1. INTRODUCTION

1.1 ATM MANAGEMENT SYSTEM

Millions of times per day around the globe people are instantly withdrawing money at automatic teller machines (ATMs). Given the fast-pace of the world today, it is not surprising that the demand for access to quick cash is so immense. The power of ATMs would not be possible without secure connections. The final act of ATM dispending cash is the result of an amazing fast burst of the customer never sees, but a trust is being done in a confidential manner.

ATM allows the customers of a bank to have access to their account without going to the bank. This is achieved by development of application like **ATM Management System.**

When the application is adopted, the user who uses this product will be able to see all the information and services provided by the ATM, when he enters the necessary pin. This project is used to Deposit and Withdraw cash also he can check the balance, the administrator can use this product to get the information of the customers and he can block and unblock the accounts when necessary. The data is stored in the database and is retrieved based on requirements. The implementation of this product needs hardware like an ATM machine to function the way its supposed to.

Development of this product includes various operations such as:

1. Home menu
2. Login/Register
3. Admin process
4. User Transaction
5. Send Email

This product has been designed in a way which asks user to enter pin to verify the account. As soon as verification is completed, either he will be logged in as the administrator or customer after which they will be provided their respective menu. For example, when user is logged in as customer, he will be provided options like deposit, withdraw or Balance enquiry as well as administrator will be provided admin menu.

1.2 OBJECTIVES

The objective of the project "ATM Management System" is to develop Automatic Teller System of the wide variety of electronic items online.

This project is mainly developed to help customers do their bank transactions under one roof easily without any trouble. Customers can visit any nearest ATM to do transactions, according to their convenience.

Millions of times per day around the globe people are instantly withdrawing money at automatic teller machines (ATMs). Given the fast-pace of the world today, it is not surprising that the demand for access to quick cash is so immense. The power of ATMs would not be possible without secure connections. The final act of ATM dispending cash is the result of an amazing fast burst of the customer never sees, but a trust is being done in a confidential manner. There are several objectives of this application are following given below.

- This application gives information of account to the customer to provide better service.
- It provides the facility to customers who want to enjoy services of bank near them.
- To reduction of fraudulent activities
- To render accurate service to customer
- To achieve speedy processing of customer data
- To reduce error processing, the guarantee of increased security

2. SYSTEM ANALYSIS

2.1 Existing System:

In existing system, the bank staffs have to maintain lot of records manually regarding the transactions of customers. Details of the customer cannot be segregated according to the user needs

Disadvantages of existing system:

- Time Consuming
- Lot of paperwork
- Prone to errors

2.2 Proposed System:

The proposed system differs from the existing system by clubbing into a single form reducing the existing so as to introduce various other properties in the software and making it easier to fill the details. And this software will be helpful in the smooth functioning of the organization due to integration of various functions, the software maintains the central database with the following information stored in different tables: customer's details, transaction details, loan details etc.

Advantages

- Less effort to complete transaction
- Less time required to complete transactions compared to existing system
- No need to maintain bulk of books / papers
- Less or no errors
- Easy to retrieve data

MODULES USED IN THE PROJECT

1. ADMIN

- Log In: Admin can login to the application by providing the valid credentials to access the application.
- Approve Accounts: Admin will approve the account requests by new customers.
- Approve Loans: Admin will approve the loans requested by existing customers.
- Update Accounts: Admin can update account details.
- Delete Accounts: Admin can delete account details.
- Block/Unblock Accounts: Admin can block/unblock accounts when required.
- View Users: Admin can view user details at any time.
- Logout: Admin can logout from application to end the session

2. Customer

- Register: customers have to first register their account to the application using card number.
- Log in: customers can login to application by providing valid card no. and password.
- Deposit: Customers can deposit money to their account.
- Withdraw: Customers can withdraw money from their account.
- Transfer: Customers can transfer money from their account to another account.
- Loan: Customers can get loans to their account or repay loans from their accounts.
- Change pin: Customers can change pin at any point of time.
- View Statement: Customers can view their transaction statements.
- Log out: Customers can logout of the application.

3. HARDWARE AND SOFTWARE REQUIREMENTS

3.1 HARDWARE INTERFACES

Processor	:	Intel i3 +
RAM	:	4GB
Hard Disk	:	80GB
Speed	:	1.2 GHz+

3.2 SOFTWARE INTERFACES

Operating System	:	Windows 7 or Higher
IDE	:	Visual Studio 2019
Language	:	Visual Basic
Framework	:	VB.NET 4.7.2
Back End	:	MY SQL

4. SOFTWARE REQUIREMENT AND SPECIFICATIONS

4.1 INTRODUCTION

Software requirement specification is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use case that describes interactions the users will have with the software. The purpose of this SRS document is to collect and analyse all assorted ideas that have come up to define the system. Its requirements with respect to consumers and also to provide a detailed overview of our software product, its parameters and goals. SRS provides an overview of the entire system with purpose, scope, definitions, acronyms, abbreviations, references. The aim of this document is to gather and analyse and give an in-depth insight of the system by defining the problem statement in detail. Also, we shall predict and sort out how this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

REASONS FOR CHOOSING .NET

Limitations of C:

- C is structured programming language.
- Does not provide efficient means of garbage collection.
- Pointers can be easily misused.
- Programmers should have strong programming knowledge.

Limitations of C++:

- Does not provide very strong type-checking.
- C++ can be thought as an Object-Oriented layer on top of C.
- It involves manual memory management.
- No built-in support for threads.
- Portability of code on another platforms.

Limitations of Java:

- Java Programmers must use Java front to back during development cycle
- It is not appropriate for many graphical or numerical intensive application.
- .NET provides solution to all the above-mentioned problems.

4.2 .NET FRAMEWORK

The .NET framework is an integral windows component that supports building and running the next generation of applications and XML web services. The key component of the .NET frame work are the common language run time and the .NET frame work class library, which includes ADO.NET, ASP.NET and windows forms. The .NET framework provides a managed execution environment simplified development and deployment and integration with a wide variety of programming languages. This framework is made up of the following parts:

- The common language runtime (CLR).
- The base class libraries.
- Object oriented internet development with ASP.NET.
- Rich client user interface using windows forms.
- RAD for the internet using web forms.

OVERVIEW OF .NET FRAMEWORK

The .NET framework is a new computing platform that simplifies application development in the highly distributed environment of the internet. The .NET framework is designed to fulfil following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally but internet- distributed or executed remotely.

- To provide a code execution environment that minimizes software deployment and versioning conflicts.
- To provide a code execution environment that guarantees safe execution of code, including code created by an unknown or semi trusted third party.
- To provide a code execution environment that eliminates the performance problem of scripted or interpreted environments.
- To make the developer experience consistent across widely types of application, such as windows-based applications and web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET framework can integrate with any other code.

The .NET framework has two main components: the common language runtime and the .Net framework class library. The common language runtime is the foundation of the .NET framework. You can think of the runtime as an agent that manages code at execution time, and removing while also enforcing strict type safely and other forms of code accuracy that ensure security and robustness in fact the concept of code management is a fundamental principle of the runtime.

Code that targets the runtime is known as managed code, while code that does not target the runtime is known as un managed code. The class library, the other main component of the .NET frameworks is a comprehensive, object-oriented collection reusable types that you can use to develop applications ranging from traditional command line or graphical user interface (FGUI) applications to application base d on the latest innovations provided by ASP.NET, such as web forms and XML web services.

The .NET framework can be hosted by unmanaged component that load the common language runtime into their processes and initiate the execution of managed code. ASP.NET works directly with the runtime to enable ASP.NET application and XML

web services, both of which are discussed later in this topic, Internet explorer is an example of unmanaged application that hosts the runtime.

Using internet explorer to the host runtime enables you to embed managed components or windows forms controls in HTML documents. Hosting the runtime in this way makes mobile code 9similar to Microsoft Active Xr controls) possible, but with significant improvement that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your application and to the overall system. The illustration also shows how managed code operated with in a larger architecture.

We can use the .NET framework to develop the following types of application and services:

- Console applications
- Window GUI application (Windows Forms) ASP.NET applications
- XML Web services
- Windows services

COMMON LANGUAGE RUNTIME (CLR)

The common language runtime (CLR) is responsible for runt-time services such as language integration; security enforcement; and memory, process and thread management. In addition, it has a roll at development time when features such as life cycle management strong type naming, cross-language exception handling, dynamic binding and so on, reduce the amount of code that a developer must write to turn the business logic the reusable component. The runtime can be hosted by high performance, server-side applications, such a s Microsoft Internet Information Services (IIS) for building web applications with ASP.NE and the next release of Microsoft SQL Server.

This infrastructure enables you to use code "managed "by the .NET framework to write your business logic, while still enjoying the superior performance of the industry's best enterprises servers that support runtime hosting.

4.3 VB .NET

VB.NET stands for Visual Basic.NET, and it is a computer programming language developed by Microsoft. It was first released in 2002 to replace Visual Basic 6. VB.NET is an object-oriented programming language. This means that it supports the features of object-oriented programming which include encapsulation, polymorphism, abstraction, and inheritance.

Visual Basic .NET runs on the .NET framework, which means that it has full access to the .NET libraries. It is a very productive tool for rapid creation of a wide range of Web, Windows, Office, and Mobile applications that have been built on the .NET framework.

ADO .NET

ADO.NET provides consistent access to data sources such a Microsoft SQL Server and XML, as well as to data sources exposed through OLE DB and ODBC. Data sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate and update the data that they contain. ADO.NET separates data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET Frame work data providers for connecting to a database, executing commands and retrieving results. Those results are either processed directly, placed in and ADO.NET Dataset objects in order to be exposed to the used in an ad hoc manner, combined with data from multiple sources or remote between tiers. The ADO.NET Dataset object can also be used independently of a .NET Framework data provider to manage data local to the application or sourced from XML.

The ADO.NET classes are found in System.Data.dll and are integrated with the XML classes found in System.Xml.dll. When compiling code that uses the System. Data, namespace reference both System.Data.dll and System.Xml.dll. ADO.NET provided functionality to developers writing managed code similar to the functionality provided to native component object model (COM) developers by ActiveX Data Objects (ADO).

ADO .NET COMPONENTS

There are two components of ADO.NET that you can use to access and manipulate data:

>>NET Framework data providers.

>> The Dataset

.NET FRAMEWORKS DATA PROVIDERS

The .NET Framework Data providers are components that have been explicitly designed for data manipulation and fast, forward-only, read-only access to data. The connection object provides connectivity to a data source. The command object enables access to database commands to return data, modify data, run stored procedures and send or retrieve parameter information. The Data Adapter provides a high-performance stream of data from the data source. Finally, the Data Adapter provides the bridge between the Dataset object and the data source.

THE DATASET

The ADO.NET Dataset is explicitly designed for data access independent of any data source. As a result, it can be used with multiple and differing data sources used with XML data or used to manage data local to the application. It contains a collection of one or more Data Table objects made up to rows and

columns of data as well as primary key, foreign key, constraint and relation information about the data in the Data Table objects.

BENEFITS OF ADO .NET

ADO.NET offers several advantages over previous versions of ADO and over other data access components. These benefits fall into the following categories:

1. Interoperability & Maintainability
2. Programmability & Scalability

MICROSOFT DATA ACCESS COMPONENTS (MDAC)

Microsoft Data Access Components (MDAC) is a collection of core files provided to help applications by providing a means of accessing data. MDAC includes core files for Open Database Connectivity (ODBC), ActiveX Data Objects (ADO), OLEDB, Network libraries and client configuration tool for SQL Server. Depending on your data access strategy, you may need to ensure that MDAC is installed on the client computers, the business servers, the Web servers or the database servers. MDAC 2.6 or later is required by the .NET Framework and at least MDAC 2.7 SP1 is recommended

4.4 MY SQL

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, MySQL. These systems allow users to create update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. MySQL stores each data item in its own fields. In MySQL, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an

occurrence). Each record is made up of number of fields. No two fields in a record can have the same field name.

During a MySQL Database design project, the analysis of your business needs to identify all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

MySQL Tables

MySQL stores records relating to each other in a table. Different tables are created for the various groups of information and are grouped together to form a database.

4.5 Functional Requirements

Functional Requirements defines a function of software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality.

Admin

In this module, admin has to login with valid card no. and pin. After login successful he can do some operations such as approve accounts, approve loans, update, block/unblock cards, and view all customer details.

Customer

In this module, there are n number of users are present. User should register before doing some operations and also set graphical authentication points while registration. After registration successful he can login by using valid card no. and pin. After Login successful he will do some operations like deposit, withdraw, transfer, get/return loans, change pin, view statement etc.

4.6 Non-Functional Requirements

Non-Functional requirements, as the name suggests, are those requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability response time and store occupancy. Alternatively, they may define constraints on the system such as the capability of the Input Output devices and the data representations used in system interfaces. Many non-functional requirements relate to the system as whole rather than to individual system features. This means they are often critical than the individual functional requirements. The following non-functional requirements are worthy of attention.

The key non-functional requirements are:

- **Security:** The system should allow secure communication between cloud server & user.
- **Platform Independence:** The application should run on any platform without recompilation.
- **Reliability:** The system should be reliable and must not degrade the performance of existing system and should not lead to the hanging of the system.
- **Response time:** The application response time should be quick.
- **Scalability:** The system should provide optional performance even if the user base grows dramatically.
- **Maintainability:** The system should support incorporation of future requirements easily.
- **Robustness:** The application should be fault tolerant with respect to illegal user/receiver inputs. Error checking has been built in the system to prevent system failure.

4.7 Feasibility Study

Feasibility study means the analysis of problem to determine if It can be solved effectively. In other words, it is the study of the possibilities of the proposed system it studies the work ability, impact on the organization ability to meet user's need and efficient use of resources.

Three aspects in which the system has to be feasible are: -

ECONOMICAL FEASIBILITY:

The economic analysis checks for the high investment incurred on the system. It evaluates development &Implementing charges for the proposed "Net Banking Project". The S/W used for the development is easily available at minimal cost & the database applied is freely available hence it results in low-cost implementation.

TECHNICAL FEASIBILITY:

This aspect concentrates on the concept of using Computer Meaning, "Mechanization" of human works. Thus, the automated solution leads to the need for a technical feasibility study. The focus on the platform used database management &users for that S/W. The proposed system doesn't require an in-depth technical knowledge as the system development is simple and easy to understand. The S/W used makes the system user friendly (GUI). The result obtain should be true in the real time conditions.

BEHAVIOURAL FEASIBILITY:

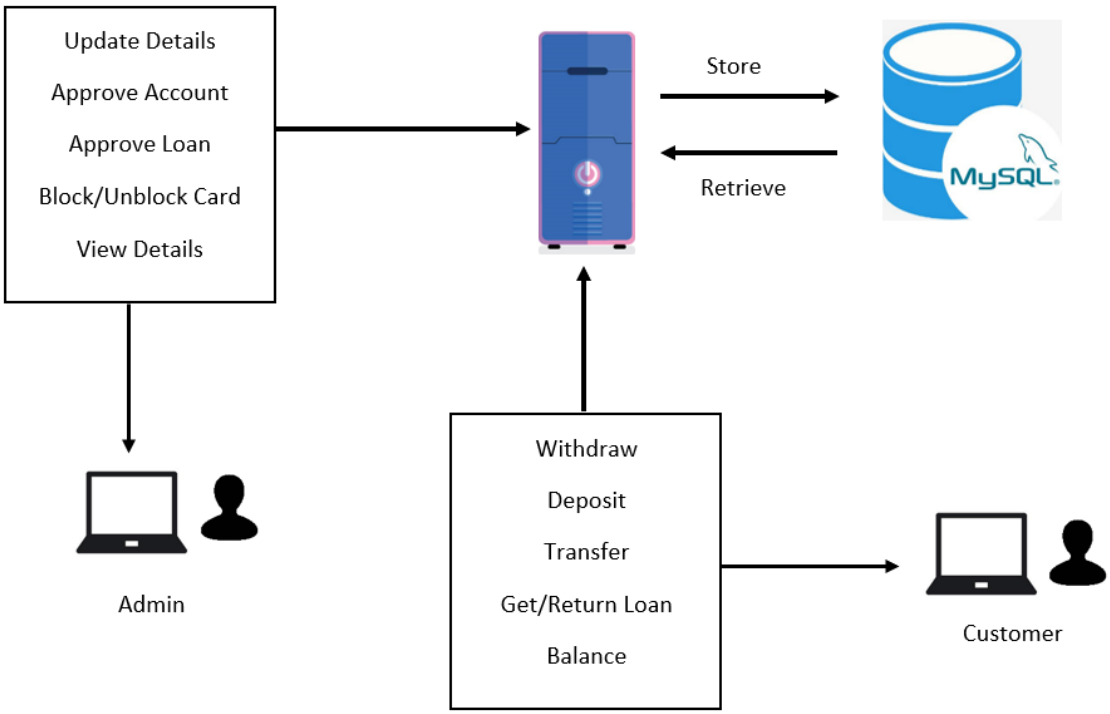
Behavioural feasibility deals with the runtime performance of the S/W the proposed system must score higher than the present in the behavioural study. The S/W should have end user in mind when the system is designed while designing s/w the programmer should be aware of the conditions user's Knowledge input, output, calculations etc. The s/w contains only a minimum no. of bugs. Care should be also taken to avoid non-working means &t buttons.

5. SYSTEM DESIGN

5.1 INTRODUCTION

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution do aaaaaa main. In other words, starting with what is needed; design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases particularly testing and maintenance.

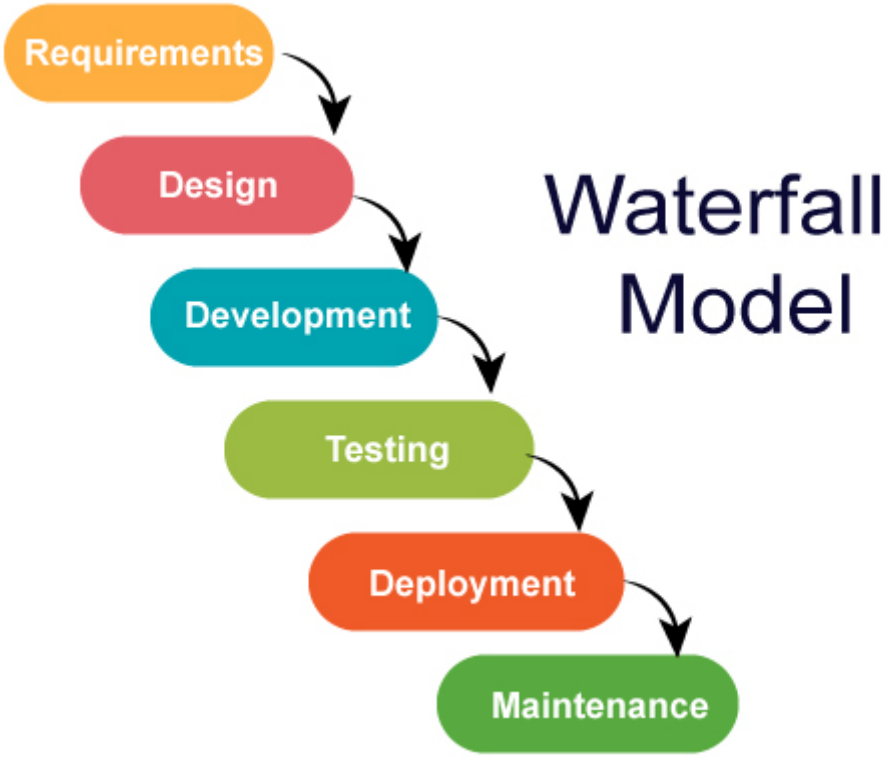
5.1.1 ARCHITECTURE DESIGN



SOFTWARE MODEL USED

5.1.2 WATERFALL MODEL

The **Waterfall model** is a sequential (non-iterative) design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance. Despite the development of new software development process models, the Waterfall method is still the dominant process model with over a third of software developers still using it.



The waterfall development model originates in the manufacturing and construction industries: highly structured physical environments in which after-the-fact changes are prohibitively costly, if not impossible. Because no formal software development methodologies existed at the time, this hardware-oriented model was simply adapted for software development.

Advantages of waterfall model:

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

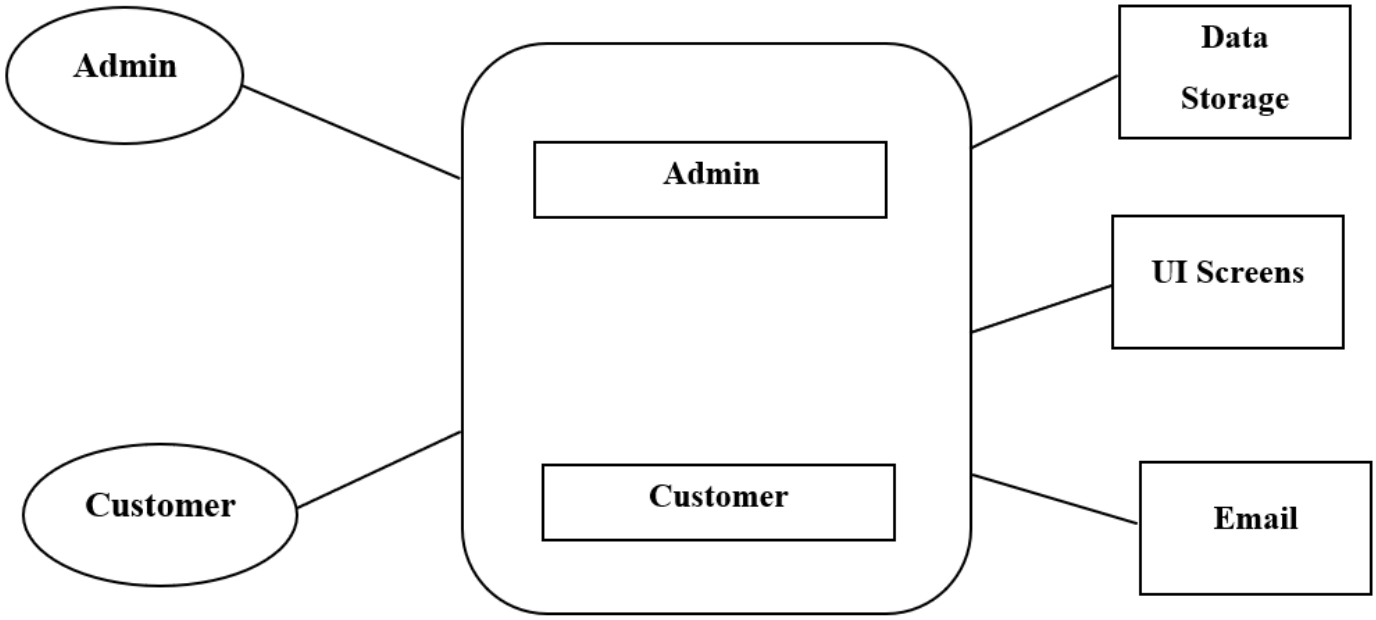
Disadvantages of waterfall model:

- Once an application is in the **testing** stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

5.2 DATAFLOW DIAGRAM

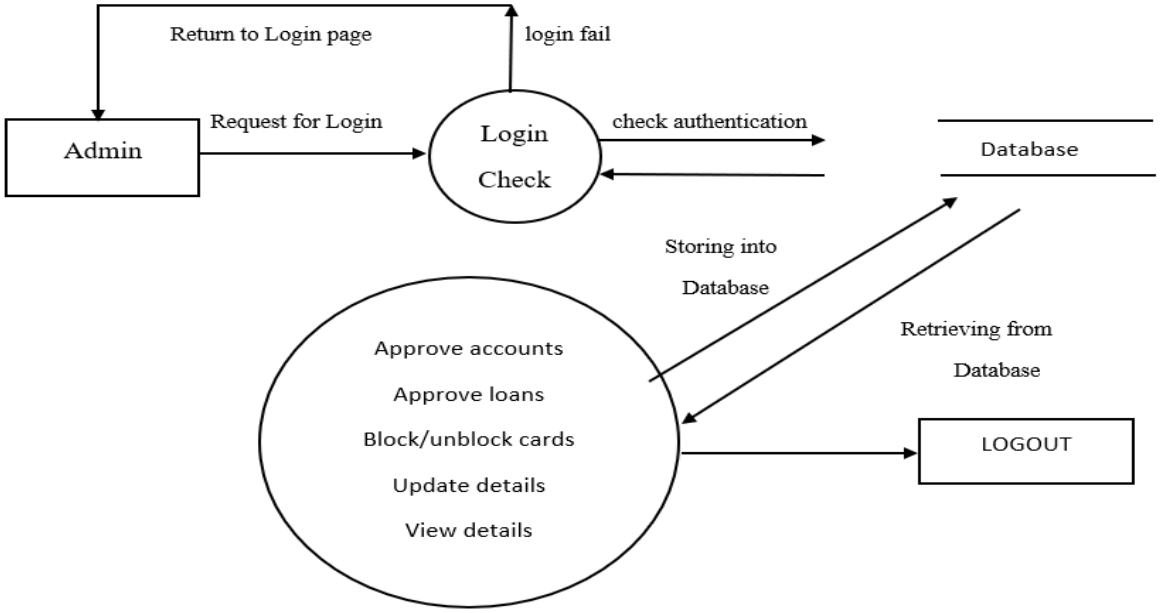
Level 0 DFD: It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

5.2.1 LEVEL 0 DFD

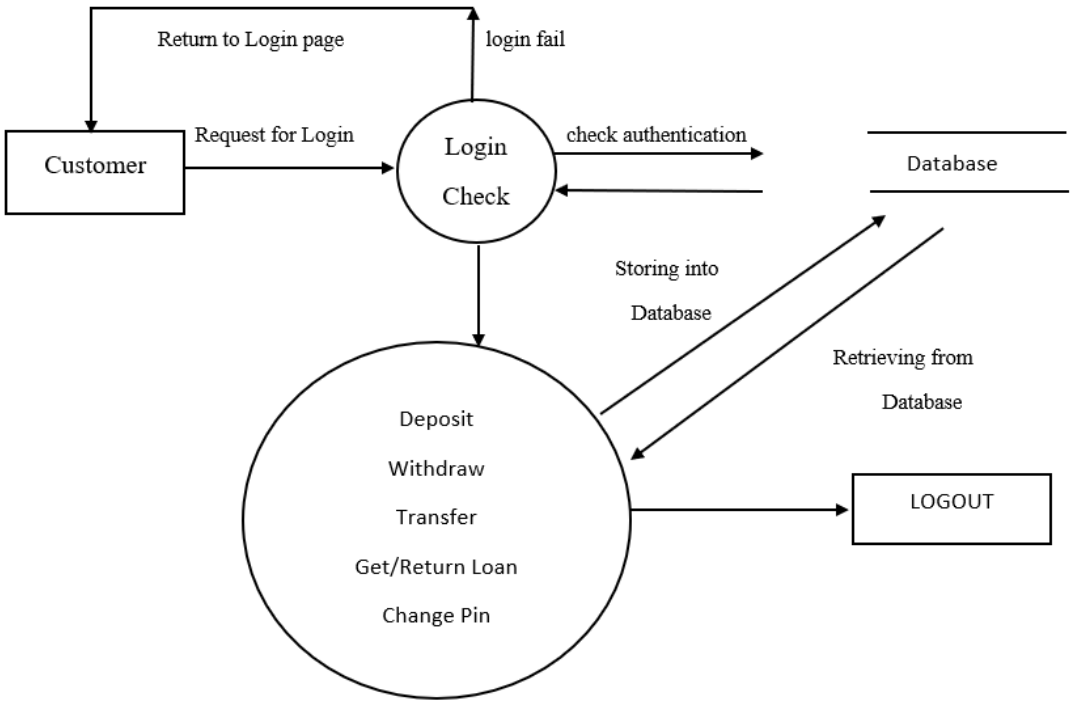


Level 1 DFD: This level shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major and high-level processes of the system and their interrelation.

5.2.2 LEVEL 1 DFD FOR ADMIN



5.2.3 LEVEL 1 DFD FOR CUSTOMER



5.3 USE CASE DIAGRAM

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users. **Use case diagrams** are usually referred to as behaviour diagrams used to describe a set of actions (use cases) that some system or 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.

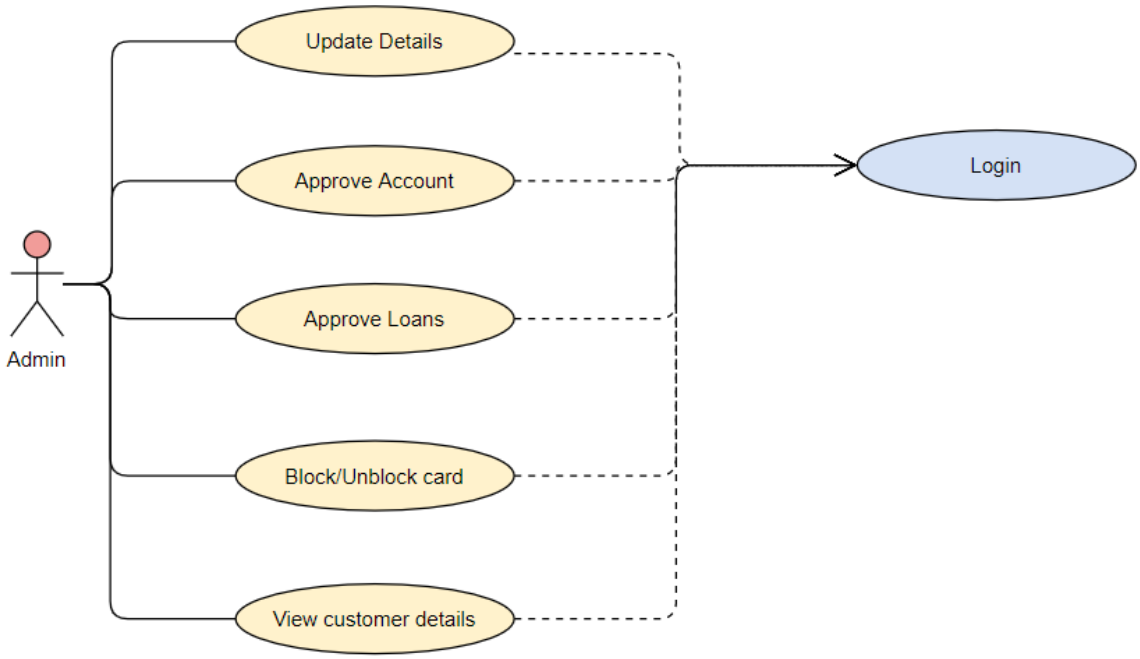
Use cases: A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

Actors: An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.

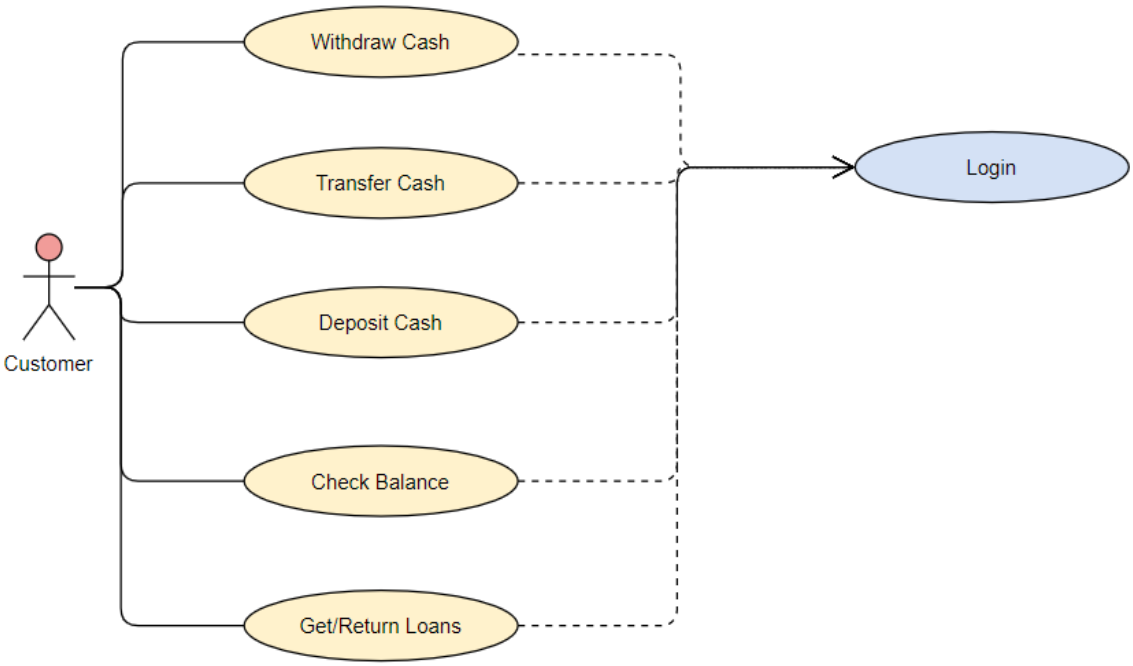
Associations: Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case.

System boundary boxes: You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system. Anything within the box represents functionality that is in scope.

5.3.1 USE CASE DIAGRAM FOR ADMIN:



5.3.2 USE CASE DIAGRAM FOR CUSTOMER:

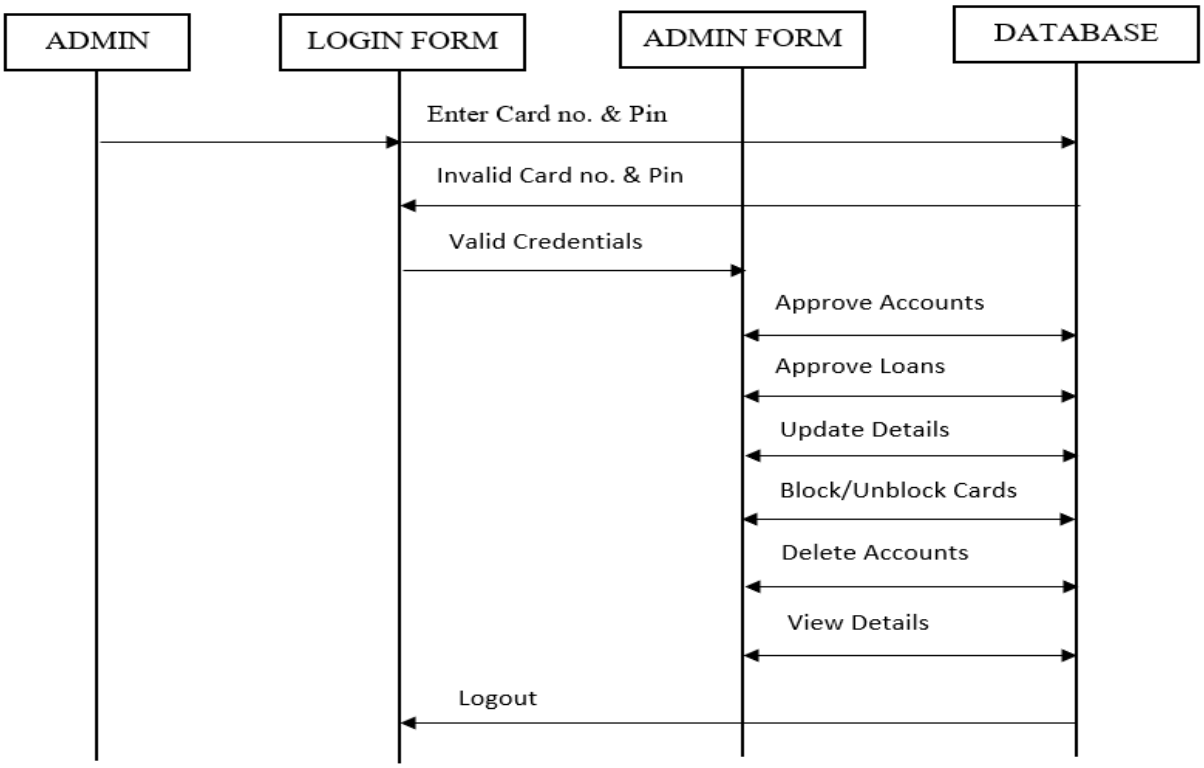


5.4 SEQUENCE DIAGRAM

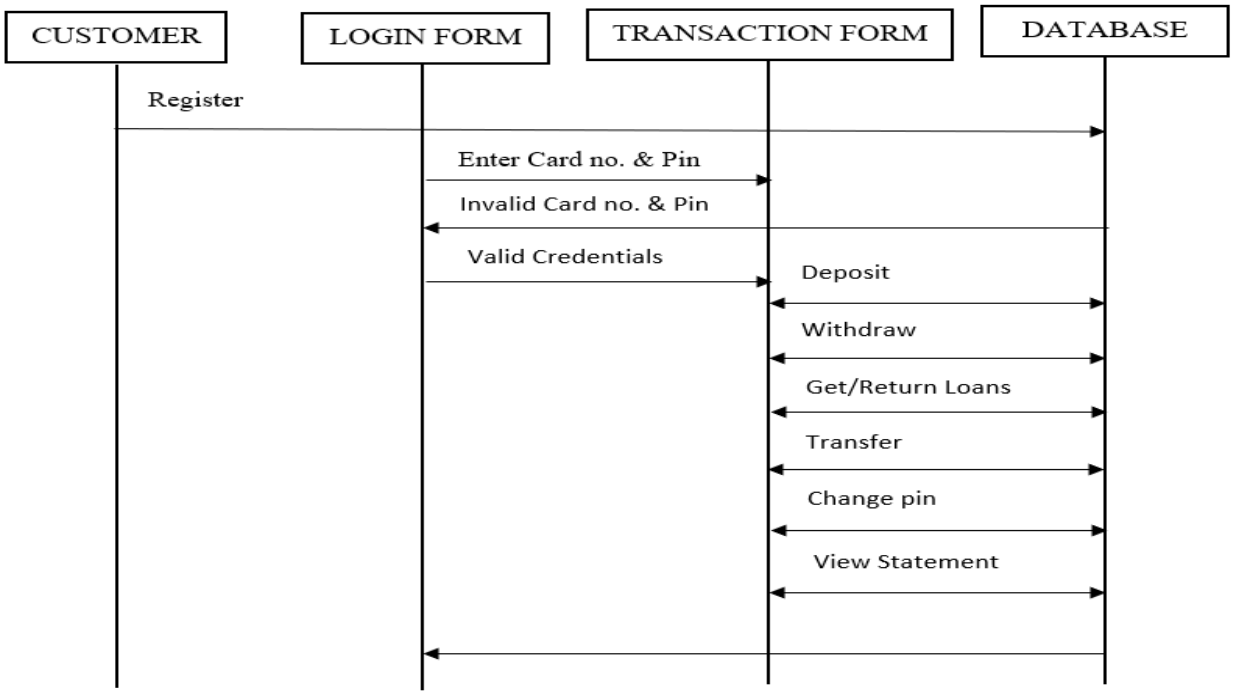
The Sequence Diagram models the collaboration of objects based on time. sequence. It shows how the objects interact with each other's in a particular scenario of a use case. With the advanced visual modelling capability, you can create complex sequence diagram in a few clicks. Besides, Visual paradigm can generate sequence diagram from the flow of events which you have defined in the use case description. The sequence diagram models the collaboration of objects based on a time sequence. It shows how the objects interacts with others in a particular scenario of a use case. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

- **Lifelines:** A sequence diagram shows, as parallel vertical lines which indicates different processes or objects that live simultaneously.
- **Message:** Messages written with horizontal arrows with the message name written above them, display interaction. The messages are written in order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.
- **Object/Activation Box/Process:** Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message.

5.4.1 Sequence Diagram for Admin



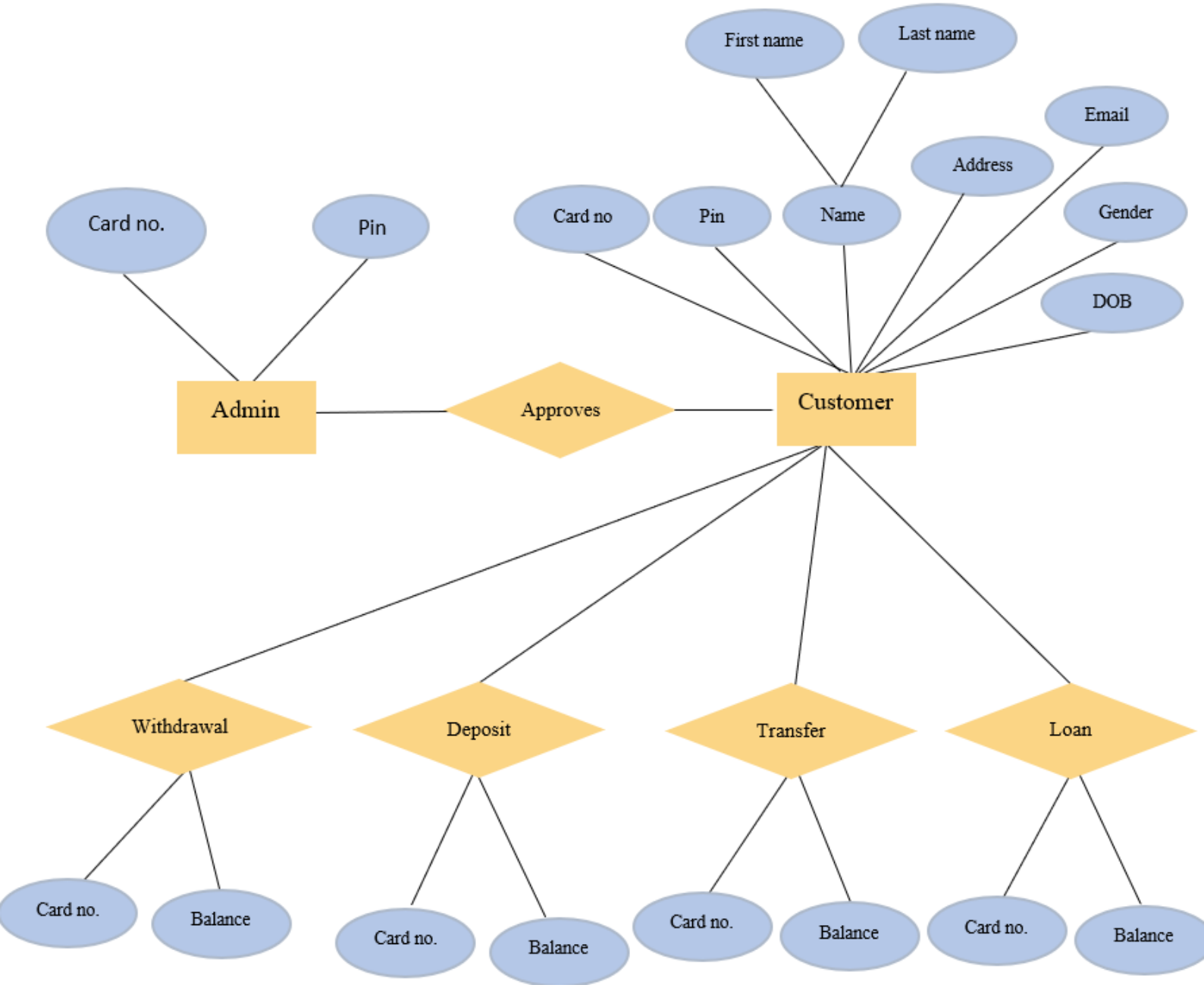
5.4.2 Sequence Diagram for Customer



5.5 E-R DIAGRAM

Entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them, and databases. ER Diagram is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored. A relationship is how the data is shared between entities. There are three types of relationships between entities:

- **One-to-One:** One instance of an entity is associated with one other instance of another entity.
- **One-to-Many:** One instance of an entity is associated with zero, one or many instances of another entity, but for one instance of entity B there is only one instance of entity A.
- **Many-to-Many:** One instance of an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity B is associated with one, zero or many instances of entity A.


First name
Last name
Email
Address
Gender
DOB
Name
Pin
Card no
Card no.
Pin
Admin
Approves
Customer
Withdrawal
Deposit
Transfer
Loan
Card no.
Balance
Card no.
Balance
Card no.
Balance
Card no.
Balance

6. IMPLEMENTATION

INTRODUCTION

Features of object-oriented paradigm:

This web application is implemented using object-oriented programming language. Object oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data structures are designed such that they characterize the objects.
- Methods that operate on the data of an object are tied together in the data structure.
- Objects may communicate with each other through methods.
- New data and methods can be easily added whenever necessary.
- Follows bottom-up approach in program design.
- Data is hidden and cannot be accessed by external functions.
- This project is implemented using three tier architecture. VB.NET is used in the presentation layer, VB classes are used in the Business logic, Table adopter is used in the data tier and MY SQL (database) is used as the backend.

6.1MODULES USED IN THE PROJECT

1. **ADMIN**
    - Log In: Admin can login to the application by providing the valid credentials to access the application.
    - Approve Accounts: Admin will approve the account requests by new customers.
    - Approve Loans: Admin will approve the loans requested by existing customers.
    - Update Accounts: Admin can update account details.
    - Delete Accounts: Admin can delete account details.
    - Block/Unblock Accounts: Admin can block/unblock accounts when required.
    - View Users: Admin can view user details at any time.
    - Logout: Admin can logout from application to end the session

2. **CUSTOMER**
    - Register: customers have to first register their account to the application using card number.
    - Log in: customers can login to application by providing valid card no. and password.
    - Deposit: Customers can deposit money to their account.
    - Withdraw: Customers can withdraw money from their account.
    - Transfer: Customers can transfer money from their account to another account.
    - Loan: Customers can get loans to their account or repay loans from their accounts.
    - Change pin: Customers can change pin at any point of time.
    - View Statement: Customers can view their transaction statements.
    - Log out: Customers can logout of the application.

6.2 DATABASE TABLES

Account details table:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
account_no	INT	☑	☑	☐	☐	☐	☐	☐	☐	
Firstname	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Lastname	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Address	LONGTEXT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Contact_no	VARCHAR(50)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Gender	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Birthday	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
pin	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
type	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
balance	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Status	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
custIMG	LONGBLOB	☐	☐	☐	☐	☐	☐	☐	☐	NULL
email	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
pan_no	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
approve	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
accType	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
income	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

Deposit Statement Table:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
acc_no	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
dep_date	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
dep_amt	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
id	INT	☑	☑	☐	☐	☐	☐	☑	☐	

Transfer Statement Table:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
acc_no	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
t_date	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
to_account	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
amt_transferred	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
id	INT	☑	☑	☐	☐	☐	☐	☑	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Withdraw Statement Table:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
acc_no	INT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
w_date	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
w_amt	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
id	INT	☑	☑	☐	☐	☐	☐	☑	☐	
		☐	☐	☐	☐	☐	☐	☐	☐	

Loan Table:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
account_no	INT	☑	☑	☐	☐	☐	☐	☐	☐	
name	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
date_taken	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
total_amt	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
status	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
email	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
months_paid	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
emi_amount	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
total_months	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
due_date	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
approve	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL

User Ratings Table:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Id	INT	☑	☑	☐	☐	☐	☐	☑	☐	
Stars	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
Suggestions	LONGTEXT	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

7. SOURCE CODE

Main Menu:

```
Imports FontAwesome.Sharp
Public Class MainMenu

 Private Sub IconButton1_Click(sender As Object, e As EventArgs) Handles IconButton1.Click
Me.hide()
            Registerfrm.Show()
   End Sub

   Private Sub IconButton2_Click(sender As Object, e As EventArgs) Handles IconButton2.Click
      Me.Hide()
      login_frm.Show()
   End Sub

   Private Sub IconButton3_Click(sender As Object, e As EventArgs) Handles IconButton3.Click
      Me.Hide()
      services.Show()
   End Sub

Private Sub IconButton4_Click(sender As Object, e As EventArgs) Handles IconButton4.Click
      Me.Hide()
      contact.Show()
   End Sub

   Private Sub IconButton5_Click(sender As Object, e As EventArgs) Handles IconButton5.Click
      Me.Hide()
      AboutUs.Show()
   End Sub

   Private Sub MainMenu_Load(sender As Object, e As EventArgs) Handles MyBase.Load
      Me.Refresh()
      lbldate.Text = day
      lbltime.Text = TimeOfDay.ToString("h:mm tt")
      progbar.Hide()
      loadlbl.Hide()
End Sub

   Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
      progbar.Increment(5)
      proglbl.Text = progbar.ProgressPercentText
```

```
    End Sub

    Private Sub btnKannada_Click(sender As Object, e As EventArgs) Handles
btnKannada.Click
 selectedkannada(
MsgBox("ಕನ್ನಡ ಭಾಷೆಯನ್ನು ಆಯ್ಕೆ ಮಾಡಲಾಗಿದೆ", MsgBoxStyle.Information,
"ಕನ್ನಡ")
    End Sub

    Private Sub hindiBtn_Click(sender As Object, e As EventArgs) Handles
hindiBtn.Click
        selectedhindi()
        MsgBox("हिंदी भाषा का चयन किया गया है", MsgBoxStyle.Information, "हिंदी")

    End Sub
End Class
```

Registration Form:

```
Imports MySql.Data.MySqlClient
Imports System.IO
Imports System.Text.RegularExpressions
Imports System.Net
Imports System.Net.Mail

Public Class Registerfrm
    Dim mysqlconn As New MySqlConnection = New
MySqlConnection("Server=localhost;user=root;password=punith123;database=atm")
    Dim command As New MySqlCommand
    Dim gender As String
    Dim income As String

    Private Sub Signup_Click(sender As Object, e As EventArgs) Handles Signup.Click
        If txtAcc.Text = "" Or txtPin.Text = "" Or txtFname.Text = "" Or txtLname.Text
= "" Or txtEmail.Text = "" Or txtAddr.Text = "" Or txtContact.Text = "" Or
txtPancard.Text = "" Or gender = "" Or cmbTypeacc.Text = "Account type" Or
cmbIncome.Text = "Select yearly income" Then
            MsgBox("One or more fileds are empty", MsgBoxStyle.Critical, "Empty
fields")
        Else
            Dim regex As Regex = New Regex("([A-Z]){5}([0-9]){4}([A-Z]){1}$")
            If Not regex.IsMatch(txtPancard.Text.Trim()) Then
                MsgBox("Incorrect Pan number", MsgBoxStyle.Information, "Pan number
error")
            Else
                Try
     Dim query As String
```

```
 = "insert into atm.tblinfo
(account_no,FirstName,LastName,Address,Contact_no,Gender,Birthday,Pin,type,bal
ance,custIMG,Status,email,pan_no,approve,accType,income) values
(@account_no,@FirstName,@LastName,@Address,@Contact_no,@Gender,@Birth
day,@Pin,@type,@balance,@custIMG,@Status,@email,@pan_no,@approve,@acc
Type,@income)"
                Dim ms As New MemoryStream
                PictureBox1.Image.Save(ms, PictureBox1.Image.RawFormat)
                mysqlconn.Open()
                command = New MySqlCommand(query, mysqlconn)
                command.Parameters.AddWithValue("@account_no", txtAcc.Text)
                command.Parameters.AddWithValue("@FirstName", txtFname.Text)
                command.Parameters.AddWithValue("@LastName", txtLname.Text)
                command.Parameters.AddWithValue("@Address", txtAddr.Text)
                command.Parameters.AddWithValue("@Contact_no", txtContact.Text)
                command.Parameters.AddWithValue("@email", txtEmail.Text)
                command.Parameters.AddWithValue("@Gender", gender)
                command.Parameters.AddWithValue("@Birthday", bdaydtp.Text)
                command.Parameters.AddWithValue("@Pin", txtPin.Text)
                command.Parameters.AddWithValue("@type", "User")
                command.Parameters.AddWithValue("@balance", "1000")
                command.Parameters.AddWithValue("@custIMG", ms.ToArray())
                command.Parameters.AddWithValue("@Status", "Active")
                command.Parameters.AddWithValue("@pan_no", txtPancard.Text)
                command.Parameters.AddWithValue("@approve", "No")
                command.Parameters.AddWithValue("@accType", cmbTypeacc.Text)
                command.Parameters.AddWithValue("@income", income)
                command.ExecuteNonQuery()
                mysqlconn.Close()
                MsgBox("Your Details have been sent for approval, plz wait for 2-3
days", MsgBoxStyle.Information, "Saved")
                send_email()
                field_clear()
            Catch ex As MySqlException
                MsgBox(ex.Message)
            End Try
        End If
    End If
  End Sub
End Class
```

Login Page:

```
Imports MySql.Data.MySqlClient
Imports System.IO
Public Class login_frm
```

```
    Private Sub loginbtn_Click(sender As Object, e As EventArgs) Handles
loginbtn.Click
        If txtPin.Text = “” Or txtAcc.Text = “” Then
            MsgBox(“One or more Fileds are empty!!”, MsgBoxStyle.Critical, "Alert")
        Else
            Dim conn As MySqlConnection
            Dim cmd As New MySqlCommand
            conn = New MySqlConnection
            conn.ConnectionString =
“server=localhost;userid=root;password=punith123;database=atm”
            Dim Reader As MySqlDataReader
            Try
                conn.Open()
                Dim query As String
                query = "select * from atm.tblinfo where account_no= '" & txtAcc.Text &
"' and Pin = '" & txtPin.Text & "' "
                cmd = New MySqlCommand(query, conn)
                Reader = cmd.ExecuteReader
                Dim count As Integer
                count = 0
                While Reader.Read
                    count = count + 1
                End While
                If count = 1 Then
                    Dim acctype = Reader.GetString("Status")
                    Dim useraccno = Reader.GetString("account_no")
                    Dim usertype = Reader.GetString(“type”)
                    Dim name = Reader.GetString("Firstname")
                    Dim approve = Reader.GetString(“approve”)
                    If approve = "No" Then
                        MsgBox("Your account has not been approved yet, please try again
later", MsgBoxStyle.Critical, "Not Approved")
                    Else
                        If acctype = "Block" Then
                            MsgBox("This Card is Blocked ! Contact Administrator",
MessageBoxIcon.Error, "Account Suspended")
                        Else
                            If usertype = "Admin" Then
                                MsgBox("Welcome " & name & " you are logged in as
Administrator ", MsgBoxStyle.Information, "Success")
                                admin_frm.labelname.Text = name
                                admin_frm.labelacc.Text = useraccno
                                admin_frm.lblacc.Text = useraccno
                                Me.Hide()
                                admin_frm.Show()
                                Me.Hide()
                                conn.Close()
                                conn.Dispose()
                            Else
```

```
                    MsgBox("Welcome " & name & "", MsgBoxStyle.Information,
"Success")
                    Me.Hide()
                    Transaction.lblName.Text = name
                    Transaction.lblAcc.Text = useraccno
                    Transaction.lblaccount.Text = useraccno
                    Transaction.Show()
                End If
            End If
        End If
      Else
        MsgBox("Wrong Username or Password!")
            End If
      conn.Close()
    Catch ex As Exception
      MessageBox.Show(ex.Message)
    Finally
      conn.Dispose()
            End Try
  End If
 End Sub
End Class
```

Deposit Form:

```
Imports MySql.Data.MySqlClient
Imports System.Net
Imports System.Net.Mail
Public Class Deposit
  Dim mysqlconn As New MySqlConnection
  Dim command As New MySqlCommand
  Dim initialbal As Integer
  Dim amt As Integer
  Dim total As Integer
 Dim dep_date As String
  Dim emailID As String
    mysqlconn.ConnectionString =
"server=localhost;userid=root;password=punith123;database=atm"
   Private Sub DepositBtn_Click(sender As Object, e As EventArgs) Handles
DepositBtn.Click
    mysqlconn = New MySqlConnection
    If txtAmt.Text = "" Then
      MsgBox("Please Enter Amount!", MsgBoxStyle.Critical, "Amount")
    Else
      Dim reader As MySqlDataReader
      Try
        mysqlconn.Open()
        Dim query As String
        query = "select * from atm.tblinfo where account_no= '" & lblAcc.Text &
"' "
        command = New MySqlCommand(query, mysqlconn)
```

```
            reader = command.ExecuteReader
            Dim count As Integer
            count = 0
            While reader.Read
               count = count + 1
            End While
            If count = 1 Then
               Dim bal = reader.GetString("balance")
               initialbal = bal
               amt = txtAmt.Text
               If amt > 20000 Then
                  MsgBox("You cant deposit more than 20000 rupees",
MsgBoxStyle.Critical, "Alert")
               ElseIf amt < 200 Then
                  MsgBox("You cant deposit less than 200 rupees",
MsgBoxStyle.Critical, "Alert")
               Else
                  total = initialbal + amt
                  updater()
                  dep_statement()
               End If
            End If
            mysqlconn.Close()
         Catch ex As MySqlException
            MsgBox(ex.Message)
         Finally
            mysqlconn.Dispose()
         End Try
         reciept.Label9.Text = initialbal
         reciept.Label11.Text = amt
         reciept.Label12.Text = total
             End If
   End Sub

   Public Sub updater()
      mysqlconn = New MySqlConnection
            Dim reader As MySqlDataReader
      Try
         mysqlconn.Open()
         Dim query As String
         query = "UPDATE atm.tblinfo SET balance='" & total & "'" & " Where
account_no='" & lblAcc.Text & "'"
         command = New MySqlCommand(query, mysqlconn)
         reader = command.ExecuteReader
         MsgBox("Amount succesfully deposited, Confirmation sent via email ",
MsgBoxStyle.Information, "Deposit")
                  mysqlconn.Close()
      Catch ex As MySqlException
         MsgBox(ex.Message, MsgBoxStyle.Critical)
      Finally
```

```
            mysqlconn.Dispose()
        End Try
        Me.Hide()
        reciept.Show()
    End Sub
End Class
```

Withdraw form:

```
Imports MySql.Data.MySqlClient
Imports System.Net
Imports System.Net.Mail
Public Class withdraw
    Dim mysqlconn As New MySqlConnection
    Dim Command As New MySqlCommand
    Dim initialbal As Integer
    Dim amt As Integer
    Dim total As Integer
    Dim wit_date As String
    Dim emailID As String
        mysqlconn.ConnectionString =
"server=localhost;userid=root;password=punith123;database=atm"
        If txtAmt.Text = "" Then


Private Sub WithdrawBtn_Click(sender As Object, e As EventArgs) Handles
withdrawBtn.Click
        mysqlconn = New MySqlConnection
            MsgBox("Please Enter Amount!", MsgBoxStyle.Critical, "Amount")
        Else
            Dim reader As MySqlDataReader
            Try
                mysqlconn.Open()
                Dim query As String
                query = "select * from atm.tblinfo where account_no= '" & lblAcc.Text &
"' "
                Command = New MySqlCommand(query, mysqlconn)
                reader = Command.ExecuteReader
                Dim count As Integer
                count = 0
                While reader.Read
                    count = count + 1
                End While
                If count = 1 Then
                    Dim bal = reader.GetString("balance")
                    initialbal = bal
```

```
                amt = txtAmt.Text
                If initialbal <= txtAmt.Text Then
                    MsgBox("Insufficient balance ", MsgBoxStyle.Critical, "Insufficient
balance")
                ElseIf initialbal > 0 Then
                    If amt > 20000 Then
                        MsgBox("You cant Withdraw more than 10000 rupees",
MsgBoxStyle.Critical, "Alert")
                    ElseIf amt < 200 Then
                        MsgBox("You cant Withdraw less than 200 rupees",
MsgBoxStyle.Critical, "Alert")
                    Else
                        total = initialbal - amt
                        withdrawal()
                        wit_statement()
                    End If
                End If
            End If
            mysqlconn.Close()
        Catch ex As MySqlException
            MsgBox(ex.Message)
        Finally
            mysqlconn.Dispose()
        End Try
        reciept.label9.Text = initialbal
        reciept.label10.Text = amt
        reciept.label12.Text = total
    End If
  End Sub
  Public Sub wit_statement()
  End Sub

  Public Sub withdrawal()
    mysqlconn = New MySqlConnection
        Dim reader As MySqlDataReader
    Try
      mysqlconn.Open()
      Dim query As String
      query = "UPDATE atm.tblinfo SET balance='" & total & "'" & " Where
account_no='" & lblAcc.Text & "'"
      Command = New MySqlCommand(query, mysqlconn)
      reader = Command.ExecuteReader
      MsgBox("Amount Withdrawn Succesfully, Confirmation sent via email ",
MsgBoxStyle.Information, "Withdraw")

      mysqlconn.Close()
      Me.Close()
  Catch ex As MySqlException
      MsgBox(ex.Message, MsgBoxStyle.Critical)
    Finally
```

```
            mysqlconn.Dispose()
            Me.Hide()
            reciept.Show()
        End Try
    End Sub

End Class
```

Balance Form:

```
Imports MySql.Data.MySqlClient
Imports System.Net
Imports System.Net.Mail

Public Class balance
    mysqlconn = New MySqlConnection
        mysqlconn.ConnectionString =
"server=localhost;userid=root;password=punith123;database=atm"
    Dim cmd As MySqlCommand
    Dim emailID As String

Private Sub Balancebtn_Click(sender As Object, e As EventArgs) Handles
Balancebtn.Click
        Dim cmd As New MySqlCommand
            Dim Reader As MySqlDataReader
        Try
            mysqlconn.Open()
            Dim query As String= "select * from atm.tblinfo where account_no= '" &
lblAcc.Text & "'"
                    cmd = New MySqlCommand(query, mysqlconn)
            Reader = cmd.ExecuteReader
            Dim count As Integer
            While Reader.Read
                count = count + 1
            End While
            If count = 1 Then
                Dim bal = Reader.GetString("balance")
                emailID = Reader.GetString("email")
                lblBal.Text = bal
                MsgBox("Balance has been sent to you via Email",
MsgBoxStyle.Information, "Balance")
                emailsender()
                mysqlconn.Close()
            End If
        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
        lblBal.Show()
    End Sub
```

```
End Class
```

Transfer Statement:

```
Imports MySql.Data.MySqlClient
Imports System.Net
Imports System.Net.Mail

Public Class transfer
        mysqlconn = New MySqlConnection
        mysqlconn.ConnectionString =
"server=localhost;user=root;password=punith123;database=atm"
    Dim command As New MySqlCommand
    Dim transfererBal As Integer
    Dim transfererEmail As String
    Dim recieverBal As Integer
    Dim recieverEmail As String
    Dim finalbal2 As String
    Dim finalbal1 As String

    Private Sub balance_loader()
        Try
            mysqlconn.Open()
            Dim query As String = "select * from atm.tblinfo where account_no='" &
lblAcc.Text & "'"
            Dim reader As MySqlDataReader
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            While reader.Read
                transfererBal = reader.GetString("balance")
                transfererEmail = reader.GetString("Email")
            End While
            mysqlconn.Close()
        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
    End Sub

Private Sub BtnTransfer_Click(sender As Object, e As EventArgs) Handles
BtnTransfer.Click

        If txtAcc.Text = "" Or txtAmt.Text = "" Then
            MsgBox("One or more fields are empty", MsgBoxStyle.Critical, "Empty")
        Else
                    If txtAmt.Text > 500 Then
            Try
```

```
            mysqlconn.Open()
            Dim query As String = "select * from atm.tblinfo where account_no='"
& txtAcc.Text & "'"
            Dim reader As MySqlDataReader
            Dim count As Integer = 0
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            While reader.Read
               count = count + 1
            End While
            If count = 1 Then
               recieverBal = reader.GetString("balance")
               recieverEmail = reader.GetString("Email")
               If transfererBal > txtAmt.Text Then
                  transferer_bal_update()
                  reciever_bal_update()
                  transfer_dB_updater()
                  MsgBox("Amount transferred succesfully",
MsgBoxStyle.Information, "Amount transferred")
               Else
                  MsgBox("Insufficient balance", MsgBoxStyle.Critical, "Low
balance")
               End If
            Else
               MsgBox("Account not found", MsgBoxStyle.Critical, "Account not
found")
            End If
            mysqlconn.Close()
         Catch ex As MySqlException
            MsgBox(ex.Message)
         End Try
      Else
         MsgBox("You cant transfer less than 500", MsgBoxStyle.Critical)
      End If
   End If
End Sub

Private Sub transferer_bal_update()
   finalbal1 = transfererBal - txtAmt.Text
   Try
      mysqlconn.Open()
      Dim query As String = "update atm.tblinfo set balance='" & finalbal1 & "'
where account_no='" & lblAcc.Text & "'"
      Dim reader As MySqlDataReader
      Dim count As Integer = 0
      command = New MySqlCommand(query, mysqlconn)
      reader = command.ExecuteReader
      mysqlconn.Close()
   Catch ex As MySqlException
      MsgBox(ex.Message)
```

```
        End Try
    End Sub

    Private Sub reciever_bal_update()
        finalbal2 = recieverBal + txtAmt.Text
        Try
            mysqlconn.Open()
            Dim query As String = "update atm.tblinfo set balance='" & finalbal2 & "'
where account_no='" & txtAcc.Text & "'"
            Dim reader As MySqlDataReader
            Dim count As Integer = 0
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            mysqlconn.Close()
        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
    End Sub

Private Sub transfer_dB_updater()

        Try
            mysqlconn.Open()
            Dim query As String = "insert into transfer_statement
(acc_no,t_date,to_account,amt_transferred) values ('" & lblAcc.Text & "','" &
lblDate.Text & "','" & txtAcc.Text & "','" & txtAmt.Text & "')"
            Dim reader As MySqlDataReader
            Dim count As Integer = 0
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            mysqlconn.Close()
        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
    End Sub
End Class
```

Statement Form:

```
Imports MySql.Data.MySqlClient
Public Class Statement
    mysqlconn = New MySqlConnection
        mysqlconn.ConnectionString =
          "server=localhost;userid=root;password=punith123;database=atm"
    Dim Command As MySqlCommand
    Dim dbdataset As New DataTable
    Dim dbdataset1 As New DataTable
```

```
Private Sub DstatementBtn_Click(sender As Object, e As EventArgs) Handles
DstatementBtn.Click
        lblstatement.Text = "Deposit"
        lblstatement.Visible = True
        Datagridview2.Visible = False
        DataGridView1.Visible = True
            Dim SDA As New MySqlDataAdapter
        Dim bSource As New BindingSource

        Try
          mysqlconn.Open()
          dbdataset.Clear()
          Dim Query As String
          Query = "select acc_no as 'Card Number',dep_date as 'Date' ,dep_amt as
'Amount' from atm.deposit_statement where acc_no='" & lblAcc.Text & "' and
dep_date between '" & FromDtp.Text & "' and '" & TODtp.Text & "' "
          Command = New MySqlCommand(Query, mysqlconn)
          SDA.SelectCommand = Command
          SDA.Fill(dbdataset)
            bSource.DataSource = dbdataset
            DataGridView1.DataSource = bSource
            SDA.Update(dbdataset)
        Catch ex As MySqlException
          MessageBox.Show(ex.Message)
        Finally
          mysqlconn.Dispose()
        End Try
    End Sub

    Private Sub WstatementBtn_Click(sender As Object, e As EventArgs) Handles
WstatementBtn.Click
        lblstatement.Text = "Withdraw"
        lblstatement.Visible = True
        Datagridview2.Visible = False
        DataGridView1.Visible = True
        Dim SDA As New MySqlDataAdapter
        Dim bSource As New BindingSource
        Try
          mysqlconn.Open()
          dbdataset.Clear()
          Dim Query As String
          Query = "select acc_no as 'Card Number',w_date as 'Date' ,w_amt as
'Amount' from atm.withdraw_statement where acc_no='" & lblAcc.Text & "' and
w_date between '" & FromDtp.Text & "' and '" & TODtp.Text & "'"
          Command = New MySqlCommand(Query, mysqlconn)
          SDA.SelectCommand = Command
          SDA.Fill(dbdataset)
          bSource.DataSource = dbdataset
          DataGridView1.DataSource = bSource
          SDA.Update(dbdataset)
```

```
            mysqlconn.Close()
        Catch ex As MySqlException
            MessageBox.Show(ex.Message)
        Finally
            mysqlconn.Dispose()
        End Try
    End Sub

    Private Sub TstatementBtn_Click(sender As Object, e As EventArgs) Handles
TstatementBtn.Click
        lblstatement.Text = "Transfer"
        lblstatement.Visible = True
        Datagridview2.Visible = False
        DataGridView1.Visible = True
        mysqlconn = New MySqlConnection
        mysqlconn.ConnectionString =
          "server=localhost;userid=root;password=punith123;database=atm"
        Dim SDA As New MySqlDataAdapter
        Dim bSource As New BindingSource
        Try
            mysqlconn.Open()
            dbdataset1.Clear()
            Dim Query As String
            Query = "select acc_no as 'Card Number',t_date as 'Date' ,amt_transferred as
'Amount', to_account as 'transferred to' from atm.transfer_statement where acc_no='"
& lblAcc.Text & "' and t_date between '" & FromDtp.Text & "' and '" & TODtp.Text
& "' "
            Command = New MySqlCommand(Query, mysqlconn)
            SDA.SelectCommand = Command
            SDA.Fill(dbdataset1)
            bSource.DataSource = dbdataset1
            DataGridView1.DataSource = bSource
            SDA.Update(dbdataset1)
            mysqlconn.Close()
        Catch ex As MySqlException
            MessageBox.Show(ex.Message)
        Finally
            mysqlconn.Dispose()
        End Try
    End Sub

End Class
```

Get or Return Loan Form:

```
Imports MySql.Data.MySqlClient
Public Class applyLoan
mysqlconn = New MySqlConnection
        mysqlconn.ConnectionString =
"server=localhost;user=root;password=punith123;database=atm"
    Dim command As New MySqlCommand
```

```
    Dim custname As String
    Dim email As String
    Dim balance As String
    Dim interest As Double
    Dim principle As Double
    Dim rate As Double
    Dim time As Double
    Dim updated_bal As Integer
    Dim penalty As Integer
    Dim months As Integer
    Dim totalmonths As Integer
    Dim income As Integer
    Private Sub getloan_Click(sender As Object, e As EventArgs) Handles
getLoan.Click
        If checkbox.Checked = True Then
            data_saver()
        Else
            MsgBox("Read Terms and condition first", MsgBoxStyle.Critical)
        End If
    End Sub


    Private Sub calcBtn_Click(sender As Object, e As EventArgs) Handles
calcBtn.Click
        principle = amt.Text
        time = periodCmb.Text / 12
        rate = interestCmb.Text
        interest = ((principle * time * rate) / 100)
        txttotal.Text = Math.Round(principle + interest, 0)
        txtemi.Text = Math.Round(txttotal.Text / periodCmb.Text, 0)
        updated_bal = balance + principle
        duedtp.Value = DateTime.Parse(loandtp.Value).AddDays(+31)
        getLoan.Enabled = True
    End Sub

    Private Sub data_saver()
Try{
            mysqlconn.Open()
            Dim query As String
            Dim reader As MySqlDataReader
            query = "insert into atm.loan_db
(account_no,name,date_taken,total_amt,status,email,months_paid,emi_amount,total_
months,due_date,approve) values ('" & crdlbl.Text & "','" & namelbl.Text & "','" &
loandtp.Value & "','" & txttotal.Text & "','Active','" & email & "','0','" & txtemi.Text
& "','" & periodCmb.Text & "','" & duedtp.Value & "','No')"
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            MsgBox("Your loan request has been sent for approval, if approved amount
will be debited to your account in 2-3 working days :)")
        Catch ex As MySqlException
```

```
            MsgBox(ex.Message)
        End Try
    End Sub

End Class
```

Admin Form:

```
Imports MySql.Data.MySqlClient
Imports System.IO
Public Class admin_frm
    mysqlconn = New MySqlConnection
        mysqlconn.ConnectionString =
          "server=localhost;userid=root;password=punith123;database=atm"
    Dim Command As New MySqlCommand
    Dim dbdataset As New DataTable
    Dim GENDER As String
    Dim accno As String

 Private Sub searchbtn_Click(sender As Object, e As EventArgs) Handles
searchbtn.Click
            Dim reader As MySqlDataReader
        Try
            mysqlconn.Open()
            dbdataset.Clear()
            Dim Query As String
            Query = "select * from atm.tblinfo where account_no='" & txtAcc.Text & "'"
            Command = New MySqlCommand(Query, mysqlconn)
            reader = Command.ExecuteReader
            Dim count As Integer = 0
            While reader.Read
                count = count + 1
            End While
            If count = 1 Then
                okBTnLoader()
            Else
              MsgBox("Account linked to entered card number not found",
MsgBoxStyle.Critical, "Alert")
                txtAcc.Text = ""
            End If
        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
    End Sub
```

```
    Private Sub loadBtn_Click(sender As Object, e As EventArgs) Handles
loadBtn.Click
        dbdataset.Clear()
       Dim SDA As New MySqlDataAdapter
        Dim bSource As New BindingSource
        Try
            mysqlconn.Open()
            Dim Query As String
            Query = "select account_no as 'Card
Number',Firstname,Lastname,Address,Contact_no as 'Phone Number',Gender from
atm.tblinfo "
            Command = New MySqlCommand(Query, mysqlconn)
            SDA.SelectCommand = Command
            SDA.Fill(dbdataset)
            bSource.DataSource = dbdataset
            DataGridView1.DataSource = bSource
            SDA.Update(dbdataset)
            mysqlconn.Close()
        Catch ex As MySqlException
            MessageBox.Show(ex.Message)
        Finally
            mysqlconn.Dispose()
        End Try
    End Sub

    Private Sub UpdateBtn_Click(sender As Object, e As EventArgs) Handles
UpdateBtn.Click
        Try
            mysqlconn.Open()
            dbdataset.Clear()
            Dim Query As String
            Query = "select * from atm.tblinfo where account_no='" & txtAcc.Text & "'"
            Command = New MySqlCommand(Query, mysqlconn)
            reader = Command.ExecuteReader
            Dim count As Integer = 0
            While reader.Read
                count = count + 1
            End While
            If count = 1 Then
                updater()
            Else
                MsgBox("Account linked to entered card number not found",
MsgBoxStyle.Critical, "Alert")
                txtAcc.Text = ""
            End If

        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
    End Sub
```

```
    Private Sub updater()
            Dim SDA As New MySqlDataAdapter
        Dim bSource As New BindingSource
        Dim reader As MySqlDataReader
        Try
            mysqlconn.Open()
            Dim query As String
            query = "update atm.tblinfo set Firstname='" & txtFname.Text &
"',Lastname='" & txtLname.Text & "',Address='" & txtAddr.Text & "',Contact_no='"
& txtContact.Text & "',pin='" & txtPin.Text & "',Gender='" & GENDER &
"',Birthday='" & bdayDtp.Text & "',account_no='" & txtAcc.Text & "'where
account_no='" & txtAcc.Text & "'"
            Command = New MySqlCommand(query, mysqlconn)
            reader = Command.ExecuteReader
            MsgBox("Account Data Updated ", MsgBoxStyle.Information, "Updated")
            refresh_db()
            mysqlconn.Close()
        Catch ex As MySqlException
            MsgBox(ex.Message, MsgBoxStyle.Critical)
        Finally
            mysqlconn.Dispose()
        End Try
    End Sub



    Private Sub blockBtn_Click(sender As Object, e As EventArgs) Handles
blockBtn.Click
        Try
            mysqlconn.Open()
            Dim Query As String
            Query = " UPDATE atm.tblinfo SET Status = 'Block'  WHERE account_no =
'" & txtAcc.Text & "'"
            Command = New MySqlCommand(Query, mysqlconn)
            READER = Command.ExecuteReader
            MsgBox("Account linked to entered card No. was not found",
MsgBoxStyle.Information, "Blocked")
            refresh_db()
            mysqlconn.Close()
        Catch ex As MySqlException
            MessageBox.Show(ex.Message)
        Finally
            mysqlconn.Dispose()

    End Try
    End Sub

  Private Sub unblockBtn_Click(sender As Object, e As EventArgs) Handles
unblockBtn.Click
```

```
        Try
          mysqlconn.Open()
          Dim Query As String
          Query = " UPDATE atm.tblinfo SET Status = 'Active'  WHERE account_no =
''' & txtAcc.Text & """
          Command = New MySqlCommand(Query, mysqlconn)
          READER = Command.ExecuteReader
          MsgBox("Account linked to entered card No. was Unblocked",
MsgBoxStyle.Information, "Unblock")
          mysqlconn.Close()
       Catch ex As MySqlException
          MessageBox.Show(ex.Message)
       Finally
          mysqlconn.Dispose()
       End Try
   End Sub
End Class
```

Approve Accounts Form:

```
Imports MySql.Data.MySqlClient
Imports System.IO
Imports System.Net
Imports System.Net.Mail
Public Class approveform
      mysqlconn = New
MySqlConnection("Server=localhost;user=root;password=punith123;database=atm")
   Dim command As New MySqlCommand


Private Sub approveBtn_Click(sender As Object, e As EventArgs) Handles
approveBtn.Click
      Try
         Dim query As String = "Update atm.tblinfo set approve='Yes' where
account_no='" & cmbacc.Text & """
          mysqlconn.Open()
          command = New MySqlCommand(query, mysqlconn)
          reader = command.ExecuteReader
          MsgBox("Account request approved", MsgBoxStyle.Information, "Approve")
          emailApprove()
          cmbacc.Text = ""
          fieldsclear()
       Catch ex As MySqlException
          MsgBox(ex.Message)
       End Try
       mysqlconn.Close()
   End Sub
End Class
```

Approve Loan Form:

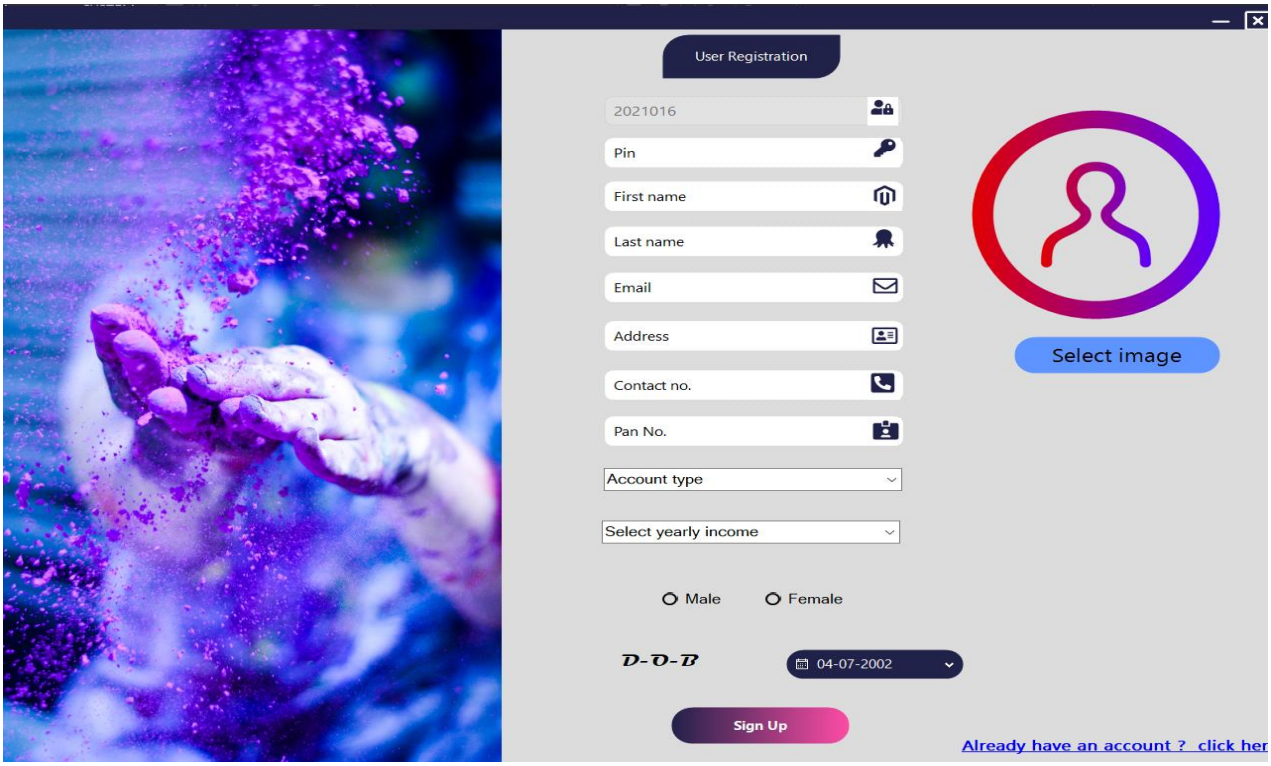
```
Imports MySql.Data.MySqlClient
Imports System.Net
Imports System.Net.Mail

Public Class Loan_approve
mysqlconn = New
MySqlConnection("Server=localhost;user=root;password=punith123;database=atm")
    Dim command As New MySqlCommand
    Dim balance As Integer
    Dim updated_bal As Integer


Private Sub approveBtn_Click(sender As Object, e As EventArgs) Handles
approveBtn.Click
        Try
            mysqlconn = New
            Dim query As String = "Update atm.loan_db set approve='Yes' where
account_no='" & cmbCard.Text & "'"
            mysqlconn.Open()
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            balance_updater()
        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
        mysqlconn.Close()
        cmb_loader()
    End Sub


    Private Sub balance_updater()
        updated_bal = balance + lblAmt.Text
              Try
            mysqlconn.Open()
            Dim query As String
            Dim reader As MySqlDataReader
            query = "update atm.tblinfo set balance='" & updated_bal & "' where
account_no='" & lblCard.Text & "'"
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            MsgBox("Loan sanctioned, amount has been transferred to your account
linked to the Card no. '" & lblCard.Text & "'", MsgBoxStyle.Information, "Loan
Sanctioned")

        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
    End Sub
```

```
Private Sub btnReject_Click(sender As Object, e As EventArgs) Handles
btnReject.Click
        Try
            Dim query As String = "delete from atm.loan_db  where account_no='" &
cmbCard.Text & "'"
            mysqlconn.Open()
            command = New MySqlCommand(query, mysqlconn)
            reader = command.ExecuteReader
            MsgBox("Rejected", MsgBoxStyle.Information, "Rejected")
            cmbCard.Text = ""
        Catch ex As MySqlException
            MsgBox(ex.Message)
        End Try
        mysqlconn.Close()
            End Sub
End Class
```

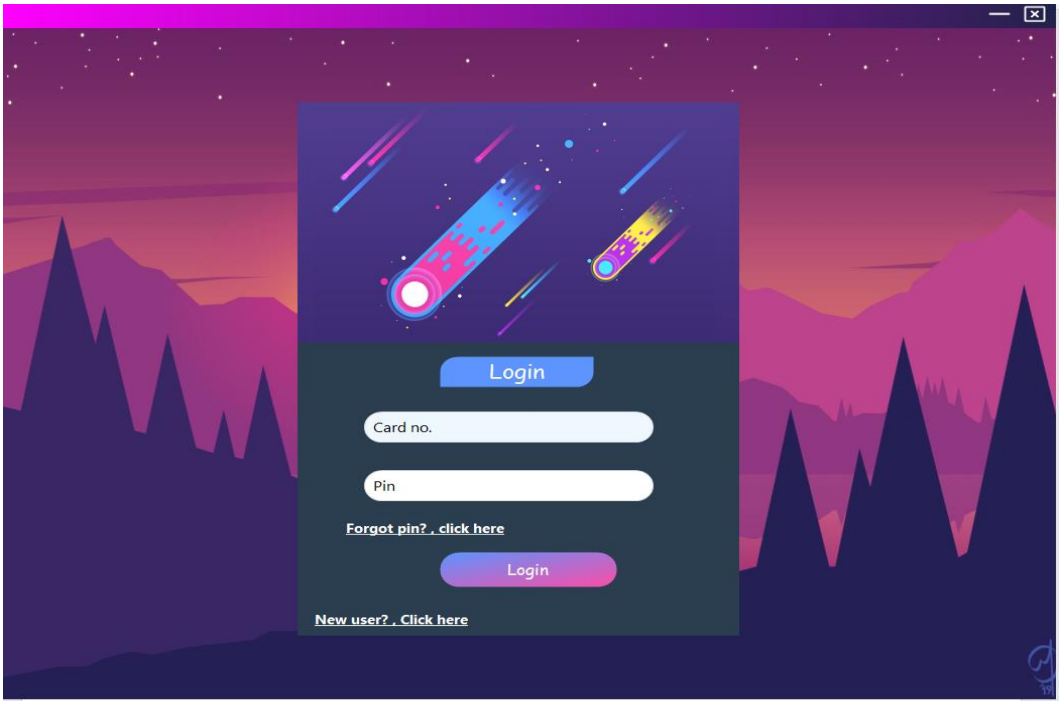
8. SCREENSHOTS

8.1.1 Main Form

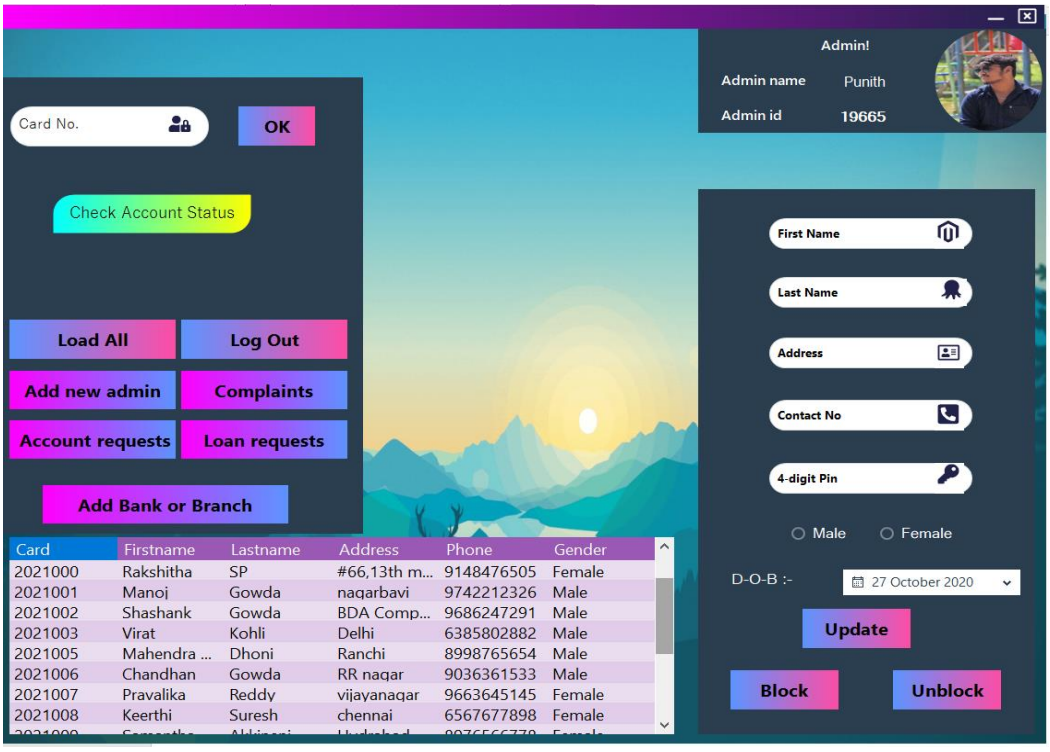
8.1.2 Registration Form



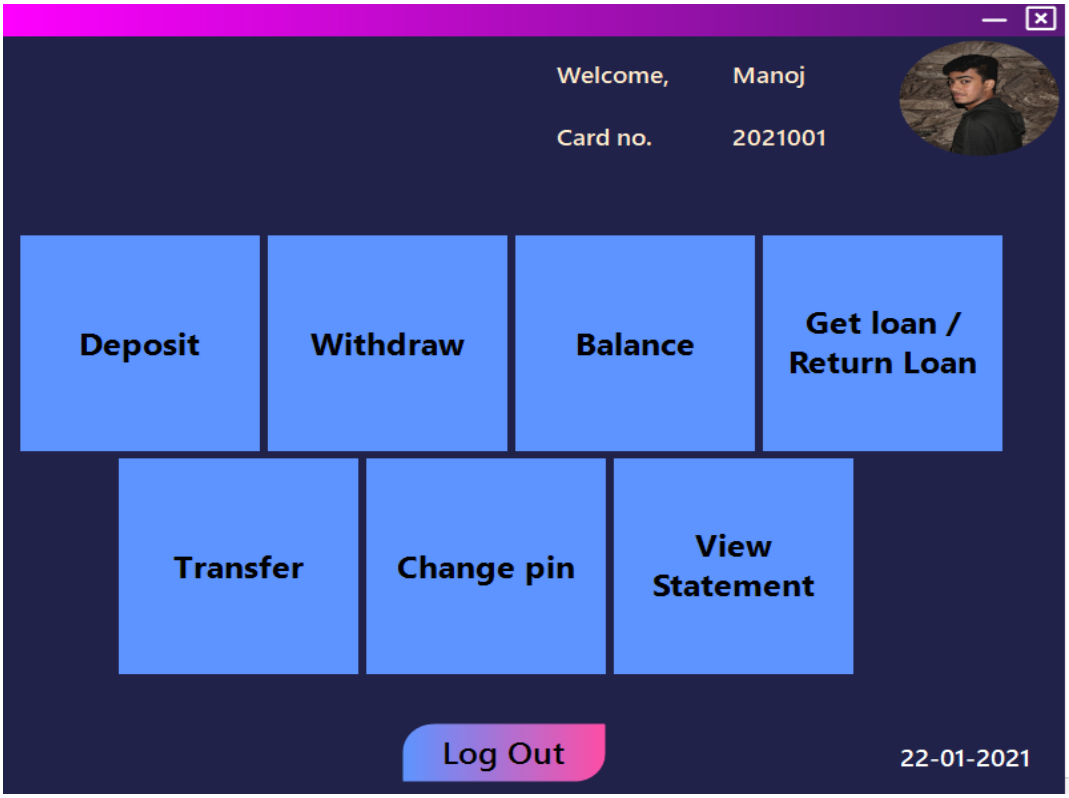
8.1.3 Login Form



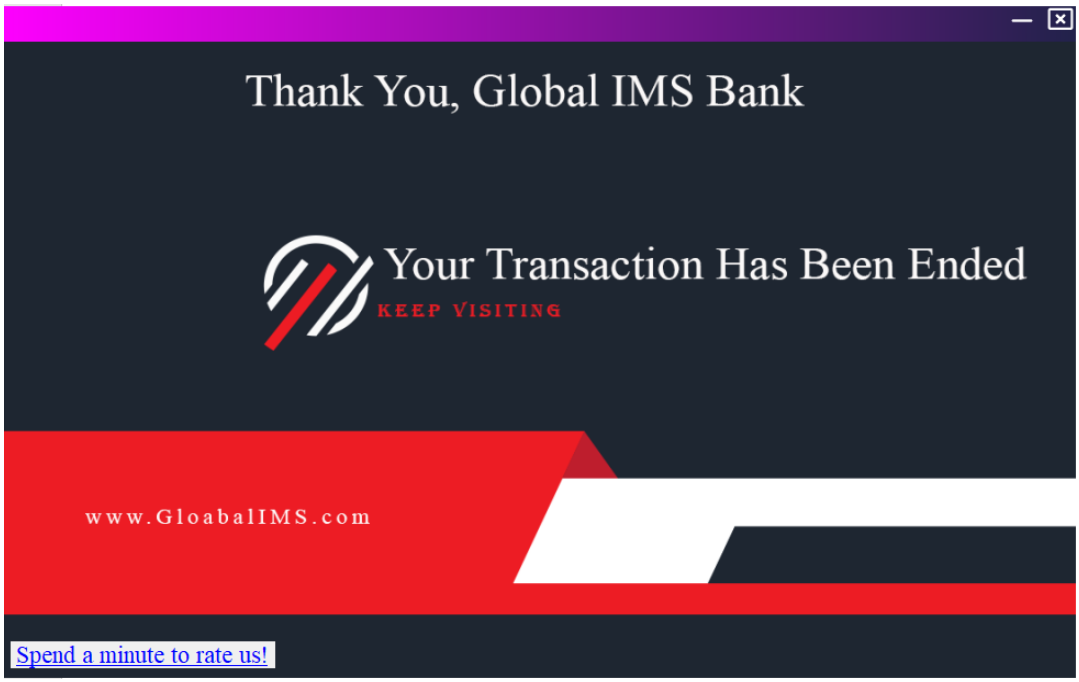
8.1.4 Admin Form



8.1.5 Transaction Form

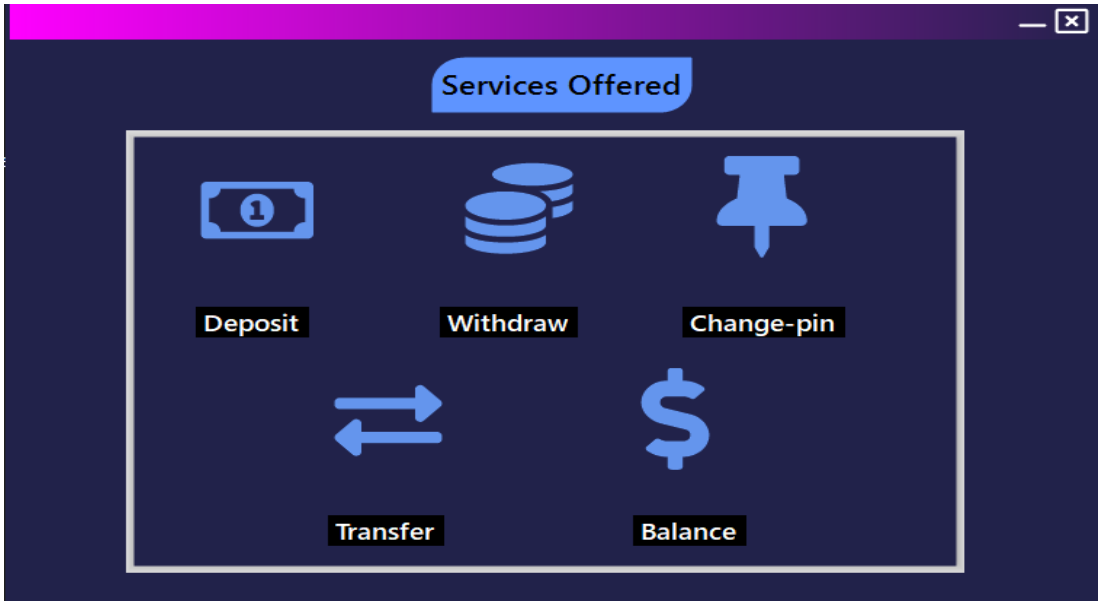


8.1.6 End Transaction

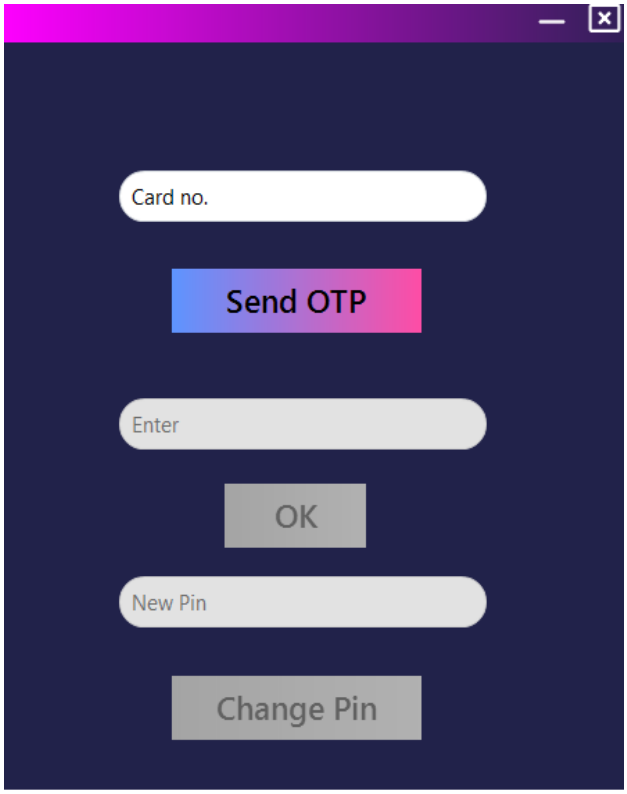


8.1.7 Contact us

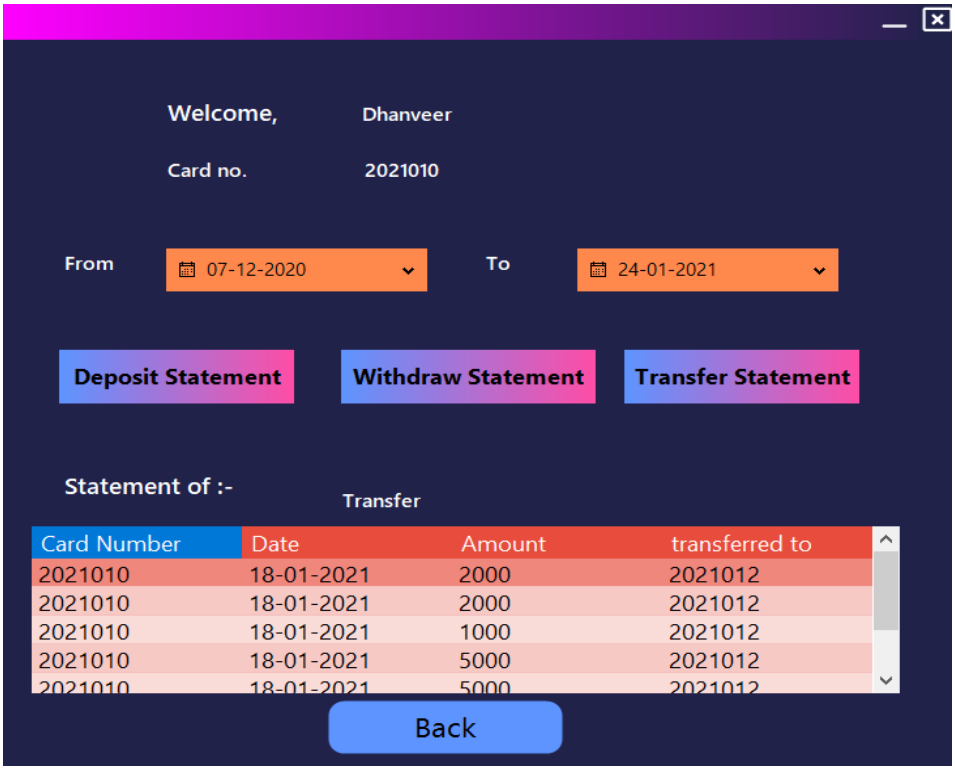
8.1.8 Services Form



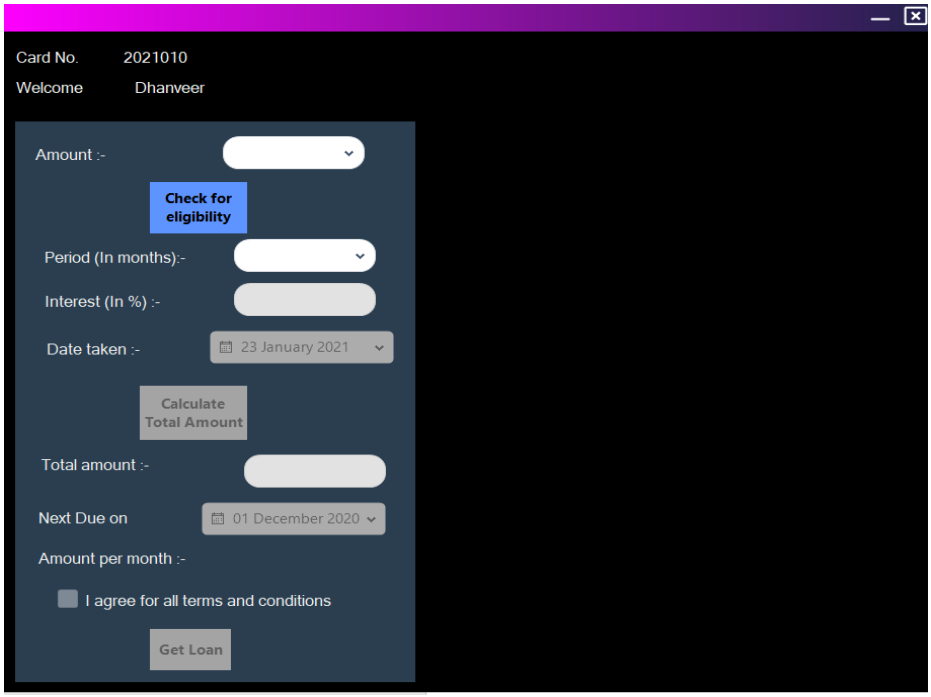
8.1.9 Forgot Password



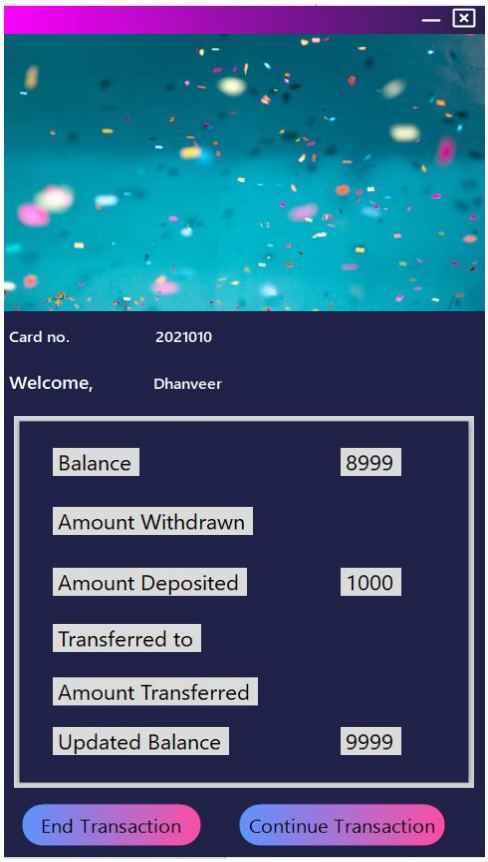
8.1.9 Statement Form



8.2.0 Get/ Return Loans



8.2.1 Receipt



9. TESTING

Software testing is performed to verify that the completed software package functions according to the expectations defined by the requirements/specifications. The overall objective to not to find every software bug that exists, but to uncover situations that could negatively impact the customer, usability and/or maintainability.

PURPOSE OF TESTING

- Finding defects in software which may get created by programmer while developing the software.
- To prevent defects.
- To make sure the end result meets business and user requirements.
- To ensure that it satisfies BRS that is Business Requirements Specification and SRS that is Software Requirements Specifications.
- To gain the confidence of the customers by providing them quality product.

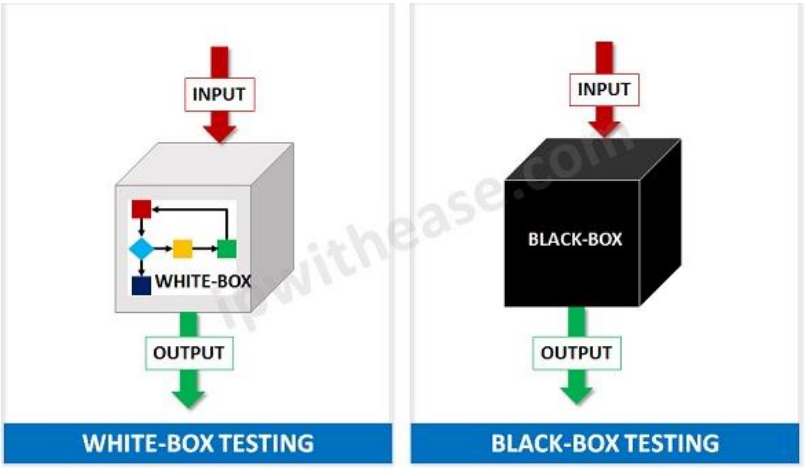
9.1 Types of testing:

1. White Box Testing

It is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions. Also known as structural testing and Glass box Testing.

2. Black box testing

Internal system design is not considered in this type of testing. Tests are based on requirements and functionality. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.



Levels of testing

UNIT TESTING

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed.

INTEGRATION TESTING

Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

SYSTEM TESTING

This is the next level in the testing and 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 Quality Standards. This type of testing is performed by a specialized testing team.

System Testing is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

ACCEPTANCE TESTING

Acceptance testing or User Acceptance Testing is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery

9.3 TEST CASES

Test Scenario 1: Enter proper credentials, login as Admin and check for successful login.

Test Case 1:

Step#	Description	Input	Expected result	Actual result	Status
1	Open the Application	N/A	'ATM Management System' Main form must be displayed	'ATM Management System' Main form is displayed	Pass
2	Click on login	N/A	Login form must be displayed	Login form is displayed	Pass
3	Enter card no. and pin and click on login	Card no: 19665 Pin: 8867	Admin form must be displayed	Admin form is displayed	Pass

Test Scenario 2: Login as admin by leaving pin field blank and check for error message

Test Case 2:

Step#	Description	Input	Expected result	Actual result	Status
1	Open the Application	N/A	'ATM Management System' Main form must be displayed	'ATM Management System' Main form is displayed	Pass
2	Click on login	N/A	Login form must be displayed	Login form is displayed	Pass

3	Enter card no. and pin and click on login	Card no: 19665 Pin:	Enter password message must be displayed	Enter password message is displayed	Pass

Test Scenario 3: Login as admin, Block card.

Test case 3:

Step#	Description	Input	Expected result	Actual result	Status
1	Login as admin	Card no: 19665 Pin:8867	Admin form must be displayed	Admin form is displayed	Pass
2	Click on card no. text box	N/A	Focus must be given to card no. text field	Focus is given to card no. text field	Pass
3	Enter proper details Click on Block	Enter card no.	Card successfully Blocked message must be displayed. Blocked card must be displayed in Card details table	Card successfully Blocked message is displayed. Blocked card is displayed in Card details table	Pass

Test Scenario 4: Login as admin, unblock a card.

Test case 4:

Step#	Description	Input	Expected result	Actual result	Status
1	Login as admin	Card no: 19665 Pin:8867	Admin form must be displayed	Admin form is displayed	Pass
2	Click on card no. text box	N/A	Focus must be given to card no. text field	Focus is given to card no. text field	Pass
3	Enter proper details Click on Unblock	Enter card no.	Card successfully Unblocked message must be displayed. Unlocked card must be displayed in Card details table	Card successfully Unblocked message is displayed. Unblocked card is displayed in Card details table	Pass

10. CONCLUSION

This is to conclude that "ATM Management System" is windows forms application which is made for customers who can't keep visiting banks every now and then.
As the technology is being advanced the way of life is changing accordance. Now a day we can visit any ATM nearby to withdraw amount. This Application also provides features like applying for loans, depositing money etc. Which helps busy customers.
We can think how the days have been changed with time. People had to stand in rows and wait to make some transactions. But with current technology we can do everything easily and without wasting time.

In future, we will try to make this application flexible and beneficial for the customer and also try to make smooth service.

11. FUTURE ENHANCEMENT

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used.

Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Since my project is based on object-oriented design, any future change can be easily adaptable.
- Based on future security issues, it can be improved using emerging technologies.
- SMS notifications for customers.
- A basic chatbot can be added to answer customer queries.
- Live chat facilities for customers with our executives.
- Intruder alert by sending pic of the intruder to the customer.
- Automated calls to notify customers about Loans or Low balance.

BIBLIOGRAPHY

REFERRED TEXTBOOKS

- "Visual Programming"- by Padma Geetha B G & Srikanth S
- "Working with Microsoft Visual Studio 2005 Team System" -by Richard Hundhausen
- "Database Management System" -by Ashwini S Diwakar
- "Software Engineering" -by Ashwini S Diwakar

REFERRED WEBSITES

- www.wikipedia.com
- www.stackoverflow.com
- www.programmingknowledgeblog.blogspot.com
- www.tutorialspoint.com
- www.codeproject.com
- www.vb.net-informations.com
- www.w3schools.com
- www.youtube.com/ProgrammingKnowledge

APPENDIX(SYNOPSIS)

TITLE: ATM MANAGEMENT SYSTEM

ABOUT THIS PROJECT

Millions of times per day around the globe people are instantly withdrawing money at automatic teller machines (ATMs). Given the fast-pace of the world today, it is not surprising that the demand for access to quick cash is so immense. The power of ATMs would not be possible without secure connections. The final act of ATM dispending cash is the result of an amazing fast burst of the customer never sees, but a trust is being done in a confidential manner

ATM allows the customers of a bank to have access to their account without going to the bank. This is achieved by development of application like **ATM Management System.**

When the application is adopted, the user who uses this product will be able to see all the information and services provided by the ATM, when he enters the necessary pin. This project is used to Deposit and Withdraw cash also he can check the balance, the administrator can use this product to get the information of the customers and he can block and unblock the accounts when necessary. The data is stored in the database and is retrieved based on requirements. The implementation of this product needs hardware like an ATM machine to function the way its supposed to.

Development of this product includes various operations such as:

1. Home menu
2. Login/Register
3. Admin process
4. User Transaction
5. Send email

OBJECTIVE

- The objective of the project "ATM Management System" is to develop Automatic Teller System of the wide variety of electronic items online.
- This project is mainly developed to help customers do their bank transactions under one roof easily without any trouble. Customers can visit any nearest ATM to do transactions, according to their convenience.

Existing System:

In existing system, the bank staffs have to maintain lot of records manually regarding the transactions of customers. Details of the customer cannot be segregated according to the user needs

Disadvantages:

- Time Consuming
- Lot of paperwork
- Prone to errors

Proposed System:

The proposed system differs from the existing system by clubbing into a single form reducing the existing so as to introduce various other properties in the software and making it easier to fill the details. And this software will be helpful in the smooth functioning of the organization due to integration of various functions, the software maintains the central database with the following information stored in different tables: customer's details, transaction details, loan details etc.

Advantages

- Less effort to complete transaction
- Less time required to complete transactions compared to existing system
- No need to maintain bulk of books / papers
- Less or no errors
- Easy to retrieve data.

MODULES USED IN THE PROJECT

1. ADMIN

- Log In: Admin can login to the application by providing the valid credentials to access the application.
- Approve Accounts: Admin will approve the account requests by new customers.
- Approve Loans: Admin will approve the loans requested by existing customers.
- Update Accounts: Admin can update account details.
- Delete Accounts: Admin can delete account details.
- Block/Unblock Accounts: Admin can block/unblock accounts when required.
- View Users: Admin can view user details at any time.
- Logout: Admin can logout from application to end the session

2. CUSTOMER

- Register: customers have to first register their account to the application using card number.
- Log in: customers can login to application by providing valid card no. and password.
- Deposit: Customers can deposit money to their account.
- Withdraw: Customers can withdraw money from their account.
- Transfer: Customers can transfer money from their account to another account.

- Loan: Customers can get loans to their account or repay loans from their accounts.
- Change pin: Customers can change pin at any point of time.
- View Statement: Customers can view their transaction statements.
- Log out: Customers can logout of the application.

SYSTEM REQUIREMENTS

HARDWARE INTERFACES

Processor	:	Intel i3 +
RAM	:	4GB
Hard Disk	:	80GB
Speed	:	1.2 GHz+

SOFTWARE INTERFACES

Operating System	:	Windows 7 or Higher
IDE	:	Visual Studio 2019
Language	:	Visual Basic
Framework	:	VB.NET 4.7.2
Back End	:	MY SQL