**Project Report**

This is a formal report talking about the idea of Block-Chains in java using MVC technique. The project is divided into 4 classes, each class has its role and functionality, and then they are controlled by another class called Main.

**The first class is our "Block":-**

The Block is the smallest building unit in a Block-Chain. It is a group of transactions that are done collected together in one block to form a block. This block as we said before is the building unit a Block-Chain, which means that any Block-Chain is formed of some blocks that have attached/connected in a specific sequence. One of the interesting parts in this smallest building unit of a Block-Chain is the strong dependencies for each block to his previous block. They are connected in a way that they depend on each other using something called "previous hash". Each Block has the timestamp that is created in it. Each block has its index number in the Block-Chain. Also, each of them has its nonce which is the number of trying in the mining process. Last but not least, each block has a hash that is unique from any other hash of other blocks.

**The second class is our "Block-Chain":-**

The Block-Chain as we have mentioned before is a group of a block attached in a specific order. Each blockchain has its difficulty; difficulty is the number of zeroes that are appended to the hashing process for each block. The more you increase this number of zeroes the more difficult it will be. Each blockchain has its ID number and has its length. I have implemented the Block-Chain as a List of blocks linked together. In this class, there are a lot of features; one of them is the Genesis Block which is the first block in each new Block-Chain. We can consider it as a special type of block as it does not need or have a previous hash to depend on it like all other blocks. Also each blockchain we can check its validity and correctness throughout a method in this class that do this task. Also, we can reach from this class to the last block and get its hash to know the next coming new block to which blockchain it belongs so we add it to his appropriate one.

**The Third Class is our "Person":-**

A person is a class that we use to represent our real case of users actions. Users can do transactions, can receive new transactions which will be as a block from another user and will be attached to their block appropriate blockchain. Any user can have more than blockchain; all of them are stored and saved for him. Also, each user has his own actively used Blockchain which is the longest blockchain in his blockchain list. We also give each user a special ID to identify him from others. Users have two more things, they have names to define their real name and to view it for us when people trying to reach each other. Also, users have their Cache-History which is we save all the transactions that have been done and have been received by/to him.

**The fourth class is our "Network":-**

Simply, a network is our community that connects people "users". We can inform people by any transaction randomly or make the transaction reach all who are in this network.

**The fifth class and our last class is the "Main":-**

It is our final class; we use it to control everything in this project. We insert some data as an input and wait to see the amazing output of our real spent effort throughout the whole project.