

PROJECT REPORT
ON
COLOR CODE AND FINGER PRINT AUTHENTICATION BASED
ATM SYSTEM

Submitted in partial fulfillment of the requirement for the award of degree in

MASTER OF COMPUTER APPLICATIONS

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted by

SHIFANA M

LNCE16MCA038

Under the guidance of

Ms. SOMYA N., MCA

Assistant Professor



DEPARTMENT OF MCA
NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE,
PAMPADY, THIRUVILWAMALA, THRISSUR-680 567
MAY 2019

COLOR CODE AND FINGER PRINT AUTHENTICATION BASED ATM SYSTEM

PROJECT REPORT

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



CENTRE: NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE, PAMPADY

MCA

2016-2019

Name : SHIFANA M
Reg.No : LNCE16MCA038
Semester : SIX

NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE, PAMPADY



CERTIFICATE

This is to certify that, the project work entitled **COLOR CODE AND FINGER PRINT AUTHENTICATION BASED ATM SYSTEM** is a bona-fide record of the original work done by **SHIFANA M, Reg. No: LNCE16MCA038**, at **NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE**, during **Jan-May 2019** in partial fulfillment of the requirement for the award of degree in **MASTER OF COMPUTER APPLICATIONS** of **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**

We also certify that the work done is original.

Project Guide

Head of the Department

Principal

External Examiner

DECLARATION

I hereby declare that the project entitled **“COLOR CODE AND FINGER PRINT AUTHENTICATION BASED ATM SYSTEM”** submitted to **APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY** in partial fulfillment of the requirement for the award of degree in **MASTER OF COMPUTER APPLICATIONS** is a record of the original work done by me under the guidance of **Ms. SOMYA N., Assistant Professor**, MCA department, during the period of study in **NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE, PAMPADY.**

Pampady

SHIFANA M

/ /

“with sincere respect and love, I dedicate this project to my parents, teachers, friends and my well-wishers for all the support and guidance showed to me in the way of my project”

ACKNOWLEDGEMENT

First and foremost, I thank the God Almighty for showering His blessings upon me and forgiving the auspicious and grace to make the right decision with dignity.

I hereby acknowledge the fact that the project entitled “**COLOR CODE AND FINGER PRINT AUTHENTICATION BASED ATM SYSYTEM**” would not have materialized without the guidance and help I received from concerned authorities. I express my sincere gratitude to all those who have spared their contribution in this effort.

I owe my sincere thanks to the **Management**, and **Prof. Dr. Ambikadevi Amma T., Principal**, NCERC for the immense support given during my course and project. I would like to express my gratitude to **Dr. Sudheer S Marar., HOD**, Department of MCA and my project guide **Ms.SOMYA N.**, Assistant Professor, MCA, whose support, stimulating suggestions and encouragement helped me in all the time of doing project.

Moreover, my sincere thanks go to my friends, teachers and other staffs of Nehru College of Engineering and research centre who has given the moral and technical support in all possible ways to complete this project. I as ever, especially indebted to my parents for their love and support throughout my life.

ABSTRACT

Color coding and Fingerprint Based ATM/shopping system are a desktop application where fingerprint of the user is used as an authentication. The finger print minutiae features are different for each human being so the user can be identified uniquely. Instead of using ATM card Fingerprint based ATM is safer and secure. There is no worry of losing ATM card and no need to carry ATM card in your wallet. You just have to use your fingerprint in order to do any banking transaction. The user has to login using his fingerprint and he has to enter the color coding based pin code in order to do further transaction. The user can withdraw money from his account. User can transfer money to various accounts by mentioning account number. In order to withdraw money user has to enter the amount he want to withdraw and has to mention from which account he want to withdraw (i.e. saving account, current account) .The user must have appropriate balance in his ATM account to do transaction. User can view the balance available in his respective account. The system will provide the user to view last 5 transactions.

CONTENTS

NO	TITLE	PAGE NO.
I.	INTRODUCTION	
	About the project	1
II.	SYSTEM ANALYSIS	
	Existing System	2
	Proposed System	2
III.	SYSTEM DESIGN AND DEVELOPMENT	
	Design process	4
	System specification	6
	About the tool	6
	Data flow diagram	9
	Table Design	18
	Module description	20
IV.	SYSTEM TESTING AND IMPLEMENTATION	
	Preparation of Test Data	21
	Test Methods	24
	Test plan	30

Implementation	30
Git log details	33
V. SYSTEM SECURITY	
Checks and Controls	35
Data Security	36
User Security	36
VI. POST IMPLEMENTATION	
System Evaluation	38
Maintenance	39
VII. CONCLUSION	
Scope for Future Enhancements	41
VIII. BIBLIOGRAPHY	
Books/Articles	42
IX. ANNEXURE	
Screen Shots	44

I. INTRODUCTION

1.1 About the Project

The ATM (Automated Teller Machine) is used by customers to access their bank deposit or credit accounts in order to make a variety of financial transactions like cash withdrawal or checking the balance etc. by any financial institution. It is an electronic telecommunication device that helps in performing the operations with a human cashier or teller. ATMs today use the magnetic strip or tape on the user's card for identification and authentication of the customer. In the current ATM system two factor authentications is obtained where security can be breached, when password is divulged to an unauthorized user or card is stolen by an imposter. Moreover, they are prone to fraud, and offer varied elements of risk such as to eves-dropping, dictionary attacks, social engineering and shoulder surfing. Furthermore, simple passwords are easy to guess while difficult password may be sniffed using sophisticated techniques; therefore, this system is not entirely secure. So that adding a third level of authentication can provide significant level of strength by relying on something that the user 'is' i.e. biometrics, which means something about that person that cannot be changed and easily mimicked. Biometric authentication is one of the most exciting technical improvements of recent history and looks set to change the way in which the majority of individuals live. To solve this problem, we added fingerprint verification to this method. Fingerprint Verification System is an easy-to-use library that allows programmers to integrate fingerprint technology into their software without specific know-how. This is based on how no two individuals can share the same morphological characteristics. Tint finger prints as their distinct identification to address this problem, finger print authentication can be combined with colors to generate session passwords. Every time a new sequence of color is generated and the user has to enter the color code that saved on the database.

II. SYSTEM ANALYSIS

2.1 Existing System

Automated teller machines, or ATMs, are machines that function like bank tellers, allowing customer to perform basic banking functions, such as making deposits, making withdraws and shifting money between different accounts. In place of identification, bank members use personalized debit cards to access their holdings. There are number of disadvantages to these machines.

The existing system the potential for identity theft is major disadvantage related to automatic teller machines. Fraudulent card readers, called skimmers, are placed over the authentic reader to transfer numbers and codes to nearby thieves. Spy cameras are also used by password voyeurs to collect access codes. Lost access cards are another potential for fraud.

2.1.1 Disadvantages

- ATM may be off-line (System down).
- You may forget your PIN number.
- Risk of robbery when you leave the ATM.
- The ATM can break down or run out of cash.
- Fees charged to use ATMs of other banks can become expensive.
- Your ATM card is protected by a PIN, keeping your money safe.

2.2 Proposed System

Fingerprint Based ATM is an application where fingerprint of the user is used as an authentication. The finger print minutiae features are different for each human being so the user can be identified uniquely. Instead of using ATM card Fingerprint based ATM is safer and secure. There is no worry of losing ATM card and no need to carry ATM card in your wallet. You just have to use your fingerprint in order to do any banking transaction.

2.2.1 Advantages

- Less effort to complete transaction.

- Safety of bank customer's money.
- Less time required.
- No need to maintain the card.
- Fairly small storage space is required for the biometric template, reducing the size of the database required.
- Each and every fingerprint including all fingers are unique, even identical twins have different fingerprints.
- Relatively offers high levels of accuracy.

III. SYSTEM DESIGN AND DEVELOPMENT

3.1 Design Process

The most creative and challenging phase of the life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementations of the candidate system. The design may be defined as “the process or a system with sufficient details to permit its physical realization”.

The designer’s goal is how the output is to be produced and in what format. Samples of the output and input are also presented. Second input data and database files have to be designed to meet the requirements of the proposed output. The processing phases are handled through the program Construction and Testing. Finally, details related to justification of the system and an estimate of the impact of the candidate system on the user and the organization are documented and evaluated by management as a step toward implementation.

The importance of software design can be stated in a single word “Quality”. Design provides us with representations of software that can be accessed for quality. Design is the only way where we can accurately translate a customer’s requirements into a complete software product or system. Without design we risk building an unstable system that might fail if small changes are made. It may as well be difficult to test, or could be one who’s quality can’t be tested. So it is an essential phase in the development of a software product.

The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The design phase is a transition from a user-oriented document to document oriented to the programmers or database personnel. System design goes through to phase of development:

- Logical Design.
- Physical Design.

The dataflow diagram shows the logical flow of the system and defines the boundaries of the system. For a candidate system, it describes the inputs(source), output(destination), database(file) and procedures(dataflow), all in a format that meets the user's requirement. In logical design, we specify the user's needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources.

Following logical design is physical design. This produces the working system by defining specifications that tell programmers exactly what the candidate system must do, in turn we write the necessary programs or modify the software package that accept input from the user, then perform the necessary operation. Logical system design is one important phase of system design, for a candidate system it describes the inputs of source, outputs or destination, database or data stores and procures all in a format that meets the user needs. When analysts prepare the logical system design, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. The logical system design covers: Reviews the current physical system its dataflow, file content, volumes, frequency etc.

Preparing output specification that determines the format, content and frequency of reports including terminal specification and location. Prepares input specification format, content and the most if the input functions. This includes determining the flow of the document from the input data source to the detailed output location. Prepares edit security and control specification this includes specifying the rules for edit correction backup procedures and the controls that ensure processing file integrity specifies the implementation plan.

Prepare the logical design walks through the information flow output, input and controls and implementation plan reviews benefits, costs, target rates and system constraints the existing file and procedure reports.

3.2 System Specification

3.2.1 HARDWARE REQUIREMENTS

Following are the minimum required for the proposed system

- Processor : Intel Core
- Hard Disk : 120GB
- RAM : 2 GB
- Monitor : Lenovo 15 inches
- Mouse : Genius Scroll Mouse
- Keyboard : 100007 keys

3.2.2 SOFTWARE REQUIREMENT

Following are the software minimum required for the proposed system

- Front End : JAVA Swing
- Software Tools : Eclipse and SQLYog
- Back End : MYSQL 5.0
- Operating System : Windows 7/10

3.3 About the Tool

3.3.1 JAVA SWING

Java SWING is a technology for developing window application that support dynamic content. Swing is the next-generation GUI toolkit that Sun Microsystems created to enable enterprise development in java. Bt *enterprise development*, we mean that programmers can use Swing to create large-scale Java applications with a wide array of powerful components. In addition, you can easily extend or modify these components to control their appearance and behavior. Swing is not an acronym. The name represents the collaborative choice of its designers when the project was kicked off in late 1996. Swing is actually part of a large family of Java products known as the Java Foundation Classes (JFC), which incorporate many of the features of Netscape's Internet Foundation Classes (IFC) as well as design aspects from IBM's Taligent division and Lighthouse Design. Swing has been in active

development since the beta period of the Java Development Kit (JDK) 1.1, circa spring of 1997. The swing APIs entered beta in the latter half of 1997 and were initially released in March 1998. When released, the swing 1.0 libraries contained nearly 250 classes and 80 interfaces. Growth has continued since then: at press time, Swing 1.4 contains 85 public interfaces and 451 public classes. Although Swing was developed separately from the core Java Development Kit, it does require at least JDK 1.1.5 to run. Swing builds on the event model introduced in the 1.1 series of JDKs; you cannot use the Swing libraries with the older JDK 1.0.2. In addition, you must have a Java 1.1-enabled browser to support Swing applets. The Java 2 SDK 1.4 release includes many updated Swing classes and a few new features. Swing is fully integrated into both the developer's kit and the runtime environment of all Java 2 release (SDK 1.2 and higher), including the Java Plug-In.

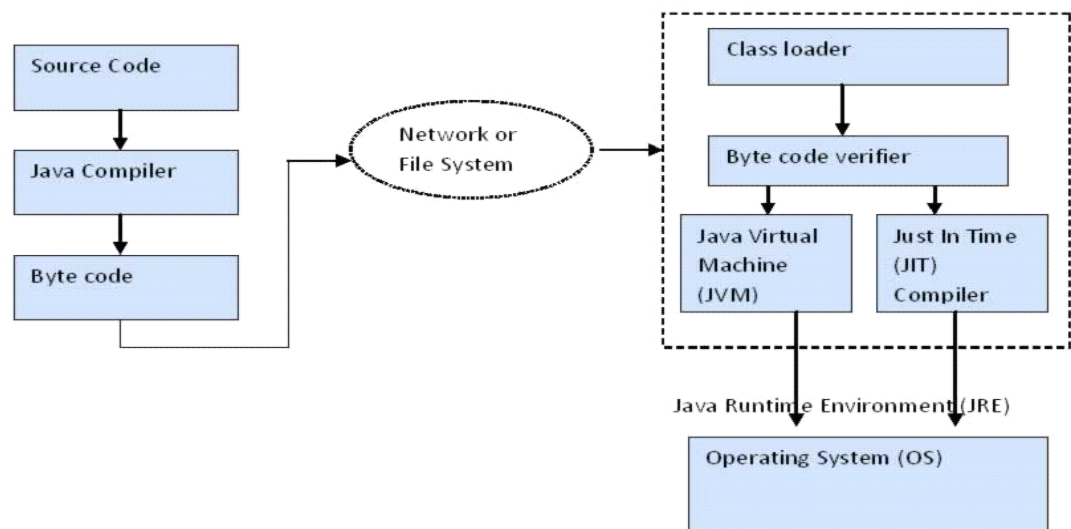


Fig 3.1 Java Architecture

3.3.2 SQLYog

SQLYog is a well-rounded, multi-functional front-end for MySQL which can be used by both newbie's and more experienced users to manage their databases. I'd clean up and reorganize the interface a little bit and remove a lot of the icons as well as list all the functions under the top menus, possibly without the icons and without repeating the same function anywhere.

A part from those small items, SQLYog is definitely worth a shot, and the Webyog team definitely did a good job in this fifth version by incorporating all the latest MySQL 5 functionalities in an already excellent program. The free version in particular offers quite a wide range of functionalities with no trail period, and this certainly helped the program to grow in popularity.

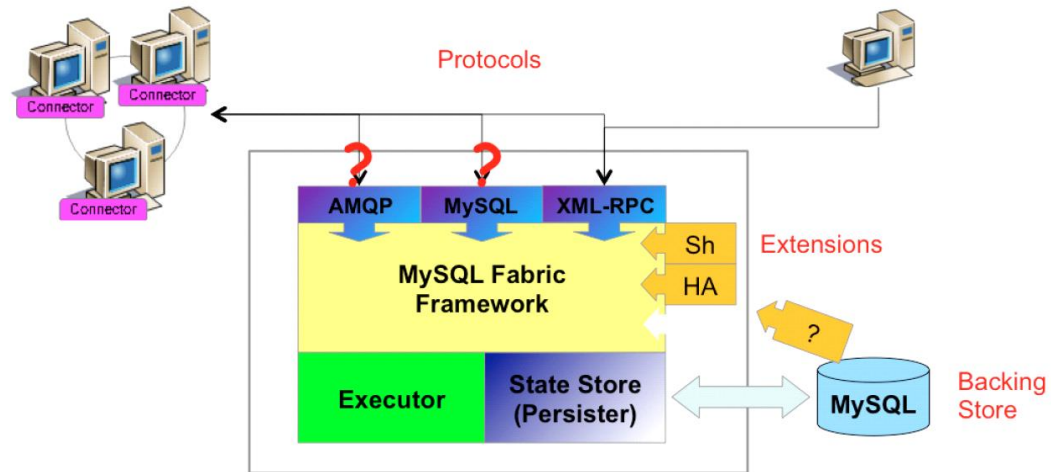


Fig 3.2 SQL Architecture

3.3.2.1 Features

- Editor with syntax highlighting and various automatic formatting options.
- Intelligent Code Completion.
- Data manipulations (INSERT, UPDATE, DELETE) may be done from a spreadsheet-like interface. Both raw table data and a result set from a query can be manipulated.
- Visual Schema Designer.
- Visual Query Builder.
- Query Formatter.
- Connectivity options: Direct client/server using MySQL API (SSL supported), HTTP/HTTPS Tunneling, SSH Tunneling.

- Wizard driven Tool for import of data from ODBC-databases
- Backup Tool for performing unattended backups. Backups may be compressed and optionally stored as a file-per-table as well as identified with a timestamp.
- 'SQL Scheduler and Reporting Tool' - a tool for scheduling and automating execution of any sequence of SQL statements. Result of queries may be sent as HTML-formatted reports.
- Schema/Structure Synchronization and Data Synchronization.
- Query Profiler and Redundant Index Finder.
- All automated jobs have mail alerting and reporting options.
- Full character set/Unicode support.
- A 'Data Search feature using a Google-type search syntax translated transparently for user to SQL.
- Form view to display one row at a time - a great way to enter/edit data.
- Foreign key lookup.
- Visual Data Compare

3.4 Data Flow Diagram

The database may be defined as an organized collection of related information. The organized information serves as a base from which further recognizing can be retrieved desired information or processing the data. The most important aspect of building an application system is the design of tables. The data flow diagram is used for classifying system requirements to major transformation that will become programs in system design. This is starting point of the design phase that functionally decomposes the required specifications down to the lower level of details. It consists of a series of bubbles joined together by lines.

Bubbles: Represent the data transformations.



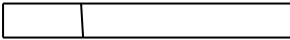

Lines: Represents the logic flow of data.

Data can trigger events and can be processed to useful information. System analysis recognizes the central goal of data in organizations. This dataflow analysis tells a great deal about organization objectives are accomplished.

Dataflow analysis studies the use of data in each activity. It documents this finding in DFD's. Dataflow analysis give the activities of a system from the viewpoint of data where it originates how they are used or hanged or where they go, including the stops along the way from their destination. The components of dataflow strategy span both requirements determination and system's design. The first part is called dataflow analysis.

As the name suggests, we didn't use the dataflow analysis tools exclusively for the analysis stage but also in the designing phase with documentation.

3.3.1 Notation used in Dataflow Diagrams

Elements References	Symbols
Data flow process	
Process	
Data store	
Source sink	

Process: describes how input data is converted to output Data

Data Store: Describes the repositories of data in a system

Data Flow: Describes the data flowing between process, Data stores and external entities.

Sources: An external entity causing the origin of data.

Sink: An external entity, which consumes the data.

3.3.2 Constructing a DFD

Several rules of thumb are used in drawing DFDs:-

- Process should be named and numbered for easy reference.
- The direction of flow is from source to destination, although they may flow back to a source. One way to indicate this is to draw a long flow line back to the source. An alternative way is to repeat the source symbol as a destination.
- When a process is exploded into lower-level details, they are numbered.
- Names of data stores, sources, and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized.

A level 0 DFD, also called a context level, represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e. Level 1 DFD. Any process, which is complex in Level 1, will be further represented into sub functions in the next level i.e. Level 2 DFD is a means of representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data process, sources or destination.

The DFD is designed to aid communication. DFD shows the minimum contents of data stores. In order to show what happens within a given process, then the detailed explosion of that process is shown. The DFD methodology is quite effective, especially when the required design is unclear and the user and the analyst need a notational language for communication.

The top-level diagram is often called a “*context diagram*”. It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

3.3.3 Types of Dataflow Diagrams

1. Physical DFD

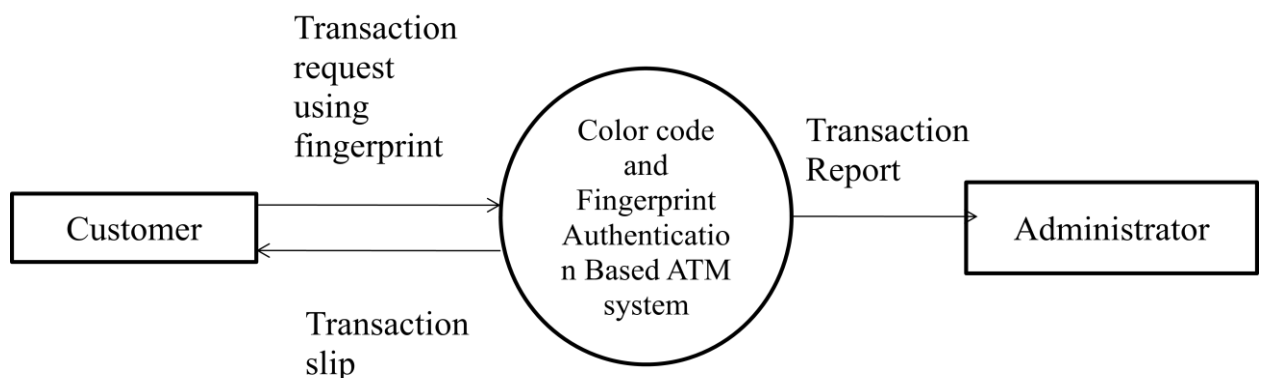
Structured analysis states that the current system should be first understood correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs show actual devices, departments, people etc., involved in the current system

2. Logical DFD

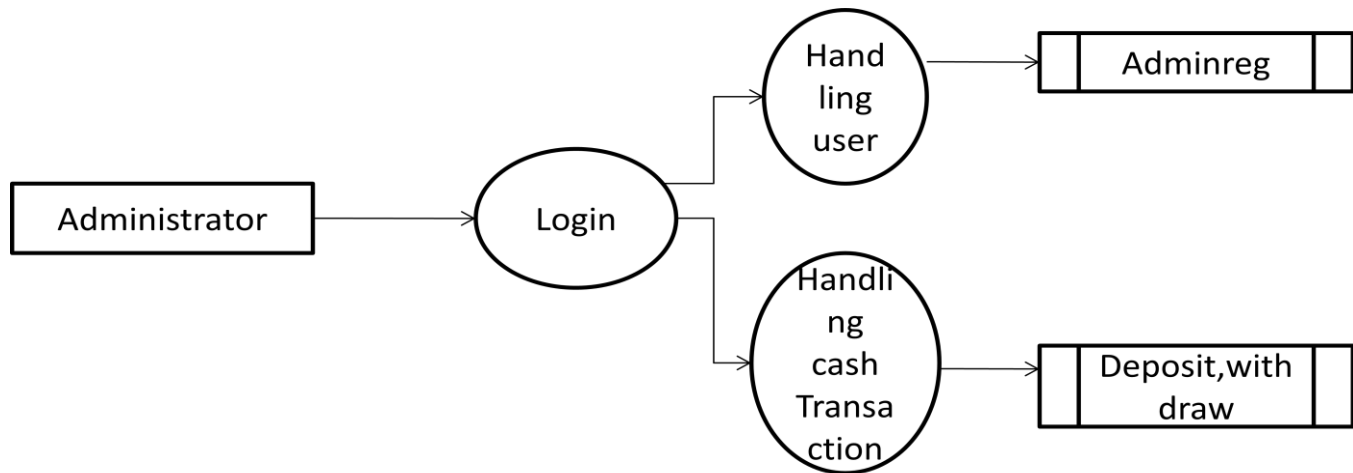
Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts.

DATA FLOW DIAGRAM

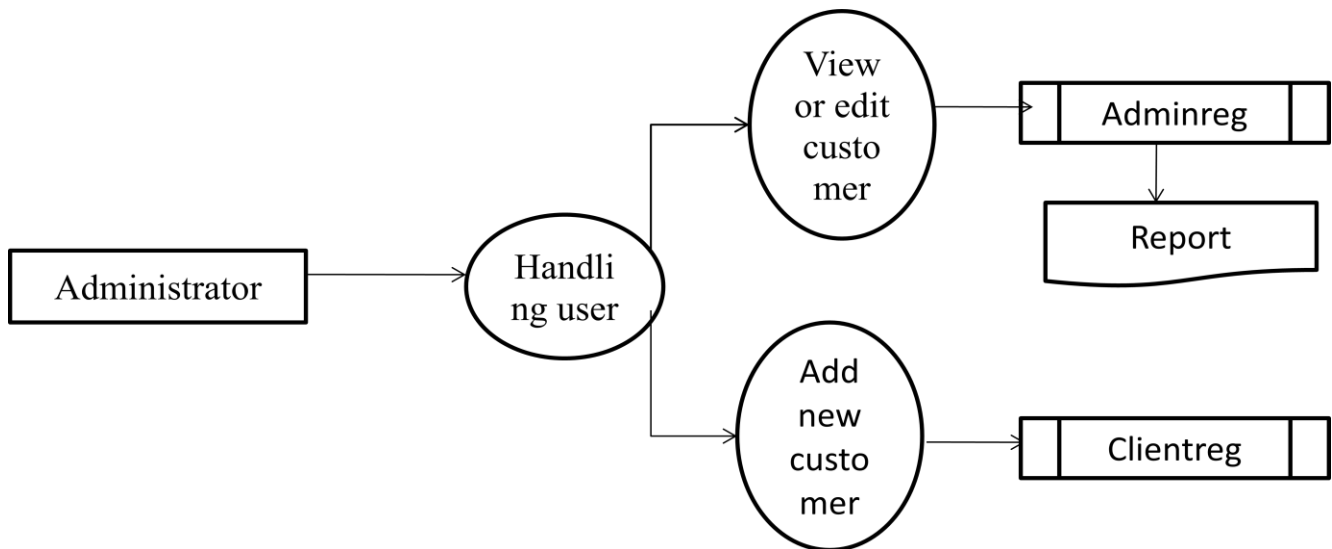
LEVEL 0



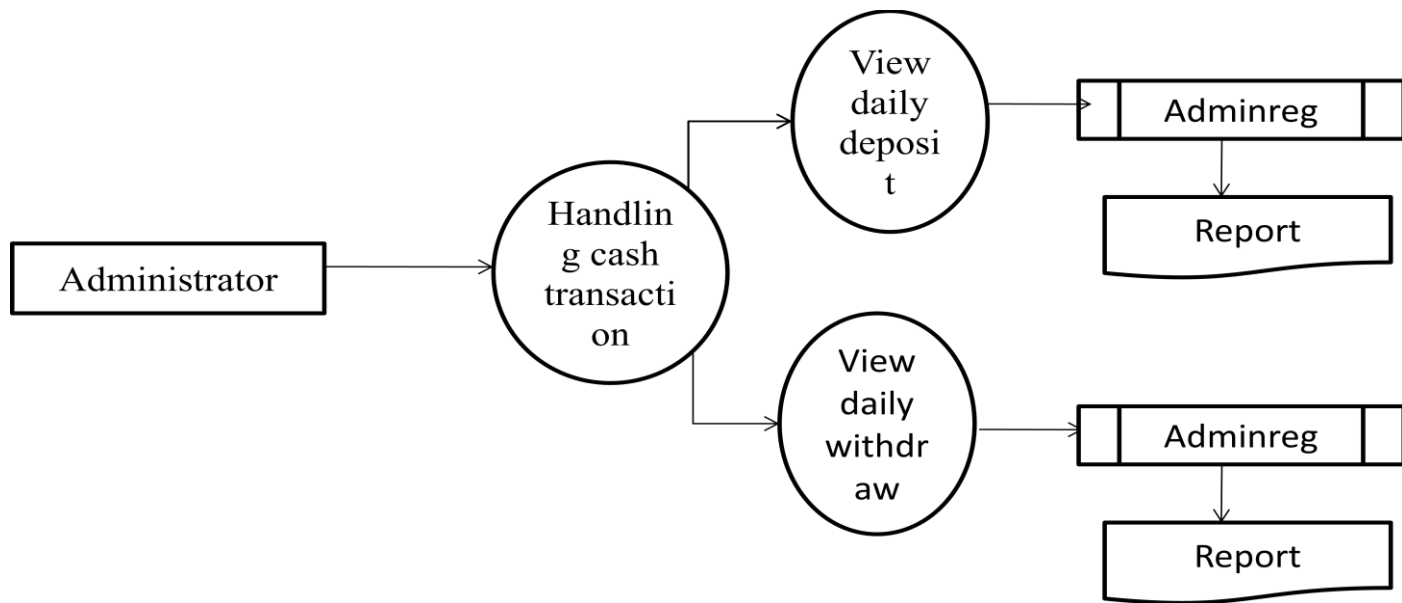
LEVEL 1



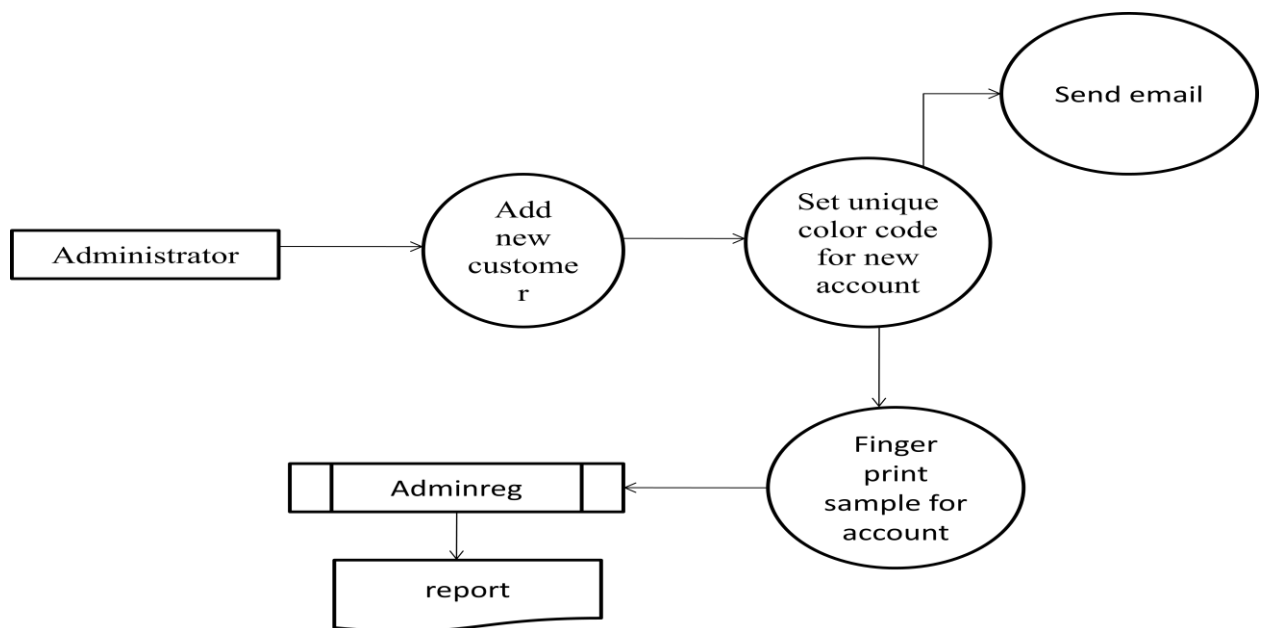
LEVEL 1.1



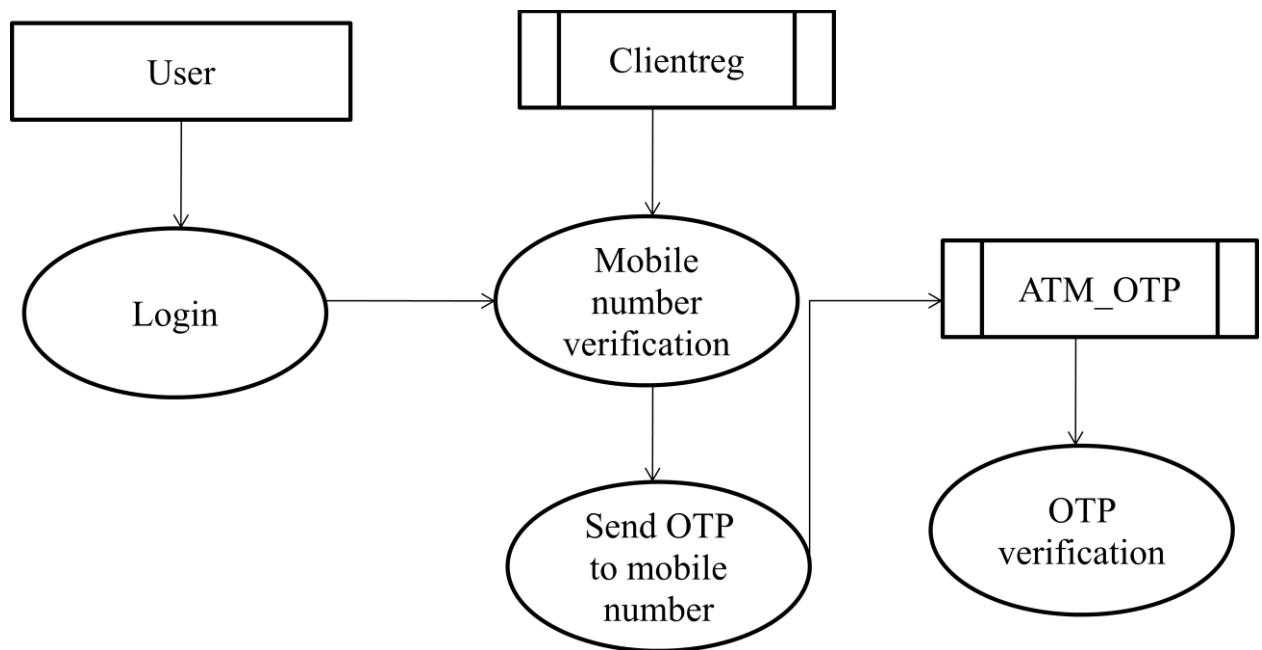
LEVEL 1.2



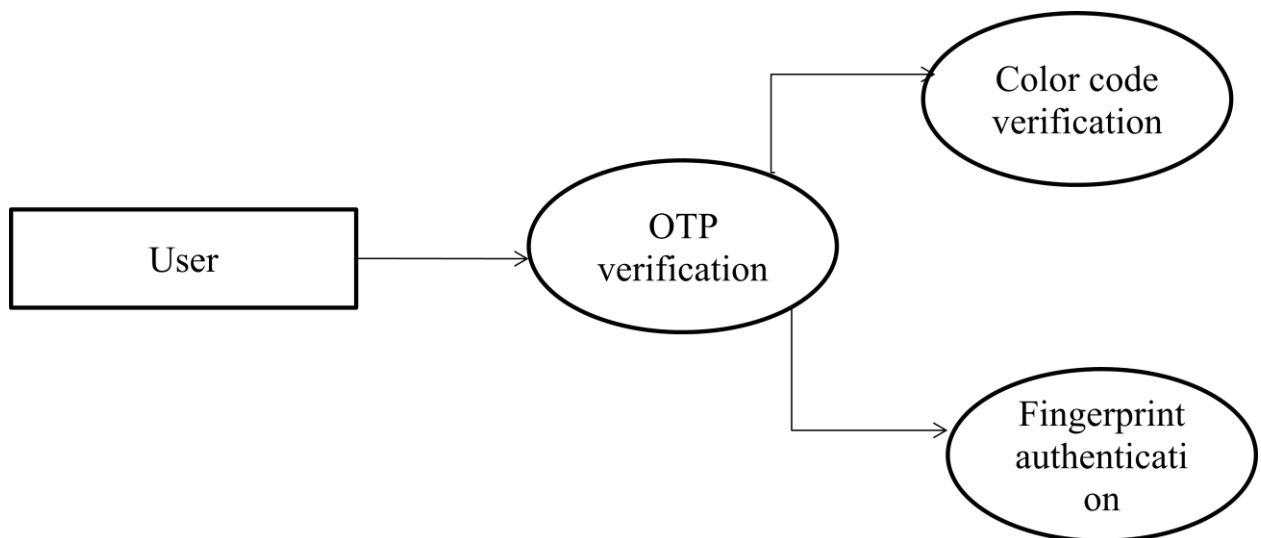
LEVEL 1.3



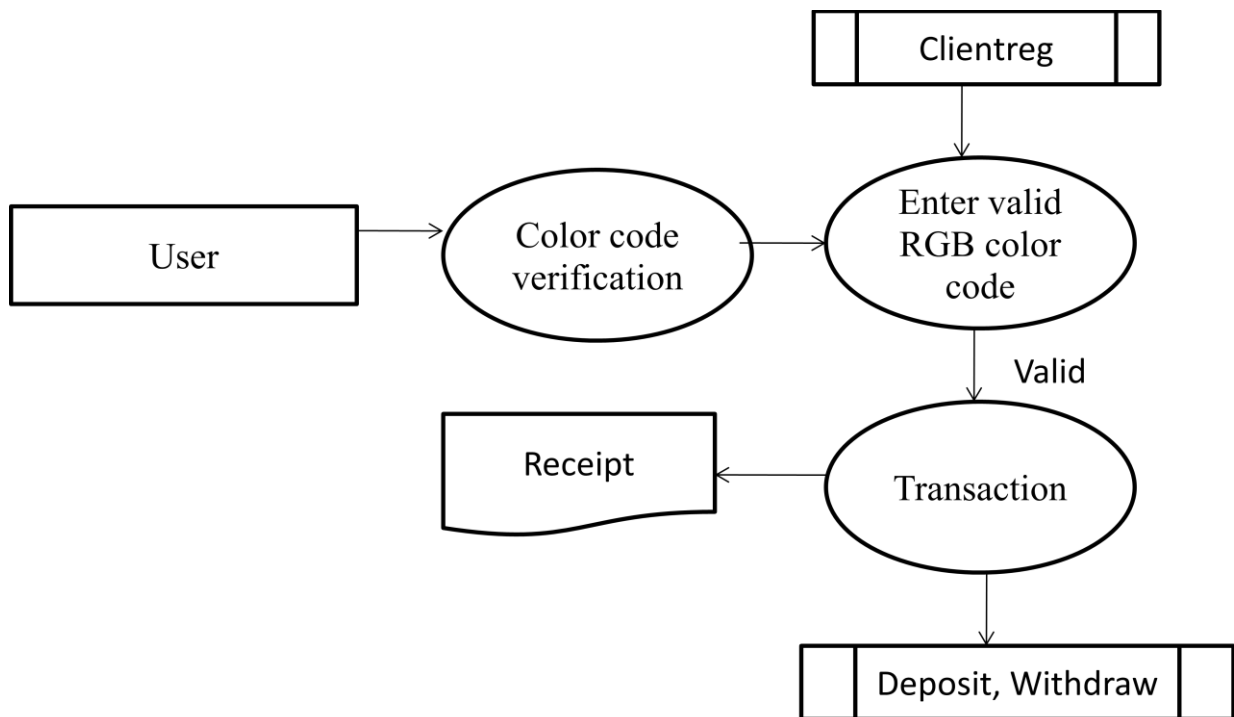
LEVEL 2



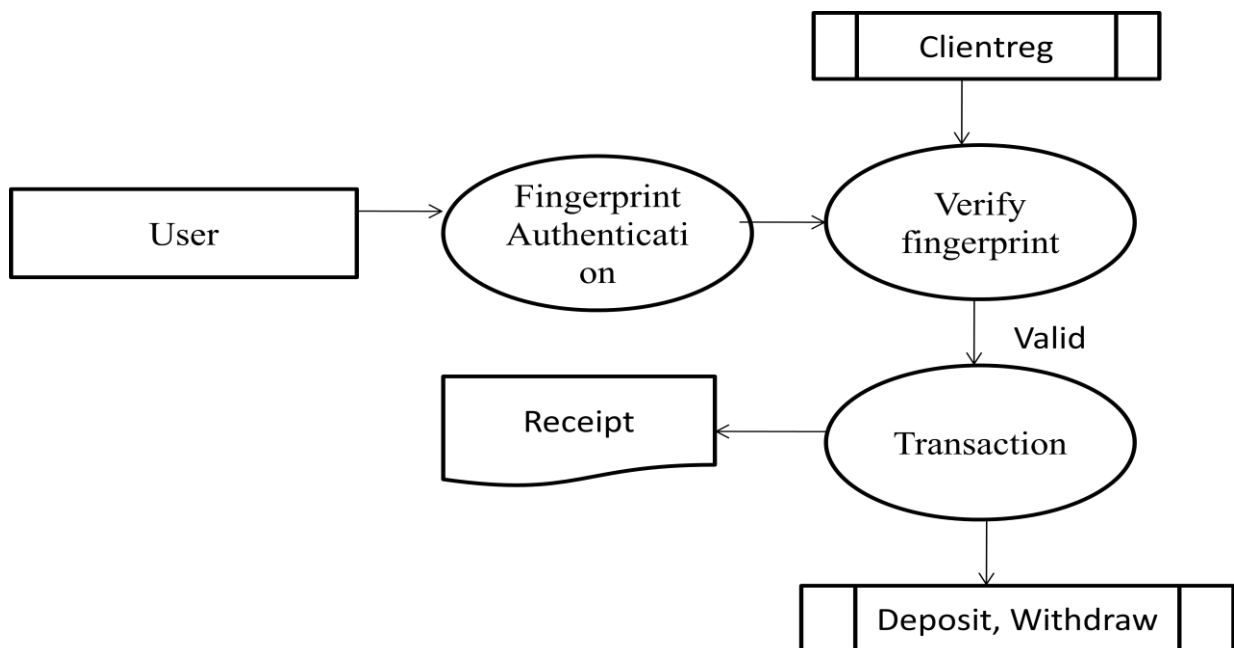
LEVEL 2.1



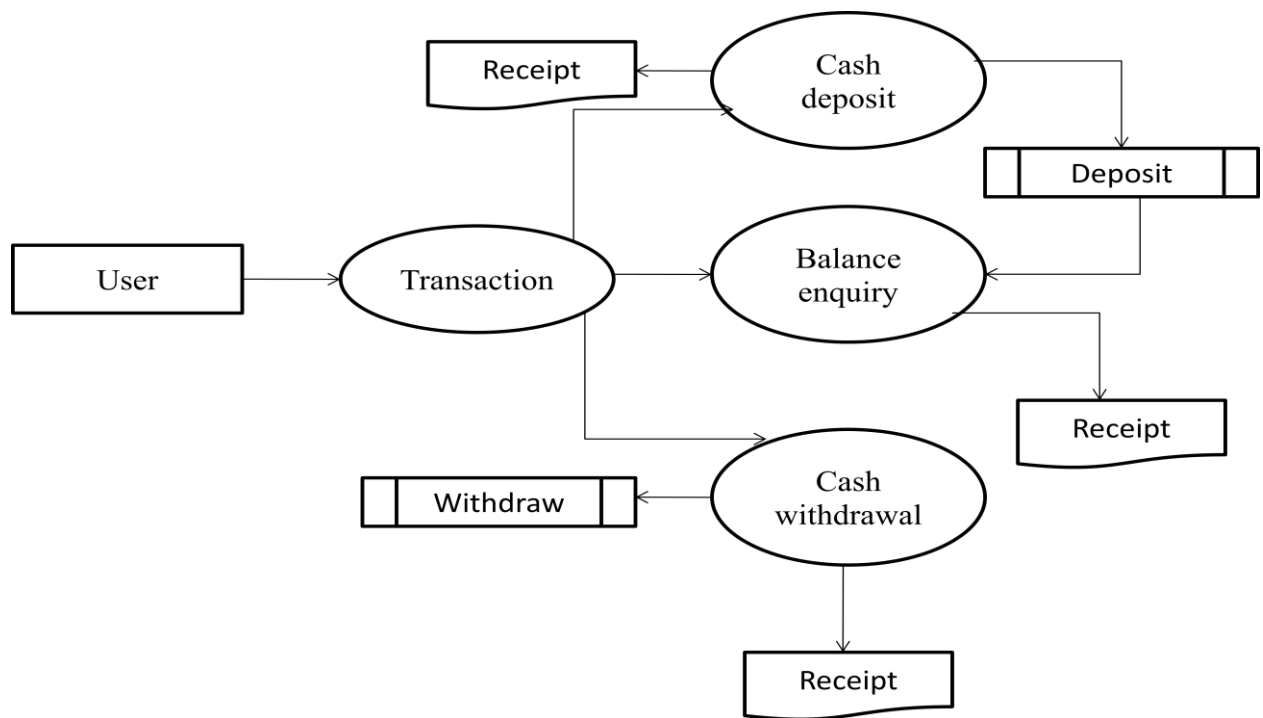
LEVEL 2.2



LEVEL 2.3



LEVEL 2.4



3.4 Table Design

Tables: A table contains a group of fields of related information that defines a single category. The table stores the data in fields. A set of fields that defines one entry is called a record.

Adminreg-Handles the Admin registration details

Fieldname	Datatype	Len	Primary key
No	Int	20	Yes
Username	Varchar	50	Yes
Firstname	Varchar	50	No
Lastname	Varchar	50	No
Pinnumber	Varchar	4	No
Bday	Varchar	2	No
Bmonth	Varchar	2	No
Byear	Varchar	4	No
Registrationdate	Varchar	10	No

N o	Usenam e	Firstnam e	Lastnam e	Pinnumbe r	Bda y	Bmont h	Byea r	Registrationdat e
1	Admin	Fne	Lne	1234	20	11	2001	11/01/2018

Clientreg-handles the Client registration details that includes RGB and fingerprint

Fieldname	Datatype	Len	Primary key
No	Int	20	Yes
Username	Varchar	50	Yes
Firstname	Varchar	50	No
Lastname	Varchar	50	No
Pinnumber	Varchar	4	No
Mobileno	Varchar	20	No
Bday	Varchar	2	No
Bmonth	Varchar	2	No
Byear	Varchar	4	No
Registrationdate	Varchar	10	No
Fingerprint	Varchar	50	No
R	Varchar	50	No

G	Varchar	50	No
B	Varchar	50	No
Balance	Varchar	50	No
Email	Varchar	50	No

Deposit- Handles the deposit details of all the clients

Fieldname	Datatype	Len	Primary key
No	Int	10	Yes
Username	Varchar	50	Yes
Depositamnt	Varchar	50	No
Depositdate	Varchar	50	No

No	Username	Depositamnt	Depositdate
1	User1	5000	19/04/2019 14:30:50
2	User2	10000	19/04/2019 14:40:57
3	User3	5000	19/04/2019 14:45:20
4	User1	6000	25/04/2019 13:15:40
5	User2	235	26/04/2019 09:57:44

Withdraw- handles the withdraw details of all the client

Fieldname	Datatype	Len	Primary key
No	Int	10	Yes
Username	Varchar	50	Yes
Withdrawamnt	Varchar	50	No
Withdrawdate	Varchar	50	No

No	Username	Withdrawamnt	Withdrawdate
1	User1	200	19/04/2019 14:31:13
2	User2	3000	25/04/2019 13:15:31
3	User3	2999	26/04/2019 09:57:54


3.6 Module Description


➤ Transaction Module (ATM)

In the Transaction process the customer must choose the Transaction type like Thumb Authentication or Color code Authentication. Next the verification panel will be displayed after verify the unique identification next step is to choose the operation type(cash deposit/cash withdrawal/balance enquiry).Once choose the operation the cash deposit and cash withdrawal process require the amount details. After the operation success customer will get the receipt.

➤ Authentication Panel (Thumb/Color Code)

In the Authentication panel the customer must enter the valid mobile number for first verification step. After verified the first step customer will receive an OTP code .next step is to verify the OTP code. If valid OTP verification proceed the payment process. Next step the authentication panel choosing panel will appear. The customers can two options for the payment process.

 **Thumb Authentication:** The customer can press their thumb in the finger print authentication device. The matching process will be started. After successful verification the customer can choose the operation type then proceed the transaction.

 **Color Authentication:** The Customer must enter the three color code for color code authentication process. Sometimes the customers forget their codes the authentication panel contain an option to recover the code through the SMS alert. Once press the forgot option the user will receive a SMS.

➤ Transaction Reports (ATM)

The admin can only rights to view the daily customer transaction details in different ways. Place wise, date wise and month and year wise.

➤ Receipt Generation (ATM Panel)

In the ATM panel customer can get a receipt after the each successful transaction process. The receipt contains the transaction type details, amount details and balance amount details etc.

IV. SYSTEM TESTING AND IMPLEMENTATION

Testing is the process by which a developer will generate a set of test data, which gives maximum probability of finding all types of errors that can occur in the software. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to a variety of tests: online response, volume, stress, recovery & security and usability tests. A series of testing are performed for the proposed system before the system is ready for user acceptance testing.

It is the process of exercising or evaluating a system by manual or automatic means to verify that it satisfies the specified requirements or to identify the difference between expected and actual results. The testing activities are aimed at convincing the customer through demonstration and actual use that the software is a solution to the original problem and that both the product and the process that created it are of high quality. It is also used to find and eliminate any residual errors from previous stages and the operational reliability of the system.

4.1 Preparation of Test Data

Software testing is a crucial element of software quality assurance and represents the ultimate review of specification, design and coding. Testing represents an interesting anomaly for the software. During earlier definition and development phases, it was attempted to build software from abstract concepts to tangible implementation. The testing responsible for ensure that the product that has built performs the way that the detailed design documentation specifies.

4.1.1 Goals and objectives

The main purpose of testing an information system is to find the errors and correct them. The scope of system testing should include both manual and computerized operations. System testing is comprehensive evaluation of the programs, manual procedures, computer operations and controls.

System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing is to be conducted in accordance to the

test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done in a systematic manner.

4.1.2 Testing Objectives:

- Testing is the process of executing a program, with the intent of finding so many errors as possible.
- A good test case is one that has a high probability of finding an as-yet-undiscovered error.
- A successful test is one that uncovers an as-yet-undiscovered error.

So, the main objective is to design tests that systematically uncover different classes of errors using minimum time and effort. Successful testing uncovers errors in software. It also shows that the software functions are working according to specifications. Also, the data collected during testing provides an indication of software reliability and software quality.

4.1.3 Statement of scope

The strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that result in the successful construction of software. The testing must co-operate with test planning, test case design, test execution and the resultant data collection and evaluation. A strategy for software testing must accommodate low level test and that are necessary to verify that a small code segment has correctly implemented as well as high level test that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. A series of testing is performed for the proposed system before the system is ready for acceptance testing.

4.1.4 Major constraints

- All tests should be traceable to the customer requirements. According to the customer, the most severe defect is that which causes the program to fail to meet its requirements.
- Tests should be planned long before the actual testing begins. All tests should be planned and designed before any code is generated.
- The Pareto principle applies to software testing. The Pareto principle implies that 80% of all errors uncovered will likely be traceable to 20% of all program components. The problem is to isolate these suspect components and thoroughly test them.

- Testing should begin ‘in the small’ and progress towards testing ‘in the large’. The first tests focus on individual components. As testing progresses, focus shifts to integrated clusters of components and then finally to the entire system.
- Exhaustive testing is not possible. The number of path combinations in even a small program is very large. So it is not possible to test all these paths. But it is possible to test the program logic and ensure that all conditions have been met.
- To be most effective, testing should be conducted by an independent third party. The software engineering who created the program is not the best person to conduct tests for the software. So, in order to find the maximum number of errors in the software, an independent third party (who had no hand in developing the software) is preferred.

There are several rules that can serve as testing objectives:

- A good test is not redundant. Testing time and the resources are limited. So, a test that has the same purpose as another test need not be conducted. Every test should have a different purpose.
- A good test should be ‘best of breed’. There can exist a group of tests having the same intention. In such cases, only a subset of these tests is used. Thus, the test that has the highest chance of uncovering a whole class of errors should be used.
- A good test should neither be too simple nor be too complex. It is possible to combine a series of tests into one test. But this can lead to masking certain errors. Hence all the tests should be executed separately.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are subject to variety of tests on-line response, volume, stress, recovery and security and usability tests. Once the system performs flawlessly on artificial data, we switch to ‘Live Data’ or real data taken from the organization. A system is generally tested in a hierarchical fashion starting at the bottom and working up.

First each program is tested; next a series of modules is tested; then each individual program with all its modules; finally the entire system consisting of a series of programs is tested. In this way, problems at the module level can be corrected before programs are tested and problems at the program level can be corrected before the entire system is used. A series of tests are performed before the system is ready for user acceptance testing.

4.2 Testing methods

The wide diffusion of Internet has produced a significant growth of the demand of Web-based applications with more and more strict requirements of reliability, usability, interoperability and security. Due to market pressure and very short time-to-market, the testing of Web-based applications is often neglected by developers, as it is considered too time-consuming and lacking a significant payoff. This depreciable habit affects negatively the quality of the applications and, therefore triggers the need for adequate, efficient and cost effective testing approaches for verifying and validating them. Though the testing of Web-based applications (Web applications, in the remaining of the paper) shares the same objectives of ‘traditional’ application testing, in most cases, traditional testing theories and methods cannot be used just as they are, because of the peculiarities and complexities of Web applications. Indeed, they have to be adapted to the specific operational environment, as well as new approaches for testing them are needed.

In common a web based application is tested using:

- Performance testing
- Functionality testing
- Compatibility testing
- Accessibility testing
- Security testing
- Usability testing

4.2.1 Performance testing

Performance testing objective is to verify specified system performances (e.g. response time, service availability). It is executed by simulating hundreds, or more, simultaneous users accesses over a defined time interval. Information about accesses are recorded and then analyzed to estimate the load levels exhausting system resources. For Web applications, system performances is a critical issue because Web users don’t like to wait too long for a response to their requests, also they expect that services are always available. Performance testing of Web applications should be considered as an everlasting activity to be carried out by analyzing data from access log files, in order to tune the system adequately. Failures uncovered by performance testing are mainly due to running environment faults (such as scarce resources, or not well deployed resources, etc.), even if any software component of the application level may contribute to inefficiency.

This testing Include

- **Connection Speed**
- **Load**
- **Stress**

4.2.2 Functional testing

The functionality of the application like, calculation, business logic, validation links and navigation should be proper. In web based application the following functional tests are carried out

- Links
- Internal Links
- External Links
- Mail Links
- Broken Links
- Forms
- Field validation
- Error message for wrong input
- Optional and Mandatory fields
- Database
- Testing will be done on the database integrity.
- Cookies
- Testing will be done on the client system side, on the temporary Internet files.

4.2.3 Compatibility testing

Compatibility testing will have to uncover failures due to the usage of different Web server platforms or client browsers, or different releases or configurations of them. The large variety of possible combinations of all the components involved in the execution of a Web application does not make it feasible to test all of them, so that usually only most common combinations are considered. As a consequence, just a subset of possible compatibility failures might be uncovered. Both the application and the running environment are responsible for compatibility failures.

4.3 Accessibility testing

It can be considered as a particular type of usability testing whose aim is to verify that access to the content of the application is allowed even in presence of reduced hardware/software configurations on the client side of the application (such as browser configurations disabling graphical visualization, or scripting execution), or of users with physical disabilities (such as blind people). In the case of Web applications, accessibility rules such as the one provided by the Web Content Accessibility Guidelines have been established, so that accessibility testing will have to verify the compliance to such rules. The application is the main responsible for accessibility, even if some accessibility failures may be due to the configuration of the running environment (e.g., browsers where the execution of scripts is disabled).

4.4 Security testing

The objective of security testing is to verify the effectiveness of the overall Web system defenses against undesired access of unauthorized users, as well as their capability to preserve system resources from improper uses, and to grant the access to authorized users to authorized services and resources. System vulnerabilities affecting the security may be contained in the application code, or in any of the different hardware, software, middle-ware components of the systems. Both the running environment and the application can be responsible for security failures. In the case of Web applications, heterogeneous implementation and execution technologies, together with the very large number of possible users, and the possibility of accessing them from anywhere may make Web applications more vulnerable than traditional ones, and security testing more difficult to be accomplished.

4.4 Usability testing

Usability testing aims at verifying to what extent an application is easy to use. Usability testing is mainly centered on testing the user interface: issues concerning the correct rendering of the contents (e.g. graphics, text editing format, etc.) as well as the clearness of messages, prompts and commands are to be considered and verified. Usability is a critical issue for a Web application: indeed, it may determine the success of the application. As a consequence, the front end of the application and the way users interact with it often are the aspects that are devoted greater care and attention along the application development process. When Web applications usability testing is carried on, issues about the completeness, correctness and conciseness of the navigation along application are to be considered and verified too. This type of testing should be a continuing activity carried out to improve the

usability of a Web application; techniques of user profiling are usually used to reach this aim. The application is mainly responsible for usability failures.

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing present an interesting anomaly for the software. Testing is vital to the success of the system. Errors can be injected at any stage during development. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. During testing, the program to be tested is executed with set of test data and the output of the program for the test data is evaluated to determine if the program is performing as expected. A series of testing are performed for the proposed system before the system is ready for user acceptance testing. The testing steps are:

- Unit Testing
- Integration Testing
- Validation testing
- Output Testing
- Acceptance Testing

4.4.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of the software design, the module this is known as module testing. Since the proposed system has modules the testing is individually performed on each module. Using the details design description as a guide, important control paths are tested to uncover errors within the boundary of the module. This testing was carried out during programming stage itself. In this testing step each module is found to be working satisfactorily as regards to the expected output from the module.

4.4.2 Integration Testing

Data can be test across an interface; one module can have adverse effect on another, sub function when combined may not produced the desired function. Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated within the interface.

4.2.3 Validation Testing

Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

1. The function or performance characteristics confirm to specifications and are accepted
2. A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

4.2.4 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output generator or displayed by the system under consideration is tested by asking the users about the format required by them. Here the output is considered in two ways: One is on screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. As far as hardcopies are considered it goes in terms with the user requirement. Hence output testing does not result any correction in the system.

4.2.5 Acceptance Testing

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required.

Two types of testing are white-box testing and black-box testing.

4.2.5.1 White-Box testing

White-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. White-box testing of software is predicted on close examination of procedural detail.

4.2.5.2 Black-Box testing

The black-box testing focuses on the functional requirements of the software. It helps to find out errors in incorrect or missing functions, interface errors, errors in data structures, performance errors and initialization and termination errors. The black-box testing is applied during the later stages for the functional requirement evaluation.

4.3 Test Plan

A test plan is a document detailing the objectives, target market, internal beta team and processes for a specific beta test for a software or hardware product. The plan typically contains a detailed understanding of the eventual workflow. A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements.

4.4 Implementation

System implementation is the final phase i.e., putting the utility into action. Implementation is the state in the project where theoretical design turned into working system. The most crucial stage is achieving a new successful system and giving confidence in new system that it will work efficiently and effectively. The system is implemented only after thorough checking is done and if it is found working in according to the specifications.

It involves careful planning, investigation of the current system and constraints on implementation, design of methods to achieve. Two checking is done and if it is found working according to the specification, major task of preparing the implementation are educating, training the users.

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. The most important in implementation stage is, gaining the users confidence that the new system will work and be effective. The system can be implemented only after through testing is done. This method also offers the greatest security since the existing system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

4..4.1 Implementation Methods

There are several methods for handling the implementation and consists for changing from the old to the new computerized system. The most secure method for conversion from the old system is to run the old and new system in parallel .In this approach; a person may operate in the manual processing system as well as start operating the new computerized system.

Another commonly used method is a direct cut over the existing manual system to the computerized system. The change may be within a week or a day. This strategy requires planning. A working version of the system can also be implemented in one part of the organization and the changes can be made as and when required, but this method is less preference due to the loss of entire system. After the system is Implementation, a review should be conducted to determine whether the system is meeting expecting where improvements are needed.

Implementation is the process of bringing a developed system into operational use and turning it over to the user. Implementation includes all those activities that take place to convert from old system to new. At this stage the theoretical design is turned into a working system. The crucial stage in achieving a successful new system and giving confidence on the system for the users that will work efficiently and effectively.

The implementation stage involves the following tasks:

- Careful planning
- Investigation of the current system and its constraints
- Design of methods to achieve the changeover

- Training of staff in the overall procedures
- Evaluation of changeover

4.4.2 Implementation Plan

Implementation plan includes a description of all activities that must occur to implement the new system and to put into operation. It defines the person responsible for the activities and prepares a time chart for implementing the system. The Implementation plan should anticipate possible problems and must be able to deal with them. The usual problem may be missing documents, missed data formats between current and new files, errors in data translation, missing data etc. Training plans are an important element of the implementation plan. Their purpose is to ensure that all the persons who are associated with the computer based business system possess the necessary knowledge and skills.

4.4.3 Operator Training

Operator training is completed in conjunction with its installation and checkout. Operators must become familiar with operational requirements of the new systems. Well prepared manuals provide a ready reference to specific duties and step by step operation instruction.

4.4.4 User Training

After the system is implemented successfully, training of the user is one of the most important subtasks of the developer. Even well designed and technically elegant systems can succeed or fail, because of the way they are operated and used. For this purpose user manuals are prepared and handed over to the user to operate the developed system.

Thus the users are trained to operate the developed system. Both the hardware and software securities are made to run the developed systems successfully in future. In order to put new application system into use, preparation of user and system documentation, conducting user training with demo, test run for some period to ensure smooth switching over the system are to be prepared.

4.5 GIT Log

The screenshot shows a web browser window displaying the GitHub repository page for 'shifanawas/Colour-code-Atmsys'. The browser's address bar shows the URL 'https://github.com/shifanawas/Colour-code-Atmsys'. The repository page includes a header with the repository name, a 'Watch' button with 0 notifications, a 'Star' button with 0 stars, and a 'Fork' button with 0 forks. Below the header is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The main content area shows the repository name 'mca project' with an 'Edit' button. Below this is a section with statistics: '7 commits', '1 branch', '0 releases', and '1 contributor'. A 'Branch: master' dropdown and a 'New pull request' button are also visible. A table of files and their commit history is shown below, with the latest commit being '72c36d9 9 days ago'.

File	Commit	Time
11SCREENSHOT.docx	screenshot	9 days ago
5SYSTEMDEVELOPMENT.docx	commit	11 days ago
AESAlgorithm.java	third update	20 days ago
ATMSession.java	third update	20 days ago
AddOrEditAdmin.java	first update	20 days ago
Admin.java	first update	20 days ago
AdminHome.java	first update	20 days ago
AdminRegister.java	first update	20 days ago

V. SYSTEM SECURITY

System security is a branch of technology known as information security as applied to computers and networks. The objective of system security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The terms system security, means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The technologies of system security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behavior.

Internet is a part of everyday life, web applications are an essential component of every business activity. Customers and trading partners expect fast, accurate and secure applications with robust functionality. Companies want sites that are easy to maintain and update, yet cost effective. Auditors and security officers want to ensure that the web applications are controlled and that there is strong data integrity. All of these requirements need to be blended to ensure that each web application meets the company's goals, satisfies the customers and trading partners, and is secure and reliable.

The major security issues of web applications are:

Types of unauthorized access

- Network security
- Firewalls
- Routers
- Intrusion detection and monitoring
- Virus detection and monitoring
- Encryption
- Operating system security
- Business continuance and disaster preparedness
- Hacker sites
- Control and security checklists

5.1 Checks and controls

This is the process to determine that an Information System protects data and maintains functionality as intended. The six basic security concepts are:

Confidentiality

A security measure which protects against the disclosure of information to parties other than the intended users that is by no means the only way of ensuring.

Integrity

A measure intended to allow the receiver to determine that the information which it receives has not been altered in transit or by other than the originator of the information. Integrity schemes often use some of the same underlying technologies as confidentiality schemes, but they usually involve adding additional information to a communication to form the basis of an algorithmic check rather than the encoding all of the communication.

Authentication

A measure designed to establish the validity of a transmission, message, or originator. Allows a receiver to have confidence that information it receives originated from a specific known source.

Authorization

This is the process of determining that a requester is allowed to receive a service or perform an operation. Access control is an example of authorization.

Availability

Assuring information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.

Non-repudiation

A measure intended to prevent the later denial that an action happened, or a communication that took place etc. In communication terms this often involves the interchange of authentication information combined with some form of provable time stamp.

5.2 Data security

The focus behind data security is to ensure privacy while protecting personal or corporate data. Data is the raw form of information stored as columns and rows in our databases, network servers and personal computers. This may be a wide range of information from personal files and intellectual property to market analytics and details intended to top secret. Encryption has become a critical security feature for thriving networks and active home users alike. This security mechanism uses mathematical schemes and algorithms to scramble data into unreadable text. It can only be decoded or decrypted by the party that possesses the associated key.

Data security wouldn't be complete without a solution to backup your critical information. Though it may appear secure while confined away in a machine, there is always a chance that your data can be compromised. You could suddenly be hit with a malware infection where a virus destroys all of your files.

Someone could enter your computer and steal data by sliding through a security hole in the operating system. Perhaps it was an inside job that caused your business to lose those sensitive reports. If all else fails, a reliable backup solution will allow you to restore your data instead of starting completely from scratch.

5.3 User security

User security lets your application use security rules to determine what it displays. It has two elements:

Authentication

Ensures that a valid user is logged-in, based on an ID and password provided by the user. ColdFusion (or, in some cases if you use web server authentication, the web server) maintains the user ID information while the user is logged-in.

Authorization

Ensures that the logged-in user is allowed to use a page or perform an operation. Authorization is typically based on one or more *roles* (sometimes called groups) to which the user belongs. For example, in an employee database, all users could be members of either the employee role or the contractor role. They could also be members of roles that identify their

department, position in the corporate hierarchy, or job description. For example, someone could be a member of some or all of the following roles such as Employees, Human Resources, Benefits, and Managers. You can also use the user ID for authorization.

Authenticating users

You can use either, or both, of the following forms of authentication to secure your Cold Fusion application:

- Web server authentication, where the web server authenticates the user and does not allow access to the website by users without valid login IDs.
- Application authentication, where the Cold Fusion application authenticates the user and does not allow access to the application by users without valid login IDs.

VI. POST IMPLEMENTATION

A Post-Implementation Review (PIR) is an assessment and review of the completed working solution. It will be performed after a period of live running; some time after the project is completed. There are three purposes for a Post-Implementation Review:

- To ascertain the degree of success from the project, in particular, the extent to which it met its objectives, delivered planned levels of benefit, and addressed the specific requirements as originally defined.
- To examine the efficiency of all elements of the working business solution to see if further improvements can be made to optimize the benefit delivered.
- To learn lessons from this project, lessons which can be used by the team members and by the organization to improve future project work and solutions.

6.1 System Evaluation

The system evaluation involves the hardware and software as a unit. The hardware selection is based on performance categories. The evaluation phase ranks vendor proposal and determines the one suited to the user's needs. It looks in to items such as price, availability and technical support.

In the operation phase, the system performance must be monitored not only to determine whether or not they perform as planned, but also to determine if they should be modified to meet changes in the information needs of the business.

In the evaluation phase, the first step adopted was to look at the criteria listed earlier and rank them in the order of importance. Three sources of information are used in evaluating hardware and software. They are benchmark program, experience of other users and product reference manuals.

6.2 Maintenance

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a

modified environment. Maintenance covers a wide range of activities, including correcting, coding and design errors, updating documentation and test data and upgrading user support. Maintenance means restoring something to its original condition.

After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance begin. The maintenance phase of the software cycle is the time in which a software product performs the useful work. If the new information is inconsistent with the design specification, then changes have to be made. The importance of maintenance is to continue to bring the new system to standards.

The system should be maintained and upgraded according to the technological advancements. It ensures the data integrity, data control and security. The system must be protected from fire and other natural calamities. The backup copies of data must be maintained daily so that we can prevent the loss of data due to various reasons.

Types of changes that can be encountered during the maintenance phase:

Corrective maintenance

Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct the defects.

Adaptive maintenance

Over time, the original environment (CPU, Operating System, Business Rules, External Product Characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

Enhancement maintenance

As software is used, the user will recognize additional functions that will provide the benefit. Perfect maintenance extends the software beyond its original functional requirements.

Preventive maintenance

Computer software deteriorates due to change, and because of this preventive maintenance often called software re-engineering, must be conducted to enable the software to serve the needs of its end users. Preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted and enhanced.

Activities of a Maintenance Procedure

Maintenance activities begin where conversion leaves off. Maintenance is handled by the same planning and control used in a formal system projects. The maintenance staff receives a request for service from an authorized user, followed by a definition of the required modifications.

The source program and written procedures for the system acquired from the programming library. Program changes are then tested and submitted to the user for approval. Once approved, the modified documentation is filled with the library and a project completion notice is sent to the user, signaling the termination of the project .Although software doesn't ware out like a piece of hardware, it ages and evenly fails to perform because of cumulative maintenance .A major problem with the software maintenance is its labor-intensive nature and therefore the likelihood of errors.

VII. CONCLUSION

It is vital step to understand the basics, i.e. the advantages, disadvantages, requirements and most importantly the feasibility of a biometric based security system. The implementation of ATM security system by using biometric method is a crucial procedure, as well as very challenging and difficult. But for security purposes and to have a control on the criminal records it is very important to bring this system in motion. Fingerprints have intrinsic features that do not change for whole life and are different individually. They are easy to use, cheap and provide the most suitable miniaturization. Biometrics is one of the most popular and effective means for identification/verification of an individual and is used as forensic evidence. It becomes important to take help of two technologies, namely embedded system and biometrics in order to provide enough security. Hybridization, along with the above two technologies is useful for fingerprint verification since it refers to the automated method of verifying/ identifying a match or similarities between two human fingerprints.

7.1 Scope for Future Enhancement

The system can be extended using a GSM module. The GSM module sends alert messages to the respective authorities when unauthorized person's finger print is detected. Improved feature enhancement and matching algorithm. Securing fingerprint based biometric system. Performance can be improved in terms of speed and memory. A speaking voice alarm can be used to indicate unauthorized person accessing the ATM. System can be made to communicate with modems or mobile phones.

VIII. BIBLIOGRAPHY

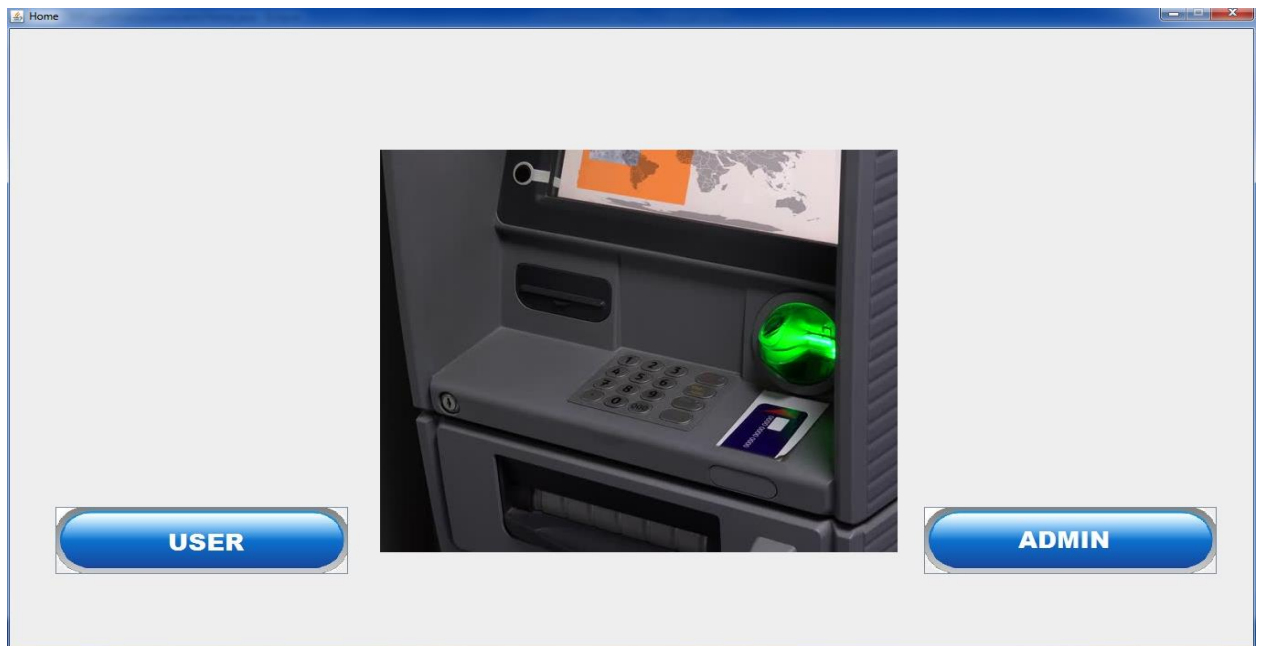
8.1 Books/Articles

1. G.Sambasiva Rao, C. NagaRaju, L. S. S. Reddy and E. V. Prasad, (2008), “A Novel Fingerprints Identification System Based on the Edge Detection”, International Journal of Computer Science and Network Security, vol. 8, pp. 394-397. New York City.
2. Robert Hastings, (2007), “Ridge Enhancement in Fingerprint Images Using Oriented Diffusion”, IEEE Computer Society on Digital Image Computing Techniques and Applications, pp. 245-252. Pune, India.
3. Jinwei Gu, Jie Zhou, and Chunyu Yang, (2006), “Fingerprint Recognition by Combining Global Structure and Local Cues”, IEEE Transactions on Image Processing, vol. 15, no. 7, pp. 1952 – 1964. Mumbai, India.
4. V. Vijaya Kumari and N. Suriyanarayanan, (2008) , “Performance Measure of Local Operators in Fingerprint Detection”, Academic Open Internet Journal, vol. 23, pp. 1-7. Mumbai University, India.
5. Raju Sonavane and B. S. Sawant, (2007), “Noisy Fingerprint Image Enhancement Technique for Image Analysis: A Structure Similarity Measure Approach”, Journal of Computer Science and Network Security, vol. 7 no. 9, pp. 225-230. Nagpur, India.

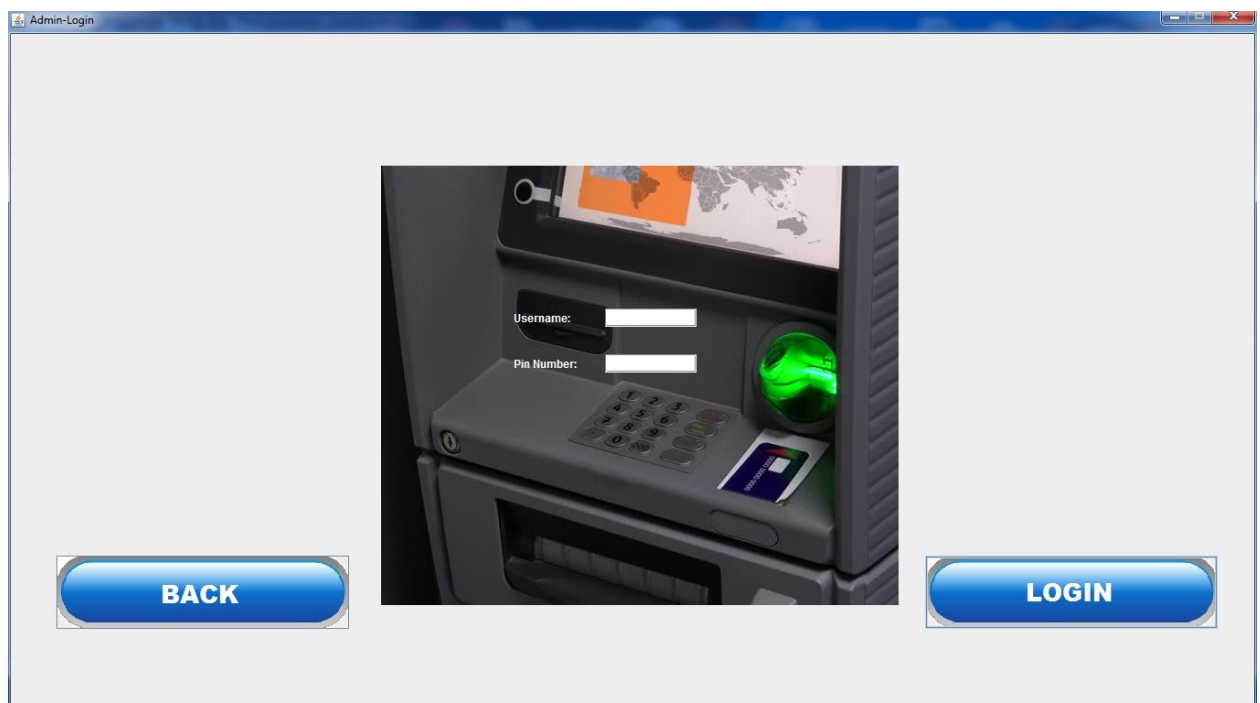
IX. ANNEXURE

9.1 Screenshots

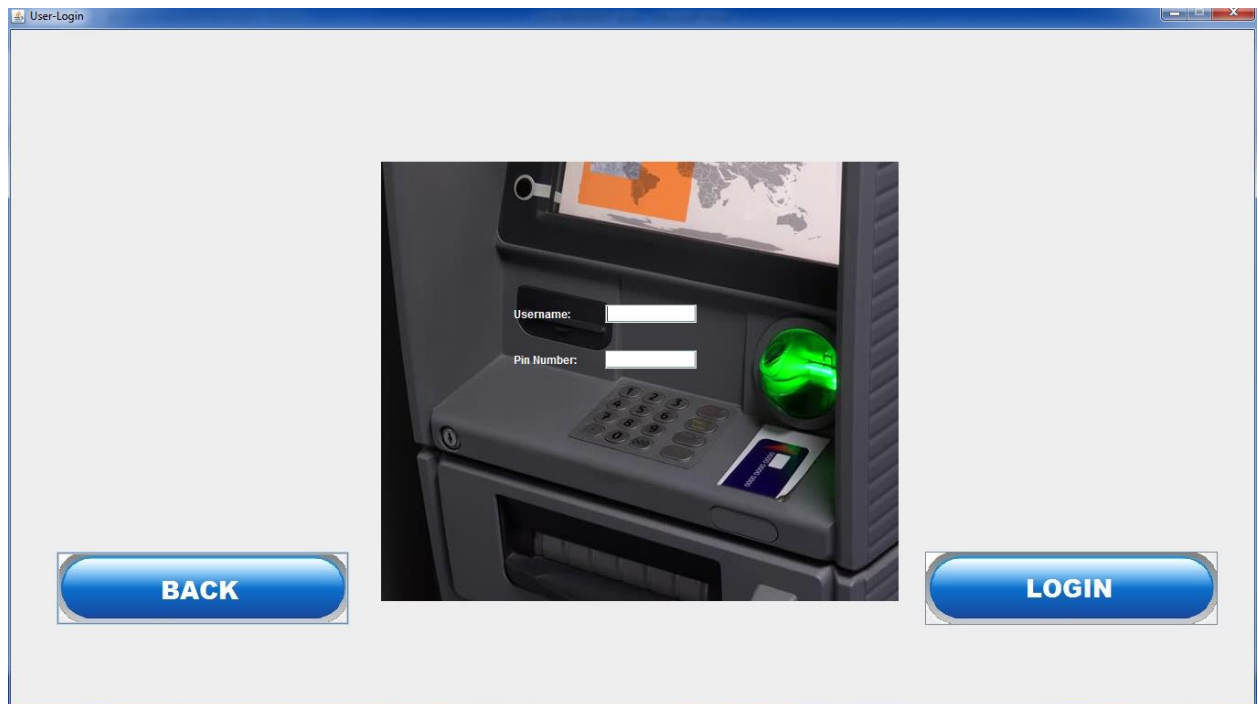
HOME PAGE



ADMIN LOGIN



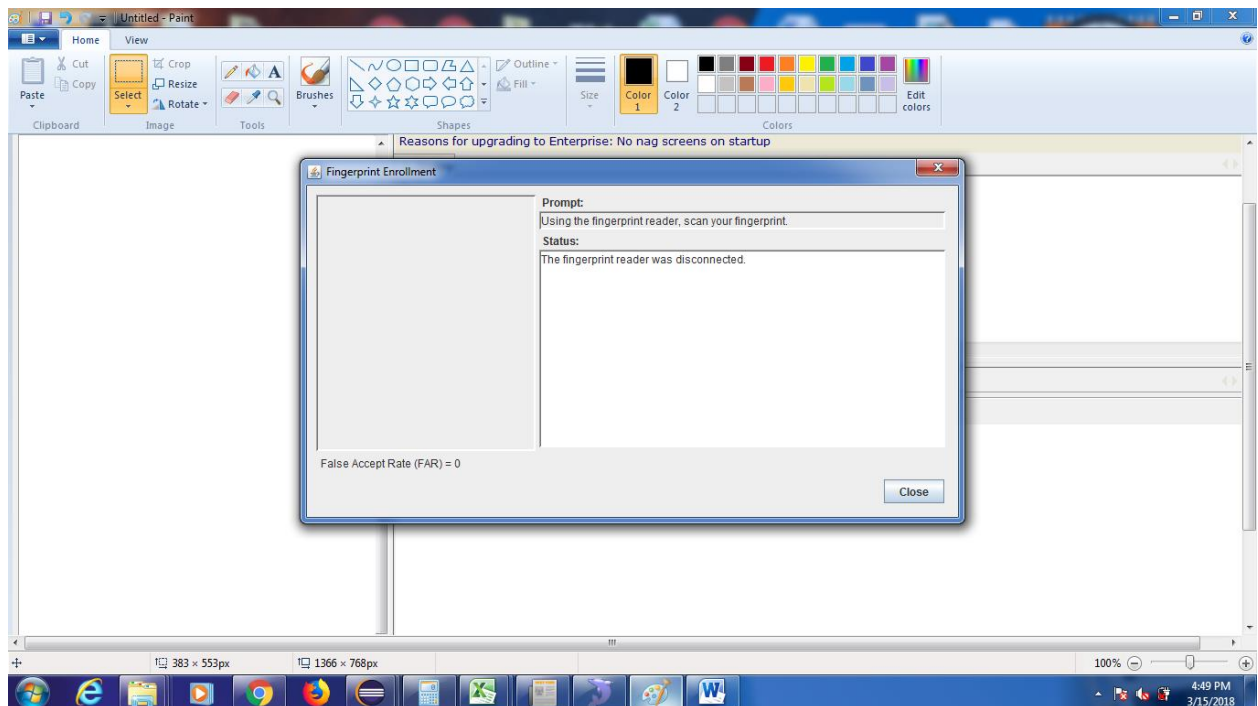
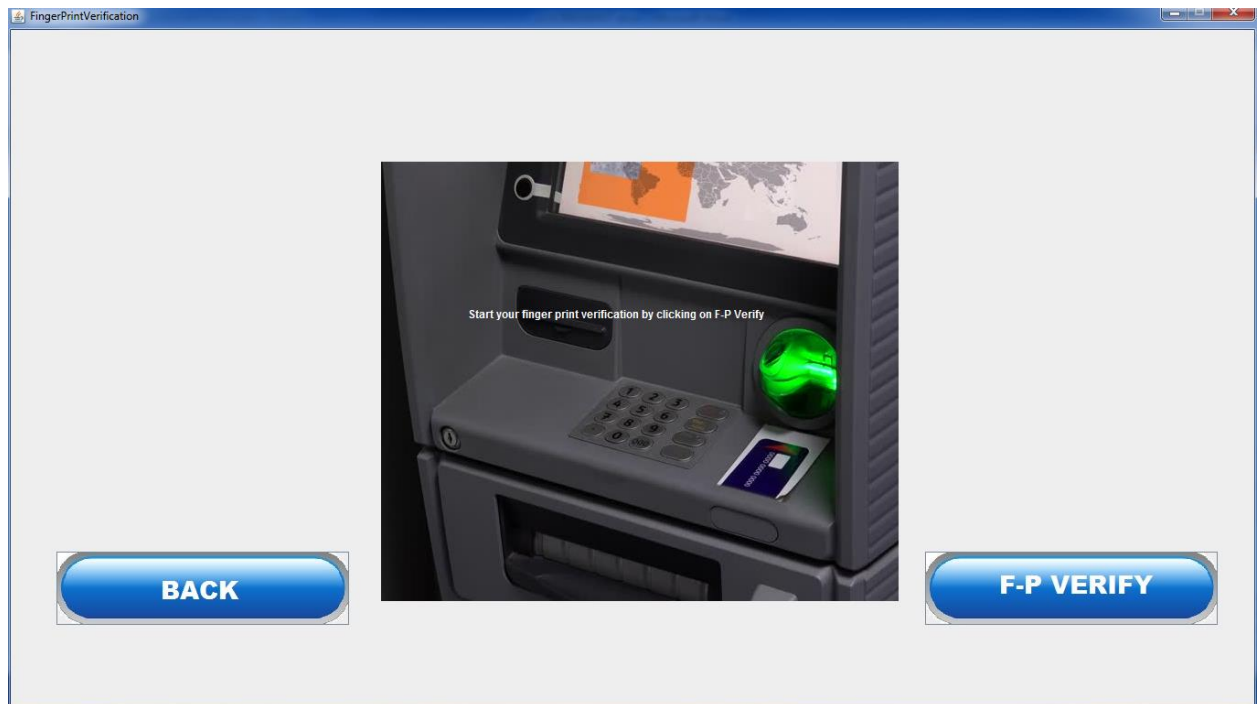
USER LOGIN



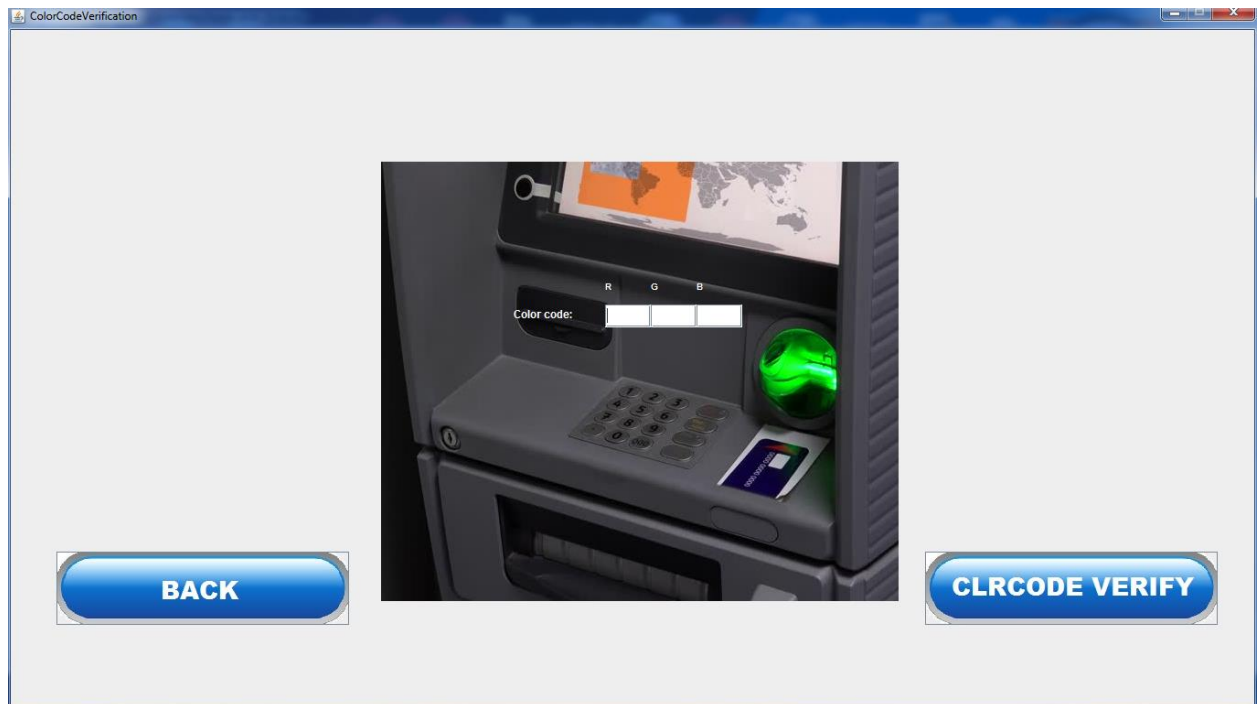
MOBILE VERIFICATION



FINGER PRINT AUTHENTICATION



COLOR CODE AUTHENTICATION



TRANSACTION

