

Online Banking System

Under Guidance Of
Mr. Joyjit Guha Biswas
Subject Matter Expert(PYTHON)

Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified)
CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064
A

Project Report
Submitted In Partial Fulfillment Of The Requirements
For The Award Of the
Bachelor of Technology
Project Carried Out At



Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified)
CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

Submitted By

Arup Bhowmik
Anirban Samanta
Surjan Mukherjee
Pradhyumna Bhattacharya

Department Of Computer Science and Engineering

Future Institute Of Technology
Boral-700154
West Bengal | INDIA |

(Note: All entries of the proforma of approval should be filled up with appropriate and complete information of approval in any respect will be summarily rejected.)

1. Name of the Student With Group: 1: Arup Bhowmik
2: Anirban Samanta
3: Surjan Mukherjee
4: Pradhyumna Bhattacharya
2. Title of the Project : **Online Banking System**
3. Name and Address of the Guide : **Mr. Joyjit Guha Biswas**
Sr. Subject Matter Expert & Technical Head(PYTHON)
Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified)
CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064
4. Educational Qualification of the Guide : Ph.d* M.tech* B.E*/B.Tech * MCA* M.Sc*
- | | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
5. Working and Teaching experience of the Guide :Years
6. Software used in the Project:
- a. VS-Code
b. Python-Django
c. SQL-DB-Browser

- 1.
- 2.
- 3.
- 4.

Signature of the Student

Date:

For Office Use Only

Approved

Not Approved

Signature of the Guide

Date:

Name: Mr. Joyjit Guha Biswas

Subject Matter Expert.

Signature, Designation, Stamp of the Project
Proposal Evaluator

Project Responsibility Form

Online Banking System

SERIAL NUMBER	NAME OF MEMBER	RESPONSIBILITY
1	ANIRBAN SAMANTA	PROJECT LEADER AND CODING.
2	SURJAN MUKHERJEE &PRADHYUMNA BHATTACHARYA	WEB PAGE DESIGNING.
3	ARUP BHOWMIK	SYSTEM ANALYSIS, DOCUMENTATION

Self Certificate

This is to certify that the dissertation/project proposal entitled “**Online Bankin System**” is done by us, is an Authentic work carried out for the partial fulfillment of the requirements for the award of the certificate of **Bachelor of Technology** under the guidance of **Mr. Joyjit Guha Biswas**. The matter embodied in this project work has not been submitted earlier for award of any certificate to the best of our knowledge and belief.

Name of the Student

☐ **Arup Bhowmik**

☐ **Anirban Samanta**

☐ **Surjan Mukherjee**

☐ **Pradhyumna Bhattacharya**

Signature of the students

- a.
- b.
- c.
- d.

Certificate by Guide

This is to certify that this project entitled “**Online Banking System**” submitted in partial fulfillment of the certificate of Bachelor of Computer Application through **Ardent Computech Pvt Ltd**, done by the Group Members

- ☐ **Arup Bhowmik**
- ☐ **Anirban Samanta**
- ☐ **Surjan Mukherjee**
- ☐ **Pradhyumna Bhattacharya**

is an authentic work carried out under my guidance & best of our knowledge and belief..

a.

b.

c.

d.

Signature of the students

Date:

Signature of the Guide

Date:

Certificate of Approval

This is to certify that this proposal of Minor project, entitled “**Online Banking System**” is a record of bona-fide work, carried out by: 1. Arup Bhowmik, 2. Anirban Samanta, 3. Surjan Mukherjee, 4. Pradhyumna Bhattacharya under my supervision and guidance through the Ardent Computech Pvt Ltd. In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by the Future Institute Of Technology as per regulations of the **Ardent®**. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for Bachelor of Technology.

Guide/Supervisor

Mr. Joyjit Guha Biswas

Subject Matter Expert & Technical Head
(PYTHON)

Ardent Computech Pvt Ltd (An ISO 9001:2008 Certified)
CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

External Examiner(s)

Head of the Department
Department of Computer Science

**Kolkata, W.B (Affiliated
to WBUT, WB)**



OBS

TABLE OF CONTENTS

<u>S.NO.</u>	<u>Name of the Topic</u>	<u>Page No.</u>
1.	Company Profile	10
2	Introduction	12-14
2.a	Objective	13
2.b	Scope	14
3	System Analysis	15-28
3.a	Identification of Need	16-17
3.b	Feasibility Study	18
3.c	Work flow	19-22
3.d	Study of the System	23
3.e	Input & Output	24
3.f	Software Requirements Specification(SRS)	25-26
3.g	Software Engineering Paradigm Applied	27-28
4	System Design	29-52
4.a	Data Flow Diagram(DFD)	30-34
4.b	Sequence Diagram	35-36
4.c	Entity-Relationship Diagram	37-41
4.d	Use-case Diagram	42-45
4.e	Modularization Details	46-47
4.f	Database Design	48-52
5	Output Screen	53-70
5.a	User Interface Design	54
5.b	Snapshots	55-70
6	Implementation & Testing	71-85
6.a	Introduction	71
6.b	Objectives of Testing	72-73

6.c	Test Cases	74-79
6.d	White Box Testing	80
6.e	Black Box Testing	81
6.f	Output Testing	82
6.g	Goal of Testing	83
6.h	Integration Test Reports	84-85
7	Gantt Chart
8	System Security Measures	86-89
8.a	Database security measures	87
8.b	System security measures	88
8.c	Limitations	89
9	Conclusion	90
10	Future Scope & Further Enhancements	91
11	Bibliography/References	92

1. ARDENT COMPUTECH PVT.LTD.

Ardent Computech Private Limited is an ISO 9001-2008 certified Software Development Company in India. It has been operating independently since 2003. It was recently merged with ARDENT TECHNOLOGIES.

Ardent Technologies

ARDENT TECHNOLOGIES is a Company successfully providing its services currently in UK, USA, Canada and India. The core line of activity at ARDENT TECHNOLOGIES is to develop customized application software covering the entire responsibility of performing the initial system study, design, development, implementation and training. It also deals with consultancy services and Electronic Security systems. Its primary clientele includes educational institutes, entertainment industries, resorts, theme parks, service industry, telecom operators, media and other business houses working in various capacities.

Ardent Collaborations

ARDENT COLLABORATIONS, the Research Training and Development Department of ARDENT COMPUTECH PVT LTD is a professional training Company offering IT enabled services & industrial trainings for B-Tech, MCA, BCA, MSc and MBA fresher's and experienced developers/programmers in various platforms. Summer Training / Winter Training / Industrial training will be provided for the students of B.TECH, M.TECH, MBA and MCA only. Deserving candidates may be awarded stipends, scholarships and other benefits, depending on their performance and recommendations of the mentors.

Associations

Ardent is an ISO 9001:2008 company.

It is affiliated to National Council of Vocational Training (NCVT), Directorate General of Employment & Training (DGET), Ministry of Labor & Employment, and Government of India.

OBS

(ONLINE BANKING SYSTEM)

2. INTRODUCTION

Online Banking System is a system that manages the record of transactions regarding credit, debit and money transfer operations.

It can be used by any banking organization or finance companies to maintain the records of customers easily. This system provides a simple interface for the maintenance of customer information from the time of account opening till the time of account closing .

Online banking system deals with all kind of customer details, account details, transaction details and other related details too. It tracks all the details of a customer from the day one to the end of the account which can be used for all reporting purpose, tracking of transaction, progress in the balance.

2a. OBJECTIVE

The objective of Online Banking System is to allow the administrator and super user of any bank to edit and view each and every minute detail record of customers, transactions, bank details, batches and payment structure of an organization easily.

Without using this system, the information remains scattered, can be redundant and collecting relevant information may be time consuming. All these problems are solved by using this website.

Throughout the project the focus has been on presenting information in an easy and intelligent manner.

The website provides facilities like online registration and profile creation of customers thus reducing paperwork and automating the record generation process in a finance organization.

2b. SCOPE

Without online banking system, managing and maintaining details of the customers, transactions, debit cards, statement creation is a tedious job for any organization.

According to our website, role of admin is to validate and add money to an account after login process and the role of super admin is to insert and modify the details of the admin and provide them the login credentials.

SYSTEM ANALYSIS

3a. IDENTIFICATION OF NEED

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process .The system studies the minutest detail and gets analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The System is viewed as a whole and the input to the system are identified. The outputs from the organization are traced to the various processes. System analysis is concerned with becoming aware of the problem ,identifying the relevant and Decisional variables ,analysis and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system .Now the existing system is subjected to close study and problem area are identified .The designer now function as a problem solver and tries to sort out the difficulties that the enterprise faces. The solution are given as proposals .The proposal is then weighed with the existing system analytically and the best one is selected .The proposal is presented to the user for an endorsement by the user .The proposal is reviewed on user request and

suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

3b. FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose the organization for the amount of work.

Effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A Feasibility study of a system proposal is according to its work ability, which is the impact on the organization, ability to meet their user needs and effective use of resources .Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various area that were considered very carefully during the feasibility study of this project such as Technical , Economic and operational feasibilities.

3c. WORK FLOW

This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

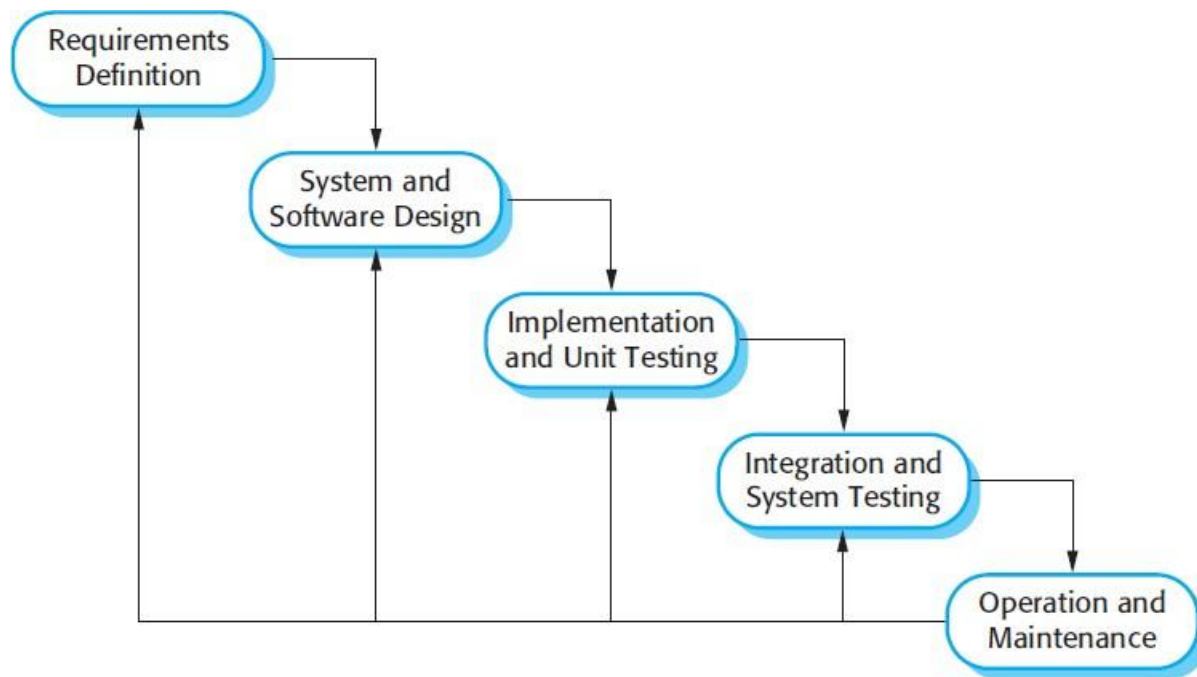
Waterfall model is the earliest SDLC approach that was used for software development .

The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap.

Waterfall Model design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model.



The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
- **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of

each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

3d. STUDY OF THE SYSTEM

Modules:

The modules used in this software are as follows:

- **LOGIN:** This module is for ADMIN and CUSTOMER.
Super-ADMIN has the authority to Insert, Update and Deleteadmin, user, transaction, cards, payment details. ADMINISTRATOR has the authority to validate , customer details and balance to account.
- **HOME :**This page contains an overview of highlights for other pages.
- **ABOUT-US:**This page contains the information about OBS.
- **RECOVERY:** This page contains the recovery form for password and mpin.
- **CARDS:**This page contains the information of the card which are provided by the bank.
- **AD_LOGIN:** This page contains the admin login form from where admin can validate any customer .
- **MONEY TRANSFER:** This page for a customer to send money to another customer.
- **STATEMENT:**This page contains the all transaction information of a customer.
- **VIRTUAL DEBIT CARD :**This page contains the virtual debit card of the customer.

3e. INPUT AND OUTPUT

The main inputs ,outputs and the major function the details are :
INPUT

- Customer can login using username and password.
- Admin can login using admin-id and password.
- Admin validate and add balance in the customers account.
- Customer can modify personal details and change mpin and password.

OUTPUT

- Customer can view personal details, transactions and balance.
- Admin can view all the the details of customer and transactions.

3f. SOFTWARE REQUIREMENT SPECIFICATIONS

Software Requirements Specification provides an overview of the entire project. It is a description of a software system to be developed, laying out functional and nonfunctional requirements. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have clear and thorough understanding of the project to be developed. This is prepared after the detailed communication with project team and the customer.

The developer is responsible for:-

- ✓ Developing the system, which meets the SRS and solving all the requirements of the system?
- ✓ Demonstrating the system and installing the system at client's location after acceptance testing is successful.
- ✓ Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- ✓ Conducting any user training that might be needed for using the system.
- ✓ Maintain the system for a period of one year after installation.

HARDWARE REQUIREMENTS:

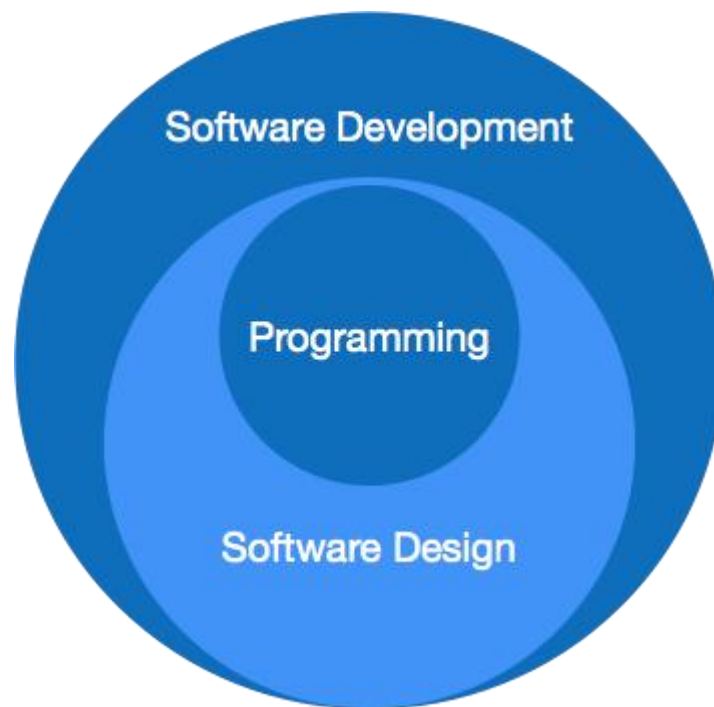
- Computer that has a 1.6GHz or faster processor
- 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine)
- HDD 20 GB Hard Disk Space and Above Hardware Requirements
- 5400 RPM hard disk drive
- DVD-ROM Drive

SOFTWARE REQUIREMENTS:

- WINDOWS OS (XP/2000/2003 or 2000 Server/Vista or 7)
- SQL-DB lite(3)
- Python-Django
- VS-code

3g. SOFTWARE ENGINEERING PARADIGM APPLIED

Software paradigms refer to the methods and steps, which are taken while designing the software. There are many methods proposed and are in work today, but we need to see where in the software engineering these paradigms stand. These can be combined into various categories, though each of them is contained in one another.



Programming paradigm is a subset of Software design paradigm which is further a subset of Software development paradigm.

There are two levels of reliability. The first is meeting the right requirement. A carefully and through systems study is needed to satisfy this aspect of reliability. The second level of systems reliability involves the actual working delivered to the user. At this level, the systems

reliability is interwoven with software engineering and development. There are three approaches to reliability.

1. Error avoidance: Prevents errors from occurring in software.
2. Error detection and correction: In this approach errors are recognized whenever they are encountered and correcting the error by effect of error of the system does not fail.
3. Error tolerance: In this approach errors are recognized whenever they occur, but enables the system to keep running through degraded perform or Applying values that instruct the system to continue process.

Maintenance:

The key to reducing need for maintenance, while working, if possible to do essential tasks.

1. More accurately defining user requirement during system development.
2. Assembling better systems documents.
3. Using some effective methods for designing, processing, and login and communicating information with project team members.
4. Making better use of existing tools and techniques.

SYSTEM DESIGN

4a. DATA FLOW DIAGRAM

A **data flow diagram (DFD)** is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A **DFD** is often used as a preliminary step to create an overview of the system, which can later be elaborated.

DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient system. Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report.

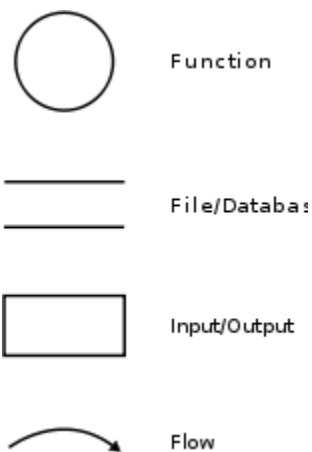
How any system is developed can be determined through a data flow diagram model.

In the course of developing a set of *leveled* data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to identify the transaction data in the data model.

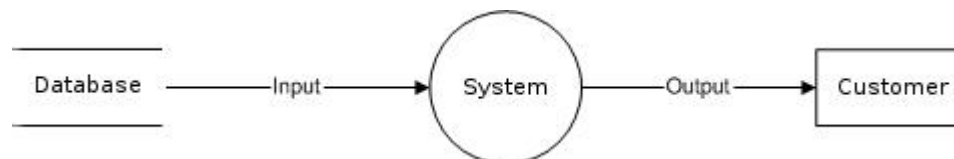
Data flow diagrams can be used in both Analysis and Design phase of the SDLC.

There are different notations to draw data flow diagrams. defining different visual representations for processes, data stores, data flow, and external entities.^[6]

DFD NOTATION



DFD EXAMPLE



Steps to Construct Data Flow Diagram:-

Four Steps are generally used to construct a DFD.

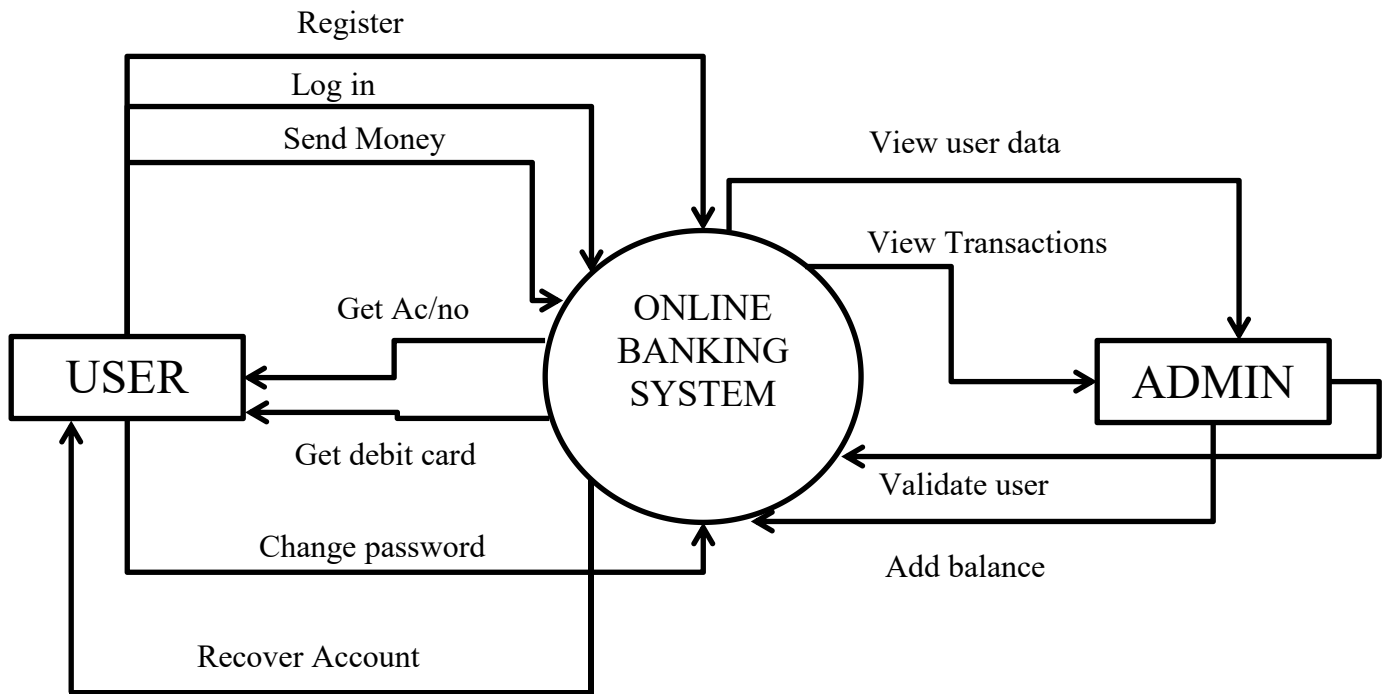
- Process should be named and referred for easy reference. Each name should be representative of the reference.
- The destination of flow is from top to bottom and from left to right.
- When a process is distributed into lower level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram:-

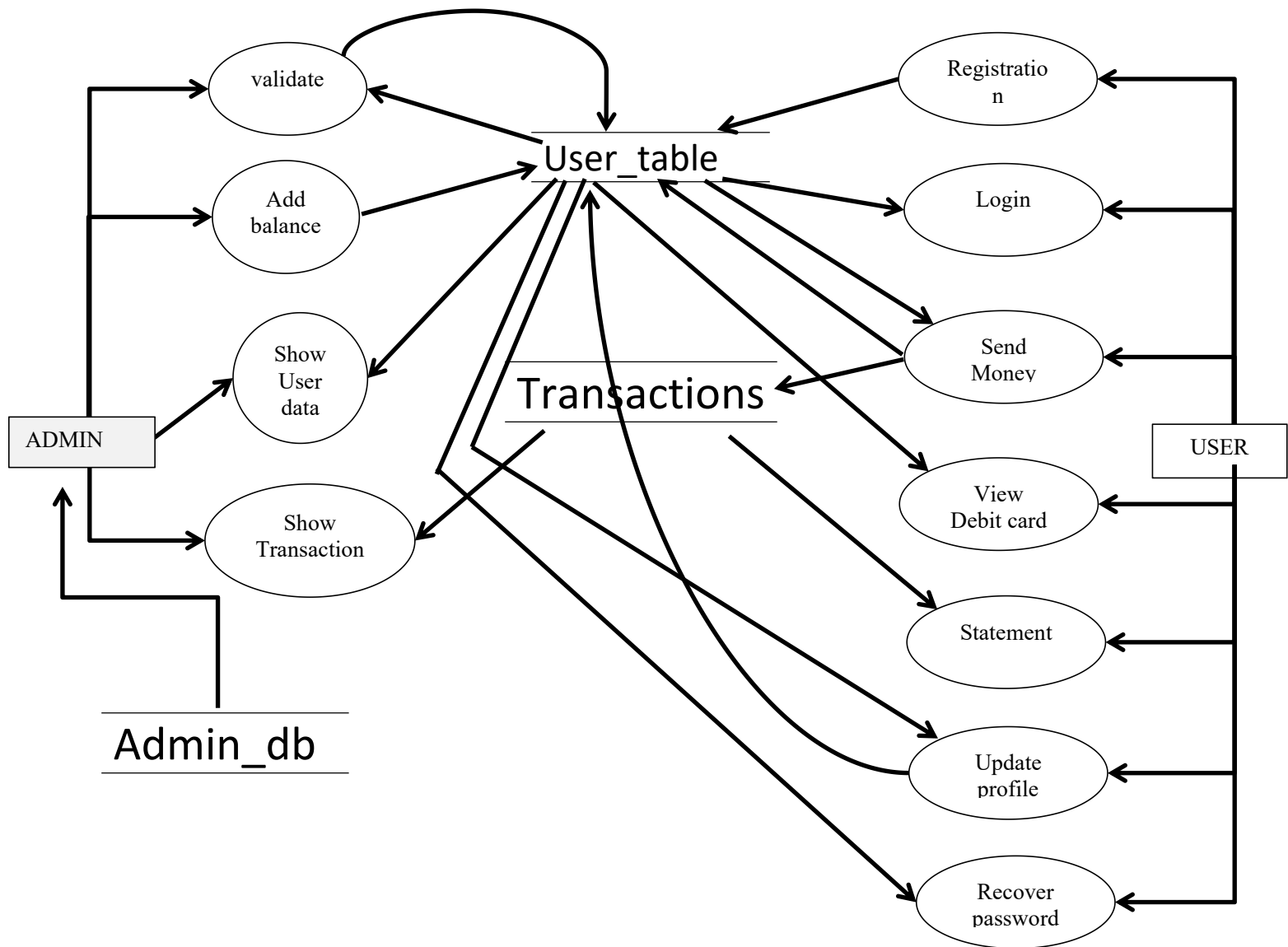
- Arrows should not cross each other.
- Squares, Circles, Files must bear a name.
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

DATA FLOW DIAGRAM

LEVEL-0 DFD DIAGRAM



LEVEL-1 DFD DIAGRAM



4b. SEQUENCE DIAGRAM

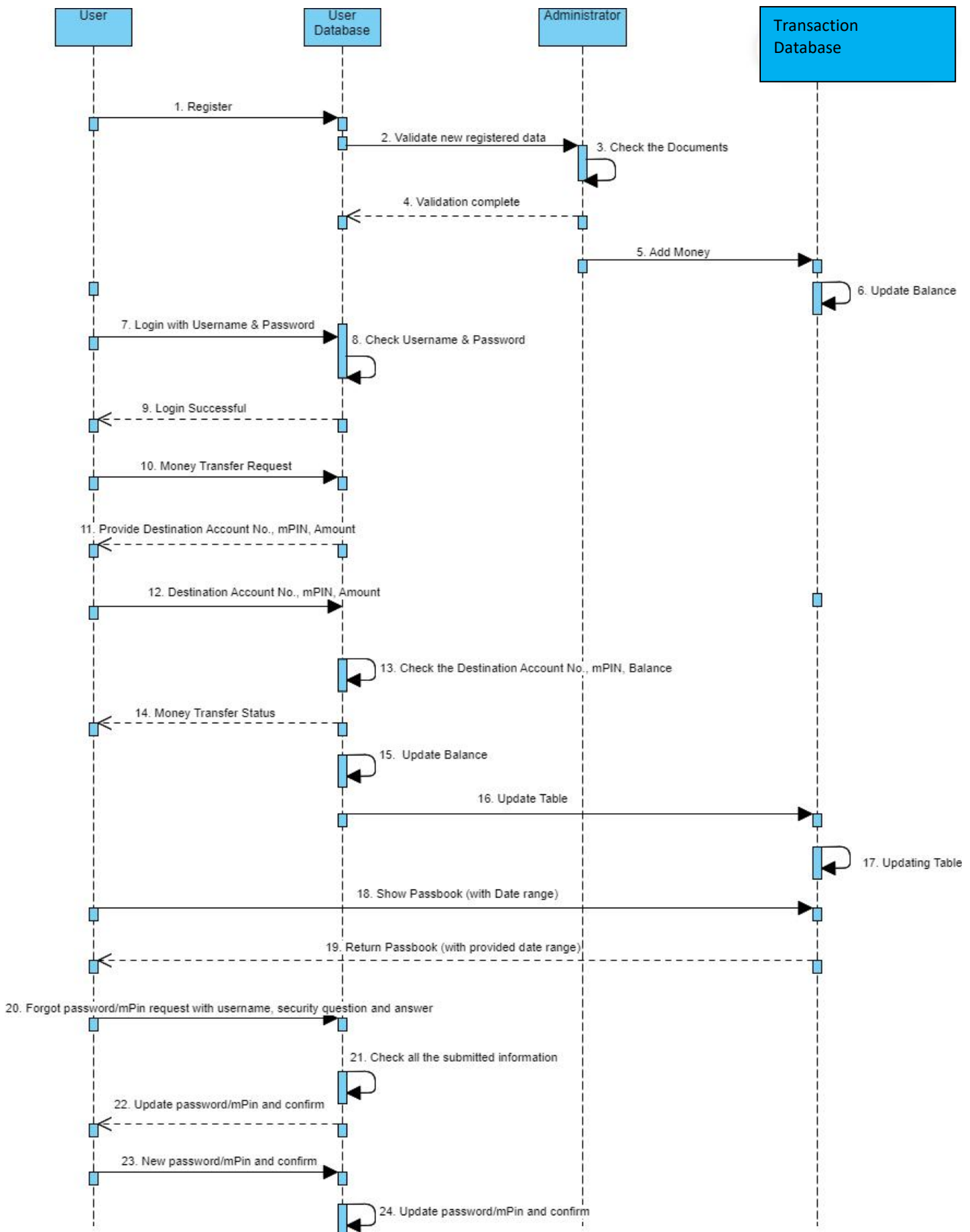
A **Sequence diagram** is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.

Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines.

The following nodes and edges are typically drawn in a **UML sequence diagram**: lifeline, execution-specification, message, fragment, interaction, state invariant, continuation, destruction occurrence.



4c. ENTITY RELATIONSHIP DIAGRAM

In software engineering, an **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. The main components of ER models are entities (things) and the relationships that can exist among them.

An entity–relationship model is the result of using a systematic process to describe and define a subject area of business data. It does not define business process; only visualize business data. The data is represented as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity–relationship diagrams.

An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers are the physical implementation of the relationships.

The three schema approach to software engineering uses three levels of ER models that may be developed.

Conceptual data model

The conceptual ER model normally defines master reference data entities that are commonly used by the organization. Developing

an enterprise-wide conceptual ER model is useful to support documenting the data architecture for an organization. A conceptual ER model may be used as the foundation for one or more logical data models. The purpose of the conceptual ER model is then to establish structural metadata commonality for the master data entities between the set of logical ER models. The conceptual data model may be used to form commonality relationships between ER models as a basis for data model integration.

Logical data model

The logical ER model contains more detail than the conceptual ER model. In addition to master data entities, operational and transactional data entities are now defined. The details of each data entity are developed and the relationships between these data entities are established. The logical ER model is however developed independent of technology into which it can be implemented.

Physical data model

One or more physical ER models may be developed from each logical ER model. The physical ER model is normally developed to be instantiated as a database. Therefore, each physical ER model must contain enough detail to produce a database and each physical ER model is technology dependent since each database management system is somewhat different.

The physical model is normally instantiated in the structural metadata of a database management system as relational database objects such as database tables, database indexes such as unique keyindexes, and

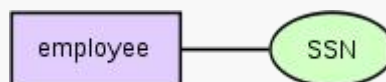
database constraints such as a foreign key constraint or a commonality constraint. The ER model is also normally used to design modifications to the relational database objects and to maintain the structural metadata of the database.

The first stage of information system design uses these models during the requirements analysis to describe information needs or the type of information that is to be stored in a database. The data modeling technique can be used to describe any ontology (i.e. an overview and classifications of used terms and their relationships) for a certain area of interest. In the case of the design of an information system that is based on a database, the conceptual data model is, at a later stage (usually called logical design), mapped to a logical data model, such as the relational model; this in turn is mapped to a physical model during physical design. Note that sometimes, both of these phases are referred to as "physical design". It is also used in database management system.

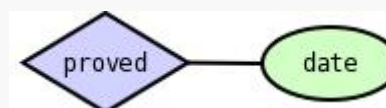
Entity–relationship modeling



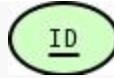
Two related entities



An entity with an attribute



A relationship with an attribute

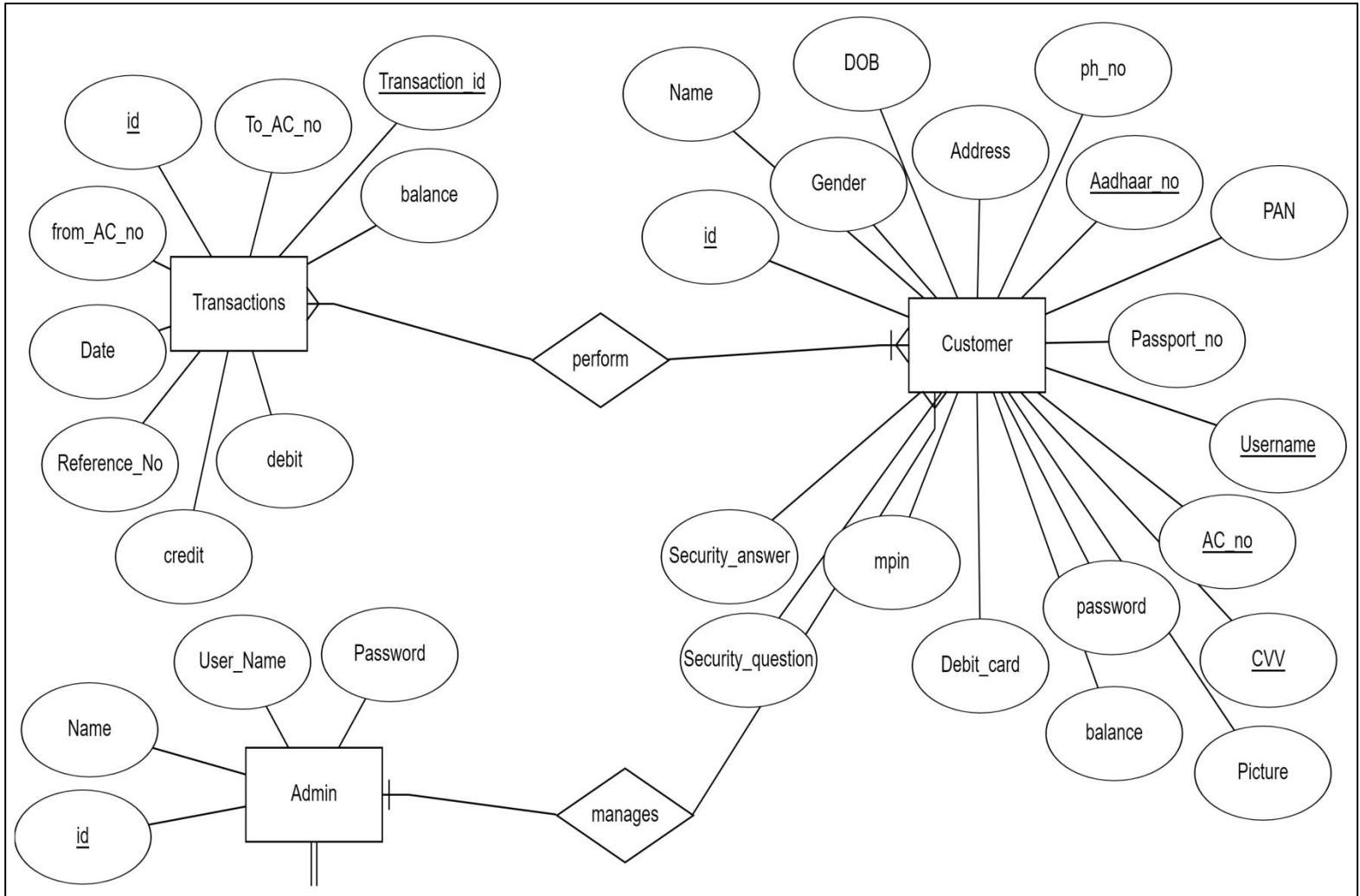


[Primary key](#)

Cardinality constraints are expressed as follows:

- A double line indicates a *participation constraint*, totality or subjectivity : all entities in the entity set must participate in *at least one* relationship in the relationship set;
- an arrow from entity set to relationship set indicates a key constraint, i.e. injectivity: each entity of the entity set can participate in *at most one* relationship in the relationship set;
- A thick line indicates both, i.e. bijection: each entity in the entity set is involved in *exactly one* relationship.
- An underlined name of an attribute indicates that it is a key: two different entities or relationships with this attribute always have different values for this attribute.

ER-DIAGRAM



4d. USE CASE DIAGRAM

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So to model the entire system numbers of use case diagrams are used.

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view.

So in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

How to draw Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases.

So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system.

The actors can be human user, some internal applications or may be some external applications. So in a brief when we are planning

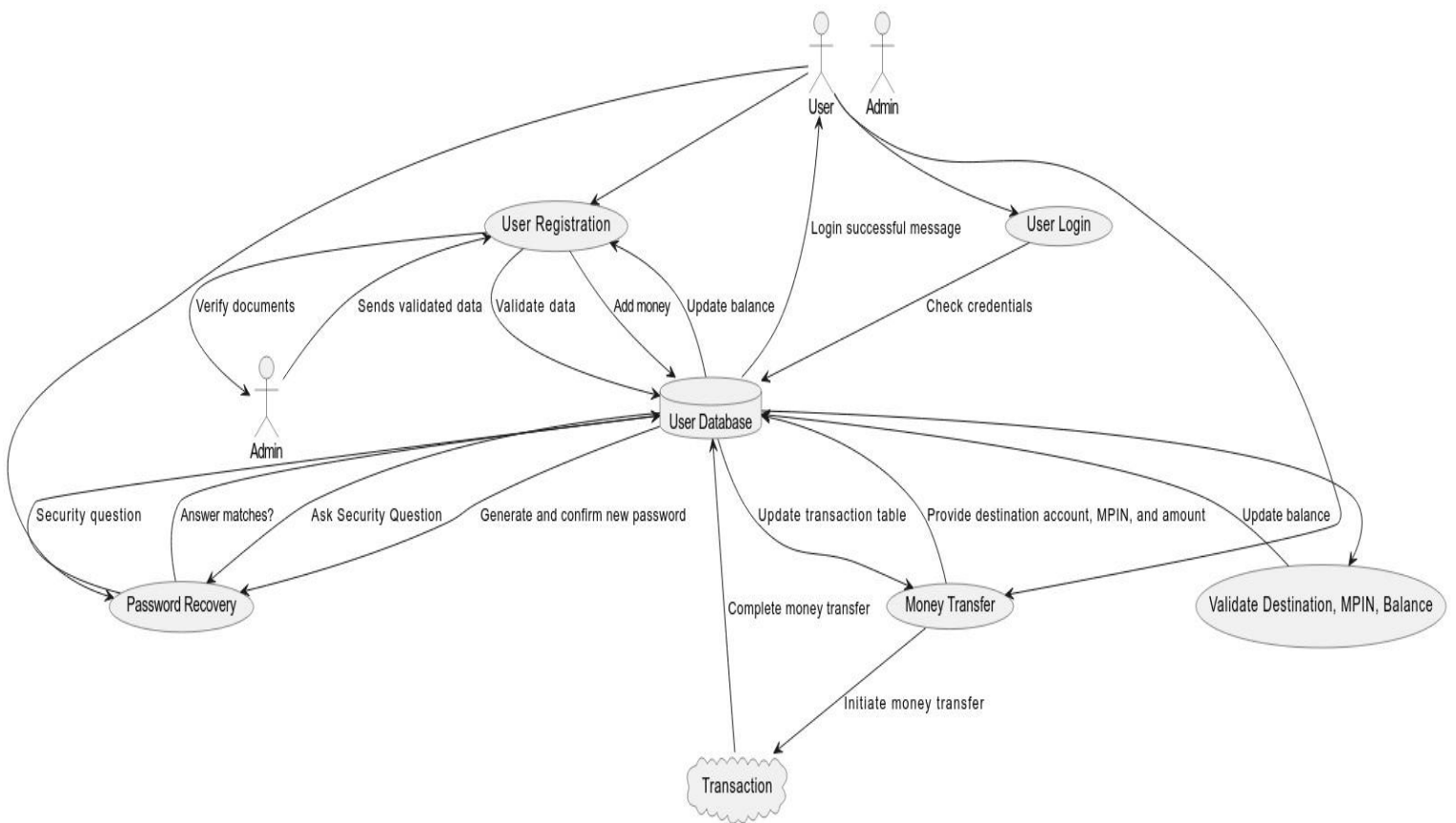
to draw a use case diagram we should have the following items identified.

- Functionalities to be represented as an use case
- Actors
- Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. So after identifying the above items we have to follow the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important. So the name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships. Because the main purpose of the diagram is to identify requirements.
- Use note whenever required to clarify some important points.

USE CASE DIAGRAM



4.e MODULARIZATION DETAILS

As Modularization has gained increasing focus from companies outside its traditional industries of aircraft and automotive, more and more companies turn to it as strategy and product development tool. I intend to explain the importance aspects of modularization and how it should be initiated within a company. After determining the theoretical steps of modularization success described in literature, I intend to conduct a multiple case study of companies who have implemented modularization in order to find how real world modularization was initiated and used to improve the company's competitiveness. By combining theory and practical approach to modularization I will derive at convergence and divergence between theoretical implementation to modularization and real world implementation to modularization. This gives a valuable input for both implantations in companies as well as new aspects to be further.

DATA INTEGRITY AND CONSTRAINTS

Data integrity is normally enforced in a database system by a series of integrity constraints or rules. Three types of integrity constraints are an inherent part of the relational data model: entity integrity, referential integrity and domain integrity:

- Entity integrity concerns the concept of a primary key. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.
- Concerns the concept of a foreign key. The referential integrity rule states that any foreign-key value can only be in one of two states. The usual state of affairs is that the foreign-key value refers to a primary key value of some table in the database.

Occasionally, and this will depend on the rules of the data owner, a foreign-key value can be null. In this case we are explicitly saying that either there is no relationship between the objects represented in the database or that this relationship is unknown.

- *Domain integrity* specifies that all columns in a relational database must be declared upon a defined domain. The primary unit of data in the relational data model is the data item. Such data items are said to be non-decomposable or atomic. A domain is a set of values of the same type.

4f. DATABASE DESIGN



























A database is an organized mechanism that has capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is two level processes. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called information Level design and it is taken independent of any individual DBMS.











In the following snapshots we display the way we have used SQL Server as the back-end RDBMS for our project and the various entities that have been used along with their table definition and table data.

DATA DICTIONARY




User registration form table

 First_name	varchar(50)	"First_name" varchar(50) NOT NULL
 Middle_name	varchar(50)	"Middle_name" varchar(50) NOT NULL
 Last_name	varchar(50)	"Last_name" varchar(50) NOT NULL
 Gender	varchar(20)	"Gender" varchar(20) NOT NULL
 Dob	date	"Dob" date NOT NULL
 Ph_no	integer	"Ph_no" integer NOT NULL
 Email	varchar(254)	"Email" varchar(254) NOT NULL
 Address	varchar(200)	"Address" varchar(200) NOT NULL
 City	varchar(30)	"City" varchar(30) NOT NULL
 District	varchar(50)	"District" varchar(50) NOT NULL
 State	varchar(30)	"State" varchar(30) NOT NULL
 Pin	integer	"Pin" integer NOT NULL
 Aadhaar	integer	"Aadhaar" integer NOT NULL
 PAN	varchar(15)	"PAN" varchar(15) NOT NULL
 passport	varchar(20)	"passport" varchar(20) NOT NULL
 pic	varchar(100)	"pic" varchar(100)
 username	varchar(10)	"username" varchar(10) NOT NULL
 password	varchar(15)	"password" varchar(15) NOT NULL
 mpin	integer	"mpin" integer NOT NULL
 question	varchar(60)	"question" varchar(60) NOT NULL
 answer	varchar(20)	"answer" varchar(20) NOT NULL
 ac_no	varchar(15)	"ac_no" varchar(15) NOT NULL
 balance	integer	"balance" integer NOT NULL
 flag	integer	"flag" integer NOT NULL
 card	integer	"card" integer NOT NULL
 ...	integer	"..." integer NOT NULL

Transaction Table

 Transaction	CREATE TABLE "Transection" ("id" integer NOT NULL PRI	
 id	integer	"id" integer NOT NULL
 fac_no	varchar(15)	"fac_no" varchar(15) NOT NULL
 tac_no	varchar(15)	"tac_no" varchar(15) NOT NULL
 reference	varchar(200)	"reference" varchar(200) NOT NULL
 tr_id	varchar(30)	"tr_id" varchar(30) NOT NULL
 credit	integer	"credit" integer NOT NULL
 debit	integer	"debit" integer NOT NULL
 balance	integer	"balance" integer NOT NULL
 date	date	"date" date NOT NULL

Admin login table

▼  AdminDb	CREATE TABLE "AdminDb" ("id" intege	
 id	integer	"id" integer NOT NULL
 name	varchar(50)	"name" varchar(50) NOT NULL
 username	varchar(20)	"username" varchar(20) NOT NULL
 password	varchar(20)	"password" varchar(20) NOT NULL

Admin table data

	id	name	username	password
	Filter	Filter	Filter	Filter
1	1	Arup	arup	12345
2	2	Anirban	anirban	54321

Transaction table data

	id	fac_no	tac_no	reference	tr_id	credit	debit	balance	date
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	5	482655224994575	1234		jUchk95554	0	20	80	2023-08-22
2	6	482655224994575	1234		QgfBgpGQUH06974	0	20	60	2023-08-22
3	7	482655224994575	12345		TZpqXcNKI77238	0	10	50	2023-08-23
4	8	482655224994575	12345		dvCtZrATPPUqwk07880	0	10	40	2023-08-23
5	9	424281872682052	12345		yElLDkOb25196	0	100	100	2023-08-23
6	10	482655224994575	12345		QmzePqpGPkBsQJ97802	0	10	30	2023-08-23
7	11	482655224994575	1234		DmWvaDI90360	0	10	20	2023-08-23
8	12	482655224994575	1234		vpCuvSCPejj83799	0	10	10	2023-08-23
9	13	482655224994575	1234		ohfxcZEB44475	0	2	8	2023-08-23
10	14	482655224994575	123		IBSWGYhvYZ56075	0	1	7	2023-08-23
11	15	123	482655224994575		IBSWGYhvYZ56097	2	0	9	2023-08-23
12	16	482655224994575	1234		QsbvhnzEWN86660	0	1	6	2023-08-23
13	17	548953158242886	1234		JHJENyfvnNaGz63870	0	30	70	2023-08-23
14	18	548953158242886	123		OXxjSujROeVofbb46666	0	10	60	2023-08-23
15	19	12345	482655224994575		OXxjSujROeVofbb46677	15	0	150	2023-08-29
16	20	482655224994575	12345		HXUPAh65901	0	10	140	2023-08-29
17	21	self	482655224994575		uwtPfZ02456	0	0	150	2023-08-29
18	22	self	482655224994575		pkOEB88582	5	0	155	2023-08-29

User Table

PAN	passport	pic	username	password	mpin	question	answer	ac_no	balance	flag	card	cvv
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
asdfghjk	asdfghjk	ARUP2.jpg	arghya	1234	1111	2	cat	166073235627923	310	1	5209651127170780	697
jkhsdfhksdf	klhjsdhjsdf	3rd.webp	saikat	12345	1111	2	cat	604580762026875	40	1	6557572904906246	173

OUTPUT SCREEN

5a. USER INTERFACE DESIGN

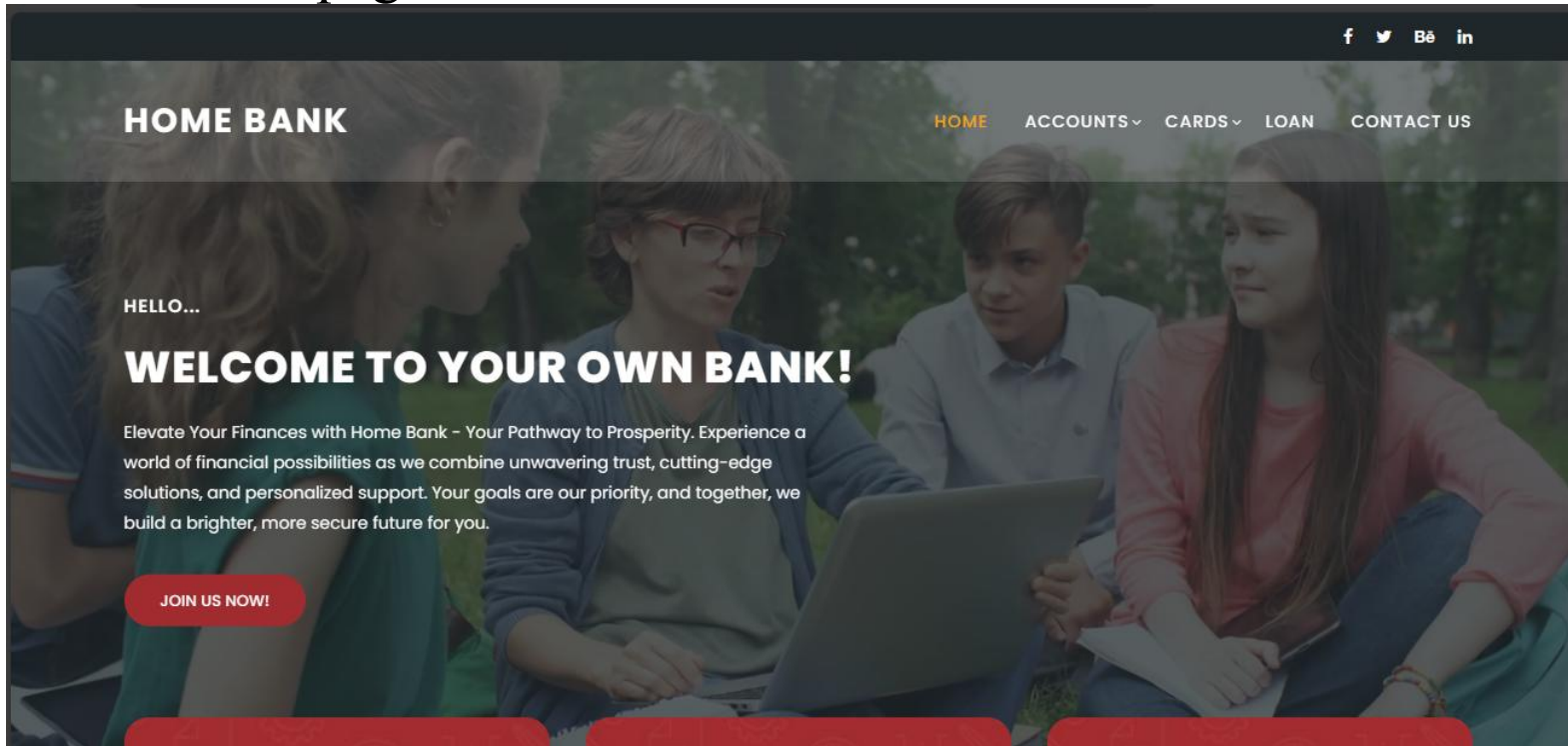
User interface design (UID) or **user interface engineering** is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge. As a result, designers tend to specialize in certain types of projects and have skills centered on their expertise, whether that be software design, user research, web design, or industrial design.

SNAPSHOTS

Home page



Admin login page

A screenshot of the Admin login page. The page has a dark header with the text 'Welcome back, please log in' in white. Below the header, there is a white login form with a shadow. The form contains two input fields: 'Username' and 'Password'. Below the 'Password' field is a dark button with the text 'Log In' in white.

Admin_login code

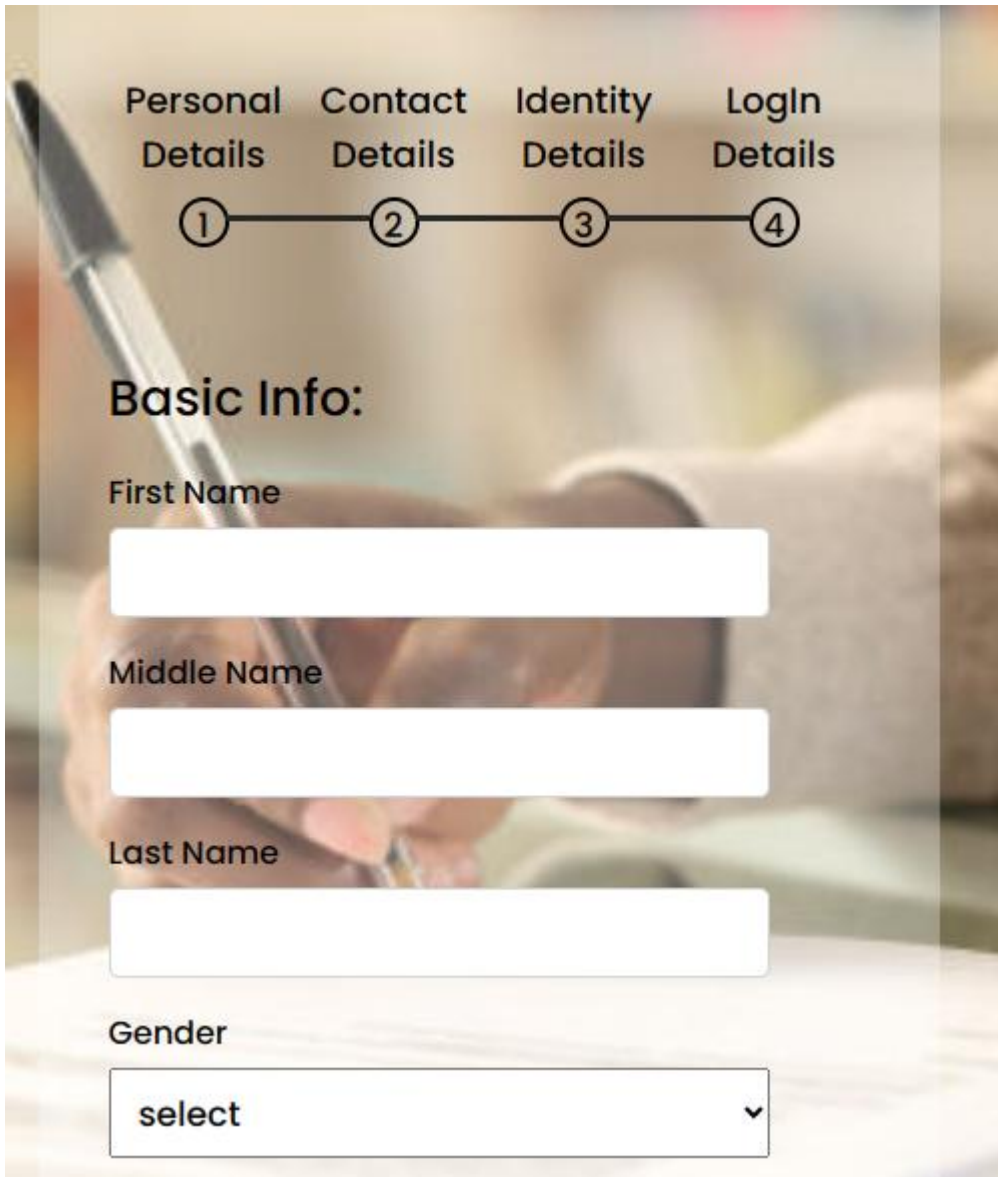
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
  }
  .header {
    background-color: #333;
    color: white;
    text-align: center;
    padding: 1rem;
  }
  .container {
    max-width: 400px;
    margin: 2rem auto;
    padding: 1rem;
    background-color: white;
    border-radius: 5px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
  }
}
```



```
.form-group {
  margin-bottom: 1rem;
}
label {
  display: block;
  font-weight: bold;
  margin-bottom: 0.5rem;
}
input[type="text"],
input[type="password"] {
  width: 100%;
  padding: 0.5rem;
  border: 1px solid #ccc;
  border-radius: 3px;
}
button {
  background-color: #333;
  color: white;
  padding: 0.5rem 1rem;
  border: none;
  border-radius: 3px;
  cursor: pointer;
}
button:hover {
  background-color: #555;
}
</style>
</head>
<body>
```

```
<div class="header">
  <h1>Welcome back, please log in</h1>
</div>
<div class="container">
  <form method="post">
    {% csrf_token %}
    {% if error_message %}
      <p class="error">{{ error_message }}</p>
    {% endif %}
    <div class="form-group">
      <label for="username">Username</label>
      <input type="text" id="username" name="username" required>
    </div>
    <div class="form-group">
      <label for="password">Password</label>
      <input type="password" id="password" name="password" required>
    </div>
    <button type="submit">Log In</button>
  </form>
</div>
</body>
</html>
```

User Registration page

A mockup of a user registration page. At the top, there is a progress bar with four steps: 1. Personal Details, 2. Contact Details, 3. Identity Details, and 4. Login Details. Below this, the 'Basic Info:' section contains four input fields: First Name, Middle Name, Last Name, and Gender. The Gender field is a dropdown menu with 'select' as the current value and a downward arrow icon.

Personal Details Contact Details Identity Details Login Details

① — ② — ③ — ④

Basic Info:

First Name

Middle Name

Last Name

Gender

select

▼

**Personal
Details**

**Contact
Details**

**Identity
Details**

**Login
Details**



Contact Info:

Phone Number

Email Address

Address

City

Personal Details

Contact Details

Identity Details

Login Details

3

4

Aadhaar No.

PAN

Passport No.

Upload photo

Choose File

No file chosen

PREVIOUS

NEXT

Personal Details

Contact Details

Identity Details

Login Details

4

Login Details:

Username

Password

Confirm Password

4 digit mpin

Security Questions?

User registration page code

```
{% load static %}
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Home bank/Register</title>
    <link rel="stylesheet" href="{% static 'mcss/style.css' %}">
    <script src="https://kit.fontawesome.com/a076d05399.js"></script>
  </head>
  <body>
    <div>
      <nav class="nav">
        <i class="uil uil-bars navOpenBtn"></i>
        <a href="index" class="logo">Home Bank</a>
        <ul class="nav-links">
          <i class="uil uil-times navCloseBtn"></i>
          <li><a href="index">Home</a></li>
          <li><a href="#">Contact Us</a></li>
        </ul>
        <i class="uil uil-search search-icon" id="searchIcon"></i>
      </nav>
    </div>
    <div class="container">
      <header>Register Here!</header>
```

```

<h4>{{m}}</h4>
<div class="progress-bar">
  <div class="step">
    <p>Personal Details</p>
    <div class="bullet">
      <span>1</span>
    </div>
    <div class="check fas fa-check"></div>
  </div>
  <div class="step">
    <p>Contact Details</p>
    <div class="bullet">
      <span>2</span>
    </div>
    <div class="check fas fa-check"></div>
  </div>
  <div class="step">
    <p>Identity Details</p>
    <div class="bullet">
      <span>3</span>
    </div>
    <div class="check fas fa-check"></div>
  </div>
  <div class="step">
    <p>LogIn Details</p>
    <div class="bullet">
      <span>4</span>
    </div>
    <div class="check fas fa-check"></div>
  </div>
</div>

```

```

<div class="form-outer">
  <form action="register">
    <div class="page slide-page">
      <div class="title">Basic Info:</div>
      <div class="field">
        <div class="label">First Name</div>
        <input type="text" name="fname">
      </div>
      <div class="field">
        <div class="label">Middle Name</div>
        <input type="text" name="mname">
      </div>
      <div class="field">
        <div class="label">Last Name</div>
        <input type="text" name="lname">
      </div>
      <div class="field">
        <div class="label">Gender</div>
        <select name="gen">
          <option value="">select</option>
          <option value="Male">Male</option>
          <option value="Female">Female</option>
          <option value="Other">Other</option>
        </select>
      </div>
      <div class="field">
        <div class="label">Date of Birth</div>
        <input type="Date" name="dob">
      </div>
      <div class="field">
        <button class="firstNext next">Next</button>

```



```
</div>
    <div class="field">
        <h4><a href="sign_in">already have an account? click here</a></h4>
    </div>
</div>
```

```
<div class="page">
    <div class="title">Contact Info:</div>
```

```
    <div class="field">
        <div class="label">Phone Number</div>
        <input type="Number" name="ph">
    </div>
```

```
        <div class="field">
            <div class="label">Email Address</div>
            <input type="text" name="email">
        </div>
```

```
            <div class="field">
                <div class="label">Address</div>
                <input type="text" name="add">
            </div>
```

```
                <div class="field">
                    <div class="label">City</div>
                    <input type="text" name="city">
                </div>
```

```
                    <div class="field">
                        <div class="label">District</div>
                        <input type="text" name="dist">
                    </div>
```

```
                        <div class="field">
                            <div class="label">State</div>
```

```
<select name="state">
    <option value="">Select state</option>
<option value="AN">Andaman and Nicobar Islands</option>
<option value="AP">Andhra Pradesh</option>
<option value="AR">Arunachal Pradesh</option>
<option value="AS">Assam</option>
<option value="BR">Bihar</option>
<option value="CH">Chandigarh</option>
<option value="CT">Chhattisgarh</option>
<option value="DN">Dadra and Nagar Haveli</option>
<option value="DD">Daman and Diu</option>
<option value="DL">Delhi</option>
<option value="GA">Goa</option>
<option value="GJ">Gujarat</option>
<option value="HR">Haryana</option>
<option value="HP">Himachal Pradesh</option>
<option value="JK">Jammu and Kashmir</option>
<option value="JH">Jharkhand</option>
<option value="KA">Karnataka</option>
<option value="KL">Kerala</option>
<option value="LA">Ladakh</option>
<option value="LD">Lakshadweep</option>
<option value="MP">Madhya Pradesh</option>
<option value="MH">Maharashtra</option>
<option value="MN">Manipur</option>
<option value="ML">Meghalaya</option>
<option value="MZ">Mizoram</option>
<option value="NL">Nagaland</option>
<option value="OR">Odisha</option>
<option value="PY">Puducherry</option>
<option value="PB">Punjab</option>
```

```
<option value="RJ">Rajasthan</option>
<option value="SK">Sikkim</option>
<option value="TN">Tamil Nadu</option>
<option value="TG">Telangana</option>
<option value="TR">Tripura</option>
<option value="UP">Uttar Pradesh</option>
<option value="UT">Uttarakhand</option>
<option value="WB">West Bengal</option>
```

```
</select>
```

```
</div>
```

```
<div class="field">
```

```
<div class="label">Pin Code</div>
```

```
<input type="Number" name="pin">
```

```
</div>
```

```
<div class="field btns">
```

```
<button class="prev-1 prev">Previous</button>
```

```
<button class="next-1 next">Next</button>
```

```
</div>
```

```
</div>
```

```
<div class="page">
```

```
<div class="field">
```

```
<div class="label">Aadhaar No.</div>
```

```
<input type="text" name="an">
```

```
</div>
```

```
<div class="field">
```

```
<div class="label">PAN</div>
```

```
<input type="text" name="pan">
```

```
</div>
```

```
<div class="field">
```

```

    <div class="label">Passport No.</div>
    <input type="text" name="pno">
</div>
<div class="field">
    <div class="label">Upload photo</div>
    <input type="file" name="pic" accept="image/*">
</div>

<div class="field btns">
    <button class="prev-2 prev">Previous</button>
    <button class="next-2 next">Next</button>
</div>
</div>

<div class="page">
    <div class="title">Login Details:</div>
    <div class="field">
        <div class="label">Username</div>
        <input type="text" name="uname">
    </div>
    <div class="field">
        <div class="label">Password</div>
        <input type="password" id="password" name="password">
    </div>
        <div class="field">
            <div class="label">Confirm Password</div>
            <input type="password" id="con_password" name="con_password">
        </div>
    <div class="field">
        <div class="label">4 digit mpin</div>
        <input type="password" name="mpin" minlength="4" maxlength="4">

```

```

    </div>
        <div class="field">
            <div class="label">Security Questions?</div>
            <select name="scq">
                <option value="0">Mother's maiden name.</option>
                <option value="1">Name of town where you were
born.</option>
                <option value="2">Name of first pet.</option>
            </select>
        </div>
        <div class="field">
            <div class="label">Answer</div>
            <input type="text" name="sa">
        </div>
        <div class="field">
            <div>terms & Conditions</div>
            <input type="checkbox">
        </div>
        <div class="field btns">
            <button class="prev-3 prev">Previous</button>
            <button class="submit">Submit</button>
        </div>
    </div>
</form>
</div>
</div>

<script>
    var password = document.getElementById("password")
, confirm_password = document.getElementById("con_password");

```

```
function validatePassword(){  
  if(password.value != confirm_password.value) {  
    confirm_password.setCustomValidity("Passwords Don't Match");  
  } else {  
    confirm_password.setCustomValidity("");  
  }  
}
```

```
password.onchange = validatePassword;  
confirm_password.onkeyup = validatePassword;  
</script>  
<script src="{% static 'mjs/script.js' %}"></script>
```

```
</body>  
</html>
```

IMPLEMENTATION AND TESTING

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" product validation. It should be thorough enough to be useful but not so thorough that no one outside the test group will read it.

6a. INTRODUCTION

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well-planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

The implementation is the final and important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

6b. OBJECTIVES OF TESTING:

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

Process Overview

The following represents the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
2. Identify which particular test(s) will be used to test each module.
3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test.

5. Document the test case configuration, test data, and expected results.
6. Perform the test(s).
7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
8. Successful unit testing is required before the unit is eligible for component integration/system testing.
9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

6c. TEST CASES

A test case is a document that describe an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular such as test case identifier, test condition, input data

Requirement expected results. The process of developing test cases can help find problems in the requirement or design of an application, since it requires completely thinking through the operation of the application.

TESTING STEPS

Unit Testing:

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

Integration Testing:

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

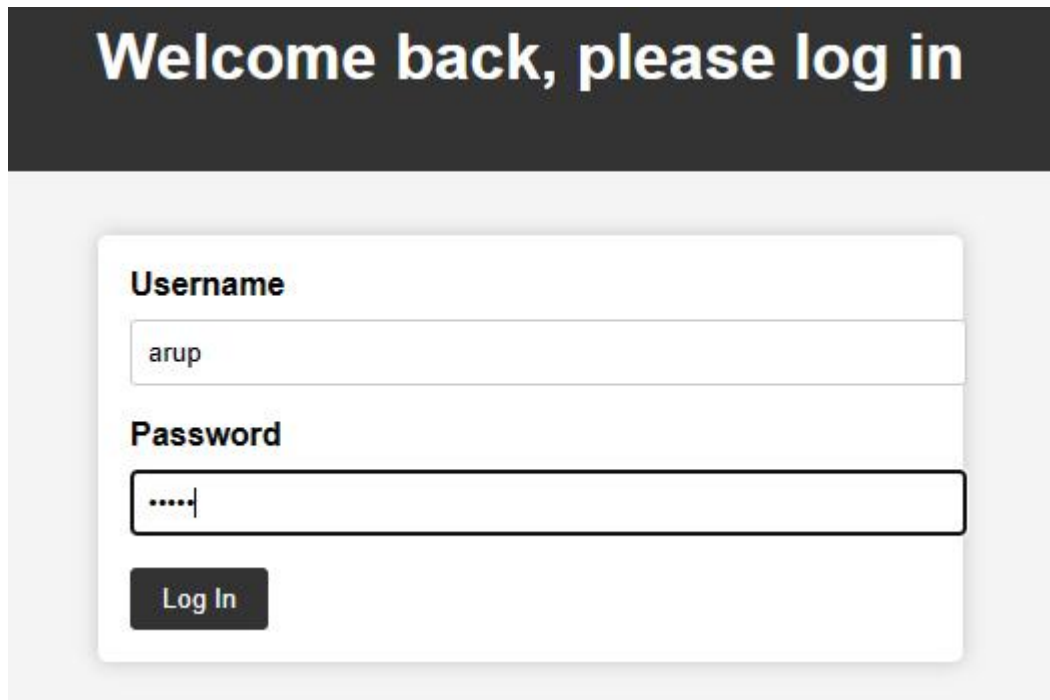
Validation:

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

- a)The function or performance characteristics confirm to specification and are accepted.
- b)A deviation from specification is uncovered and a deficiency lists is created.
- c)Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

Tested By:		Arup Bhowmik
Test Type		Unit Testing
Test Case Number		1
Test Case Name		Admin Identification
Test Case Description		The admin should enter his/ her accurate admin id and password so that he/she can able to go for the further options. The test case will check the application for the same since a user can only login with the correct user id, password.
Item(s) to be tested		
1	Verification of the user id and password with the record in the database.	
Specifications		
Input		Expected Output/Result
1) Correct User id and password		1) Successful login
2) Incorrect Id or Password		2) Failure Message

SNAPSHOT OF ADMIN LOGIN



A screenshot of an admin login interface. At the top, a dark grey banner contains the text "Welcome back, please log in" in white. Below this, a white login form is centered on a light grey background. The form has two input fields: "Username" with the text "arup" and "Password" with masked characters ".....". A dark grey "Log In" button is positioned at the bottom of the form.

Welcome back, please log in

Username

arup

Password

.....

Log In

Tested By:		Anirban Samanta
Test Type		Unit Testing
Test Case Number		2
Test Case Name		Registration of user ,send money, statement check , logout, debit card check.
Test Case Description		Admin will enter the details in the registration form and the credentials will be checked by the validators used in the form. After all the credentials are validated then only the form will be submitted and data will be stored in the database
Item(s) to be tested		
1	Required fields in the form are not empty, validation of proper credentials	
Specifications		
Input		Expected Output/Result
1) User id, name, password, phone, address, email.		1) Successful registration
2) Empty field, Invalid entry		2) Failure Message

SNAPSHOTS OF STUDENT REGISTRATION

Personal Details Contact Details Identity Details Login Details

① — ② — ③ — ④

Basic Info:

First Name

Middle Name

Last Name

Gender

select

▼

6 d. WHITE BOX TESTING

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

6e. BLACK BOX TESTING

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case we will focus only on function validation and performance. And in both cases we will use the black-box method of testing.

6f. 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 a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

User Acceptance Testing:

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

This is done in regard to the following point:

- a) Input Screen Design
- b) Output Screen Design
- c) Format of reports and other outputs.

6g. GOAL OF TESTING

"Program testing can be used to show the presence of bug, but never to show their absence." If the results delivered by the system are different from the expected ones then the system is incorrect and these bugs should be fixed.

6h. INTEGRATION TEST REPORTS

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

- Whether the software matches with the specification (i.e. process base) to verify the product.
- Whether this software in one client what wants (i.e. product base) to validate the product.

Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behaviour if exists. Later all the module was integrated and flat file was generated.

FUNCTIONAL TESTING

These are the points concerned during the stress test:


- Nominal input: character is in putted in the place of digits and the system has to flash the message "Data error"
- Boundary value analysis: exhaustive test cases have designed to create an output report that produces the maximum (and minimum) allowable number of table entries.

Testing Method Used

We have adopted a testing method which is a mix of both (structural) and black box (functional) testing. For modules we have adopted white box testing. Then we integrated the module into sub - systems and further into the system. These we adopted black box testing for checking the correctness of the system.

Requirements Validated and Verified:

- The data is getting entered properly into database.
- The Screens are being loaded correctly
- The Various functions specified are being performed completely.



SYSTEM SECURITY MEASURES

8.a DATABASE SECURITY

System security measure is meant to be provided to make your system reliable and secured from unauthorized user may create threats to the system. So you should follow some security measures. We have used security levels in database level at system level.

8.b SYSTEM SECURITY

If we talk about the system security in our proposed system we have implemented with the help of maintain the session throughout the system's use. Once a user has logged out than he/she will not be able to perform any task before signing back again.

A high level of authentic login is given to the system so this is a very tedious task to enter without authorization and authentication.

8c. LIMITATIONS:

- ✓ Since it is an online project, customers need internet connection to use it.
- ✓ People who are not familiar with computers can't use this software.
- ✓ Customer must have valid email and phone number.

9. CONCLUSION

This project has been appreciated by all the users in the organization. It is easy to use, since it uses the GUI provided in the user dialog. User friendly screens are provided. The usage of software increases the efficiency, decreases the effort. It has been efficiently employed as a Site management mechanism. It has been thoroughly tested and implemented.

10. FUTURE SCOPE AND FURTHER ENHANCEMENTS

In future we would like to keep working on this project and make new additions to provide users with more advanced features and more detailed information. We have set our sights on the following additions in future:-

1. Adding Loan Section to the project.
2. Online bill payments.
3. Add upi option.

11. BIBLIOGRAPHY

- <https://www.w3schools.com>
- <https://www.slideshare.com>
- <https://www.scribd.com>
- <https://www.tutorialspoint.com>
- <https://www.youtube.com>

THANK YOU

