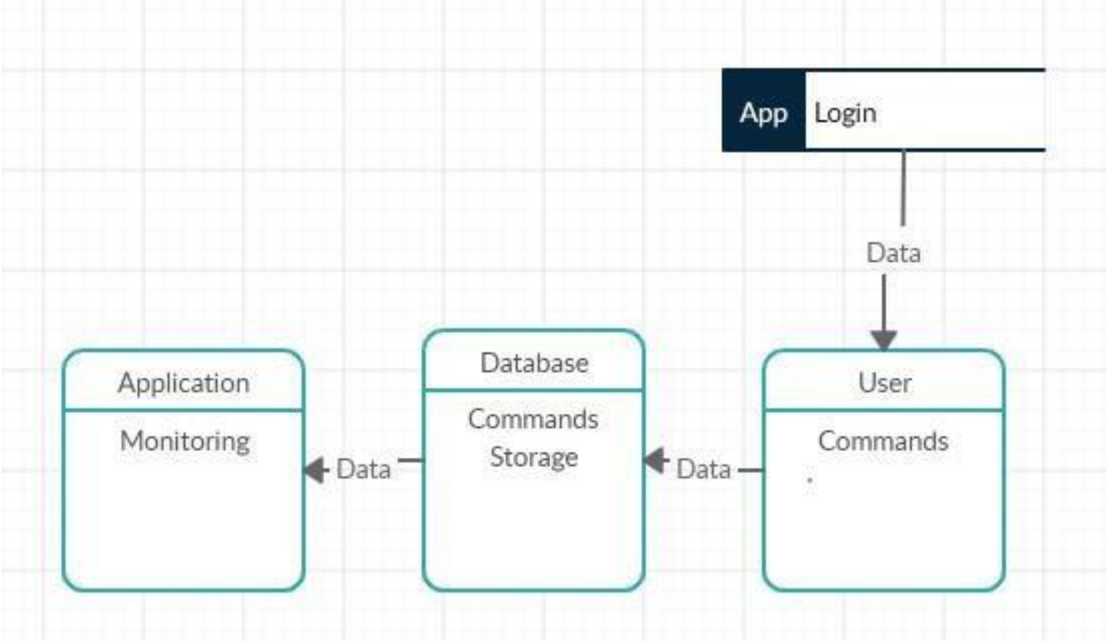
A data flow has a noun phrase label more than one data flow noun phrase can appear on a single arrow if all of the flows on the same arrow move together as one pack.

LEVEL1:

Login:

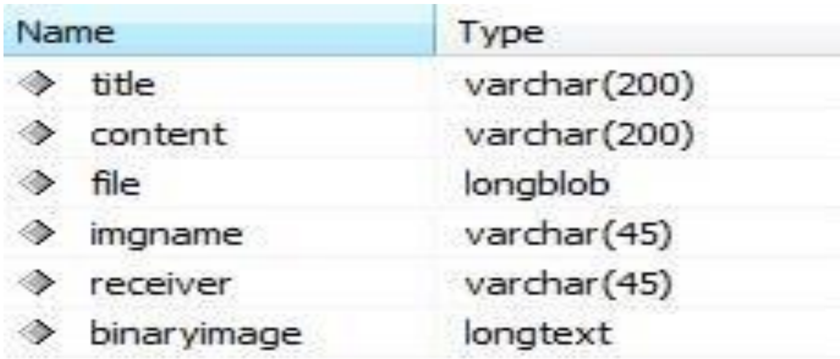


LEVEL 2:
User:

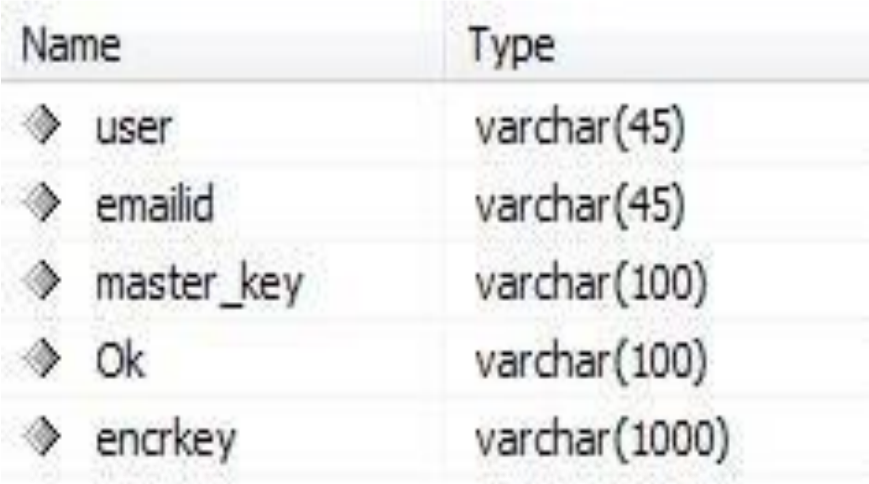


7.5 DATA DICTIONARY

A data dictionary, or metadata repository, as defined in the SQL Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format". Oracle defines it as a collection of tables with metadata.



Name	Type
title	varchar(200)
content	varchar(200)
file	longblob
imgname	varchar(45)
receiver	varchar(45)
binaryimage	longtext



Name	Type
user	varchar(45)
emailid	varchar(45)
master_key	varchar(100)
Ok	varchar(100)
encrkey	varchar(1000)

7.6 UML DIAGRAMS

The **use case** diagrams describe the system functionality as a set of tasks that the system must carry out and **actors** who interact with the system to complete the tasks.

Use Case:

Each **use case** on the diagram represents a single task that the system needs to carry out. *Buy a Product*, *Add Client*, *Make Purchase* and *Validate Order Information* are all Verifyples of use cases. Some use cases may include or extend a task represented by another use case.

For Verifyple, to make a purchase, the order information will need to be validated.

Actor:

An **actor** is anything outside the system that interacts with the system to complete a task. It could be a user or another system. The actor "uses" the use case to complete a task. *System Administrator*, *Credit Authentication System*, *Accounting System* and *Web Client* are all Verifyples of actors. Often, it is useful to look at the set of use cases that an actor has access to -- this defines the actor's overall role in the system.

Association:

The **association** is the link that is drawn between and actor a use case. It indicates which actors interact with the system to complete the various tasks.

Includes:

Use the **includes** link to show that one use case includes the task described by another use case. For Verifiable, saving a Visual Case project includes saving the diagrams and saving the project settings. Sometimes the word "Uses" is used instead of "Includes" Generalization**:**

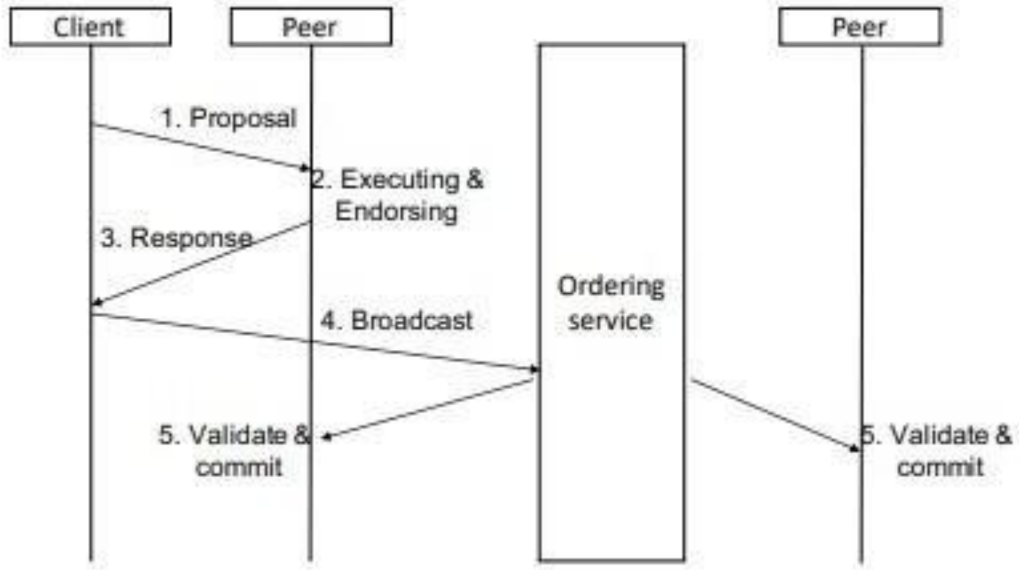
The **generalization** link is an informal way of showing that one use case is like another use case, but with a little bit of extra functionality. One use case inherits the functionality represented by another use case and adds some additional behavior to it.

Extends:

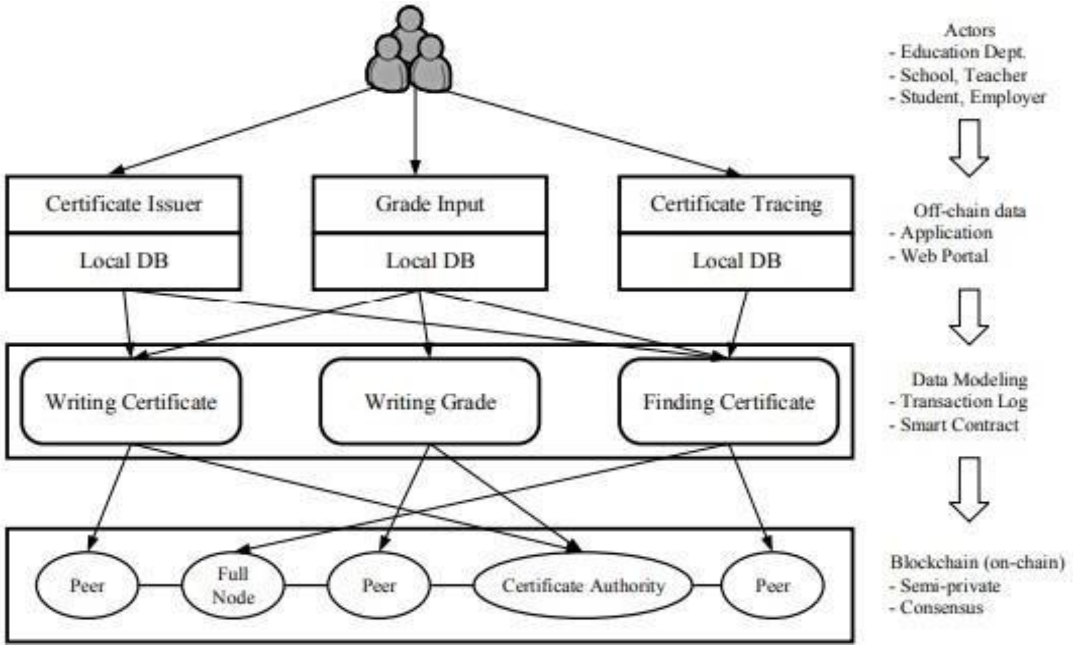
The extends link is used to show that one use case extends the task described by another use case. It's very similar to generalization but is much more formalized.

The use case that is extended is always referred to as the **base use case** and has one or more defined **extension points**. The extension points show exactly where extending use cases are allowed to add functionality. The extending use case doesn't have to add functionality at all of the base use case's extension points. The extension link indicates which extension points are being used.

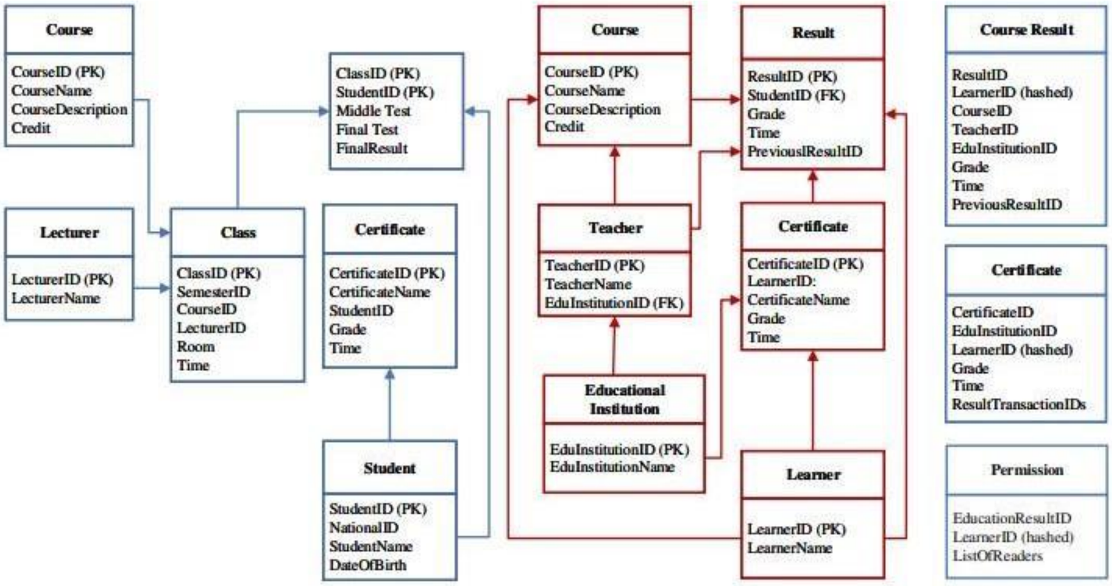
UML Diagrams: Overview Use Case:



7.7 ACTIVITY DIAGRAM:



7.8 Class Diagram

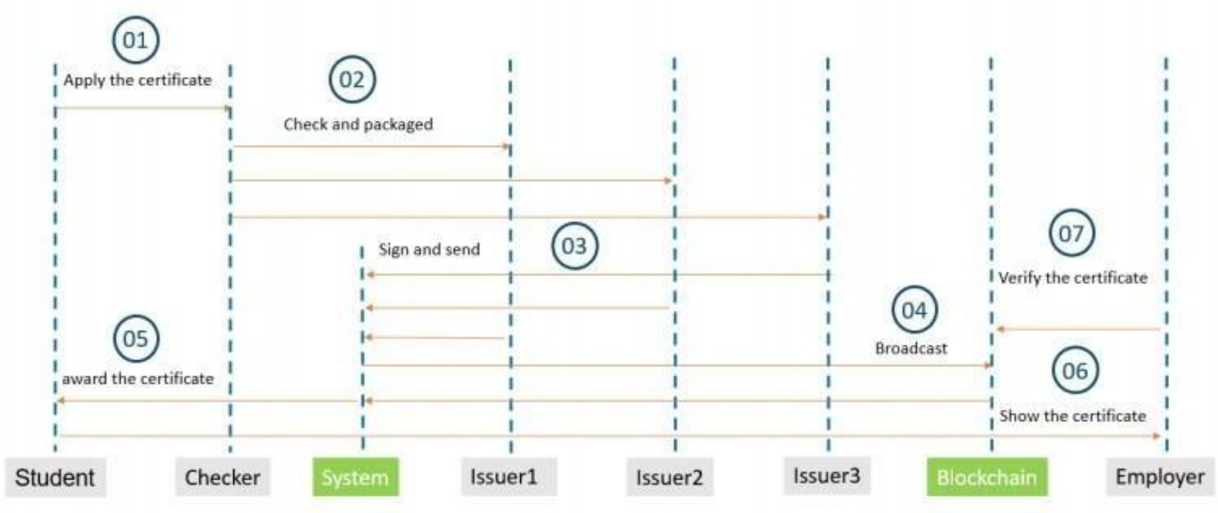


A. Example input database

B. Common data schema

C. Assets in blockchain

7.9 SEQUENCE DIAGRAM:



8. SYSTEM TESTING AND IMPLEMENTATION

8.1 INTRODUCTION

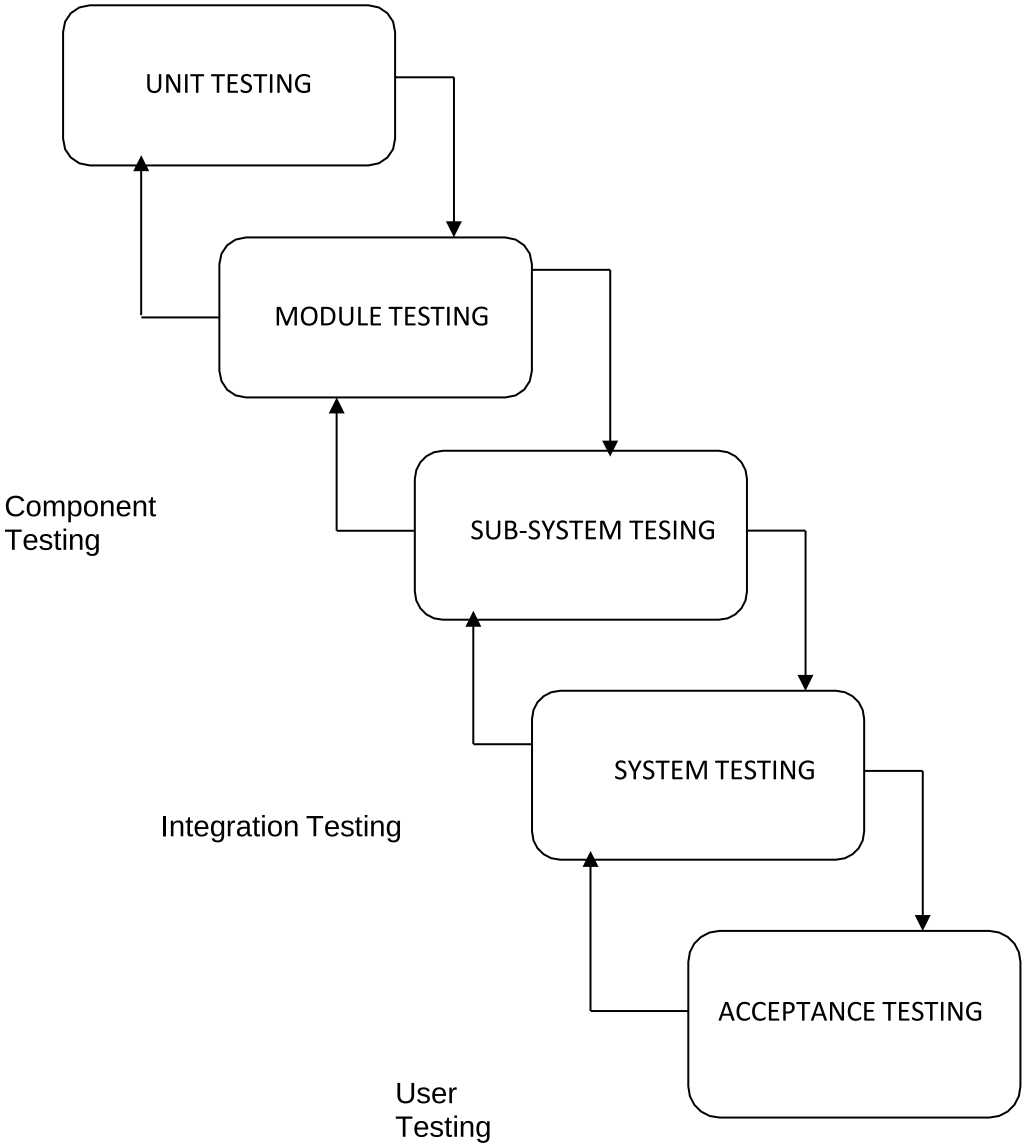
Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

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

8.2. STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction at each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing will progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.



VALIDATION TEST

8.3. UNIT TESTING

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

1. WHITE BOX TESTING

This type of testing ensures that,

- All independent paths have been exercised at least once.
- All logical decisions have been exercised on their true and false sides.
- All loops are executed at their boundaries and within their operational bounds.
- All internal data structures have been exercised to assure their validity.

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

2. BASIC PATH TESTING

The Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:
Use the design of the code and draw correspondent flow graphs.
Determine the Cyclomatic complexity of the resultant flow graph, using formula:
V (G) =E-N+2 or
V (G) =P+1 or
V (G) =Number of Regions
Where V (G) is Cyclomatic complexity,
E is the number of edges,
N is the number of flow graph nodes, P
is the number of predicate nodes.
Determine the basis of set of linearly independent paths.

3.CONDITIONAL TESTING

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

4.DATA FLOW TESTING

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

5.LOOP TESTING

- In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:
- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loop test the innermost loop first and then work outwards.
- Of concatenated loops the values of dependent loops were set with the help of a connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.
- Each unit has been separately tested by the development team itself and all the input has been validated.

9.FEASIBILITY STUDY

INTRODUCTION:

The preliminary investigation Verifyines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging the oldest running system. All systems are feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

9.1. TECHNICAL FEASIBILITY

Technical Feasibility centers on the existing computer system hardware, software, etc. and to some extent it can support the proposed addition. This involves financial considerations to accommodate technical enhancements. Technical support is also a reason for the success of the project. The techniques needed for the system should be available and it must be reasonable to use. Technical Feasibility is mainly concerned with the study of function, performance, and constraints that may affect the ability to achieve the system. By conducting an efficient technical feasibility, we need to ensure that the project works to solve the existing problem area.

Since the project is designed with PYTHON STUDIO with JAVA as Front end and SQL Server as Back end, it is easy to install on all the systems wherever needed. It is more efficient, easy, and user-friendly to understand by almost everyone. Huge amounts of data can be handled efficiently using SQL Server as back end. Hence this project has good technical feasibility.

9.2. OPERATIONAL FEASIBILITY

People are inherently instantly to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have towards the development of the computerized system.

The staff is accustomed to computerized systems. These kinds of systems are becoming more common. day by day for evaluation of the software engineers. Hence, this system is operationally feasible. As this system is technically, economically, and operationally feasible, this system is judged feasible.

9.3. ECONOMICAL FEASIBILITY

The role of interface design is to reconcile the differences that prevail among the software engineer's design model, the designed system meets the end user requirement with an economical way at minimal cost within the affordable price by encouraging more of the proposed system. Economic feasibility is concerned with comparing the development cost with the income/benefit derived from the development system. In this we need to derive how this project will help the management to take effective decisions.

Economic Feasibility is mainly concerned with the cost incurred in the implementation of the software. Since this project is developed using PYTHON STUDIO with JAVA and SQL Server, which is more commonly available and even the cost involved in the installation process is not high. Similarly, it is easy to recruit people for operating the software since almost all the people are aware of PYTHON STUDIO with JAVA and SQL Server. Even if we want to train the people in these areas the cost involved in training is also very low. Hence this project has good economic feasibility.

The system once developed must be used efficiently. Otherwise, there is no meaning for developing the system. For this a careful study of the existing system and its drawbacks is needed. The user should be able to distinguish the existing one and proposed one, so that one must be able to appreciate the characteristics of the proposed system; the manual one is not highly reliable and is considerably fast. The proposed system is efficient, reliable, and also responding quickly.

CHAPTER 10: CONCLUSION

As of now we are using internet (which is decentralized online platform) to sharing information. But when we transfer money; we are using old-fashioned, centralized financial establishments like banks. In other areas we are also using centralized system to share information (like education- where university has full control).

Blockchain technology provides a way to eliminate this "middleman/central authority. It does this by filling three important roles – recording transactions, establishing identity and establishing contracts. Information security is one of the most important features of Blockchain [**6**].

Blockchain can be used to store any type of digital information (e.g. computer code) rather than cryptocurrency usages .Previous work in the field of the blockchain, which is mainly focused on the cryptocurrency and it‟s mining. In 2017, the blockchain rose to a high level, Most of the attention has been on cryptocurrencies such as Bitcoin and Ethereum as investors try to catch the next wave. Now it is going to different sector-Education, Product registry, Banking Share marking….

For truly digitization process in Banking and other sectors, we can use Blockchain technology as a base. It will build trust and provide a way that someone can verify the other person documents in less time and validate the originality.

If we use blockchain in Education/Product Registry/ID card verification/Banking sector, then it will be a "**1st step towards corruption free country."**

REFERENCES

[1] Lyndon Lyons and Andreas Bachmann Jan Seffinga, "The Blockchain (R)evolution –The Swiss Perspective," , SwitgerProduct, 2017.

[2] Don Tapscott and Alex Tapscott, "Realizing the Potential of Blockchain-A Multistakeholder Approach to the Stewardship of Blockchain and Cryptocurrencies," in *World Economic Forum*, 2017.

[3] Alex Tapscott, BLOCKCHAIN REVOLUTION:Understanding the 2nd Generation of The Internet and the New Economy, 2017.

[4] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008, White Paper.

[5] George F. Hurlburt and Irena Bojanova, "Bitcoin: Benefit or Curse?," in *IEEE*, 2014.

[6] Nicola Dimitri, The Blockchain Technology: Some Theory and Applications, 2017, MSM-Working Paper No.
2017/03.

[7] Deokyoon Ko, Sujin Choi, Sooyong Park, Kari SmoProducter Jesse Yli-Huumo, "Where Is Current Research on Blockchain Technology?—A Systematic Review," October 2016.

[8] Nirmala Singh and Sachchidanand Singh, "Blockchain: Future of financial and cyber security," in *IEEE*, Noida, 2016.

[9] Engin Zeydan and Suayb Sb Arslan Gültekin Berahan Mermer, "An overview of blockchain technologies: Principles, opportunities and challenges," in *IEEE*, Turkey, 2018.

[10] Narn-Yih Lee , Chien Chi and Yi-Hua Chen Jiin-Chiou Cheng, "Blockchain and smart contract for digital QR Code," in *IEEE*, Japan, 2018.

[11] Henrique Rocha ,Marcus Denker and Stephane Ducasse Santiago Bragagnolo, "SmartInspect: solidity smart contract inspector," in *IEEE*, Itly, p. 2018.

[12] GWYN D'MELLO. (2017, Dec.) https://www.indiatimes.com/technology/news. [Online].

https://www.indiatimes.com/technology/news/bitcoin-miners-are-using-more-electricity-thanireProduct- other-159-countries-no-kidding-335114.html

[13] Abdul Wadud Chowdhury. (2017, Nov.) https://medium.com. [Online]. https://medium.com/oceanize-geeks/blockchain-and-the-future-of-digital-trust-354acc279acc

[14] Nick Grossman. (2015, June) https://www.nickgrossman.is. [Online]. https://www.nickgrossman.is/2015/the-blockchain-as-time/