**CHAPTER 1**

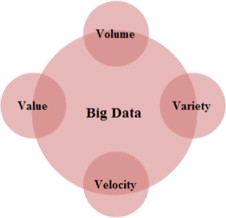
**INTRODUCTION**

* 1. **GENERAL**

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. The challenges include analysis, capture, duration, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime and so on. So we can implement big data in our project because every employ has instructed information so we can make analysis on this data.

## CHARACTERISTICS OF BIG DATA

Big data is a term utilized to refer to the increase in the volume of data that are difficult to store, process, and analyze through traditional database technologies. The nature of big data is indistinct and involves considerable processes to identify and translate the data into new insights. The term “big data” is relatively new in IT and business. However, several researchers and practitioners have utilized the term in previous literature. For instance, referred to big data as a large volume of scientific data for visualization. Several definitions of big data currently exist.” Meanwhile and defined big data as characterized by three V’s: volume, variety, and velocity. The terms volume, variety, and velocity were originally introduced by Gartner to describe the elements of big data challenges. IDC also defined big data technologies as “a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling the high velocity capture, discovery, and analysis.” specified that big data is not only characterized by the three V’s mentioned above but may also extend to four V’s, namely, volume, variety, velocity, and value This 4V definition is widely recognized because it highlights the meaning and necessity of big data.

[](http://www.sciencedirect.com/science/article/pii/S0306437914001288#gr1)

***Fig 1.1 Four V’s Of Big Data***

**VOLUME** - Refers to the amount of all types of data generated from different sources and continue to expand. The benefit of gathering large amounts of data includes the creation of hidden information and patterns through data analysis Collecting longitudinal data requires considerable effort and underlying investments. Nevertheless, such mobile data challenge produced an interesting result similar to that in the examination of the predictability of human behaviour patterns or means to share data based on human mobility and visualization techniques for complex data.

**VARIETY-**Refers to the different types of data collected via sensors, smart phones, or social networks. Such data types include video, image, text, audio, and data logs, in either structured or unstructured format. Most of the data generated from mobile applications are in unstructured format. For example, text messages, online games, blogs, and social media generate different types of unstructured data through mobile devices and sensors. Internet users also generate an extremely diverse set of structured and unstructured data.

**VELOCITY**-Refers to the speed of data transfer. The contents of data constantly change because of the absorption of complementary data collections, introduction of previously archived data or legacy collections, and streamed data arriving from multiple sources

**VALUE-** is the most important aspect of big data; it refers to the process of discovering huge hidden values from large datasets with various types and rapid generation

## BIG DATA STORAGE SYSTEM

The rapid growth of data has restricted the capability of existing storage technologies to store and manage data. Over the past few years, traditional storage systems have been utilized to store data through structured RDBMS. However, almost storage systems have limitations and are inapplicable to the storage and management of big data. A storage architecture that can be accessed in a highly efficient manner while achieving availability and reliability is required to store and manage large datasets. Hence Big Data is used.

**HADOOP BACKGROUND**

Hadoop is an open-source Apache Software Foundation project written in Java that enables the distributed processing of large datasets across clusters of commodity. Hadoop has two primary components, namely, HDFS and Map Reduce programming framework. The most significant feature of Hadoop is that HDFS and Map Reduce are closely related to each other; each are co-deployed such that a single cluster is produced. Therefore, the storage system is not physically separated from the processing system.

HDFS is a distributed file system designed to run on top of the local file systems of the cluster nodes and store extremely large files suitable for streaming data access. HDFS is highly fault tolerant and can scale up from a single server to thousands of machines, each offering local computation and storage. HDFS consists of two types of nodes, namely, a name node called “master” and several data nodes called “slaves.” HDFS can also include secondary name nodes. The name node manages the hierarchy of file systems and director namespace (i.e., metadata). File systems are presented in a form of name node that registers attributes, such as access time, modification, permission, and disk space quotas. The file content is split into large blocks, and each block of the file is independently replicated across data nodes for redundancy and to periodically send a report of all existing blocks to the name node.

Map Reduce is a simplified programming model for processing large numbers of datasets pioneered by Google for data-intensive applications. The Map Reduce model was developed based on GFS is adopted through open-source Hadoop implementation, which was popularized by Yahoo. Apart from the Map Reduce framework, several other current open-source Apache projects are related to the Hadoop ecosystem, including Hive, Hbase, Mahout, Pig, Zookeeper, Spark, and Avro. Twister provides support for efficient and iterative Map Reduce computations. An overview of current Map Reduce projects and related software is shown in Map Reduce allows an inexperienced programmer to develop parallel programs and create a program capable of using computers in a cloud. In most cases, programmers are required to specify two functions only: the map function (mapper) and the reduce function (reducer) commonly utilized in functional programming. The mapper regards the key/value pair as input and generates intermediate key/value pairs. The reducer merges all the pairs associated with the same (intermediate) key and then generates an output. Summarizes the process of the map/reduce function.

* 1. **PROBLEM DESCRIPTION**

In the “big data era,” a massive amount of data is available for all kinds of industrial applications. For example, the cloud service can be considered as a data warehouse which provides a useful source of data. Wireless sensor networks [e.g., radio frequency identification (RFID), near field communications] can be used to collect useful data ubiquitously. Extracting and utilizing useful information from huge and dynamic databases for “big data” is far from easy. Since these data are linked to real-time events, they can be employed, if properly (e.g., via BI schemes), for rescheduling or re-planning activities in business applications which finally reduce the level of risk and improve profitability and efficiency of the operations.  Developed a service optimization model for handling big data stored in cloud systems when privacy is a critical concern (e.g., the medical data). Service quality may be compromised if a cloud server refuses to provide the data due to the privacy issue. Such optimization model can maximize the service quality and is verified by a simulation study. Another application of big data is on smart grids. Simmhan et al. predicted the demand of a cloud-based smart grid system and derived the optimal pricing strategy, based on the big data on real-time consumption. The approach is possible due to the data mining algorithm the authors developed. Owing to the importance of big data analytics for business applications, this paper is developed. With respect to the core topic on big data analytics for business operations and risk management, we organize this paper into three big sections, namely: 1) BI and data mining; 2) industrial systems reliability and security; and 3) business operational risk management (ORM). Each of these sections: 1) examines some carefully selected papers; 2) outlines the related research challenges; and 3) proposes the future research directions.

* 1. **OBJECTIVE**

The objective of the proposed system are

* To access multiple account using a single personal card linking it with aadhaar card and providing security to the user.
* To fetch the transaction history from database using Hadoop.
  1. **SCOPE OF PROJECT**

The scope of the project is to integrate the multiple bank account in the single ATM card by using big data. The user can withdraw the money beyond the limit from any account in the card by introducing the formula authentication factor. Meanwhile the user can see the last transaction history using hadoop technology where the transaction done in big data is updated dynamically in database.

* 1. **ORGANIZATION OF THE THESIS**

This report consists of 6 chapters, the contents of which are described below:

Chapter 1 provides a brief introduction of the project. It specifies the idea of the project and provide information about it. It defines the objective and scope of the project. Chapter 2 is a literature survey that elucidates on the research works of the existing systems. The issues faced by the existing system are explored and a new system is proposed, that attempts to provide a better approach that the existing one. Chapter 3 delineates the system design and the essential information about algorithms used and the flowchart of the various modules specified. The functional architecture is represented diagrammatically. Chapter 4 outlines the system implementation and the various techniques and algorithms that are transformed into a code. It provides details about the development platform used and the various steps involved in the implementation process are portrayed in the form of screenshots from the actual systems. Chapter 5 outlines the testing and performance analysis of the proposed system. It takes into account the different test cases and tabulates the findings based on the outcome. It also explains about various measures considered for increasing the performance of the system. Chapter 6 gives the conclusion that summarizes the efforts undertaken and states the findings and also the shortcoming of the proposed system.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 GENERAL**

Literature survey is the documentation of a comprehensive review of the publishers and the unpublished word from the secondary sources data in the areas of specific interest to the researcher. It is the process of analyzing, summarizing, organizing and presenting novel conclusions from the results of technical review of large number of recently published scholarly articles. The purpose of literature survey is to review the critical points of current knowledge including substantive findings as well as theoretical and methodological contributions to a particular topic. It is a study and review of relevant literature materials in relation to a topic that have been given.

**2.2 EXISTING SYSTEMS**

In the existing system, user has to have the corresponding ATM card to withdraw money from ATM Machine. Also user cannot withdraw money if the daily withdrawal limit is crossed.

Gary M. Gaukler et.al [1] presents a model to help evaluate the impact of an introduction of item-level radio-frequency identification (RFID) in a retail environment where stock-out-based substitution is common. There are two main thrust areas in this work. First, we examine the impact of RFID in a centralized setting where retailer and manufacturer are one entity. This thrust area is concerned with evaluating the profitability of RFID and exploring which product properties favor an RFID implementation. Second, we examine the impact of RFID in a decentralized setting, where retailer and manufacturer independently maximize their profits. We investigate the problem of sharing the costs of RFID, from both the perspective of tag costs and fixed costs. Our research shows that the presence of substitution at the shelf plays a major role in determining the expected benefits of an RFID implementation, as well as in determining the optimal allocation of these benefits among retailer and manufacturer. It is therefore critically important that decision makers make strong efforts to correctly account for substitution effects when evaluating potential item-level RFID implementations in the retail sector.

Olatunde A. Durowoju et.al [2] tells that cloud computing introduces flexibility in the way an organization conducts its business. On the other hand, it is advisable for organizations to select cloud service partners based on how prepared they are owing to the uncertainties present in the cloud. This study is a conceptual research which investigates the impact of some of these uncertainties and flexibilities embellished in the cloud. First, we look at the assessment of security and how it can impact the supply chain operations using entropy as an assessment tool. Based on queuing theory, we look at how scalability can moderate the relationship between cloud service and the purported benefits. We aim to show that cloud service can only prove beneficial to supply partners under a highly secured, highly scalable computing environment and hope to lend credence to the need for system thinking as well as strategic thinking when making cloud service adoption decisions.

Kannan Govindan, et.al [3] states that in the big data era, systems reliability is critical to effective systems risk management. In this paper, a novel multi objective approach, with hybridization of a known algorithm called NSGA-II and an adaptive population-based simulated annealing (APBSA) method is developed to solve the systems reliability optimization problems. In the first step, to create a good algorithm, we use a co-evolutionary strategy. Since the proposed algorithm is very sensitive to parameter values, the response surface method is employed to estimate the appropriate parameters of the algorithm. Moreover, to examine the performance of our proposed approach, several test problems are generated, and the proposed hybrid algorithm and other commonly known approaches (i.e., MOGA, NRGA, and NSGA-II) are compared with respect to four performance measures: 1) mean ideal distance; 2) diversification metric; 3) percentage of domination; and 4) data envelopment analysis. The computational studies have shown that the proposed algorithm is an effective approach for systems reliability and risk management.

Naresh Manwani et.al [4] tells us that he explores noise-tolerant learning of classifiers. We formulate the problem as follows. We assume that there is an unobservable training set that is noise free. The actual training set given to the learning algorithm is obtained from this ideal data set by corrupting the class label of each example. The probability that the class label of an example is corrupted is a function of the feature vector of the example. This would account for most kinds of noisy data one encounters in practice. We say that a learning method is noise tolerant if the classifiers learnt with noise-free data and with noisy data, both have the same classification accuracy on the noise-free data. In this paper, we analyze the noise-tolerance properties of risk minimization (under different loss functions). We show that risk minimization under 0–1 loss function has impressive noise-tolerance properties and that under squared error loss is tolerant only to uniform noise; risk minimization under other loss functions is not noise tolerant. We conclude this paper with some discussion on the implications of these theoretical results.

Hing Kai Chan et.al [5] states that many optimization techniques have been proposed over the years to improve the performance of supply chains. Although these approaches have been shown to be effective, most of them were developed without considering uncertainties in supply chains to simplify the analysis. In fact, uncertainties can deteriorate the performance of supply chains, such as increase in total cost, or drop in fill rate, of the whole system. In distributed supply chains, participating members are not under a sole control by a central authority, the problem is even more stringent due to incomplete information sharing, or so called asymmetrical information sharing. One way to improve the system performance is to achieve coordination among participating parties through establishment of contracts. The objectives of this paper are i) to evaluate the effects of demand uncertainty in a distributed supply chain, which is modelled as an agent-based system; ii) to propose a coordination mechanism with early order completion contract to minimize the negative impacts of demand uncertainty; and iii) to present associated simulation results. Performance of the system is measured in terms of costs and fill rate. Simulation results indicate that the proposed contract approach is able to improve the performance measures of the system

Bin Shen, el.al [6] tells us that, motivated by the popular markdown money policy (MMP) in the textiles and clothing (TC) industry, in this paper, we explore how this policy performs in a two-stage TC/fashion supply chain with an upstream risk-averse manufacturer (supplier) and a downstream risk-neutral retailer. Specifically, we investigate both the optimal decisions of the risk-averse supplier with respect to the MMP contract parameters and the optimal ordering decision of the risk-neutral retailer so that the whole supply chain can be coordinated (i.e., optimized). We then conduct a numerical study with the real data from two companies to explore the performance of the optimal MMP proposed in our paper. Important insights and specific implications to the industry practitioners are discussed.

**2.3 ISSUES IN THE EXISTING SYSTEM**

The following are the main issues in the existing systems

* User has to have the corresponding ATM card to withdraw a money from ATM machine.
* Integration of multiple accounts into a single user interface is not implemented.
* User will not be able to withdraw the money beyond the limit.

**2.4 PROPOSED SYSTEM**

In this implementation, all of the Users accounts are integrated using Aadhaar card. Using this application user can do ATM transaction using input of Aadhaar card. Once Aadhaar number is authenticated then user has to be displayed with list of banks connected to that account. Now user can choose any bank & do any normal transaction like withdrawal of money. Formula password is given to the main server initially. If user withdraws amount beyond the permitted withdrawal limit then Formula based authentication is verified. Amount withdrawal is permitted only after successful formula authentication and SMS alert is initiated regarding amount withdrawal. All the user transaction with respect to Aadhaar card is updated to the main server. Hadoop Big data is implemented for data analysis of overall transaction of any user in any bank. Complete tracking would happen through this interface.

**2.5 SUMMARY**

This chapter covered the issues in the existing system and also covers the survey paper which gives the technical results. It also highlights the brief overview of the proposed system for which the design layout is covered in the next chapter.

**CHAPTER 3**

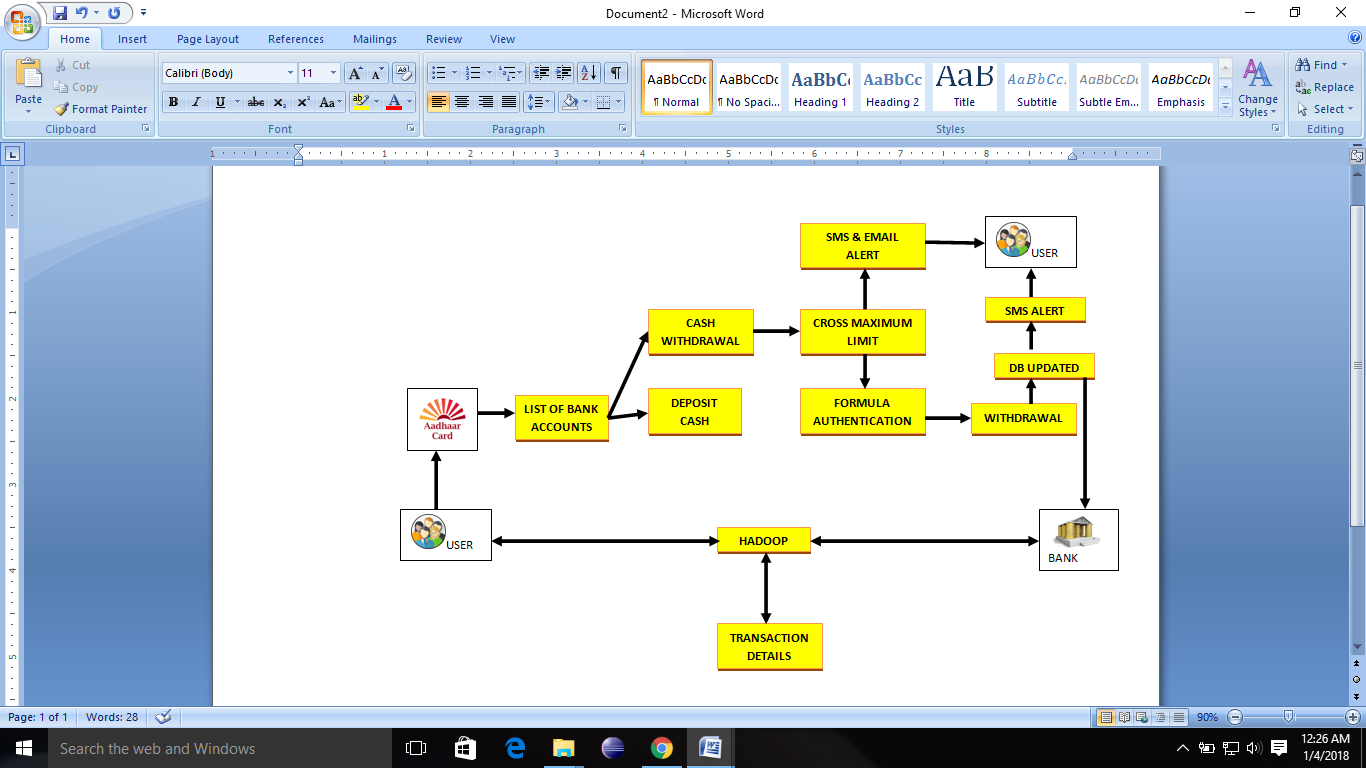
**SYSTEM DESIGN**

* 1. **GENERAL**

System design is the process or art of defining the architecture, components, modules, interfaces and date for a system specified requirements. The main purpose of the design phase is to plan a solution for the problem specified by the required document. The design phase takes an input as the requirement of the Analysis phase. For each requirement, a set of one or more design elements will be produced as a result. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance.

* 1. **SYSTEM ARCHITECTURE**

To overcome the limitations in the existing system, a new system has been proposed which will be easily accessible by all ATM user’s. In this system, first the user has to register himself to get the aadhaar number by entering his personal details and image and using the aadhaar number as a reference he can register with any bank which he wishes to have an account. During registration each bank provides the user with a pin and a formula which can be used during ATM transaction. During the ATM process first the user has to swipe his personal card enter his aadhaar number and password which will then be verified. Then the list of banks will be displayed where the user can select the bank he wants and enter the pin and withdraw or deposit cash. During withdrawal if the user crosses the bank limit he will first be intimated with a message and formula authentication provided to the user during registration must be used to withdraw the amount. Formula authentication is a formula which is made up of 26 alphabets and 2 operators where each alphabet is given a cryptographic value and the values should be calculated using the formula given and value should be entered to withdraw cash. After the process is over and the cash is collected the user will be intimated using a SMS and Email. All these process gets updated dynamically in the database which is under the tutelage of bank and government. Hadoop is used to process the user queries and transaction history to the user in case of any information needed by user.

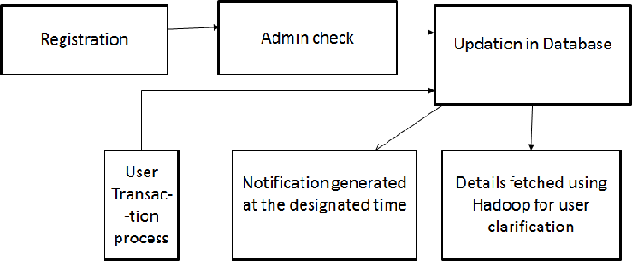


**Figure 3.1** System architecture

**3.3 FUNCTIONAL ARICHTECTURE**

The functional architecture is an architectural model that identifies enterprise functions and interactions. It gives an outline of how each and every part of the depicted system works and that function aspects of the proposed design coincide with that of the overall architecture. Generally, more than one architecture can satisfy the requirements. It serves as a bridge between the software engineers and architects and shows the clear view of overall input and output for every module of the project. It is used to support functional and performance test development.

The functional architecture of the proposed system depicts that the registration process is taken place where the details are collected and stored in a database. This process done by the administrator. A message queue is maintained using round robin scheduling which is used to send the customized notifications for checkup dates. The messages are arranged in an order. Based on the time slot notification is sent to the mobile. If the checkup date is preponed then it is updated in the database and the notification is sent accordingly. Figure 3.2 depicts the functional architecture for the system.

****

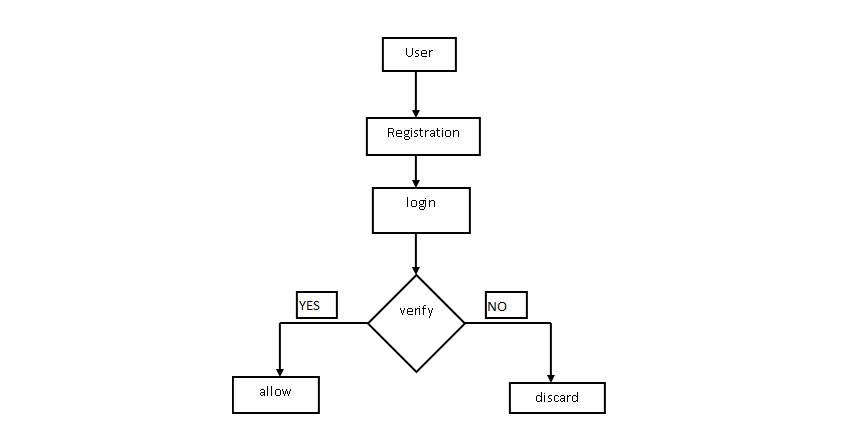
**Figure 3.2** Functional architecture

**3.4 MODULAR DESIGN**

The proposed system consists of the following modules:

* User registration
* User authentication
* Database update and Query retrieval

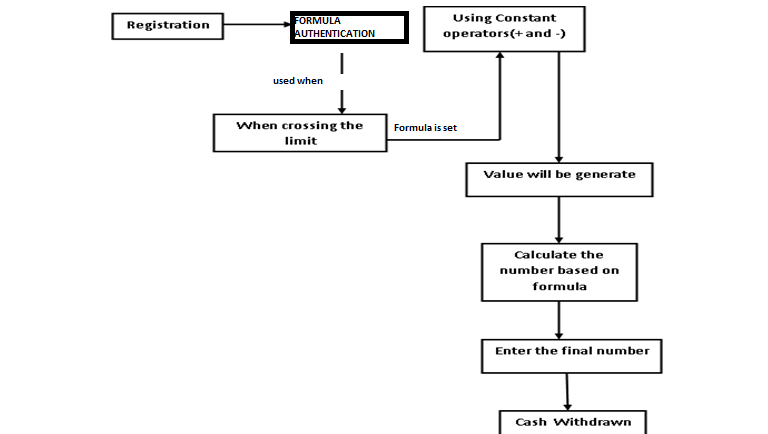
**3.4.1 USER REGISTRATION**

****

**Figure 3.3** Module diagram for User registration

In this module, first the user registers all his details where he will first get his aadhaar number and a passcode for that number by entering his personal information. Then using this number as a reference he can create bank account in the respective banks. After registration in the bank he will get a pin and formula which can be used during the withdrawal process. During ATM transaction first the user swipe’s his personal card which is linked with the aadhaar card and he will enter his aadhaar number and password which gets verified with the server

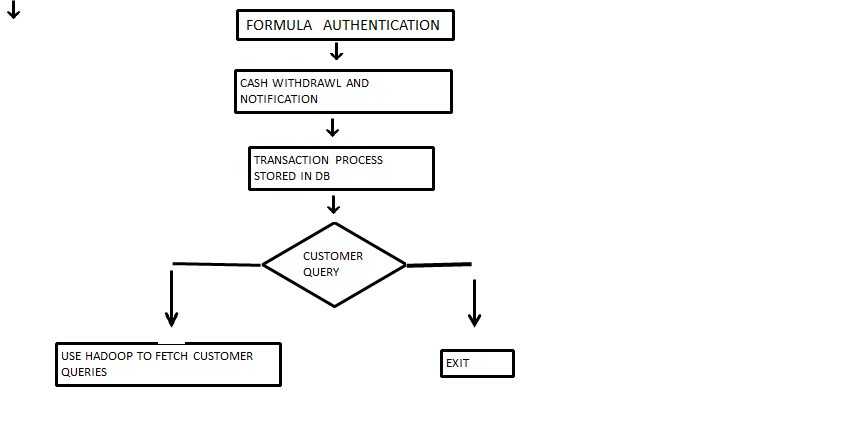
**3.4.2 USER AUTHENTICATION**

****

**Figure 3.4** Module diagram for User authentication

After verification of aadhaar details the user is allowed to move forward if the details are valid and if invalid the process is discarded. After validation the user is displayed with list of banks where he can choose where to carry on the process and enter the pin. In case of withdrawal if the amount exceeds the daily withdrawal limit the user is intimated used SMS first and has to calculate the values entered on the screen using his formula provided during registration. After entering the value after calculation it is checked and if the value is correct amount is allowed to withdraw else the process is discarded. After the withdrawal is done the user is intimated using a SMS.

**3.4.3 DATABASE UPDATION AND QUERY RETREIVAL**

****

**Figure 3.5** Module diagram for Database updation and Query retreival

After withdrawal of amount using formula authentication all the data’s are stored in the database dynamically. This database is accessible by both the government and bank. The database consist of all the user details and transaction history. Since large amount of details are present this is called Big data and can be accessed a tool called Hadoop. User queries like recent transaction, last 5 transactions, bank-wise transactions, month-wise transaction, user profile etc., can be viewed.

**3.5 SYSTEM REQUIREMENTS**

**3.5.1 Hardware specification**

System : Core i5 CPU 2.4 GHz

RAM : 8 GB

**3.5.2 Software specification**

Operating System : Windows

Programming code : Java

Database : MySQL, Apache Hadoop -2.7.0

**3.6 SUMMARY**

The design methodology adopted for the proposed system is presented in this chapter. The functional architecture gives the overview of the system. All the modules and sub-modules for the system are explained in detail. The design documents such as Functional Architecture, Modular design which describes about each module are explained in this chapter.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

**4.1 GENERAL**

This chapter brings out the implementation of the projects. System implementation refers to the realization of the technical specification or algorithm as a program or software components, through computer programming and deployment.

The development platform for implementing the project is given below:

**Development platform:**

Operating System : Windows 7 or higher

Language : Java, SQL

Tool : Net beans, Apache Hadoop-2.7.00

**4.2 OVERVIEW OF THE PLATFORM**

**4.2.1 NETBEANS**

**4.2.1.1 OVERVIEW**

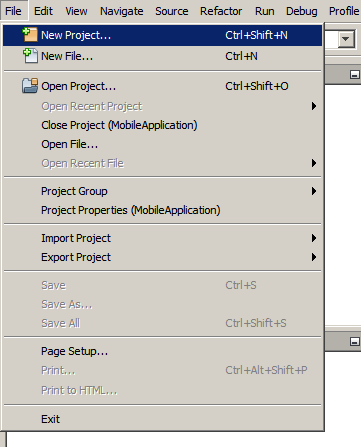
## NetBeans is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). NetBeans allows applications to be developed from a set of modular [software components](https://en.wikipedia.org/wiki/Software_component) called modules. NetBeans runs on [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [mac OS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux) and [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)). In addition to Java development, it has extensions for other languages like [PHP](https://en.wikipedia.org/wiki/PHP), [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B) and [HTML5](https://en.wikipedia.org/wiki/HTML5)., [Javadoc](https://en.wikipedia.org/wiki/Javadoc) and [Java script](https://en.wikipedia.org/wiki/JavaScript). Applications based on NetBeans, including the NetBeans IDE.

## The Net beans Team actively supports the product and seeks feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.

## 4.2.1.2 CREATING PROJECTS IN NETBEANS

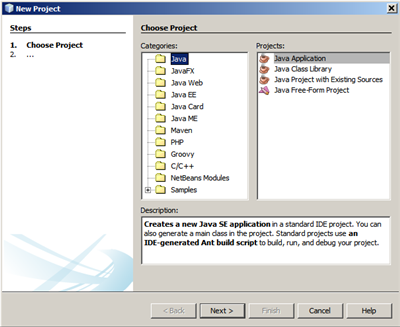
To create an IDE project:

1. Start Net beans IDE.
2. In the IDE, choose File > New Project, as shown in the figure below.



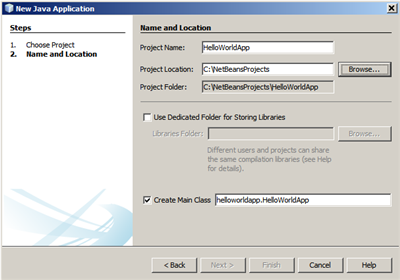
**Fig 4.1 New project**

1. In the New Project wizard, expand the Java category and select Java Application as shown in the figure below. Then click Next.

[](https://netbeans.org/images_www/articles/72/java/quickstart/proj-wizard.png)

**Fig 4.2 Java application**

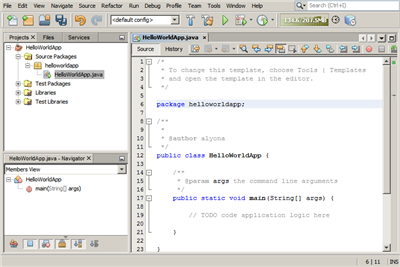
1. In the Name and Location page of the wizard, do the following (as shown in the figure below):
   * In the Project Name field, type HelloWorldApp.
   * Leave the Use Dedicated Folder for Storing Libraries checkbox unselected.
   * In the Create Main Class field, type helloworldapp.HelloWorldApp.

[](https://netbeans.org/images_www/articles/72/java/quickstart/proj-wizard2.png)  
**Fig 4.3 Creating Name and location for application**

1. Click Finish.

The project is created and opened in the IDE. You should see the following components:

* The Projects window, which contains a tree view of the components of the project, including source files, libraries that your code depends on, and so on.
* The Source Editor window with a file called HelloWorldApp open.
* The Navigator window, which you can use to quickly navigate between elements within the selected class.

[](https://netbeans.org/images_www/articles/72/java/quickstart/proj-opened.png)

**Fig 4.4 Navigator Window**

**4.2.2 JAVA**

## 4.2.2.1 OVERVIEW

Java is one of the world’s most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown. As of 2015, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers using and working on it.

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications. The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be Write Once, Run Anywhere.

**4.2.2.2 FEATURES OF JAVA**

* **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
* **Platform Independent** – Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
* **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
* **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
* **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
* **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
* **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
* **Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
* **High Performance** − With the use of Just-In-Time compilers, Java enables high performance.

**4.2.2.3 PACKAGES**

Packages are used in Java in order to prevent naming conflicts, to control access, to make searching/locating and usage of classes, interfaces, enumerations and annotations easier, etc. A Package can be defined as a grouping of related types (classes, interfaces, enumerations and annotations) providing access protection and namespace management.

Some of the existing packages in Java are

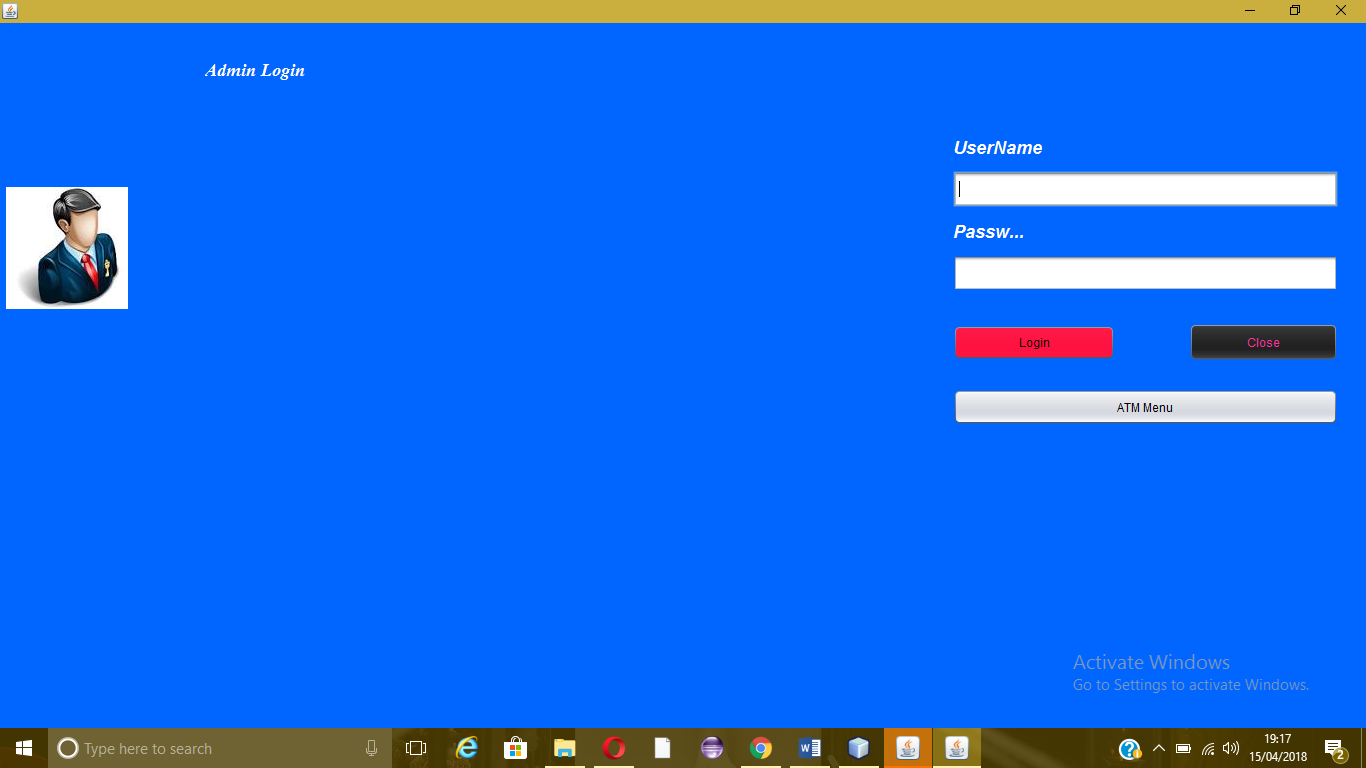
* **java.lang** − bundles the fundamental classes
* **java.io** − classes for input , output functions are bundled in this package

Programmers can define their own packages to bundle group of classes/interfaces, etc. It is a good practice to group related classes implemented by you so that a programmer can easily determine that the classes, interfaces, enumerations, and annotations are related.

Since the package creates a new namespace there won't be any name conflicts with names in other packages. Using packages, it is easier to provide access control and it is also easier to locate the related classes.

**4.3 MODULAR IMPLEMENTATION**

This project considers registration phase that will take all the necessary details of the user. The registration details mainly consist of aadhaar details and Bank details. These details helps to identify the person uniquely. The registration field mainly consist of two phases. One is the aadhaar registration and another is the bank registration. Aadhaar registration is mainly done in cases where the person does not have an registered aadhaar number. Bank registration depends on the user’s wish where he can create/integrate accounts depending on his preference. Once the registration is done all the details get dynamically stored in the database.

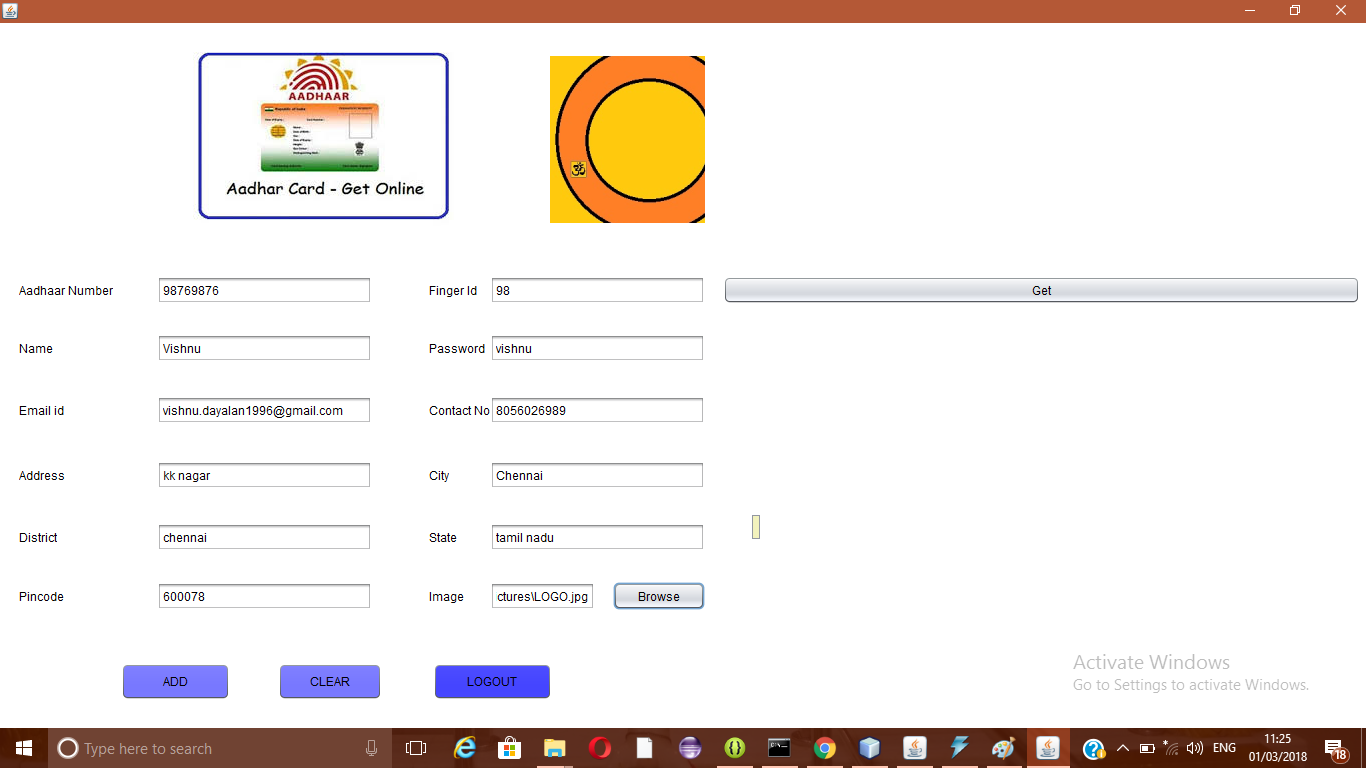


**Figure 4.5** Admin page

The first page that pop’s up in this implementation is the admin page where the admin can view the registration details of the user, add or remove details, check every day happenings etc., also the admin grants permission for the user to enter into the registration phase. Admin shares the login page where the user has to enter his credentials and start registration. After logging in the user finds two stage registration. First among them is aadhaar registration.

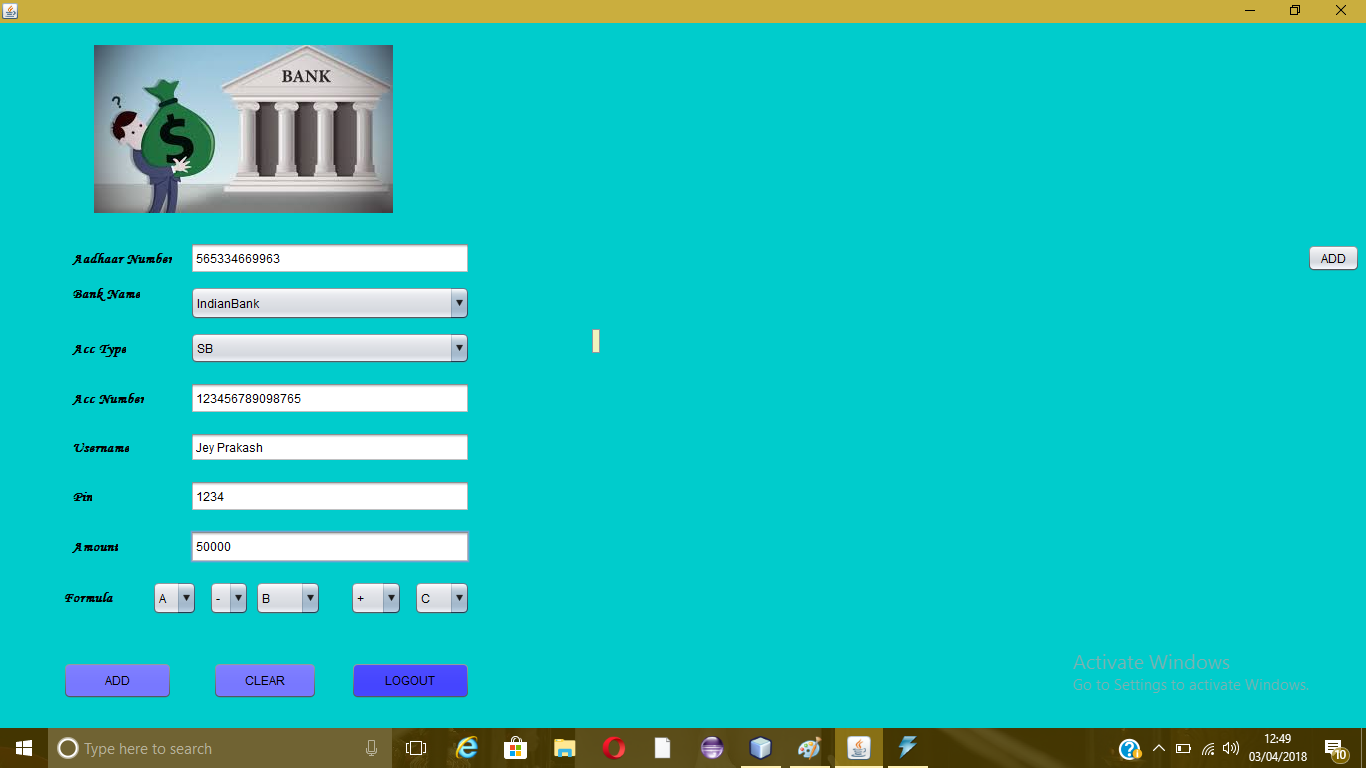
**4.3.1 USER REGISTRATION**

This registration phase involves obtaining the input data from the user. The registration details mainly consist of aadhaar details and Bank details. These details helps to identify the person uniquely.



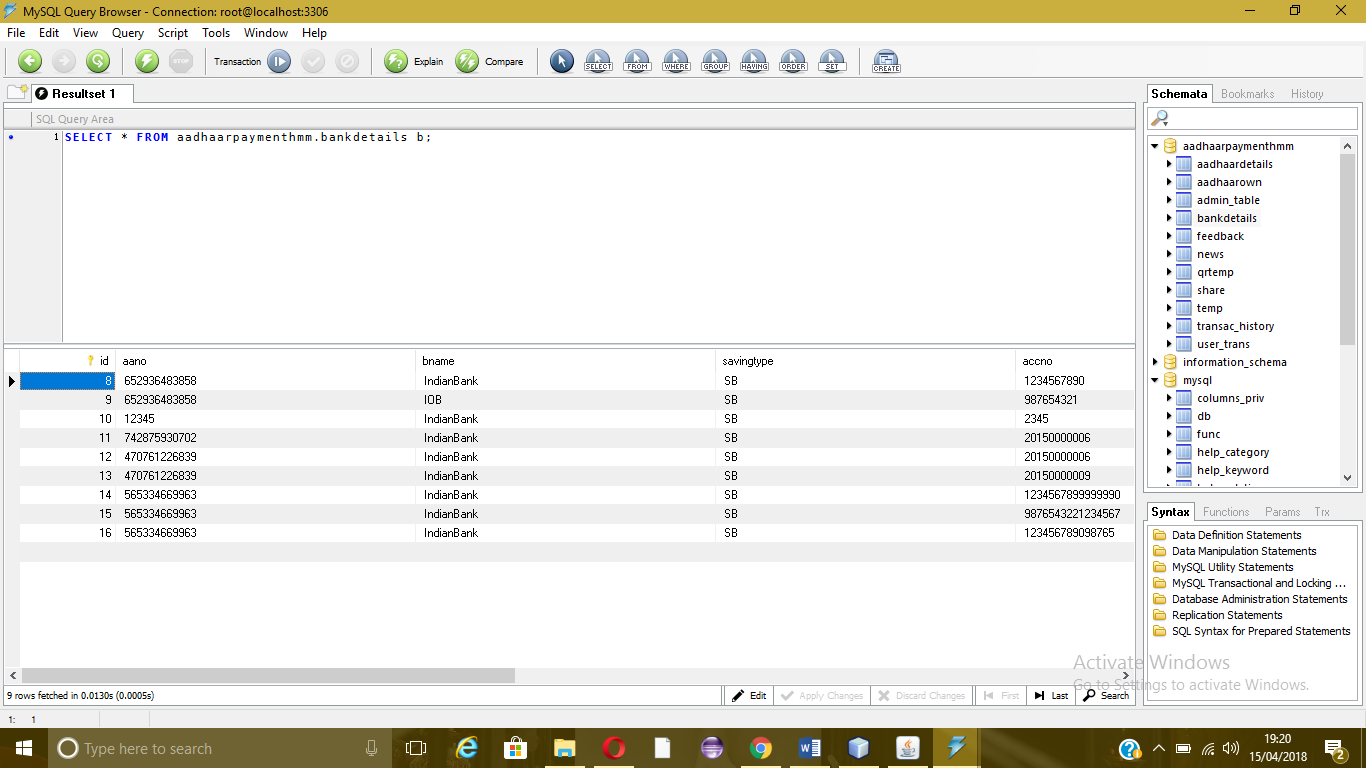
**Figure 4.6** Aadhaar Registration by user

After logging in the first step is the aadhaar registration where it is not mandatory to register if you already have an aadhaar number. If you don’t have one then you have to start with the registration process. After entering the aadhar registration phase the aadhaar number and finger id for user gets generated automatically and the other personal details like Name, Password, Email id, Contact number, Address, City, District, State, Pincode and Image of the user must be entered. After verifying the entered details the user can press ADD to save the details to the database. This aadhaar details are social details and hence saved in governmental database.



**Figure 4.7** Bank Registration of user

After aadhaar registration is done the next phase of registration is the bank registration where the person has to create bank accounts with reference to bank accounts or can also integrate the created accounts with respect to aadhaar number. While creating bank account first the person has to enter his aadhaar number and select the bank which he wishes to have an account then enter his account number and pin. Username and amount is also entered. Then the new process implemented here is the creation of formula where the user can create his own formula or will be provided with a random formula. For example, A+B-C is the preferred formula using which the person can withdraw money if the limit exceeds.

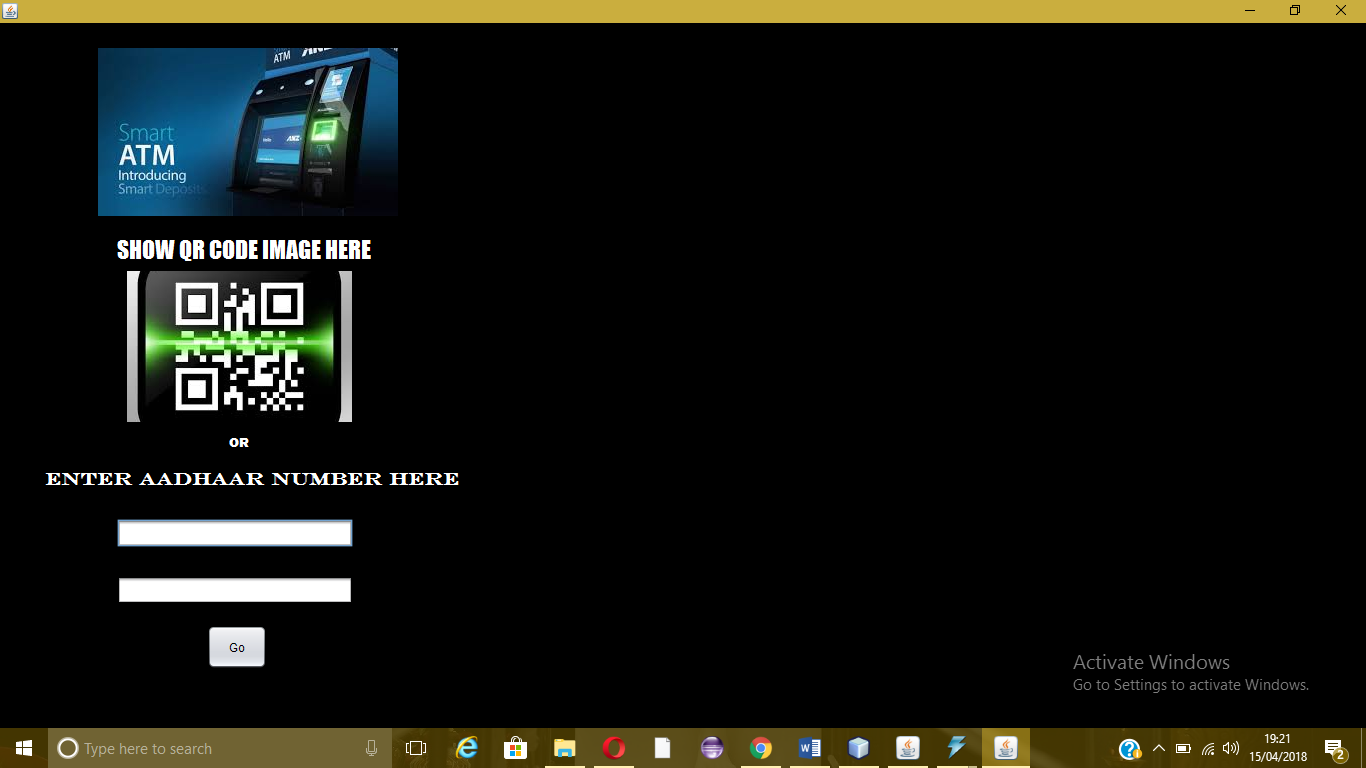


**Figure 4.8** Registered Data

After registration of both aadhaar details and bank details all the details gets displayed in the database. This database is accessible only by the admin and not prescribed for the user. The admin can check and modify user details in case of any user query or admin query.

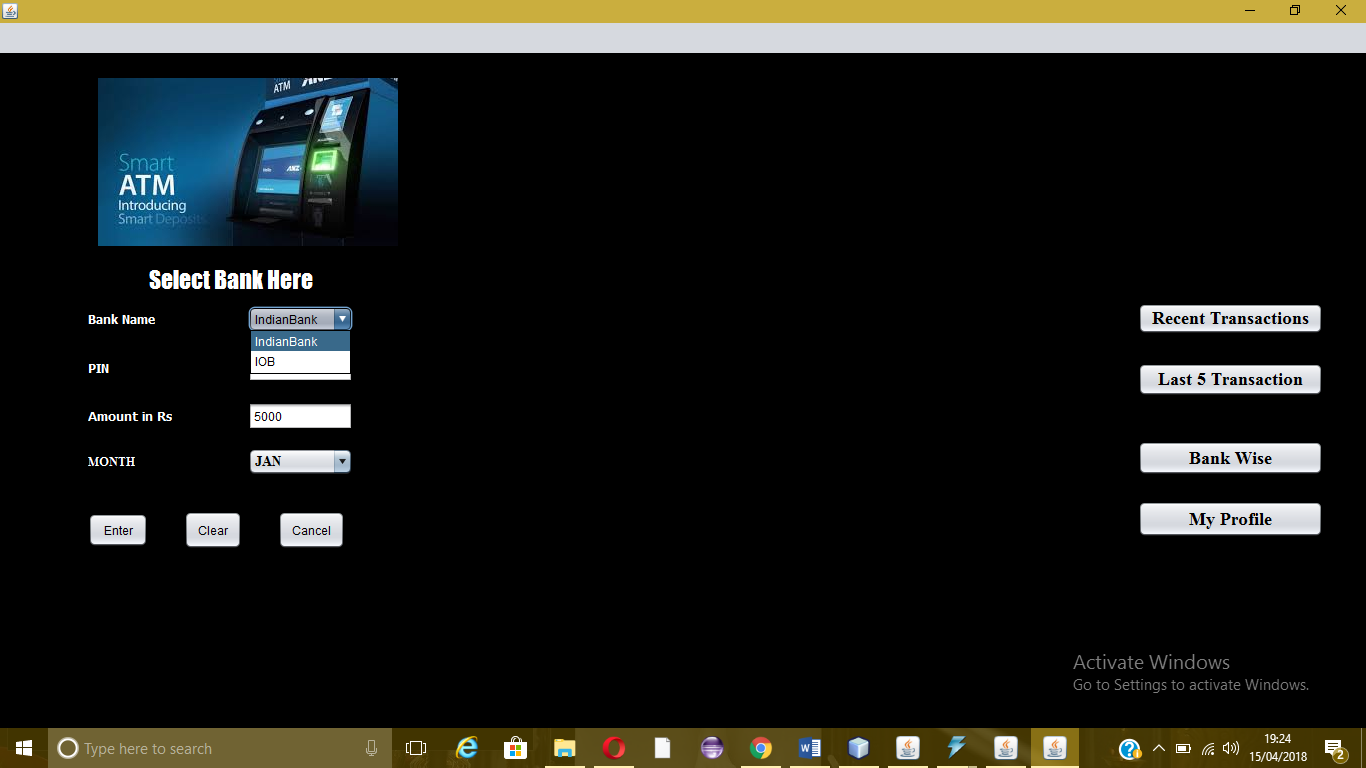
**4.3.2 USER AUTHENTICATION**

After registration of details and storing in database the next phase is the ATM transaction and Formula Authentication. Here the user first swipe his personal card in the ATM machine and the card is verified and a welcome message is shown. After this the ATM transaction process starts.



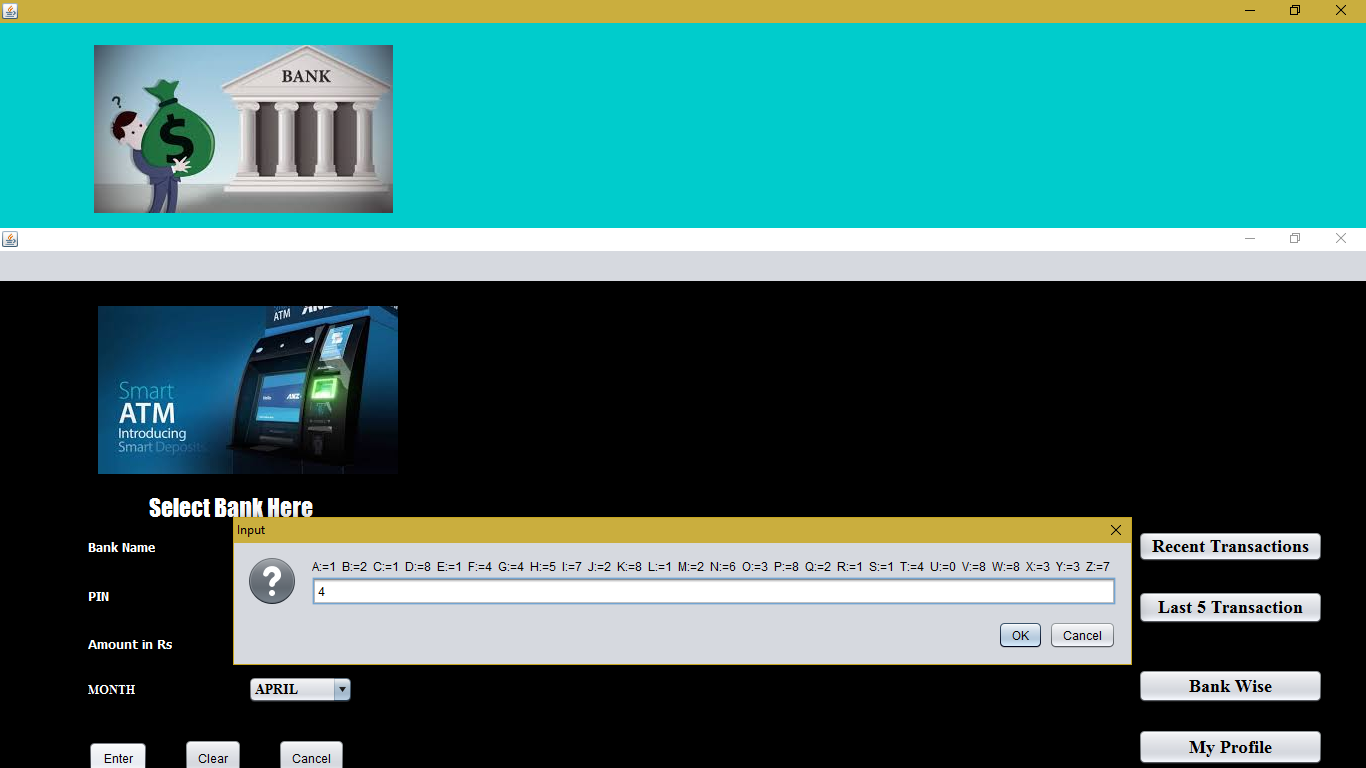
**Figure 4.9** ATM Transaction login

After welcome message the user is asked to enter his aadhaar number and password to login to his account. The login credentials are checked with the database and is validated. If the entered details are correct the user is allowed to proceed further else the process is discarded.

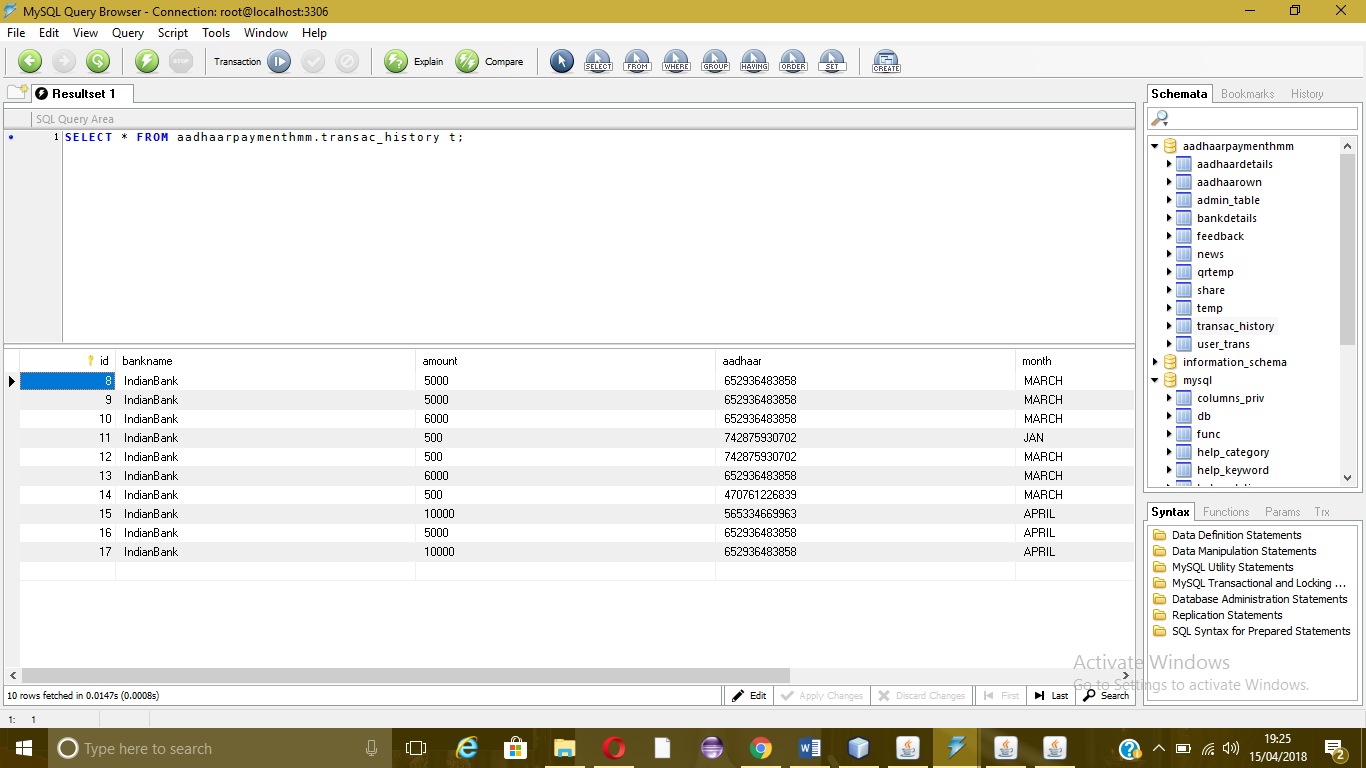


**Figure 4.10** Bank selection and withdrawal page

After logging in the user has to select the bank which he prefers to withdraw money and enter the pin. Later the pin is verified and asks the user to enter the amount. If the withdrawal amount crosses the daily withdrawal limit then the user is notified using a message to his mobile then he has to perform the formula authentication. The user is given the formula during registration and the machine displayed some random cryptographic values and the values must be calculated using his formula and the value must be entered. For example, A+B-C is the formula for a particular user and the values displayed are A=5, B=6, C=4 then user can calculate the value and enter the answer as 7. If the entered answer is correct you will get a message stating to collect your cash else if the answer is wrong the process is discarded and should be started from the first.



**Figure 4.11** Formula Authentication

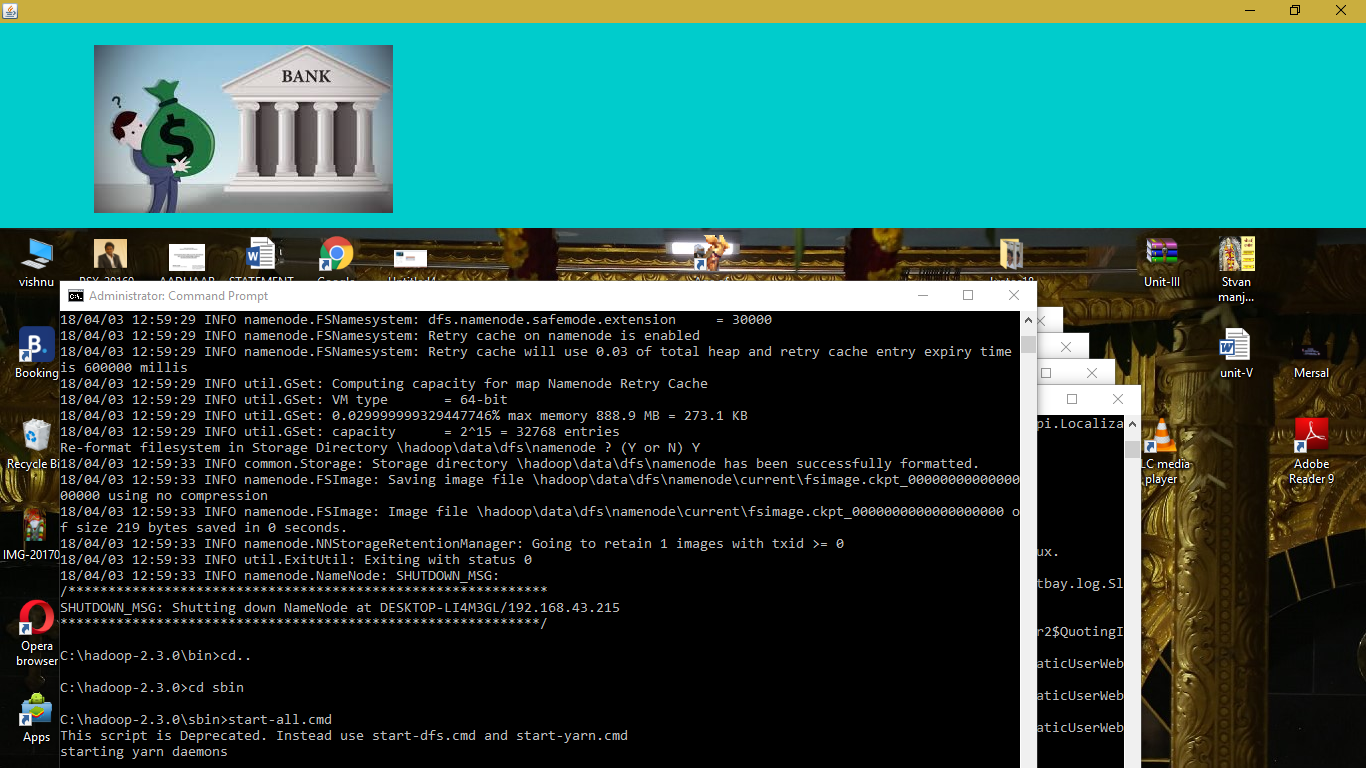


**Figure 4.12** Transaction updated in database

After the withdrawal process is done the transaction history is stored in the database and can be accessed by admin.

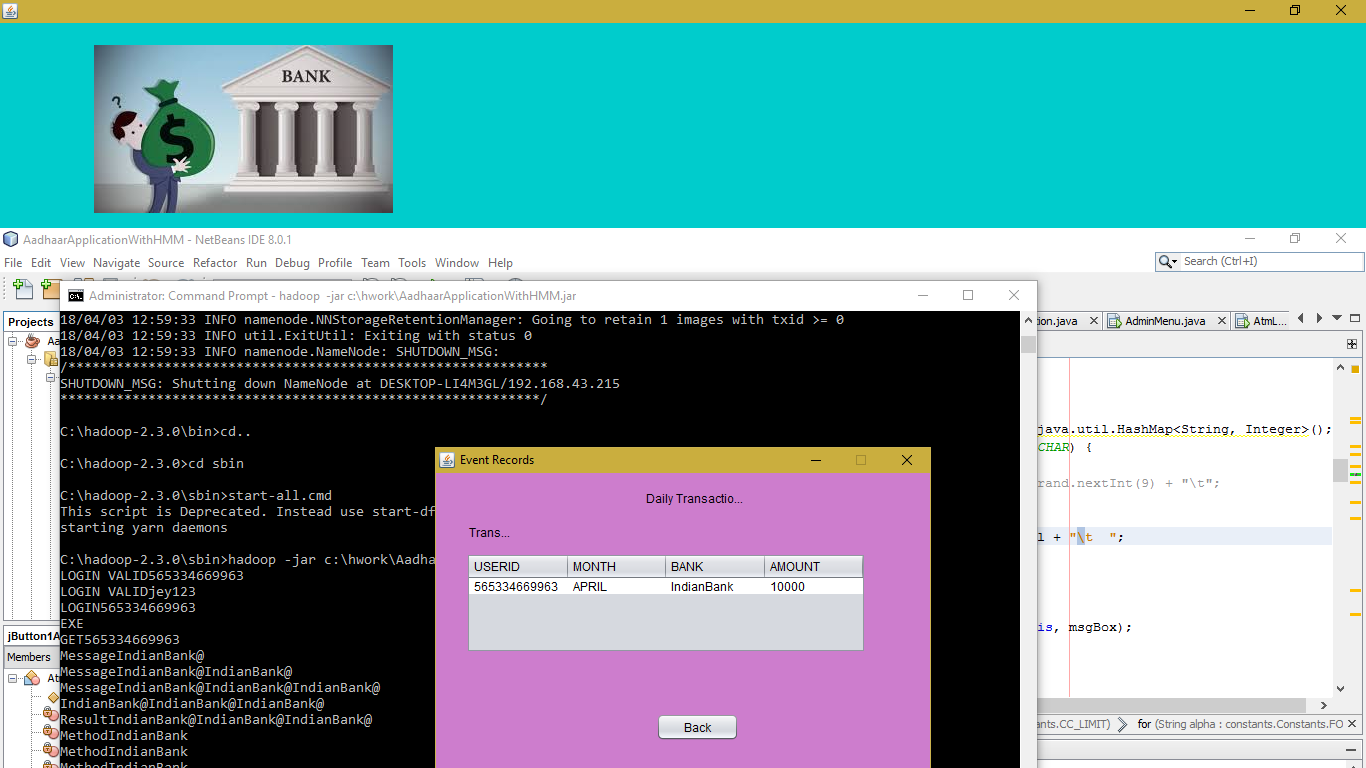
**4.3.3 DATABASE UPDATE AND QUERY RETREIVAL**

After the ATM process is done and the transaction history is stored in the database the next phase is that the transaction details can be viewed by the user for further clarifications.



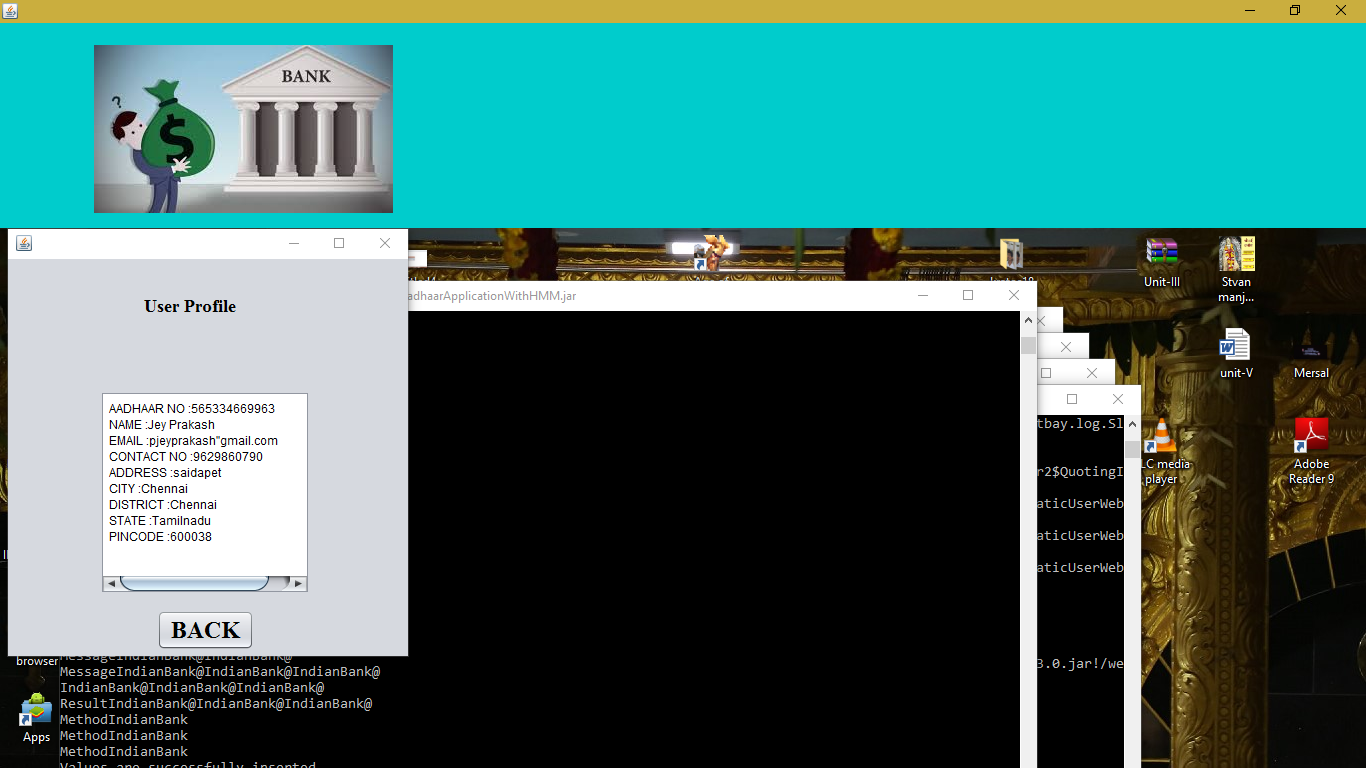
**Figure 4.13** Start Hadoop

To view the user transaction and other queries stored in the database the admin has to first run hadoop. Hadoop is a database retrieval tool which is used to retrieve large amount of data at a much faster rate. Since large datasets are stored in the database including user registration and transaction history we use Hadoop to retrieve data at a faster rate since it is 10 times faster than RDBMS.

****

**Figure 4.14** Daily transaction details of user

After running Hadoop as per the user query the daily transaction of a particular user has been shown. Here the account number, month, bank name and amount are displayed.



**Figure 4.15** User profile

Here user profile has been displayed using the data retrieval tool called Hadoop.

**4.4 SUMMARY**

This chapter has brought out the implementation details of the proposed system. It also describes the overview of the platform. In this chapter the snapshots of different scenarios at different instants are also presented.

**CHAPTER 5**

**SYSTEM TESTING AND PERFORMANCE ANALYSIS**

**5.1 GENERAL**

Testing is the process of trying to find out every conceivable fault or weakness in a work product. It provides a way to check functionality of components, subassemblies, assemblies and finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations.

Performance Analysis is a specialist discipline involving systematic observations to enhance performance and improve decision making, primarily delivered through the provision of objective, statistical (Data Analysis) and visual feedback (Video Analysis).

**5.2 TEST CASES**

**5.2.1 TEST CASE-I: USER REGISTRATION**

The data is obtained from the user using a registration form maintained by the administrator. It collects information about the user’s details such as Name, Age, Address, Mobile number, User name, Password, Image, bank details, pin, amount, etc. These information help to identify the user uniquely. Based on the data, each user details is stored separately using his aadhaar details and bank details. Testing is done based on the specification that are stored in database. This is illustrated in the Table 5.1.

**Table 5.1** Test Case: User Registration

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST DESCRIPTION** | **TEST INPUT** | **EXPECTED RESULT** | **ACTUAL RESULT** |
| Checking for missing values | Values for all specification given | Should accept all data | Data is accepted |
| Checking for missing values | Name, Username, Password and Contact | Should accept all data | Data is accepted |
| Checking for missing values | If some of details are not entered in the registration form | Should not accept data | It accepts the user and the data that is entered and leave other data field blank |
| Checking for form validation | The Name is entered | Should not accept characters more than 15 | More than 15 characters are not accepted and shows warning |
| Checking for form validation | The contact number is entered | Should contain only numbers and cannot exceed 10 numbers | Shows warning for entering characters |
| Checking for missing values | All specifications are stored in database | Should accept all data | All the details are present in the database |

**5.2.2 TEST CASE-II: USER AUTHENTICATION**

This process mainly deals with the ATM transaction. First the login credentials should be checked and pin number should be checked based on which the process continues. If withdrawal limit is crossed then formula authentication phase is entered where values from A to Z is displayed and the values is entered. This is illustrated in the Table 5.2

**Table 5.2** Test Case: User Authentication

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST DESCRIPTION** | **TEST INPUT** | **EXPECTED RESULT** | **ACTUAL RESULT** |
| Checking for the generation of message | Mobile number is entered | Should accept the data of 10 numbers | Data is accepted |
| Checking for the login credentials | Aadhaar number and password | Should match the data stored in the database | Proceed further if the details match |
| Formula authentication if limit is crossed | Values for alphabets are displayed from A to Z | Calculate values as per formula | Values should match for that particular formula |

**5.2.3 TEST CASE-III: DATABASE UPDATE AND QUERY RETREIVAL**

In this phase, Hadoop is run by the administration to retrieve data from database. This process works according to user queries. Also details won’t be shown if not entered in database correctly or data is erased. This is illustrated in the Table 5.3

**Table 5.3 Test Case:** Database Update and Query retreival

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST DESCRIPTION** | **TEST INPUT** | **EXPECTED RESULT** | **ACTUAL RESULT** |
| Checking for details | Login to database to check if details are present | Data is present | Data is accepted |
| Start Hadoop | Enter commands to start the process | Starting Hadoop, Yarn, Name node, HDFS | Verifies the process and runs Hadoop. |
| Completing user queries | Taken from the Database | Shows the details requested by the user from database | Displayed in the tabular column format. |

* 1. **PERFORMANCE MEASURES**

The following are the performance measures in this project:

* Time

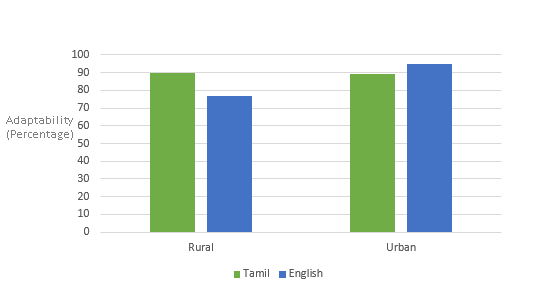
Timeis a quantity of various measurements used to sequence events and to compare the duration of events. Time is a series of snapshots that have been taken at regular intervals.

* Battery level

This parameter determines the percentage of the consumption of power.

* 1. **PERFORMANCE ANALYSIS**
* **User adaptability in rural verses urban areas**

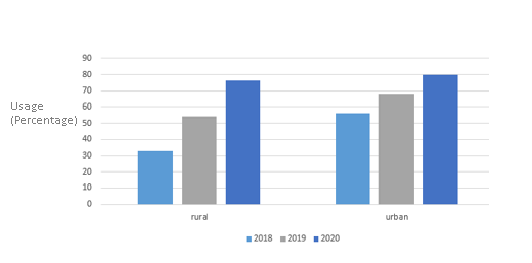
This graph provides user adaptability on the new ATM transaction machine in rural and urban. The application has been tested for a group of 30 people and the feedback has been collected from them. It has been found that about 90 percent of user in rural areas are more adaptable in using Tamil ATM transaction rather than English ATM transaction. Whereas when it comes to urban areas they are comfortable in using the ATM transaction machine in both the languages. Figure 5.1 shows the graph.



**Figure 5.1** User adaptability in rural verses urban areas

* **Usage of ATM Transaction in rural verses urban areas**

This graph provides an analysis about the total percentage of rural and urban users using the application. The number of users increases gradually for every year. It shows that an increase of 22% of users in case of rural and 35% of users in case of urban. Figure 5.2 shows a prediction for the usage of the application in the following years.



**Figure 5.2** Usage of ATM transaction in rural verses urban areas

* **Security versus Threat**

The below mentioned graph shows the security and threat comparison year by year. As years pass by both security and threat grow by but the security is insufficient compared to the threat. This process helps to meet the threat and cannot be hacked easily. Figure 5.4 shows the graph of time verses the battery level.



**Figure 5.3** Security versus Threat

**5.5 SUMMARY**

In this chapter, the test case scenario of the project is mentioned. Test case strategy and different types performed for this project is defined and the results are briefed. Appropriate performance measures for the project are identified and defined. Performance analysis and the results are obtained.

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

**6.1 CONCLUSION**

This process of multibank account integration was mainly introduced to provide security to user’s bank accounts and to prevent it from unwanted access. Also this process reduces the burden of people who carrying multiple ATM card in-hand by providing a single personal card which is linked with aadhaar card. Also aadhaar card is used as the main process of verifying and validating the user as the user is asked to enter the aadhaar number and password as first step of security and the details are checked with the server. This is one of the major enhancement which is formulated because according to recent reports a huge amount of money is being forged by hiding it from the government. This process help the government to monitor the database which in turn will reduce these forgeries. Also security is provided by means of formula authentication when the withdrawal limit gets exceeded. This is used as the second step of security since this formula is unique for each and every user and the values displayed are cryptographic values. Finally the amount is intimated by means of SMS and Email.

**6.2 FUTURE WORK**

The data’s stored here are large voluminous data because it consist of all the user’s transaction details as well as personal details hence Hadoop is used to process these data’s and solve user queries. In future, this personal card can not only be used as a ATM card but can all be used in retail, healthcare, ID card etc., where all the user needs are present in one single place. RDBMS which was earlier replaced by Hadoop which is comparatively faster and secure and in future can be replaced by Spark which is 100 times faster than Hadoop.

**APPENDIX**

package aadhaarapplication;

import com.constant.Userdetails;

import com.constant.constant;

import com.entity.AddDataoperation;

import com.entity.UserEntity;

import com.nura.sms.SendSMS;

import com.security.Encrypt;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class AtmLogin extends javax.swing.JFrame {

String fingerid="";

public AtmLogin() {

initComponents();

GetImage();

/\* java.io.FileInputStream fis = null;

try {

initComponents();

Encrypt e=new Encrypt();

try {

String mainfunc=e.main("505295380133");

System.out.println("MAIN FUNC"+mainfunc);

} catch (Exception ex) {

Logger.getLogger(AtmLogin.class.getName()).log(Level.SEVERE, null, ex); }

// tf\_rfid.setEditable(false);

fis = new java.io.FileInputStream(constant.RFID\_FILE\_LOCATION);

java.util.Scanner scan = new java.util.Scanner(fis);

while (scan.hasNextLine()) {

//tf\_rfid.setText(scan.nextLine());

fingerid=scan.nextLine(); }

} catch (FileNotFoundException ex) {

//Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex); }

finally {

try {

fis.close();

} catch (IOException ex) {

//Logger.getLogger(UserLogin.class.getName()).log(Level.SEVERE, null, ex);

}

}\*/

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

jPanel1 = new javax.swing.JPanel()

jLabel4 = new javax.swing.JLabel();

jLabel1 = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jTextField2 = new javax.swing.JTextField();

jPasswordField1 = new javax.swing.JPasswordField();

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jScrollPane1.setViewportView(jTextArea1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(0, 0, 0));

jPanel1.setToolTipText("");

jPanel1.setName("HomePage"); // NOI18N

jLabel4.setIcon(newjavax.swing.ImageIcon(getClass().getResource("/com/images/atm.jpg")));

jLabel1.setFont(new java.awt.Font("Monotype Corsiva", 3, 14)); // NOI18N

jLabel1.setText("Aadhaar Number");

jLabel9.setIcon(new javax.swing.ImageIcon(getClass().getResource("/com/images/qrscan.jpg")));

jLabel9.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jLabel9MouseClicked(evt);

}

});

jLabel2.setFont(new java.awt.Font("Impact", 0, 24));

jLabel2.setForeground(new java.awt.Color(255, 255, 255));

jLabel2.setText("SHOW QR CODE IMAGE HERE");

jLabel3.setFont(new java.awt.Font("Franklin Gothic Heavy", 0, 14));

jLabel3.setForeground(new java.awt.Color(255, 255, 255));

jLabel3.setText("OR");

jLabel5.setFont(new java.awt.Font("Engravers MT", 1, 18)); // NOI18N

jLabel5.setForeground(new java.awt.Color(255, 255, 255));

jLabel5.setText("Enter Aadhaar Number Here");

jButton1.setText("Go");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup( jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup() .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(117, 117, 117) .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel1) .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jPasswordField1, javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 236, Short.MAX\_VALUE))))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(98, 98, 98)

.addComponent(jLabel4))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(127, 127, 127)

.addComponent(jLabel9))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(229, 229, 229)

.addComponent(jLabel3))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(45, 45, 45)

.addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 437, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(207, 207, 207)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 60, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(43, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup( jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(jLabel4)

.addGap(18, 18, 18)

.addComponent(jLabel2) .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED\_SIZE, 151, javax.swing.GroupLayout.PREFERRED\_SIZE) .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel3)

.addGap(18, 18, 18)

.addComponent(jLabel5)

.addGap(29, 29, 29)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel1) .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 35, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 44, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(59, 59, 59)));

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE));

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE));

pack();

}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

//JOptionPane.showMessageDialog(rootPane, "Place ur Finger at Finger Print Device");

//String aadhaarno=jTextField3.getText().toString().trim();

String aadhaarno=jTextField2.getText().toString().trim();

String pass=jPasswordField1.getText().toString().trim();

AddDataoperation ad=new AddDataoperation();

if(ad.LoginValid(aadhaarno, pass)){

this.dispose();

Userdetails.setUseraano(aadhaarno);

UserEntity.setAadhaarno(aadhaarno);

new AtmBankMenu().setVisible(true);

}

else{

JOptionPane.showMessageDialog(rootPane, "InvalidCredentials");

}

}

private void jLabel9MouseClicked(java.awt.event.MouseEvent evt) {

try {

// TODO add your handling code here:

AddDataoperation ad=new AddDataoperation();

ResultSet rs=null;

rs=ad.GetQRId();

if(rs.next()){

String aano=rs.getString(1);

System.out.println(aano);

jTextField2.setText(aano);

}

} catch (SQLException ex) {

Logger.getLogger(AtmLogin.class.getName()).log(Level.SEVERE, null, ex);

}

}

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) { java.util.logging.Logger.getLogger(AtmLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) { java.util.logging.Logger.getLogger(AtmLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) { java.util.logging.Logger.getLogger(AtmLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) { java.util.logging.Logger.getLogger(AtmLogin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new AtmLogin().setVisible(true);

}

});

}

public void GetImage(){

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPasswordField jPasswordField1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JTextField jTextField2;

// End of variables declaration

}

**REFERENCES**

1. Albert Levi and Omer Mert Candan. (2017) “Robust Two-Factor Smart Card Authentication”, IEEE International Black Sea Conferance on Communications and Networking

2. Mrs.G.Aalin Joys, S.Saran Kumar, N.Sathish, SP.Thiruelavazhagan. (2017) “Integration of Multi Bank Multi User in Single Card with User Behaviour Monitoring Using Face Recognition”, International Journal for Science and Advance Research in Technologies (IJSART)

3. Aditya Bodake, Ashwini Bodake, Viraj Baviskar and Prof.N.J.Kulkarnis. (2016) “Multipurpose Smart Card System”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)

4. JL.Ganesh and Yoso Adi Setyoko. (2016) “Multipurpose Smart Card System”,6th International Conference on Advances in Computing & Communications, ( ICACC)

5. Jiaying Weng and Derek S. Young. (2017) “Some dimension reduction strategies for the analysis of survey data”, Journal of Big Data (Springer)

6. Karim Abouelmehdi , Abderrahim Beni-Hessane and Hayat Khaloufi. (2018) “Big healthcare data: preserving security and privacy”, Journal of Big Data (Springer)

7. K.L. Man, W. Chong, and C.O. Chan. (2016) “Social Networking Analysis Using Big data”, “proceeding of the international MultiConference of Engineering and Computer Scientists”

8. Tao Lei, Ao Zhou, Jinglin Li, Qibo Sun, Cunqun Fan and Fangchun Yang. (2016) “A security authentication method based on trust evaluation in VANETs”, EURASIP Journal on Wireless Communications and Networking

9. Zhiyu Zhou, Guoyong Zhao. (2017) “Design and Implementation of the Online Shopping System”, Springer-Verlag Berlin Heidelberg

10. Zhenyu Wu ,Haining Wang, Sushil Jajodia, Xin Ruan. (2016) “Intergration of Multi bank user in single card behaviour monitoring using face Recognitions”, IEEE Transactions on Information Forensics and Security (2016)

11. Tao Lei, Ao Zhou, Jinglin Li, Qibo Sun, Cunqun Fan and Fangchun Yang. (2016) “A security authentication method based on trust evaluation in VANETs”, EURASIP Journal on Wireless Communications and Networking

12.OnlineSource:“<https://en.wikipedia.org/wiki/Java_(programming_language)>”

13.Online Source: “<https://en.wikipedia.org/wiki/Netbeans> ”

14.Online Source: “<https://developer.hadoop.com/setup/index.html>”