

1. INTRODUCTION

ATM is an abbreviation of Automated Teller Machine. It is introduced in the year 1959 for encouraging self-service in retail banking. This makes people to deposit, withdraw and transfer amount without the help of banking personnel and it can be done at anytime and anywhere. At first, the ATM was made to transact for the particular bank customers but later on the ATMs are connected to interbank network, so that it enables people to deposit, withdraw and transfer amount from the ATM machines not belonging to that particular bank (i.e.) any one can access any banks ATM machine to carry out their transactions. ATMs rely on authorization of a financial transaction by the card issuer or other authorizing institution via the communication network. At present every customer has an individual ATM card for each and every bank in which he/she maintains account. So handling the cards, their passwords play a major role here. So to overcome these difficulties we embedded more than one bank account of the user in a single ATM smart card, so that the user can swipe the card and can select the bank from which he/she are interested to carry out transaction.

To overcome this of holding so many Cards it is proposed to design a single card with many Accounts. Different banks can create and enter the individuals' bank account numbers in the same card. Initially bank can give a unique PIN code in the beginning. Here we are developing an application for banking sector particularly for Debit/ATM card section. User can create His account and get ATM card from the bank. He can integrate all his account in other bank can be integrated in this single card with unique PIN number.

OBJECTIVE AND SCOPE OF THE SYSTEM

2. OBJECTIVE AND SCOPE OF THE SYSTEM

2.1 Objective of Smart ATM Card System:

The Objective of carrying out this project is to implement a Smart ATM card System for handling multiple accounts of the same user with a single smart card which embeds all the accounts of the user and providing high end security using a personnel identification number (PIN). This project is using a radically new set of banking transaction protocols with new architectural design. I believe that the implementation of this model will make banking transactions more user friendly, more inexpensive and more secure.

2.2 Scope of Smart ATM Card System:

- The scope of this specific application to create the user of various Banks and do their account access and transactions using this solution.
- The Admin will add Bank details and can update the existing details of the bank.
- The Bank will accept or reject registration of a Customer using the previous details.
- The Bank can access this to see the all Customer transactions, Transfer status, etc
- Provides Response for the queries related to the Customers.
- The Customers can request for multiple bank account access to the Administrator. He
 can view the Account related information.
- The customer is able to perform transactions, can view his Account balance in that particular bank, view his transaction details and statements, change his/her ATM Pin and Bank Access Pin

- This software offers benefits such cash withdrawals, balance transfers, deposits, inquiries, credit card advances and other banking related operations for customers. It also allows the administrator to fix the tariffs and rules as and when required.
- The software takes as input the ATM card number of the user for login purposes. The outputs then comprise of an interactive display that lets the user select the desirable function that he wants to perform.

THEORETICAL BACKGROUND DEFINITION OF PROBLEM

3. THEORETICAL BACKGROUND DEFINITION OF PROBLEM

3.1 Theoretical Background:

Automated Teller Machine (ATM) is an electronic banking outlet, which allows customers to carry out basic transactions without the aid of a branch representative or teller. The basic transactions at an ATM include financial transactions (cash withdrawal) and non-financial transactions (balance inquiry, mini-statement, pin change, etc.).

In order to access an ATM, banks issue debit cards to their account holders. An ATM of a bank can serve debit cards of their own bank as well as other banks. Transactions done at a bank's ATM by their own cards are referred to as On-Us transactions while those done by other bank's cards are Off-Us (or cross ATM or third-party ATM) transactions. Today there are more than 170 thousand ATMs in India and the interoperable feature of these ATMs allow debit card holders to access their bank accounts through any of the ATMs in the country.

RBI introduced one of the most effective regulatory measure in expanding the ATM network it made all transactions on third-party ATMs free. Later, RBI allowed banks to limit the number of free cross ATM cash withdrawals to five a month (and later to five free ATM transactions). RBI's subsequent moves were primarily to address some business model issues of the banks. Furthermore, with the freedom given to choose any bank's ATM to withdraw cash and to additionally carry out non-financial transactions, the system has currently converged where the probability of a debit card being used at a cross ATM instead of own bank's ATM is one in three.

3.2 Drawback in Existing System:

- In the existing system the user has to carry the ATM cards all time
- As you never know which card may be used at what time?
- Suppose the user is at some place and forget to carry the card with him/her, then the
 user has to go back to collect it, which intern wastes his / her precious time as well as
 money.
- Moreover there is a limitation in transaction for the other bank customers in using the
 ATM of some other bank crossing the limit they have to pay transaction fees.
- Different ATM cards to access different bank accounts.
- Difficulty in maintenance of many ATM cards.

3.3 Proposed System:

Banking Transactions with ATM depends on authorization of a financial transaction by the card issuer or other authorizing institution via the communication network. At present every customer has an individual ATM card for each and every bank in which they maintain an account. Hence, handling the ATM cards with confidentiality of their passwords play a major role here. To overcome this difficulty or complexity, embedding all their bank accounts of the users in a single

ATM smart card in such a way that, the users can swipe the card and can select the bank from which they are interested to carry out transactions. Hence, a combination of multiple security compliments is mandatory to provide secure transaction. For security purpose pin password is required during the card authentication time. Since multiple accounts of user involved here so providing high end security is must. As a result, this project proposes for user identification and authentication in Automatic Teller Machine (ATM) using Personal

Identification Number (PIN) authentication to complete proper identification of user. Hence the idea behind this Smart ATM card is that the customers can use a single ATM card to operate different bank accounts instead of having individual card for each bank account and maintaining their pin's, carrying the cards safely which is a tedious process at present scenario. The technology behind the product of the service is that adding all the user bank accounts to a single ATM card

3.4Advantages of Proposed Statement:

- There is no need to carry different cards of different banks.
- All different bank account information will be provided in a single card.
- Provides necessary information in single click.
- Easy to maintain single card than multiple cards.
- More user friendly than the existing system and Reduces transaction cost.
- Provides high security than the present system.
- User can perform transactions for all his/her bank accounts using single ATM card and same ATM PIN, with different Bank Access Pin.

SYSTEM ANALYSIS AND DESIGN

4. SYSTEM ANALYSIS AND DESIGN

4.1 System Analysis:

System analysis is the first and foremost step performed in developing the software to solve a particular problem. In the analysis part, a software developer examines the requirements. Carrying out preliminary investigation identifies the requirements. Analysis consists of two sub phases: Planning and Requirement Definition

During planning phase, cost estimates and work schedules will be planned. Requirement definition is a specification that describes the processing environment, the required software functions, performance constrains (size, speed, machine configuration) and exception handling.

4.1.1 Principles of System Analysis:

- Understand the problem before you begin to create the analysis model.
- Develop prototypes that enable a user to understand how human machine interaction will occur.
- Record the origin of and the reason for every requirement.
- Use multiple views of requirements like building data, function and behavioral models.
- Work to eliminate ambiguity.
- System analysis is a separation of a substance into parts for study and their implementation and detailed examination.

4.1.2 Analysis of Smart ATM Card system:

- a) Functionality: ATM's handle as many as traditional teller operations as possible Traditional ATMs implement most basic daily banking functions, such as deposit, withdrawals and balance checking. These functions are designed based on the performance limitations of computing and networking, which have changed rapidly over the course of the last few years. These limited functions may not be satisfactory by modern standards
 - b) Security: ATMs act as electronic tellers, and security is always an important concern for users. Major security issues are already addressed in the modern ATM design. A password, or PIN number, is used to protect the information. After decades of use, ATMs have proven the effectiveness of this security policy.
- c) Usability: Unlike some other electronic devices, ATMs should be useful to a wide-range of users and those users should be able to use the system with limited or no assistance. This aspect is critical to the new ATM design. Although these issues are already taken into consideration in the current ATM designs, evaluation may reveal that there is room for improvement in current systems. Theoretically, it is also possible that there is no perfect design once we review the usability requirements. We may have to provide a compromised solution to suit the majority.
- **d) Efficiency:** ATMs must be both easy to use and fast. The more time a user spends at an ATM, the more inconvenienced the user feels. Wait times also increase for other users. Clearly, this is an important issue to consider when designing an ATM system.
- e) Accessibility: As ATMs are physical machines, their design must take accessibility concerns into account. Some good accessibility features can be found in current ATM design. For example, the ATM keyboard includes Braille support for the blind.

4.2 System Design:

Software design sets at 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 have 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 for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. System 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.

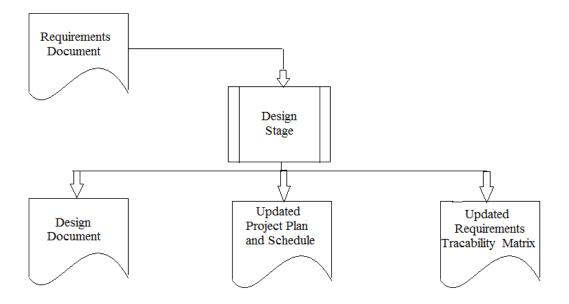


Fig.4.2 A Design Phase

4.2.1 Feasibility study:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are: Economical Feasibility, Technical Feasibility, and Social Feasibility.

- a) Economical Feasibility: This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.
- b) Technical Feasibility: This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.
- c) Social Feasibility: The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system.

This Project Includes Modules such as:

- 1) Admin Module
 - Admin Login
 - Menu-My account
 - Menu Items- Change Password, Logoff, Exit
 - Menu-Master
 - Menu Item- Master(where Admin creates different Banks)
- 2) Bank Module
 - Bank Login
 - Menu-My account
 - Menu Items- Change Password, Logoff, Exit
 - Menu-Master
 - Menu Item- Customer Master(where Bank Personnel creates account for customer)
 - Menu item- Transition Master(which gives the transaction details of customer)
- 3) User Module (ATM System)
 - Validating the User with ATM Pin
 - Selection of Bank
 - Validating the User with Bank Access Pin
 - Perform Transaction
 - Facilities such as (withdrawal, balance enquiry, mini statement, fast cash, change ATM
 Pin and change bank Access Pin).

USER REQUIREMENTS AND SYSTEM PLANNING

5. USER REQUIREMENTS AND SYSTEM PLANNING

5.1 User Requirements:

A good set of user requirements are needed for any project, especially computer system projects, to be successful. This is where many projects fail, in that they do not specify correctly what the system should do. In fact many systems have just been given a deadline for delivery, a budget to spend, and a value notion of what it should do. The root of this problem is:

- Computer system developers rarely have as good an idea of how a business runs and should run, compared with a business user.
- Business users have little idea of what a computer system could achieve for them.

As a result paralysis sets in and business management time is concentrated on meeting timescales and budgets, rather than what is going to be delivered.

5.1.1 Requirements Definition:

The truth is that you do not need a great deal of technical knowledge to specify requirements. In fact it can be a big disadvantage. A requirements for a computer system specifies what you want or desire from a system. For business in particular this is, "What you want or desire you a business advantage"

This advantage need not just be a reduction in costs, In fact many systems justified on a reduction in operating costs, fail to deliver as low skilled but relatively cheap staff, have to be replaced by high skilled, and more expensive staff. The advantage can be a reduction in time to process something, which will lead to a reduction in costs, or being able to better use the unique

knowledge base belonging to a business. As you start to specify what you want or desire, you hit up against technical language of requirements. Fear not, this is quite straightforward:

- Functional Requirements
- Nonfunctional Requirements

a) Functional Requirements of Smart ATM Card System:

- The ATM system should provide access to an authorized customer.
- The ATM system should accept customer requests and provide feedback.
- The ATM system should determine ATM responses to inputs received.
- The system should ask for account type.
- The system should check for printing receipt.
- The system should ask for amount to be withdrawn if it is withdrawal.
- Dispense cash and debit the amount if there is enough account balance.
- If there is not enough balance error message should be displayed.
- If it is balance enquiry correct account balance should be displayed.
- If it is mini statement request, a printed receipt should be given.
- b) User Interface Requirements Smart ATM Card system: The interface provided to the user should be a very user-friendly one and it should provide an optional interactive help for each of the service listed. The interface provided is a menu driven one and the following screens will be provided.
 - Login screen is provided in the beginning for entering the required ATM Card no, pin
 no. and account number.
 - An unsuccessful login leads to a reattempt (maximum three) screen for again entering
 the same information. The successful login leads to a screen displaying a list of
 supported languages from which a user can select any one.

- In case of administrator, a screen will be shown having options to reboot system, shut down system, block system, and disable any service.
- In case of reboot/ shut down, a screen is displayed to confirm the user's will to reboot
 and also allow the user to take any backup if needed.
- In case of blocking system, a screen is provided asking for the card no. By entering the card no of a particular user, system access can be blocked for him.
- Administrator is also provided with a screen that enables him to block any service provided to the user by entering the name of the service or by selecting it from the list displayed.
- After the login, a screen with a number of options is then shown to the user. It contains
 all the options along with their brief description to enable the user to understand their
 functioning and select the proper option.
- A screen will be provided for user to check his account balance.

5.2 System Planning:

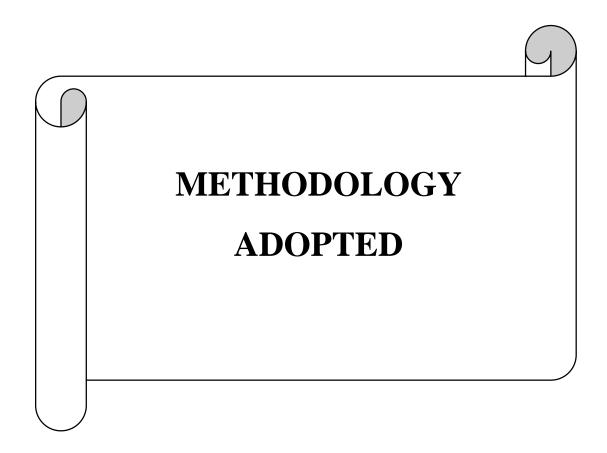
System planning should be affective so that the project begins with well-defined task.

Affective project planning helps to minimize the additional costs incurred while it is in progress.

For effective system planning, some principles are followed. These principles are listed below:

- **Planning is necessary:** Planning should be done before a project begins. For effective planning, objectives and schedules should be clear and understandable.
- **Risk Analysis:** Before starting the project, senior management and the project management team should consider the risks that may affect the system. For example, the user may desire changes in requirement while the project is in progress in such a case, the estimation of time and cost should done according to those requirements

- **Tracking of project plan:** Once the project plan is prepared, it should be tracked and modified accordingly.
- Meet quality standards and produce quality deliverables: The project plan should
 identify processes by which the project management team can ensure quality in
 software, based on the process selected for ensuring quality, the time and cost for the
 project is estimated.
- **Description of flexibility to accommodate changes:** The result of project planning is recorded in the form of a project plan, which should allow new changes to be accommodated when the project is in progress.



6. METHODOLOGY ADOPTED

A software development methodology is a framework that is used to structure, plan, and control the process of developing an information system, this includes the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weakness. Each of the available methodology frameworks are best suited to specific kinds of projects, based on various technical, organizational, project and team considerations. The methodology framework is often defined in some kind of formal documentation. To implement the project goals, the following methodologies need to be followed:

- Specifying the Application and various components of the Architecture.
- Specifying the bindings between the tasks and the resources either manually or by the design tools.
- Specifying the port interconnections between the resources.
- Analysis, extracting the data required for analysis and the doing the analysis.

6.1 Software Process Used:

6.1.1 Spiral Model:

The spiral model is similar to the incremental model, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral.

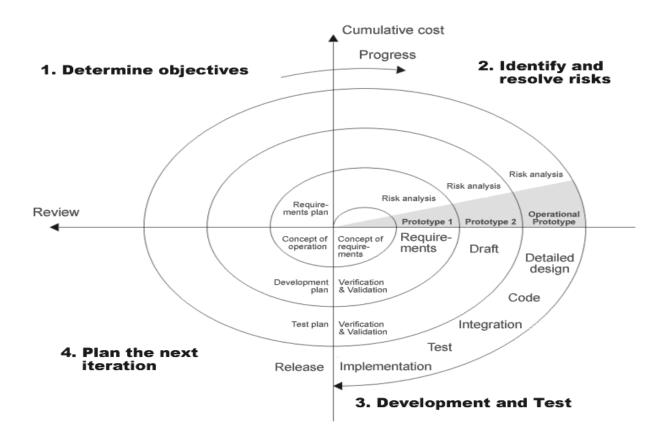


Fig 6.1 Spiral Model

- a) **Planning Phase:** Requirements are gathered during the planning phase. Requirements like 'BRS' that is 'Business Requirement Specifications' and 'SRS' that is 'System Requirement specifications'.
- b) **Risk Analysis:** In the **risk analysis phase**, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. If any risk is found during the risk analysis then alternate solutions are suggested and implemented.
- c) **Engineering Phase:** In this phase software is **developed**, along with <u>testing</u> at the end of the phase. Hence in this phase the development and testing is done.

d) **Evaluation phase:** This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

6.2 Software Description:

The Application of software is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description as functional representation, a representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

6.2.1 Java Language (Bank side):

6.2.1.1 The Java Programming Language:

Java is a high-level programming language which is Simple, Object-oriented Distributed, Interpreted, Robust, Secure, Architecture-neutral, Portable, High-performance, Multithreaded and Dynamic Java is also unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes**—the platform-independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how this works.

Java byte codes can be considered as the machine code instructions for the **Java Virtual Machine** (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM.

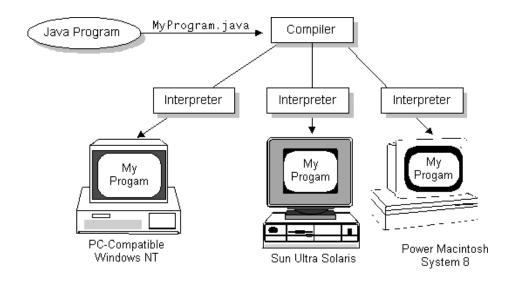


Fig6.2.1.1 Java compilation and interpretation Process

6.2.2 Client Side Scripting Language:

6.2.2.1 JavaScript:

JavaScript (JS) is a dynamic computer language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alert the document that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. The key design principles within JavaScript are taken from the self and Schema programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. JavaScript was formalized in the ECMAScript language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to computational objects within a host environment.

6.2.3 Web Technology HTML and CSS:

6.2.3.1 Hyper Text Markup Language:

The Hyper Text Markup language (HTML) is a simple markup language used to create hypertext documents that are portable from one platform to another HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. This specifications defines HTML version 4.0 HTML 4.0 aims to capture recommended practice as of early '96 and as such to be used as a replacement for HTML 3.2

- a) <HTML>...</HTML> All HTML files start and end with the tag pair.
- b) <HEAD>...</HEAD> All HTML have a pair of "HEAD" tags that indicate what the tile and other attributes of the page are going to be.
- c) <TITLE>...</TITLE> this tag indicates what the title of the HTML file
- d) <BODY>...</BODY> this tag pair is to logically separate the HTML file into the header and the body

6.2.4 Server side scripting language PHP:

PHP is an html-embedded scripting language. Much of its syntax is borrowed from c, java and Perl with a couple of unique PHP-specific features thrown in. the goal of the language is to allow web developers to write dynamically generated pages quickly." This is generally a good definition of PHP. However, it does contain a lot of terms you may not be used to. Another way to think of PHP is a powerful, behind the scenes scripting language that your visitors won't see! When someone visits your PHP webpage, your web server processes the PHP code. It then sees which parts it needs to show to visitors (content and pictures) and hides the other stuff(file

operations, math calculations, etc.) then translates your PHP into html. After the translation into html, it sends the webpage to your visitor's web browser.

6.2.4.1 PHP Syntax:

Before we talk about PHP's syntax, let us first define what syntax is referring to.

• Syntax - the rules that must be followed to write properly structured code.

PHP's syntax and semantics are similar to most other programming languages (c, java, perl) with the addition that all PHP code is contained with a tag, of sorts. all PHP code must be contained within the following...

6.2.4.3 PHP Variable:

A variable is a means of storing a value, such as text string "hello world!" or the integer value 4. A variable can then be reused throughout your code, instead of having to type out the actual value over and over again.

In PHP you define a variable with the following form:

• \$variable name = value;

If you forget that dollar sign at the beginning, it will not work. This is a common mistake for new PHP programmers.

6.2.5 Web Server Apache:

The Apache Server is the world's most widely used web server software. Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web,^[4] quickly overtaking NCSA HTTP as the dominant HTTP server, and has remained the

most popular HTTP server since April 1996. In 2009, it became the first web server software to serve more than 100 million websites.

Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Most commonly used on a Unix-like system (usually Linux), the software is available for a wide variety of operating systems, including Windows, OS X, Linux, Unix, FreeBSD, Solaris, NetWare, OS/2, TPF, OpenVMS and eComStation. Released under the Apache License, Apache is free and open-source software.

6.2.6 Database MySQL:

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in Memory but data fetching and writing would not be so fast and easy with those types of systems. So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

6.2.6.1 MySQL Database:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses.

MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company.

MySQL is becoming so popular because of many good reasons:

• MySQL is released under an open-source license. So you have nothing to pay to use it.

- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP,
 PERL, C, C++, JAVA, etc.
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.

SYSTEM IMPLEMENTATION AND DETAILS OF HARDWARE AND SOFTWARE USED

7. SYSTEM IMPLEMENTATION & DETAILS OF HARDWARE & SOFTWARE USED.

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users, which it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over an evolution, of change over methods. Apart of planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

An implementation co-ordination committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

7.1 Modules of Smart ATM Card System:

1. Admin Module: In the Admin Module the system allows the admin to login with username and password and there is a Menu Bar provided with menu items such as My Account in which there are menu items like Change Password where the admin can change his password whenever necessary, Log off and Exit and the second Menu is the Master which has a menu item Bank Master which allows the admin to add a Bank branch details such as Bank name, Bank code, address, IFSC code and provide the Password for the banks to login in the Bank

Module and once the password is given the admin cannot edit the password and it can only be done in the Bank Module by the particular bank to which the password has been provided and search the records by entering the IFCS code of that particular bank the admin can view the details of the bank and also perform updating and delete the records.

2. Bank Module: In the Admin Module the system allows the Bank Personnel to login with Bank Name, Bank Code and the password which has been provided to the bank by the Admin and there is a Menu Bar provided with menu items such as My Account in which there are menu items like Change Password where the Bank Personnel can change the password whenever necessary, Log off and Exit and the second Menu is the Master which has the Menu Items Customer Master and Transaction Master. In the Customer Master the Bank Personnel will add the customer details such as Name, Address, Mobile No, Email Id, Type of Account, Bank Name and Branch Code will be automatically added as the Bank Personnel Login with the Particular Bank and the PAN of the customer will be added, then Account number will be provided, ATM card number will be give and there are two PIN code are provided one is the ATM Pin which is for that particular ATM card and the second is the Bank Access pin which is given by the particular bank in which the user has the account, this is to access to that particular account to perform transaction. The Bank Personnel can also view the records by entering the account number of the customer and can also update and delete the particular customer records. The second menu item is the Transaction Master where the Particular Bank Personnel can view the transaction details of the customer such as balance amount by entering the account number and can credit the amount and save the details and also to approve or reject customer transaction request.

- **3. User Module:** The user module is the module where the ATM transactions in the ATM machine are done by the user which gives the user the facilities which are described below:
 - a) User Login: At the start, the user is provided with a log in screen and he is required to enter the ATM card number. And Account details which are then verified by the machine. In case of an unsuccessful attempt a user is asked again for his credentials but the maximum number of attempt given to the user is limited and if the ATM card no is correctly entered then the user has to enter the ATM pin.
 - b) Validating the User with ATM Pin: This module is used to secure our system from the unauthorized persons so it prompt the user to enter the pin no and after validating the pin number it allows the user to perform transaction. The module depicts the typical representation of the first module of proposed system where the user is checked for the authentication details, once the authentication completes successfully in this module it displays the list of banks else the "Invalid PIN" message is displayed.
 - c) Selection of Bank: This module is used to select a particular bank from the list of banks from which the user wishes to perform transaction. User can select the bank if and only if he/she enters the pin no correctly. The list of banks in which the user has the account is maintained by a centralized database and once the bank is selected the control is transferred to the particular bank server with account name and number. The bank administrator checks the account details in the database. This shows the next modular implementation where the user after authentication gets the list of banks from which he has to select a particular bank for transaction once the user selects the bank account then the account is checked for validation by the user data base.
 - d) Validating the User with Bank Access Pin: In This module, the user after selecting the particular bank in with he/she is interested to perform transaction has to enter the Bank access pin provided by that particular bank where the user is checked for the

authentication details, once the authentication completes successfully it allows the user to perform transaction and displays the transaction options such Amount withdraw, balance enquiry, mini statement, fast cash, change ATM pin, and change bank access pin else the "Invalid PIN" message is displayed.

e) Perform Transaction: From the selected bank it contacts the bank server for the account details in the particular database. Once it found enough balance to perform transaction, it starts the transaction process and all the transaction details are stored in database. Every transaction the user performs is automatically stored with date and time of access in individual database. So if there is any mischievous behavior in the network it can easily be detected. The database is periodically saved in order to prevent the loss of transaction. Each bank has its own database and hence all the data are stored in their own database which avoids collision of data. All the databases are connected to a centralized database which contains user and list of banks he/she has the account. The module shows the last stage of the implementation where when the user starts his transaction for withdrawing the amount, first user has to select withdraw cash then enters the amount and then the bank server checks the database for availability of balance if the cash is available then the amount is withdrawn else the message "insufficient balance" is displayed If any of the above validation/sequencing flow does not hold true, appropriate error messages will be prompted to the user for doing the needful.

7.2 System Architecture:

The idea behind the Smart AMT Card system is that the customers can use a single ATM card to operate different bank accounts instead of having individual card for each bank account and maintaining their PIN"s, carrying the cards safely which is a tedious process at

present scenario. The technology behind the product of the service is that adding all the user bank accounts to a universal ATM card. In this the user swipes his/her smart card in the ATM machine, then it request for authentication in the server side using PIN. After the user is authenticated successfully, then it displays the list of all banks that the user is having account. Now the user can select the bank from which he/she is willing to perform transaction. After selecting the bank the request is sent to the corresponding bank through a network and links it with the banks server for accessing the database of the user or customer so that the transaction is processed.

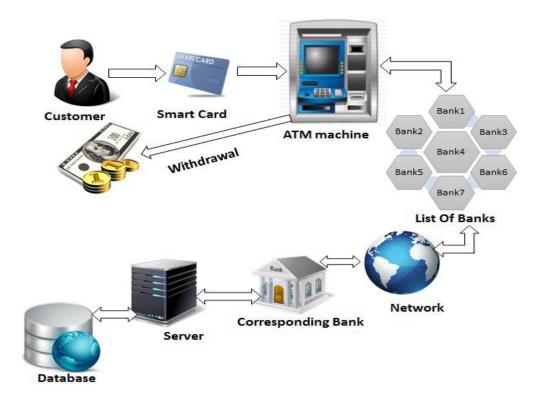


Fig 7.2 System Architecture showing Internal Connections

7.3 Hardware Requirements:

PROCESSOR : Dual Core or above

RAM : 512 MB Min

HARD DISK : 5 GB min

7.4 Software Requirements:

OPERATING SYSTEM : Windows XP or higher

LANGUAGE : JAVA (Bank side)

SERVER SIDE SCRIPTING : PHP

CLIENT SIDE SCRIPTING : JavaScript

WEB TECHNOLOGY : HTML, CSS

WEB SERVER : Apache

BACK END : MySQL

SYSTEM MAINTENANCE & EVALUATION

8. SYSTEM MAINTENANCE & EVALUATION

8.1 System Maintenance:

The results obtained from the evaluation process help the organization to determine whether its information systems are effective and efficient or otherwise. The process of monitoring evaluating and modifying of existing information systems to make required or desirable improvements may be termed as System Maintenance.

System Maintenance is an ongoing activity, which covers a wide variety of activities, including removing program and design errors, updating documentation and test data and updating user support. For the purpose of convenience, maintenance may be categorized into three classes, namely: Corrective Maintenance, Adaptive Maintenance, and Perfective Maintenance.

- **a) Corrective Maintenance:** This type of maintenance implies removing errors in a program, which might have crept in the system due to faulty design or wrong assumptions. Thus, in corrective maintenance, processing or performance failures are repaired.
- **b) Adaptive Maintenance:** In adaptive maintenance program functions are changed to enable the information system to satisfy the information needs of the user. This type of maintenance may become necessary because of organizational changes which may include:
 - Change in the organizational procedures.
 - Change in organizational objectives, goals, policies etc.
 - Change in forms.
 - Change in information needs of managers.
 - Change in system controls and security needs etc.

SMART ATM CARD SYSTEM

- c) Perfective Maintenance: Perfective Maintenance means adding new programs or modifying the existing programs to enhance the performance of the information system. This type of maintenance undertaken to respond to user's additional needs which may be due to the changes within or outside of the organization. Outside changes are primarily environmental changes, which may in the absence of system maintenance, render the information system in effective and inefficient. These environmental changes include:
 - Changes in governmental policies, laws, etc.
 - Economic and competitive conditions.
 - New technology.

8.1.1 Importance of Software Maintenance:

- The key software maintenance issues are both managerial and technical. Key management issues are: alignment with customer priorities, staffing, which organization does maintenance, estimating costs.
- Key technical issues are: limited understanding, impact analysis, testing, maintainability measurement.
- Any work done to change the software after it is in operation is considered to be maintenanc work. The purpose is to preserve the value of software over the time. The value can be enhanced by expanding the customer base, meeting additional requirements, becoming easier to use. Maintenance may span for 20 years, whereas development may be 1-2 years.

8.1.2 Software Maintenance Process:

- The implementation process contains software preparation and transition activities, such as the conception and creation of the maintenance plan; the preparation for handling problems identified during development; and the follow-up on product configuration management.
- The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it and check its validity, investigate it and propose a solution, document the request and the solution proposal, and finally, obtain all the required authorizations to apply the modifications.
- The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.
- Finally, the last maintenance process, also an event which doesn't occur on a daily basis, is the retirement of a piece of software.

8.2 Evaluation:

8.2.1 Hardware Evaluation Factors:

When we evaluate computer hardware, we should first investigate specific physical and performance characters for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These hardware evaluation factors are: Performance, Cost, Reliability, Availability, Compatibility, Modularity, Technology, Connectivity, Environmental requirements, Software. There is much more to evaluating hardware then determining the fastest and cheapest computing device. For example the question

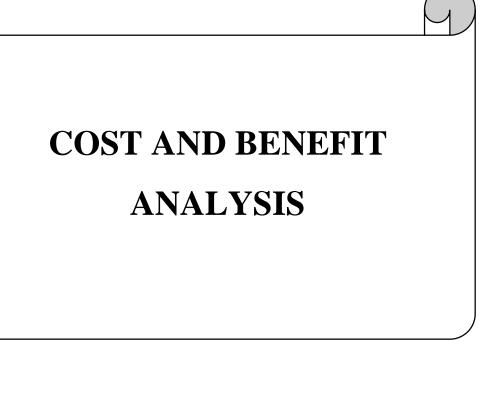
SMART ATM CARD SYSTEM

of possible obsolescence much be addressed by making a technology evaluation. The factor of ergonomics is also very important.

8.2.1 Software Evaluation Factors:

Software can be evaluated according to many factors similar to hardware evaluation. Thus the factors of Performance, Cost, Reliability, Availability, Compatibility, Modularity, Technology, Ergonomics, Connectivity, Environmental requirements, Software, Support should be used to evaluate proposed software acquisitions. In addition, however, the software evaluation factors are summarized below. For example some software packages required too much memory capacity and are notoriously slow, hard to use are poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

- a) Efficiency: It is the software a well-written system of computer instructions that does not use much memory capacity time?
- b) Flexibility: Can it handle its processing assignments easily without major modifications
- c) Security: Does it provide control procedures for error, malfunctions and improper use?
- **d) Language**: Do our computer programmers and users write it in a programing language that is used?
- e) Documentation: Is the software well documented? Does it include helpful user instructions?
- f) Hardware: Does existing hardware have the pictures required to best use this software?



9. COST AND BENEFIT ANALYSIS

9.1 Cost and benefit analysis:

Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial. Cost benefit analysis helps to give management a picture of the cost, benefits and risks. Cost benefit determines the benefits and saving that are expected the system and compares them with the expected costs. In performing cost and benefit analysis it is important to identify cost and benefit factors.

Cost and benefits can be categorized into the following categories:

- **Development Costs:** Development costs is the costs that are incurred during the development of the system. It is one time investment.
- Operating Costs: Operating costs are the expenses required for the day to day running of the system. Examples of operating costs are Wages, Supplies and Overheads.
- **Hardware/Software Costs:** It includes the cost of purchasing or leasing of computers and its peripherals. Software costs involve required S/W costs.
- **Personnel Costs:** It is the money spent on the people involved in the development of the system.

We can define benefits as Profit or Benefit = Income - Cost

Benefits can be accrued by: Increasing Income, Decreasing costs and both.

- a) **Technical Feasibility:** Technical Feasibility includes existing and new H/W and S/W requirements that are required to operate the project using JSP. The basic S/W requirement is J2EE in which the front end of the online hospital management project has been done. The basic entry forms are developed in JSP and the data stored in the MY SQL.
- **b) Operational Feasibility:** Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits.
- c) Legal Feasibility: A determination of any encroachment, violation, or liability that could result from development of the system. Legal feasibility tells that the software used in the project should be original purchased from the legal authorities and they have the license to use it.

PROCESS PHASE	COST INVOLVED
Concept Requirement	Rs. 1500/-
Requirement gathering and Designing	Rs. 1500/-
Implementation	Rs.2000/-
Designing and Coding	Rs. 2000/-
Testing and Maintenance	Rs. 1000/-
Documentation	Rs. 2000/-
Grand Total	Rs. 10000/-

Table 9.1 The detail showing the cost required to develop this project

DETAILED LIFE CYCLE OF THE PROJECT

10. DETAILED LIFE CYCLE OF THE PROJECT

10.1 Life Cycle:

Life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub processes required for developing a system. Development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the system analysis and design terminology, the system development life cycle means software development life cycle. The different phases of software development cycle: System study, Feasibility study, System analysis, System design, Coding, Database creation, Database connectivity, Testing, Implementation, Maintenance.

10.2 Entity Relationship Diagram:

The relation upon the system is structure through a conceptual ER-Diagram, which not only specifies the existential 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 descriptions. The primary purpose of the ERD is to represent data objects and their relationships.

The set of primary components that are identified by the ERD are:

- Data object
- Relationships
- Attributes
- Various types of indicators

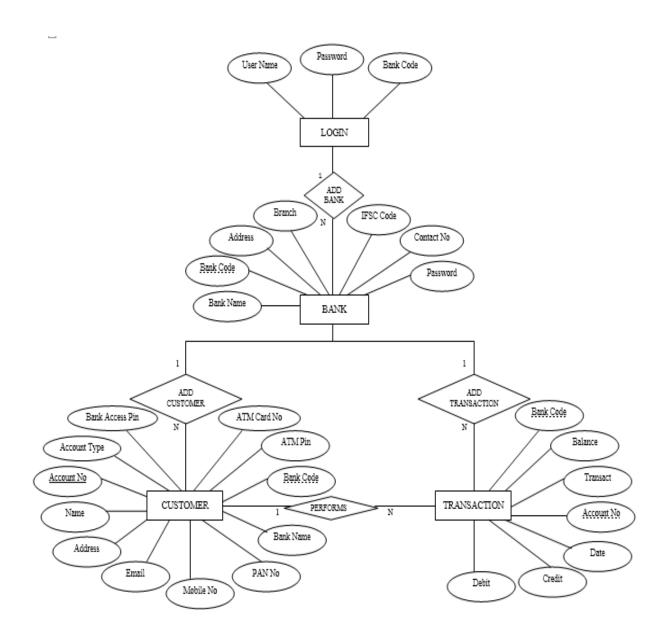


Fig 10.2 The Entity Relationship Diagram.

10.3 Data Flow Diagram:

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. In the DFD, there are four symbols

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

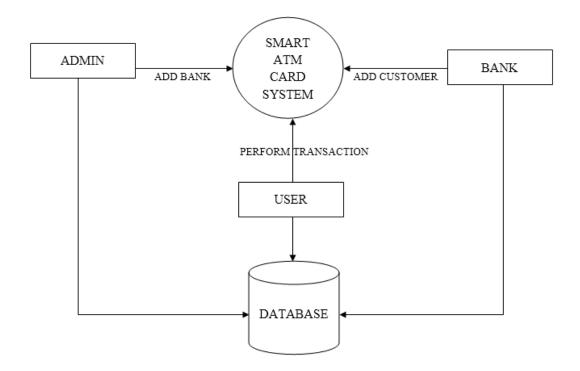


Fig 10.3 Data Flow Diagram Level-0

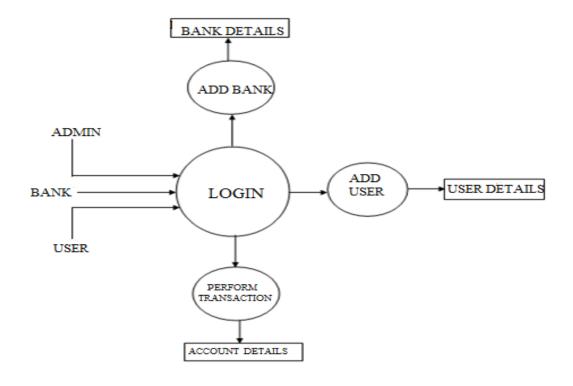


Fig 10.3 Data Flow Diagram Level-1

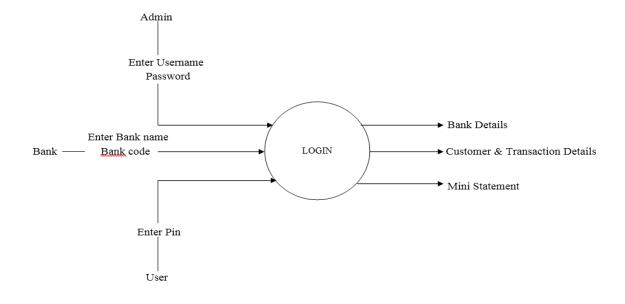


Fig 10.3 Data Flow Diagram Level-2

10.4 Use Case Diagram:

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify and organize system requirements. The use cases, which the specific roles are played by the actors within and around the system. These are usually referred as behavior diagrams used to describe a set of actions (use cases) that some systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

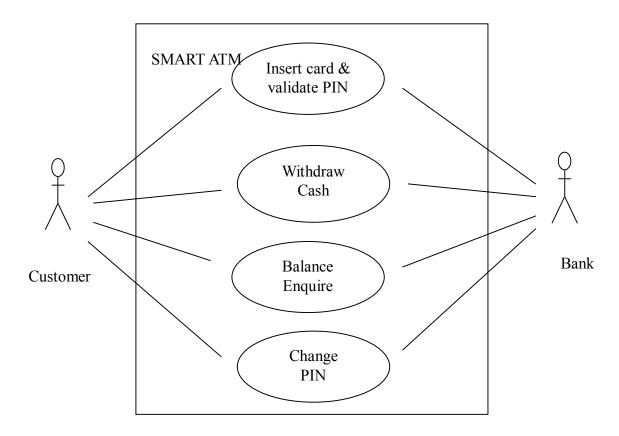
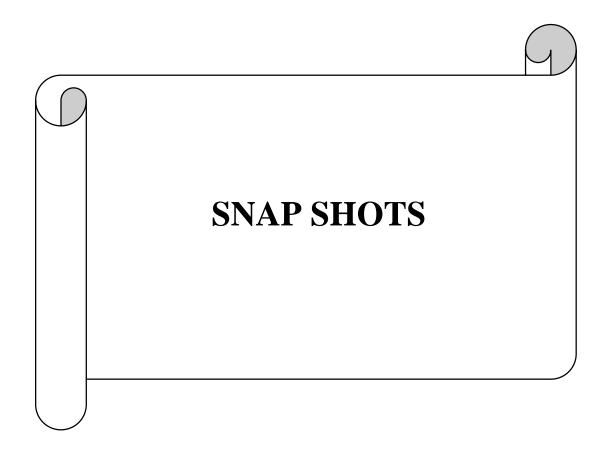
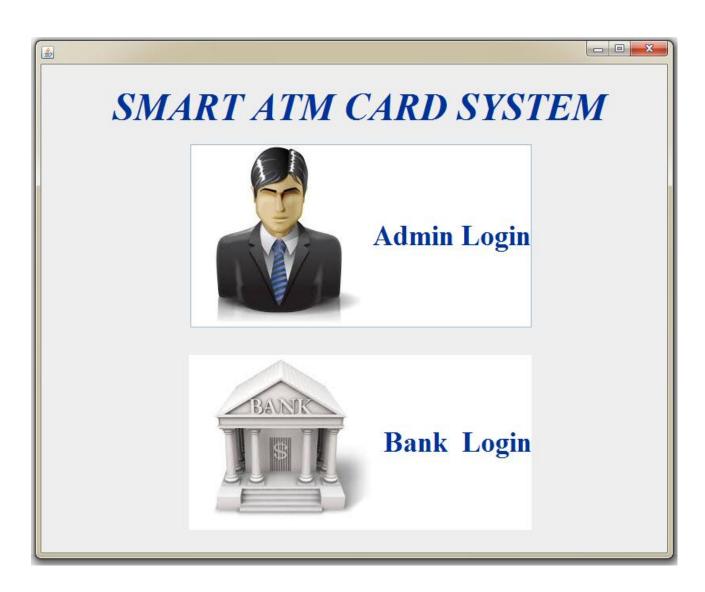


Fig 10.4 Use Case Diagram



11. SNAP SHOTS

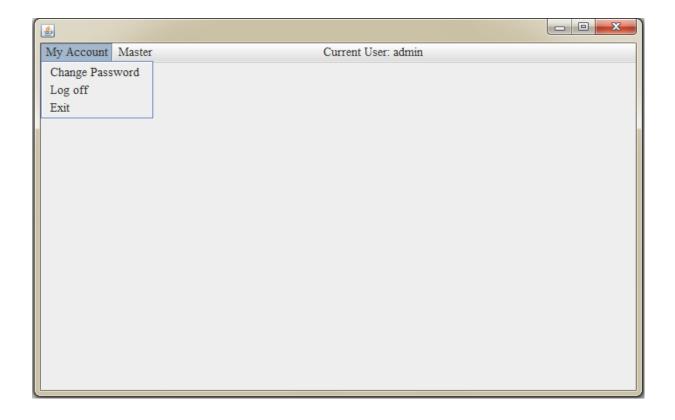
Smart ATM Card System Main Form:



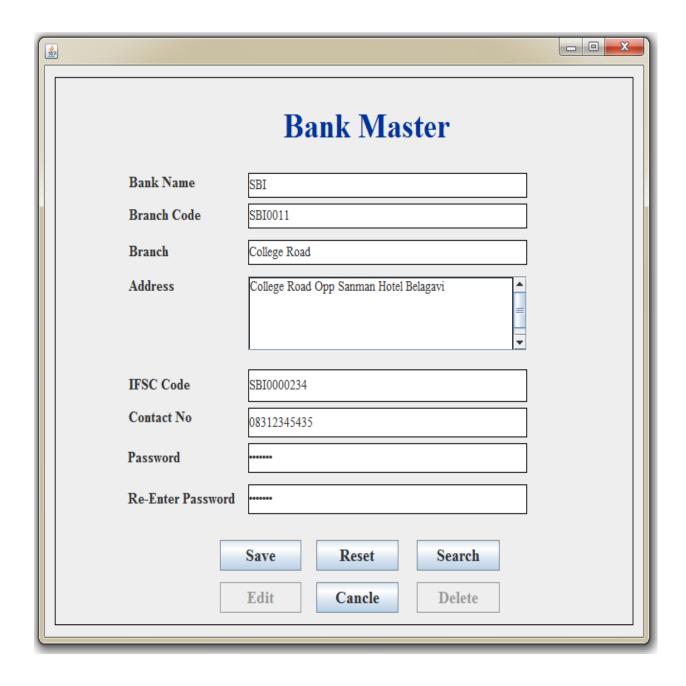
Admin Login Form:



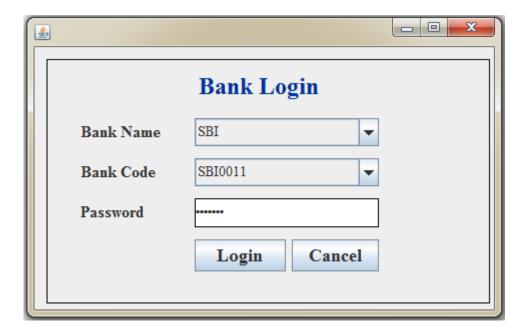
Admin Menu Form:



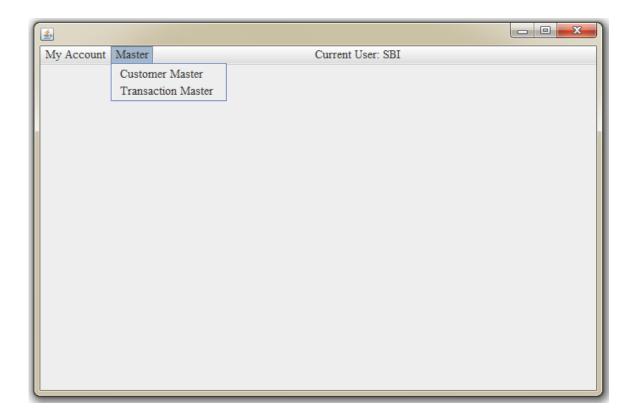
Bank Master Form:



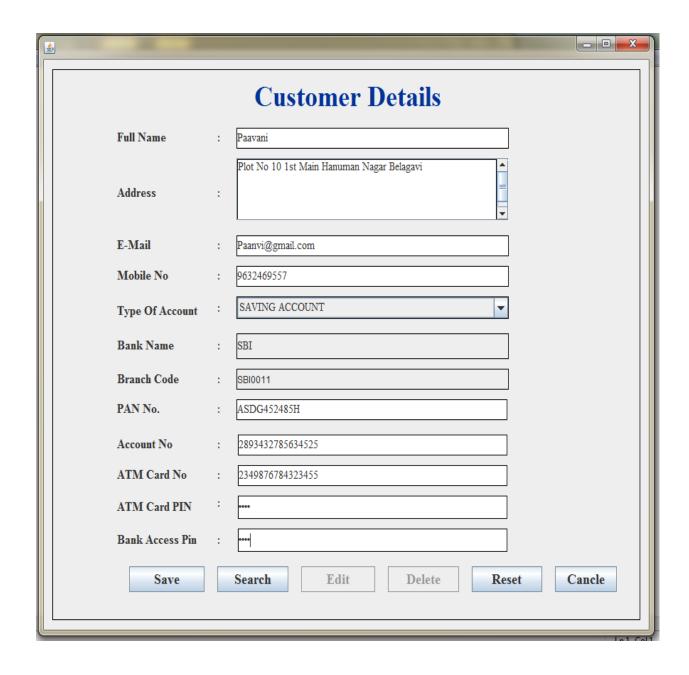
Bank Login Form:



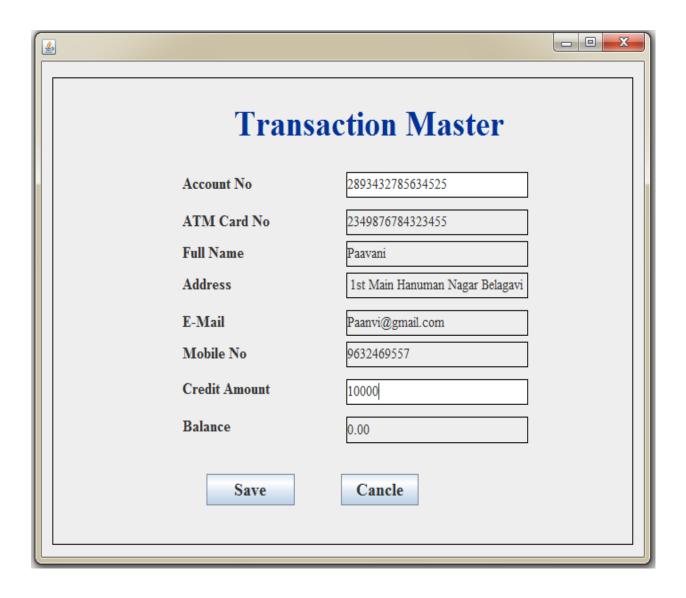
Bank Menu Form:



Customer Details Form:



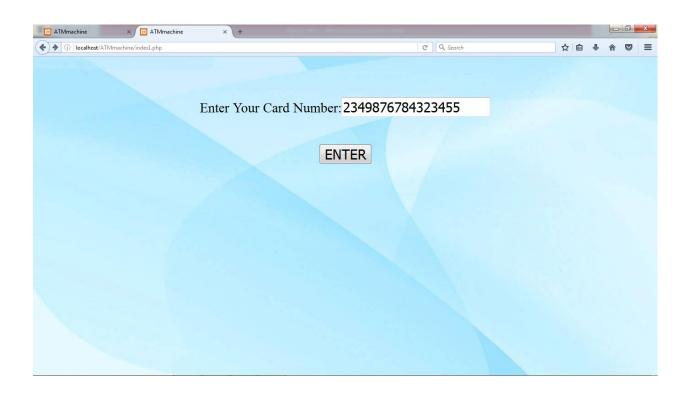
Transaction Master Form:



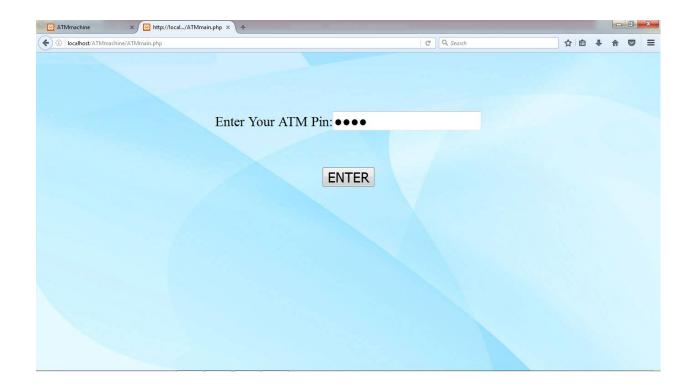
Smart ATM Card System Home Page:



Page For Inserting ATM Card Number:



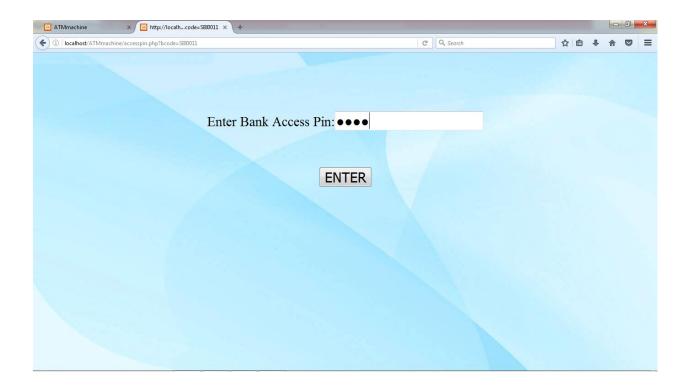
Page For Inserting ATM Pin:



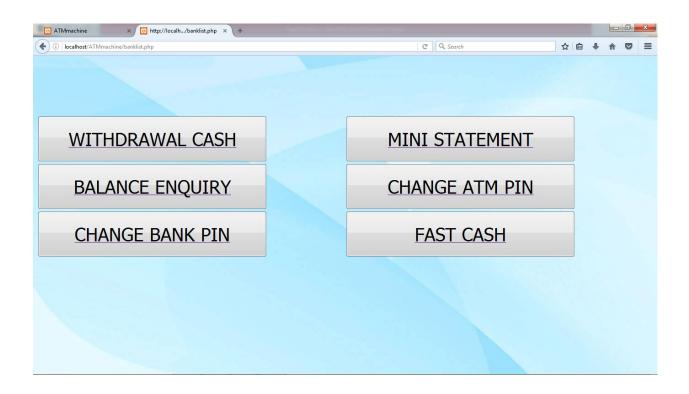
Bank List Page:



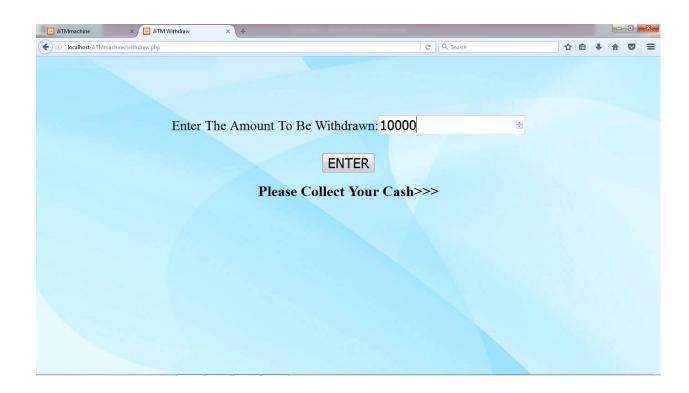
Page For Inserting Bank Access PIN:



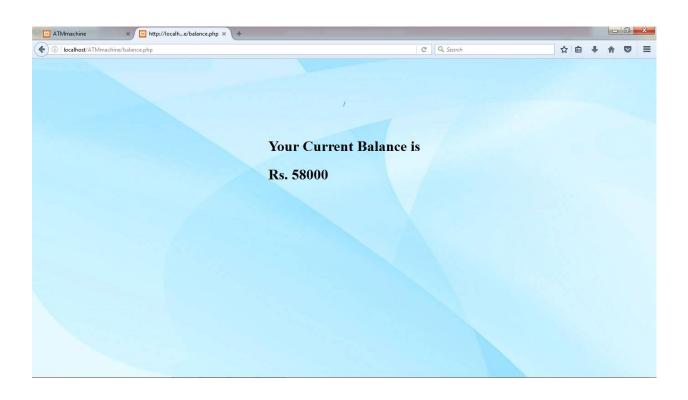
Transaction Facilitation Page:



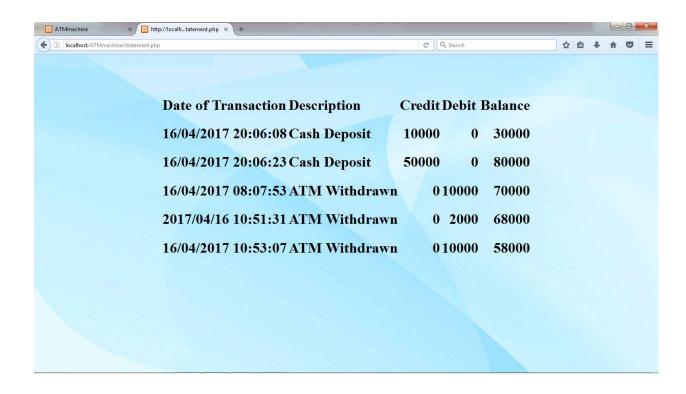
Amount Withdrawn Page:



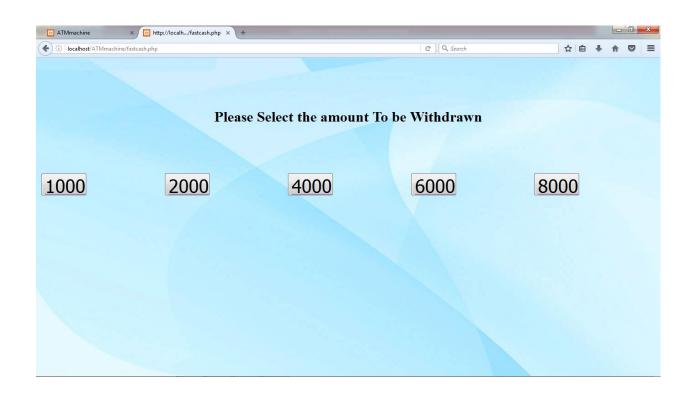
Balance Enquire Page:



Mini Statement Page:



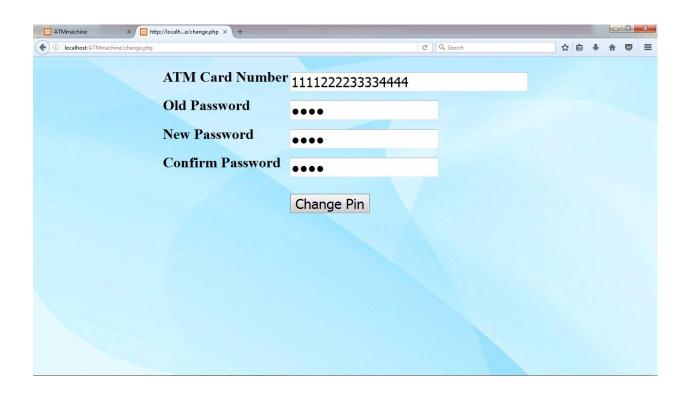
Fast Cash Page:

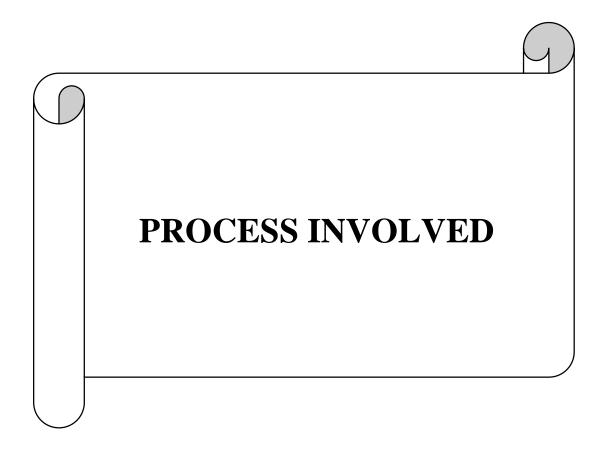


Change Bank Access Pin Page:



Change ATM PIN Page:





12. PROCESS INVOLVED

Project process is also referred to as the development and normal life period of the project. It is the actual management of a project in practical terms. This is the level when the projects inputs are transformed into outputs via the project activities following the laid down work plan. This leads to the attainment of immediate objectives.

It involves the coordinating, monitoring and control of the performance of the various project groups and the use of project resources in such a way that the project activities are completed in an orderly and optimal fashion within the constraint of time and resources available.

12.1 Approaches To Project Implementation:

12.1.1 Top-Down Approach:

The top-down methodology has been recently developed to produce provably perform ant designs relative to what is achieved in classical centralized control theory. Ideally the designer should start from the definition of an objective that involves global quantities, then devise a centralized optimization algorithm and finally proceed to the synthesis of the decentralized (agent-based) solution. The design process consists of three steps: modeling, synthesis and analysis/optimization.

SMART ATM CARD SYSTEM

- a) Modeling: in this phase the designer identifies and categorizes system's agents according to the following taxonomy derived from classical Control Theory. Modelling agents collect data from many information agents and update internal estimates of the "real world" state. Planning agents use the current world state estimates, the viable action or control options and the current goals to plan new actions to carry out. These agents may need to ask **brokering** agents to report on available resources such as additional state and action information.
- **b) Synthesis:** Agent controllers are designed following the lines of a three-stage top-down process:
 - At first, it is assumed that each agent can access remote resources local to other agents
 instantly and with infinite precision. So a first centralized solution aimed at optimizing
 a global objective is designed;
 - Next, limitations of the distributed environment are applied and so the visibility of each
 agent gradually reduced. Consequently inter-agent communication issues arise for now
 each agent needs to replace global resources with local resources with local resources.
 The result is a fully decentralized solution;
 - Finally, the obtained solution must be calibrated via parameter tuning.
 - d) Analysis/Optimization: The inter-agent communication must be optimized in order for the distributed system to perform as predicted at the beginning of the synthesis phase. The Analysis conducted in this phase may lead to review (feedback) of the original Modeling of the agent system thus creating a cycle.

12.1.2 Bottom-Up Approach:

The bottom-up design methodology is very popular for producing autonomous, scalable and adaptable systems often requiring minimal communication. The design process consists of three steps: Synthesis, Modeling and Analysis, and Optimization.

- a) Synthesis: In the Synthesis phase one has to define the agent the controller which can be described by any automation that is the behavioral representation of an agent. In the case of a reactive agent the controller can be characterized by a finite state automaton (FSA). Each state of the automaton represents the action or a behavior the agent is executing, with transitions coupling it to other states. Consequently, the behavioral dynamics of a reactive agent can be considered as an ordinary Markov process.
- **b) Modeling and Analysis:** Once a controllers for individual agents have been constructed, one need to develop a mathematical model of the collective behavior. Remarkably, the finite automaton of a single agent in many cases can be used for adequately describing the macroscopic or collective behavior of a large-scale system composed of many such controllers. The model consists of coupled differential equations describing how the average group behavior changes in time. This modelling approach is based on the theory of stochastic process.
- c) Optimization: Mathematical model can be used not only to validate the controller, but also to estimate individual parameters that optimize group-level performance. Using mathematical analysis one can finally answer a number of design questions.

METHODOLOGY USED FOR TESTING

13. METHODOLOGY USED FOR TESTING

13.1 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 that 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 in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

13.2 Testing principles:

Before applying methods to design effective test cases, a software engineer must understand the basic principle that guide software testing:

- All tests should be traceable to customer requirements
- Tests should be planned long before testing begins

SMART ATM CARD SYSTEM

- 80 percent of all errors uncovered during testing will likely be traceable to 20 percent
 of all program components. The problem, of course, is to isolate these suspect
 components and to thoroughly test them.
- Testing should being "in the small" and progress toward testing "in the large".
- Exhaustive testing is not possible
- To be most effective an independent third party should conduct testing

13.3 White box testing:

White-box testing of software is predicated on close examination of procedural detail. Providing test cases that exercise specific sets of conditions and/or loops tests logical paths through the software. The "status of the program" may be examined at various points to determine if the expected or asserted status corresponds to the actual status. Using white-box testing methods, the software engineer can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once,
- Exercise all logical decisions on their true and false sides,
- Execute all loops at their boundaries and within their operational bounds,
- Exercise internal data structures to ensure their validity.

The basis path method enables the test case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basis set of execution paths. Test cases derived to exercise the basis set are guaranteed to execute every statement in the program at least one time during testing.

13.4 Black box testing:

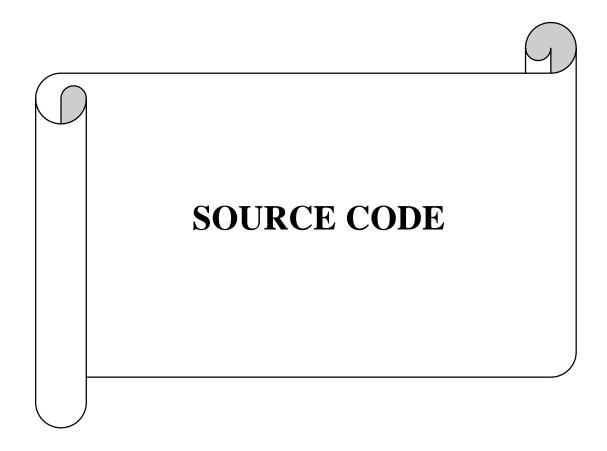
Black box called behavioral testing, focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of error than white-box methods. When computer software is considered, black box testing alludes to tests that are conducted at the software interface. Although they are designed to uncover errors, black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced and that the integrity of external information is maintained. A black-box test examines some fundamental aspect of a system with a little regard for the internal logical structure of the software. Black-box testing attempts to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external database access.
- Behavior or performance errors.
- Initialization and termination errors. By applying back-box techniques.
- a) Functional Testing: This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

SMART ATM CARD SYSTEM

- b) Unit Testing: This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team. The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.
- c) Integration Testing: Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing. In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations.
- d) System Testing: System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards. This type of testing is performed by a specialized testing team. System testing is important because it is the first step in the Software Development Life Cycle, where the application is tested as a whole to verify that it meets the functional and technical specifications.
- e) Acceptance Testing: This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirement. The QA team will have a set of prewritten scenarios and test cases that will be used to test the application. More ideas will be shared about the application and more tests can be performed on it to gauge its accuracy and the reasons why the project was initiated.

- f) Performance Testing: It is mostly used to identify any bottlenecks or performance issues rather than finding bugs in a software. There are different causes that contribute in lowering the performance of a software: Network delay, Client-side processing, Database transaction, processing, Load balancing between servers, Data rendering.
- g) Load Testing: It is a process of testing the behavior of a software by applying maximum load in terms of software accessing and manipulating large input data. It can be done at both normal and peak load conditions. This type of testing identifies the maximum capacity of software and its behavior at peak time.



14. SOURCE CODE

JAVA CODE:

FirstLogin.java

```
package imfromatm;
public class FirstLogin extends javax.swing.JFrame {
public FirstLogin()
initComponents();
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
Login1 l=new Login1();
l.setVisible(true);
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
BANKLOGIN B=new BANKLOGIN();
B.setVisible(true);
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run()
new FirstLogin().setVisible(true);
});
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton4;
private javax.swing.JColorChooser jColorChooser1;
private javax.swing.JLabel jLabel1; }
```

Login.java

```
package imfromatm;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
public class Login1 extends javax.swing.JFrame {
Statement st;
ResultSet rs;
DBConnection db=new DBConnection();
String bank;
public Login1()
initComponents();
private void formWindowOpened(java.awt.event.WindowEvent evt)
this.setBounds(480, 100, 475, 280);
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
this.setVisible(false);
FirstLogin f=new FirstLogin();
f.setVisible(true);
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String a="", b="", m="", n="", c="";
a=txtuser.getText().trim();
b=txtpwd.getText().trim();
if(a.equals("") && b.equals("")) {
JOptionPane.showMessageDialog(rootPane, "UserName and Password fields cannot
be empty", "Login", 0);
txtuser.requestFocus(true);
return;
st=db.DBC();
rs=st.executeQuery("select * from login where Username='"+a+"'
                                                                          and
password='"+b+"'");
if(rs.next())
```

```
m=rs.getString("username");
n=rs.getString("password");
c=rs.getString("bcode");}
if(a.equals(m) && b.equals(n))
main af=new main(a,c);
af.setVisible(true);
dispose();
return;
} else {
JOptionPane.showMessageDialog(rootPane, "Invalid User
                                                              Name/Password",
"Login", 0);
txtuser.setText("");
txtpwd.setText("");
txtuser.requestFocus(true);
return;
}catch (SQLException ex) {
Logger.getLogger(Login1.class.getName()).log(Level.SEVERE, null, ex);
} }
private void txtpwdKeyPressed(java.awt.event.KeyEvent evt) {
if(evt.getKeyCode()==10)
{
jButton2ActionPerformed(null);
} }
public static void main(String args[]) {
ava.awt.EventQueue.invokeLater(new Runnable() {
public void run()
new Login1().setVisible(true);
});}
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
private javax.swing.JPasswordField txtpwd;
private javax.swing.JTextField txtuser;
```

Main.java

```
package imfromatm;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
public class main extends javax.swing.JFrame {
String lgn;
String bcode;
Statement st;
ResultSet rs;
DBConnection db=new DBConnection();
public main(String x, String bc)
lqn=x;
bcode=bc;
initComponents();
private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
this.setVisible(false);
Login1 l=new Login1();
l.setVisible(true);
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
int s=JOptionPane.showInternalConfirmDialog(rootPane, "Do you really want to
quit the application", "Exit", 0, 3);
System.out.print(s);
if(s==0){
System.exit(0);
} }
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
if(lgn.equals("admin")){
BankMaster b=new BankMaster();
b.setVisible(true);}
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
ChangePassword ch=new ChangePassword(lgn,bcode);
ch.setVisible(true);}
private void formWindowActivated(java.awt.event.WindowEvent evt) {
try {
st=db.DBC();
```

```
rs=st.executeQuery("select username from login where username='"+ lgn +"'");
if(rs.next()){
user.setText( "Current User: "+rs.getString(1));
} catch (SQLException ex) {
Logger.getLogger(main.class.getName()).log(Level.SEVERE, null, ex);
} }
public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
public void run() {
new main("","").setVisible(true);
}
});
}
private javax.swing.JMenu jMenu1;
private javax.swing.JMenu jMenu2;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JMenuItem jMenuItem2;
private javax.swing.JMenuItem jMenuItem3;
private javax.swing.JMenuItem jMenuItem4;
private javax.swing.JMenu user;
}
BankLogin.java
package imfromatm;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
public class BANKLOGIN extends javax.swing.JFrame {
DBConnection db=new DBConnection();
Statement st;
ResultSet rs;
String bank;
public BANKLOGIN() {
initComponents();
private void txtbankpassKeyPressed(java.awt.event.KeyEvent evt) {
if(evt.getKeyCode() == 10) {
```

jButton3ActionPerformed(null);

```
} }
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
try {
String a="", b="", m="", n="", c="", o="";
a=cmbbank.getSelectedItem().toString();
b=cmbifsc.getSelectedItem().toString();
c=txtbankpass.getText();
if(a.equals("Select Bank"))
JOptionPane.showMessageDialog(rootPane, "Please Select Bank", "Login", 1);
cmbbank.requestFocus(true);
return;
} else if(b.equals("Select Bank Code")){
JOptionPane.showMessageDialog(rootPane,
                                             "Please
                                                            Select
                                                                        Bank
code", "Login", 1);
cmbifsc.requestFocus(true);
return;
} else if(c.equals("")){
JOptionPane.showMessageDialog(rootPane, "Please Enter Password", "Login",
txtbankpass.requestFocus(true);
return; }
st=db.DBC();
rs=st.executeQuery("select * from login where Username='"+a+"'
                                                                         and
password='"+c+"'");
if(rs.next()) {
m=rs.getString("username");
n=rs.getString("password");
o=rs.getString("bcode");
if(a.equals(m) && b.equals(o) && c.equals(n)){
main1 af=new main1(a,b);
af.setVisible(true);
dispose();
return;
} else {
                                                                   Name/Bank
JOptionPane.showMessageDialog(rootPane,
                                            "Invalid
                                                          Bank
Code/Password", "Login", 0);
return;
}} catch (SQLException ex) {
Logger.getLogger(Login1.class.getName()).log(Level.SEVERE, null, ex);
```

```
} }
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
this.setVisible(false);
FirstLogin f=new FirstLogin();
f.setVisible(true);
private void cmbbankItemStateChanged(java.awt.event.ItemEvent evt) {
try {
st=db.DBC();
bank=cmbbank.getSelectedItem().toString();
System.out.print(bank);
ResultSet r=st.executeQuery("Select bcode from tblbank where bankname=""+
bank +"'");
boolean b=r.next();
System.out.print(b);
if(b==true){
cmbifsc.removeAllItems();
cmbifsc.addItem("Select Bank Code");
ResultSet rs1=st.executeQuery("Select bcode from tblbank where bankname='"+
bank +"'");
while(rs1.next()){
cmbifsc.addItem(rs1.getString(1));
}
cmbifsc.requestFocus();
}else {
System.out.print("ouuuttt");
}} catch (SQLException ex) {
Logger.getLogger(Login1.class.getName()).log(Level.SEVERE, null, ex);
private void formWindowOpened(java.awt.event.WindowEvent evt) {
try {
this.setBounds(480, 100, 475, 300);
st=db.DBC();
rs =st.executeQuery("Select distinct(bankname) from tblbank");
cmbbank.addItem("Select Bank");
while(rs.next()){
cmbbank.addItem(rs.getString(1));
}} catch (Exception ex) {
System.out.print(ex);
}
```

```
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new BANKLOGIN().setVisible(true);
    }
    });
}

private javax.swing.JComboBox cmbbank;

private javax.swing.JComboBox cmbifsc;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JPanel jPanel2;

private javax.swing.JPasswordField txtbankpass;
}
```

PHP CODE

HomePage.PHP

```
<html>
<head>
<meta charset="UTF-8">
<title>ATMmachine</title>
</head>
<script lang="javascript">
function validate()
if(isNaN(document.f.num.value)){
alert('<h1>Enter Valid Digits</h1>');
return false;
}
return true;
}
</script>
<body style="background-image:url(Indian-Rupeenew.jpg)">
<form name="f" action="login.php" method="post" onsubmit="javascript:return</pre>
validate();">
<center><br/><br/><br/><br/></pr/></pr/>
```

```
<lable style="font-size:30px;">Enter Your Card Number:</lable><input</pre>
type="text" name="num" required="required" maxlength="16" style="font-
size:30px"/>
<br/><br/><br/><br/>
<input type="submit" value="ENTER" style="font-size:30px"/>
</center>
</form>
</body>
</html>
Card no
<html>
<head>
<meta charset="UTF-8">
<title>ATMmachine</title>
</head>
<body style="background-image:url(Indian-Rupeenew.jpg)">
<form name="f" action="index1.php" method="post">
<center>
<br/><br/><br/><br/><br/>
<label style="font-size:50px;color:darkblue">WELCOME TO SMART ATM CARD
SYSTEM</label>
<br/><br/><br/><br/><br/>
<input type="submit" value="Continue>>>" style="font-
size:30px;color:black;"/>
</center>
</form>
</body>
</html>
Atmpin.PHP
<?php
session start();
include 'dbconnection.php';
if($ SESSION['id']==null){
header('location:index.php');
else{
?>
<body style="background-image:url(Indian-Rupeenew.jpg)">
<center>
<form name="f" action="" method="post"><br/><br/><br/><br/><br/><br/>
```

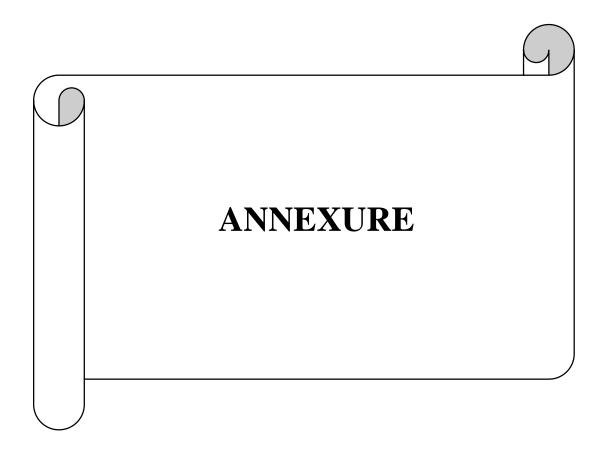
```
<lable style="font-size:30px;">Enter Your ATM Pin:</lable><input</pre>
type="password" name="atmpin" maxlength="4" required="required"
style="font-size:30px"/>
<br/><br/><br/><br/><br/>
<input type="submit" name="sub" value="ENTER" style="font-size:30px;"/>
</form>
<?php
include 'dbconnection.php';
if(isset($ REQUEST['sub']))
$p=$ POST['atmpin'];
$query="select * from tbluserdetails where atmpin='$p'";
$result=mysql query($query);
if(mysql num rows($result)>0)
header('location:bankmain.php');
}
else {
echo '<h1>Wrong ATM Pin</h1>';
}
}
?>
</center> </body> </html>
Accesspin.PHP
<html>
<body style="background-image:url(Indian-Rupeenew.jpg)">
<center>
<lable style="font-size:30px;">Enter Bank Access Pin:<input</pre>
type="password" name="bankpin" maxlength="4" required="required"
style="font-size:30px" /><br/><br/><br/><br/>
<input type="submit" name="sub" value="ENTER" style="font-size:30px"/>
<?php
session start();
$f=$ GET['bcode'];
$ SESSION['bcode']=$f;
include 'dbconnection.php';
if(isset($ REQUEST['sub']))
```

```
$a=$ POST['bankpin'];
$query="select * from tbluserdetails where accesspin='$a' and bcode='$f'";
$result=mysql query($query);
if(mysql num rows($result)>0){
$row=mysql fetch assoc($result);
$ac=$row['accno'];
$ SESSION['accno']=$ac;
header('location:banklist.php');
}else{
echo '<h1><br/>Wrong Bank Access Pin </h1>';
}
}
?>
</form> </center> </body> </html>
Banklist.PHP
<?php
session start();
2>
<body style="background-image:url(Indian-Rupeenew.jpg)">
<center>
<table style="width:100%; align-
content:center"><br/><br/><br/><br/><br/><br/><br/></pr/></pr/>
<a href="withdraw.php"><input type="button" name="wi"
value="WITHDRAWAL CASH" style="align-content:flex-start; width:500px;
height:100px; font-size:40px"/></a>
<a href="statement.php"><input type="button" name="st" value="MINI
STATEMENT" style="align-content:flex-end; width:500px; height:100px; font-
size:40px"/></a>
<a href="balance.php"><input type="button" name="bal"
value="BALANCE ENQUIRY"
                        style="align-content:flex-start; width:500px;
height:100px; font-size:40px"/></a>
<a href="change.php"><input type="button" name="chngpass" value="CHANGE
ATM PIN" style="align-content:flex-end; width:500px; height:100px; font-
size:40px"/></a>
<a href="changebpin.php"><input type="button" name="bal"
value="CHANGE BANK PIN" style="align-content:flex-start; width:500px;
height:100px; font-size:40px"/></a>
<a href="fastcash.php"><input type="button" name="bal" value="FAST
CASH" style="align-content:flex-start; width:500px; height:100px; font-
size:40px"/></a>  </center> </body>
```

Withdrawl.PHP

```
<html>
<head><title>ATM Withdraw</title></head>
<script lang="javascript">
function validate()
if(isNaN(document.f.cash.value)){
alert('Cash Should be in the Form of Digits');
return false;
if((document.f.cash.value)<100) {</pre>
alert('Please Enter Minimum 100 Rs');
return false;
return true;
</script>
<body style="background-image:url(Indian-Rupeenew.jpg)">
<center>
<form name='f' action='' method='post' onsubmit="javascript:return</pre>
validate();"><br/><br/><br/><br/><br/>
<lable style="font-size:30px;">Enter The Amount To Be
Withdrawn:</lable><input type="number" name="cash" required="required"
style="font-size:30px"/>
<br/><br/><br/>
<input type="submit" name="sub" value="ENTER" style="font-size:30px"/>
<?php
date default timezone set('Asia/Kolkata');
session start();
include 'dbconnection.php';
r=1;
if(isset($ REQUEST['sub']))
$w=$ POST['cash'];
$query2="select MAX(slno)as no from tbltransaction where
accno='{$ SESSION['accno']}' and bcode='{$ SESSION['bcode']}'";
$result2=mysql query($query2) or die(mysql error());
if(mysql num rows($result2)>0){
$r1=mysql fetch assoc($result2);
```

```
$query="select bal from tbltransaction where slno={$r1['no']} and
accno={$ SESSION['accno']} ";
$result= mysql_query($query);
if($result)
$row=mysql_fetch_assoc($result);
$a=$row['bal'];
if($a<$w)
echo 'Insuffient balance';
}
else {
$total=$a-$w;
$dt=Date('d/m/Y h:i:s');
$query2="select MAX(slno)as no from tbltransaction where
accno='{$ SESSION['accno']}' and bcode='{$ SESSION['bcode']}'";
$result2=mysql query($query2) or die(mysql error());
$r1=mysql fetch assoc($result2);
if($r1['no']!=null){
$r=$r1['no']+1;
$query1="insert into tbltransaction
values('$r','{$ SESSION['bcode']}','{$ SESSION['accno']}','$dt','ATM
Withdrawn','0','$w','$total')";
$result1= mysql query($query1) or die(mysql error());
if($result1>0)
{
echo '<h1>Please Collect Your Cash>>></h1>';
else {
echo '<h1>Your Account Can not Update at This Time!!!Please Try after
sometime</h1>';
} }
else
{
echo '<h1>You Cant Withdraw the Amount</h1>';
} }
}
?>
</form>
</re>
```



15. ANNEXURE

15.1 Brief Background Of The Organization

15.1.1 Company Name:

ZEEL Network Solution/CODE Lab

15.1.2 Company Profile:

General Company Information:

- ZEEL Network Solution/CODE Lab
- #2842, 2nd Floor, 204, Ashirwad Complex, Belagavi
- Karnataka India
- Mobile Number 9880917783
- Email: bhushandongre@zeelnet.com
- Manager: Mr. Bhushan Dongre

15.2 Data Dictionary

A Data Dictionary is a catalogue- a repository of element in a system. As the name suggest, these elements center around data and the way these are structured to meet the user requirement and system needs. In Data Dictionary we find list of all the elements are data flows, data stores, and process. A Data Dictionary contains a list of all files in the database, the number of records in each file, and the names and types of each field. Most database management system keep the data dictionary hidden form users to prevent them from accidentally destroying its contents. Data Dictionaries contains:

- The definitions of all schema objects in the database(tables, indexes, clusters, synonyms, sequences, procedures, functions, packages, triggers, and so on)
- How much space has been allocated for, and is currently used by, the schema objects
- Default values for columns
- Integrity constraints information
- Other general database information

The data dictionary is structured in tables and views, just like other database data. All the data dictionary tables and views for a given database are stored in the database's system table space.

15.2.1 Important of data dictionary:

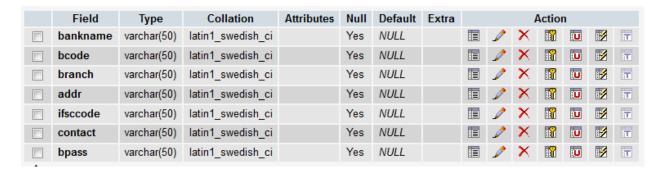
- To manage the detail in large system.
- To communicate a common meaning for all system elements.
- To Document the feature of the system.
- To Facilitates analysis of the details in order to evaluate characteristics and determine where system changes should be made.

The following tables are used in the system:

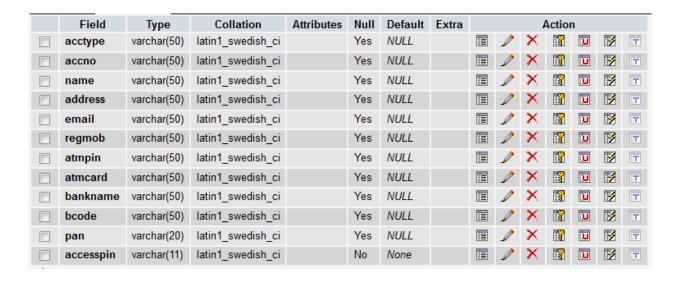
1. Login Table

Field	Туре	Collation	Attributes	Null	Default	Extra	Action						
username	varchar(50)	latin1_swedish_ci		Yes	NULL			1	×		Ü	7	T
password	varchar(50)	latin1_swedish_ci		Yes	NULL			1	X		Ü	1	T
bcode	varchar(50)	latin1_swedish_ci		Yes	NULL			1	×		U	3	T

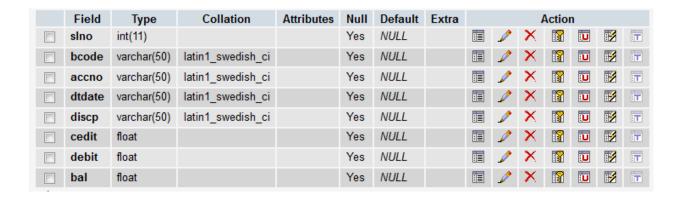
2. Bank Details Table



3. Customer Details Table



4. Transaction Details Table



15.3 List of Abbreviations, Figures and Tables:

15.3.4Abbreviations:

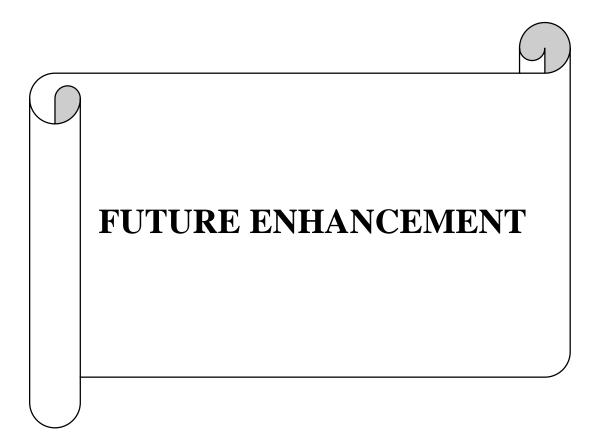
- ATM Automated Teller Machine
- PIN Personal Identification Number
- PAN Permanent Account Number
- PHP Hypertext Preprocessor
- SQL Structured Query Language
- HTML Hypertext Markup Language
- CSS Cascading Style Sheet
- RDMS Relational Database Management System
- SDLC Software Development Life Cycle
- DFD Data flow Diagram
- ERD Entity Relationship Diagram
- SQL Structured Query Language
- GPL General Public License

15.3.5 Figures:

- 4.2 Design Phase
- 6.1.1 Spiral Model
- 10.2 Entity Relationship Diagram
- 10.3 Data Flow Diagram
- 10.3 Data Flow Diagram Level-0
- 10.3 Data Flow Diagram Level-1
- 10.4 Use Case Diagram
- 6.2.1.1 Java compilation and interpretation Process
- 7.2 System Architecture showing Internal Connections

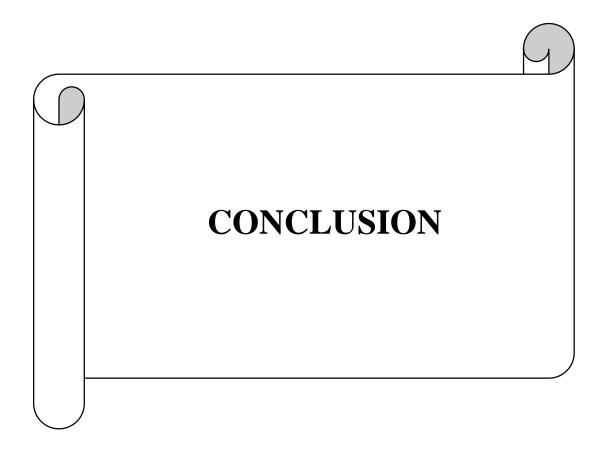
15.3.6 Tables:

• **Table** 9.1 – Details showing the cost involved in developing this project.



16. FUTURE ENHANCEMENT

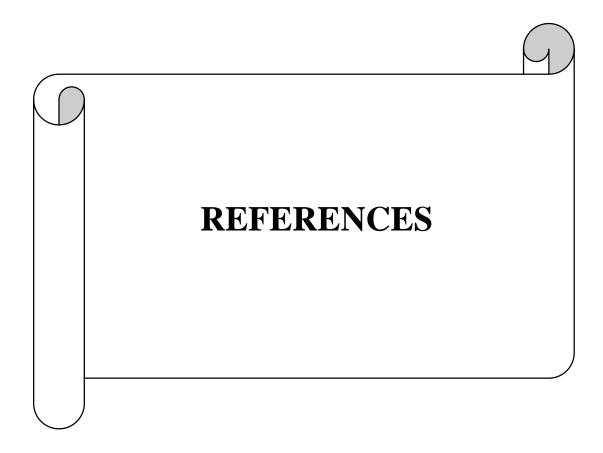
Since more than one bank accounts are being added, the existing PIN security is not sufficient enough, so we can embed a biometric scan in the smart card i.e. multicomponent card. So that the user holds the card such that the finger rests on the biometric scan reader while he swipes the card and the image is authenticated at the real time. No one other than the user and his/her nominees can use the card. Only if the thumb impression matches the next step is processed otherwise the transaction will not be allowed until the user is authenticated. Future research will help to do away with PINs completely and dwarf ATM card authorization by introducing palm and finger vein authentication which is fast, accurate and difficult to fake. No one other than the user can use the card.



17. CONCLUSION

Thus the user can manage his/her multiple accounts in various banks with the help of this single smart card which provides easy access and reduces the complexity of managing more than one ATM card and their passwords. This also leads to lessen the transaction charges that were levied on the users/ customers for transaction and decrease in the production of smart cards for each every account the user has. By implementing this the ATM fraud i.e. skimming etc can be avoided.

The system we are using for handling multiple accounts here is more user friendly than present system. This even Reduces transaction cost of handling multiple accounts of a single user. This also make banking system more inclusive than the present system. Using this the users can perform transactions for all his bank Accounts using single smart ATM card with Enhanced security system. Thus the user can manage his/her multiple accounts in various banks with the help of this single smart card which provides easy access and reduces the complexity of managing more than one ATM card and their passwords. This also leads to lessen the transaction charges that were levied on the users/customers for transaction and decrease in the production of smart cards for each every account the user has. By implementing this the ATM fraud i.e. skimming etc can be avoided.



18. REFERENCES

Bibliography

- 1) Gokul.R, Godwin Rose Samuel.W, Arul.M, Sankari.C, "Multi account Embedded ATM card", "International Journal of Scientific and Engineering Research".
- 2) Harshal M.Bajad, Sandeep E.Deshmukh, Pradnya R.Chaugule, Mayur S.Tambade, "Universal ATM Card System", "International Journal of Engineering Research and Technology".
- 3) Fumiko Hayashi, Richard Sullivan and Stuart E.Weiner, "A Guide to the ATM and Debit card Industry".
- Java The Complete Reference Book J2SE 5 Edition Developer Fusion:
 Connecting Developers Worldwide.
- 5) Introduction To Programming with PHP Complete reference
- 6) Beginning PHP and MYSQL: from Novice to Professional(Third Edition)

Web Sites

- 1) http://www.mediafire.com/?yokymw2mymz
- 2) http://www.sun.com/books/java_series.html
- 3) http://www.w3schools.com
- 4) http://php.net

