# Acknowledgement

**Keep away from people who try to belittle your ambitions. Small people always do that, but the really great make you feel that you too, can become great.**

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted co-operation and have helped me in completing this project successfully.

First of all, I would like to thank **Mr. Nitin Mehta, Director (Ducat Noida Pvt. Ltd)** for creating opportunities to undertake me in the esteemed organization.

Special thanks to **Mr. Aditya,** Project Manager for all the help and guidance extended to me by him in every stage during my training. His inspiring suggestions and timely guidance enabled me to perceive the various aspects of the project in a new light.

I am highly indebted and graceful to **Mr. Madhur (Team Leader)** for his strict supervision, constant encouragement, inspiration and guidance, which ensure the worthiness of my work. Working under them was an enrich experience. I express my sincere thanks to my senior of software development department for their encouragement and valuable suggestion.

I would also thank to my **Er. Khushboo,** HOD (Computer Science Dept.) & My internal project guide **Er. Khushboo** who guided me a lot in completing this project. I would also like to thank my parents & project mate for guiding and encouraging me throughout the duration of the project.

In all I found a congenial work environment in **Ducat Noida.** and this completion of the project will mark a new beginning for me in the coming days.

**Name: Shailendra Kumar Rai**

**Diploma: 5th Sem**

**Er No.: E20224135500015**

**DECLARATION**

I, **Shailendra Kumar Rai** hereby declare that the report of the project entitled **“Banking Automation”** has not presented as a part of any other academic work to get my degree or certificate except **Prasad** **polytechnic** for the fulfillment of the requirements for the degree of Diploma.

CERTIFICATE OF ORIGINALITY

This is to certify that the project entitled “**Banking Automation”** being submitted for the partial fulfillment of degree ofDiploma, session 2022-2023, is a record of work carried out by **Shailendra Kumar Rai** under my guidance and supervision.

This is to further certify that the student has attended the **Ducat, Noida** for the **2** months after the fifth

semester theory exams. His work has been satisfactory and

commendable.

I wish him success in his life.

**Date: 24/09/2022**

**Place: Noida**

CONTENTS

S.N. Title

1. Introduction
   1. About The Project
2. System Analysis
   * + 1. 3.1 Scope of Investigation
       2. 3.2 Problems Faced
       3. 3.3 Proposed System

3.4.1. Feasibility Study

3.4.2. Technical Feasibility

* + - 1. 3.4.3. Economic Feasibility
      2. 3.4.4. Behavioral Feasibility

1. Introduction

4.1. Logical Design

4.2. Physical Design

4.3. Module Design

4.4. Input Design

4.5. Output Design

4.6. Data Flow Diagram

4.7. Database Design

4.8. Relation Database Management system (RDBMS)

4.9. Table Structure

1. Software Environment
   * + 1. 5.1. Software and Hardware Requirement
          1. 5.2. Hardware Specifications
          2. 5.3. Software Specifications
          3. 5.4. Project Architecture and Platform
          4. 5.5. Coding Standards
          5. 5.6. Naming Convention
          6. 5.7. Labels and comments
          7. 5.8. Sample codes
2. Testing Introduction

6.1. Test Plans

6.1.1. Unit Testing

6.1.2. Integration Testing

6.1.3. Validation Testing or System Testing

6.1.4. Output Testing or User Acceptance Testing

1. Implementation
   * 1. 7.1. Implementation Procedures
     2. 7.1.1. User Training

7.1.2. Training On the Application Software

7.1.3. Operational Document

7.1.4. System Maintenance

1. Screen Shots
2. Source-Code
3. Conclusion
4. Scope For Future Enhancement
5. Bibliography

**INTRODUCTION ABOUT THE PROJECT**

In the bank all the activities are being done manually. As the bank widens its services & it finds difficult to manage its operations manually and hence this leads to the automation of some of its operations.

Banking Information system is a windows-based applications. This project mainly deals with managing accounts and their related operations.

Tasks involved in this project are opening the user accounts, recording the account holder’s transactions, depositing, withdrawing amount.

**Modules:**

1. **Login:**

In this module user enter the User id and password is checked andonly valid user id and password will get entry into member’s zone. This is a security feature to avoid entry of unauthorized users.

1. **Registration Process:**

Through this module new users can register them. After giving their details, they will get a user id and password. Then to get entry into details section they need to provide this id and password and only user with valid id and password will get entry into details zone. This is also a security feature to avoid entry of unauthorized user.

1. **Manage Transaction:**

This module deals with different types of transactions such as Deposit, Withdraw related to customer as well as Bank.

1. **Administrator:**

This is the Administrator’s module by which he keeps the eye on whole site and maintain and upgrade the site’s service for sake of users.

2. SYSTEM ANALYSIS

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 is studied to the minutest detail and 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 organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing 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 areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions 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.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

 Here in the Email to Fax server project, a detailed study of existing system is carried along with all the steps in system analysis. An idea for creating a better project was carried and the next steps were followed.

**2.1 Scope of investigation**

The aim of this project is to design, build and test a “Banking Automation” System. This will be a vastly complex software development project which will take approximately 5 months to complete. The project will be split up into stages and documented thoroughly throughout.

Project management is a key factor of this task to ensure the strict deadlines are adhered to. It is also of paramount importance that tried and tested practices and techniques from the field are adhered to ensure that no common development project mistakes are reproduced.

**2.2 Problems Faced:**

**2.3 Proposed System**

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

**2.4.1 FEASIBILITY STUDY**

Feasibility study is made to see if the project on completion will serve the purpose of 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 workability, 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 provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

**2.4.2 TECHNICAL FEASIBILITY**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

* Does the existing technology sufficient for the suggested one?
* Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

**2.4.3 ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

**2.4.4 BEHAVIORAL FEASIBILITY**

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

**3. DESIGN INTRODUCTION**

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

**3.1 LOGICAL DESIGN:**

The logical flow of a system and define the boundaries of a system. It includes the following steps:

* Reviews the current physical system – its data flows, file content, volumes, Frequencies etc.
* Prepares output specifications – that is, determines the format, content and Frequency of reports.
* Prepares input specifications – format, content and most of the input functions.
* Prepares edit, security and control specifications.
* Specifies the implementation plan.
* Prepares a logical design walk through of the information flow, output, input, Controls and implementation plan.
* Reviews benefits, costs, target dates and system constraints.

**3.2 PHYSICAL DESIGN:**

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

* Design the physical system.
* Specify input and output media.
* Design the database and specify backup procedures.
* Design physical information flow through the system and a physical design
* Plan system implementation.
* Prepare a conversion schedule and target date.
* Determine training procedures, courses and timetable.
* Devise a test and implementation plan and specify any new hardware/software.
* Update benefits, costs, conversion date and system constraints

**3.3 Design/Specification activities:**

* Concept formulation.
* Problem understanding.
* High level requirements proposals.
* Feasibility study.
* Requirements engineering.
* Architectural design.

**3.4 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
* It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**3.5 OUTPUT DESIGN**

* A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.
* Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
* Select methods for presenting information.
* Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**3.6 Data Flow Diagram**

The Data flow Diagram shows the flow of data. It is generally made of symbols given below:

(1) A **square** shows the Entity: -

(2) A **Circle** shows the Process: -

(3) An **Open-Ended Rectangle** shows the data store: --

(4) An **arrow** shows the data flow: -

The DFD can be up to several levels.

The 0 level DFD states the flow of data in the system as seen from the outward in each module.

The first level DFD show more detail, about the single process of the 0 level DFD

The second level DFD can show even more details and so on.

**Context Level DFD**

Registration

Login

Check balance

deposit

withdraw

**ADMINISTRATOR**

Edit Information

View Regd. Users

**User**

**3.7 DATABASE DESIGN**

A database is an organized mechanism that has the 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 a two level process. 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 second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design.

The organization of the data in the database is aimed to achieve the following two major objectives.

* Data Integrity
* Data independence

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible.

Normalization is carried out in this system for the following reasons.

* To structure the data so that there is no repetition of data , this helps in

saving.

* To permit simple retrieval of data in response to query and report request.
* To simplify the maintenance of the data through updates, insertions,
* Deletions.
* To reduce the need to restructure or reorganize data which new application
* Requirements arise.

**3.7.1 RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS):**

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

**RELATIONS, DOMAINS & ATTRIBUTES:**

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

## RELATIONSHIPS:

* Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
* Entity Integrity enforces that no Primary Key can have null values.
* Referential Integrity enforces that no Primary Key can have null values.
* Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.
* Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity.

## NORMALIZATION:

As the name implies, it denoted putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These includes:

* Normalize the data.
* Choose proper names for the tables and columns.
* Choose the proper name for the data.

**First Normal Form**:

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values.

The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each nonatomic attribute or nested relation. This eliminated repeating groups of data.

A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

**Second Normal Form**:

According to Second Normal Form, for relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.

In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that

are fully functionally dependent on it. This step helps in taking out data that is only dependant on apart of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

**Third Normal Form**:

According to Third Normal Form, Relation should not have a nonkey attribute functionally determined by another nonkey attribute or by a set of nonkey attributes. That is, there should be no transitive dependency on the primary key.

In this we decompose and set up relation that includes the nonkey attributes that functionally determines other nonkey attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key.

A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non key attribute.

**3.8 tables structure**

**USERS**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| USER\_NAME | VARCHAR2(50) |
| ACC\_NO (Primary Key) | NUMBER (10) |
| USER\_PASSWORD | VARCHAR2(50) |
| USER\_EMAIL | VARCHAR2(50) |
| USER\_MOBILE | VARCHAR2(10) |
| USER\_BAL | NUMBER (10) |
| USER\_GENDER | VARCHAR2(10) |

**4.1 SOFTWARE AND HARDWARE SPECIFICATION**

**Hardware Specification:**

Microprocessor : - Pentium-4 class processor, 450 megahertz (MHz)

RAM : - 256 MB of RAM

Hard Disk : - 40 gigabytes (GB) on installation drive

**Software Specification:**

* Windows 98 / XP /7 or any operating system
* SQLite database
* Python 3.8 or above

# 4.2 Project Architecture and Platform

**Front End : Python tkinter**

**Back End** **:**

Business Logic : Python

Database : SQLite

##### **4.2.1 CODING STANDARDS**

The standard used in the development of the system is Microsoft Programming standards. it includes naming conversions of variables, constants and objects, standardized formats or labelling and commenting code, spacing, formatting and indenting.

##### **4.2.2 NAMING CONVENTION**

Classes’ names and interface names will start with capital letter. The function names will start with small letters and the first letter of each word in the function name will be in capital letter.

##### **4.2.3 LABELS AND COMMENTS**

Sufficient labels and comments are included in the description of it for the benefits if the developer and other programmers who might examine it later.

##### User Interface

1. For all the entry screen frames are used which will show the type of the user who is currently logged in and the menus.
2. Standard actions are used for standard actions.
3. Same font is related properties are used for similar screens.

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the users are trained in the new procedures.

**5. TESTING INTRODUCTION**

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term’s verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation **:** Are we doing the right job?

Verification **:** Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis

looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives.

They are

* Testing is a process of executing a program with the intend of finding an error.
* A good test case is one that has high possibility of finding an undiscovered error.
* A successful test is one that uncovers an undiscovered error.

     If a testing is conducted successfully according to the objectives as stated above, it would have uncovered errors in the software also testing demonstrate that the software function appears to be working according to the specification, that performance requirement appears to have been met.

There are three ways to test program.

* For correctness
* For implementation efficiency
* For computational complexity

Test for correctness is supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

**5.1 TEST PLAN**

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

* Unit testing
* Integration Testing
* Data validation Testing
* Output Testing

**5.1.1 UNIT TESTING**

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified.

**5.1.2 INTEGRATION TESTING**

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After unit testing in Sell-Soft System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

**5.1.3 VALIDATION TESTING OR SYSTEM TESTING**

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

**5.1.4 OUTPUT TESTING OR USER ACCEPTANCE TESTING**

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points

* Input Screen Designs,
* Output Screen Designs,
* Online message to guide the user and the like.

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Validation Checking:**

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected, and a final series of software test-validation checks may begin. Validation can be defined in many ways, but a simple definition (Albeit Harsh) is that validation succeeds when software functions in a manner that can be reasonably expected by a customer. Software validation is achieved through a series of black-box tests to be conducted and a test procedure defines specific test cases that will be used in attempt to uncover errors in conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied; all performance requirements are achieved; documentation is correct and human –Engineered and other requirements are met. Once the application was made free of all logical and interface errors, inputting dummy data to ensure that the software developed satisfied all the requirements of the user did validation checks. However, the data are created with the intent of determining whether the system will process them correctly.

In the proposed system, if the clients click the send button after selecting a file from his file list, then the system will show the confirmation message for sending files. Similarly, if a client makes an attempt to download a file from the server file list, then also the system will show the confirmation message for downloading. This is how the data validations were made in the proposed system.

**6. IMPLEMENTATION**

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

Implementation is the stage of the project where the theoretical design is tuned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

* Careful planning.
* Investigation of system and constraints.
* Design of methods to achieve the changeover.
* Training of the staff in the changeover phase.

**6.1 Implementation Procedures**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that

* The active user must be aware of the benefits of using the new system.
* Their confidence in the software is built up.
* Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

**6.1.1 User Training**

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

**6.1.2 Training on the Application Software**

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered.

It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

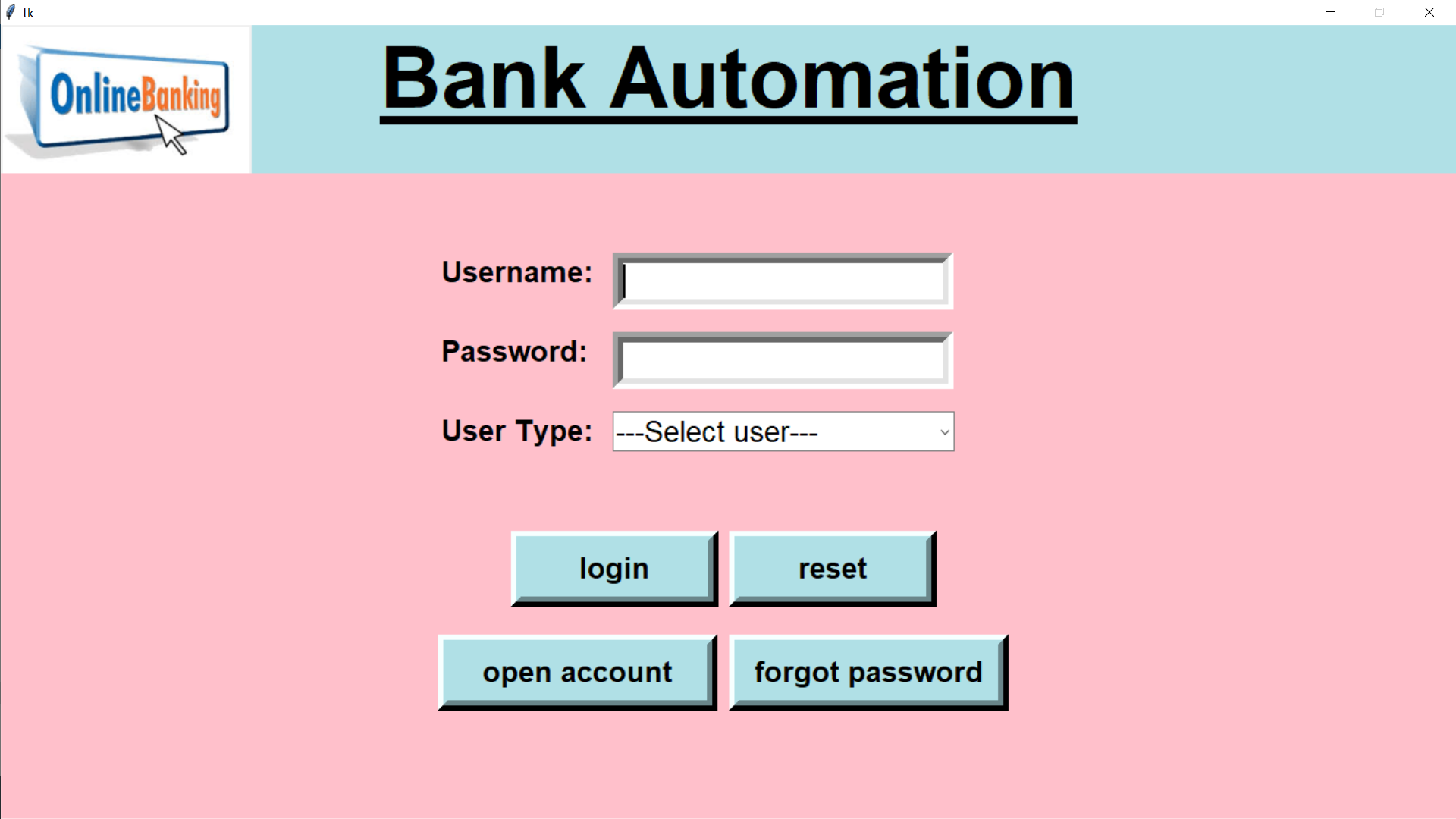
**6.1.3 Operational Document**

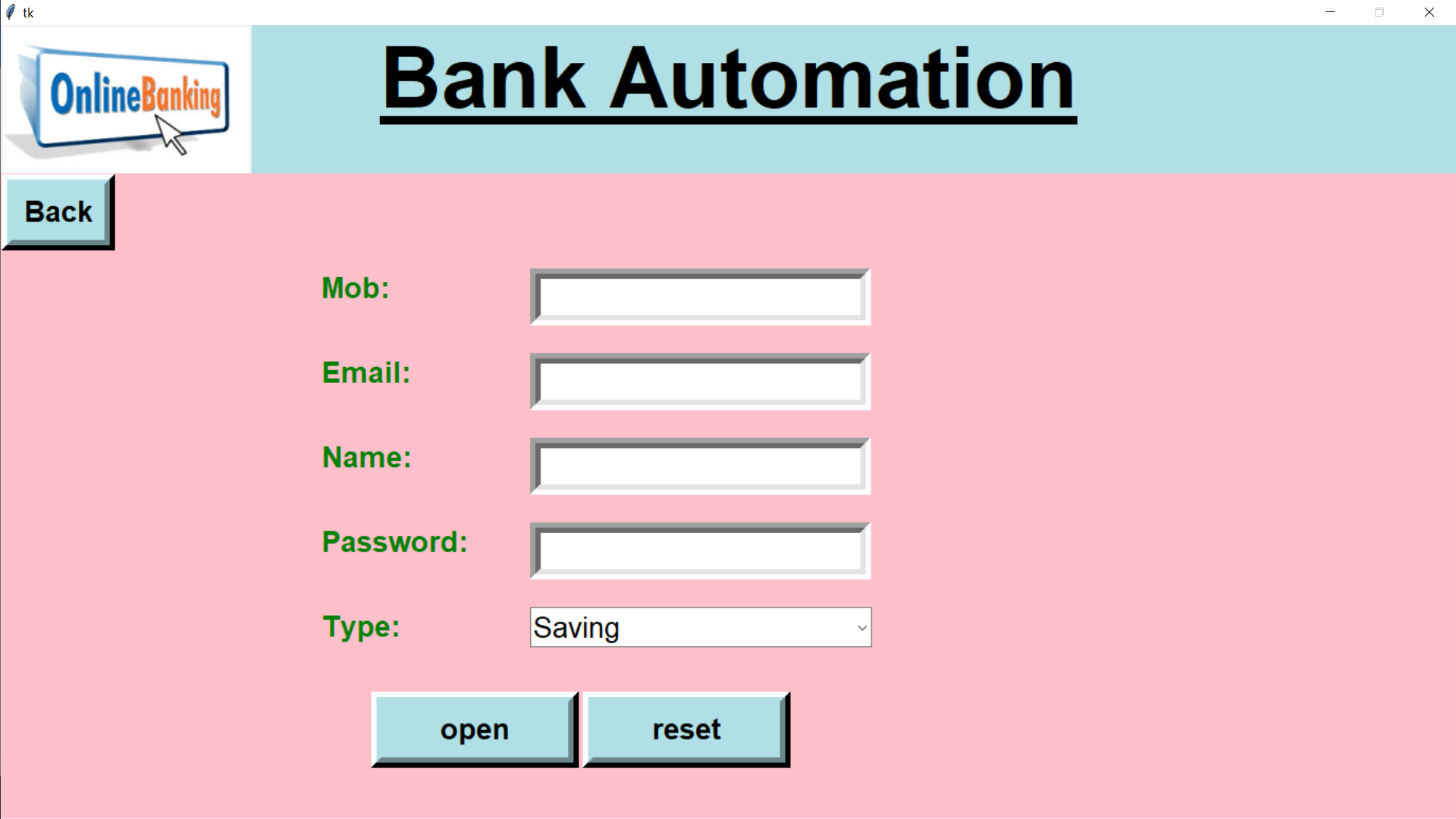
Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. Education involves right atmosphere and motivating the user. A documentation providing the whole operations of the system is being developed in such a way that the user can work with it in well consistent way. The system is developed user friendly so that the user can work the system from the tips given in the application itself. Useful tip and guidance is given inside the application itself to help the user. Users have to be made aware that what can be achieved with the new system and how it increases the performance of the system. The user of the system should be given a general idea of the system before he uses the system.

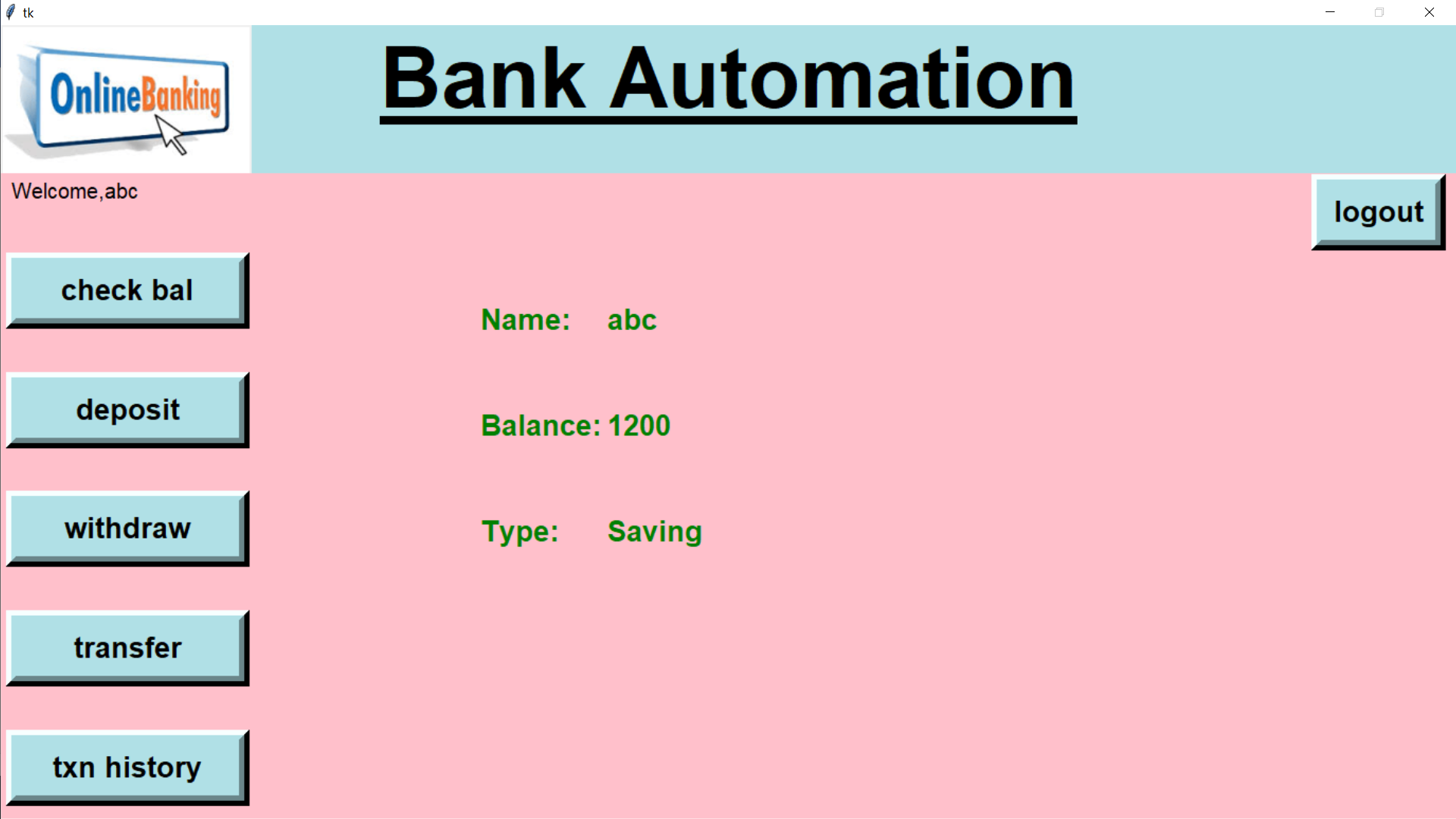
**6.1.4 System Maintenance**

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes". Maintenance may be defined by describing four activities that are undertaken after a program is released for use

**7. SCREEN SHOTS**







**8.Source Code**

**from tkinter import \***

**from tkinter.ttk import Combobox,Treeview,Style,Scrollbar**

**from PIL import Image,ImageTk**

**from datetime import datetime**

**import sqlite3**

**from tkinter import messagebox**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**table1="create table accounts(account\_no integer primary key autoincrement,account\_name text,account\_pass text,account\_email text,account\_mob text,account\_type text,account\_bal float,account\_opendate text)"**

**table2="create table txn(txn\_account\_no int,txn\_amt float,txn\_update\_bal float,txn\_date text,txn\_type text)"**

**try:**

**cur.execute(table1)**

**cur.execute(table2)**

**print("Tables created")**

**except:**

**print("something went wrong in db,might be table(s) already exists")**

**con.commit()**

**con.close()**

**win=Tk()**

**win.state("zoomed")**

**win.configure(bg="#5bccf6")**

**lbl\_title=Label(win,text="National Bank",bg="#5bccf6",font=('Arial',70,'bold','underline'))**

**lbl\_title.pack(pady=10,padx=50)**

**img=Image.open("Bank.png").resize((375,150))**

**imgtk=ImageTk.PhotoImage(img,master=win)**

**lbl\_bank=Label(win,image=imgtk)**

**lbl\_bank.place(x=1,y=2)**

**def login\_screen():**

**frm=Frame(win)**

**frm.configure(bg="#fcde67")**

**frm.place(relx=0,rely=.2,relwidth=1,relheight=.8)**

**def newuser():**

**frm.destroy()**

**newuser\_screen()**

**def forgot():**

**frm.destroy()**

**forgotpass\_screen()**

**def reset():**

**e\_acn.delete(0,"end")**

**e\_pass.delete(0,"end")**

**e\_acn.focus()**

**def login\_db():**

**acn=e\_acn.get()**

**pwd=e\_pass.get()**

**if(acn=="" or pwd==""):**

**messagebox.showwarning("login","Please fill both fields")**

**else:**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select \* from accounts where account\_no=? and account\_pass=?",(acn,pwd))**

**global tup**

**tup=cur.fetchone()**

**if(tup==None):**

**messagebox.showerror("login","Invalid ACN or Pass")**

**else:**

**frm.destroy()**

**welcome\_screen()**

**lbl\_acn=Label(frm,text="Account No",bg="#fcde67",font=('Arial',20,'bold'))**

**lbl\_acn.place(relx=.3,rely=.1)**

**e\_acn=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_acn.place(relx=.42,rely=.1)**

**e\_acn.focus()**

**lbl\_pass=Label(frm,text="Password",bg="#fcde67",font=('Arial',20,'bold'))**

**lbl\_pass.place(relx=.3,rely=.23)**

**e\_pass=Entry(frm,font=('Arial',20,'bold'),bd=5,show="\*")**

**e\_pass.place(relx=.42,rely=.23)**

**btn\_login=Button(frm,text="Login",command=login\_db,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_login.place(relx=.42,rely=.35)**

**btn\_reset=Button(frm,text="Reset",command=reset,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_reset.place(relx=.51,rely=.35)**

**btn\_fp=Button(frm,text="Forget Password",command=forgot,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_fp.place(relx=.42,rely=.48)**

**btn\_new=Button(frm,text="Open New Account",command=newuser,width=15,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_new.place(relx=.42,rely=.6)**

**def newuser\_screen():**

**frm=Frame(win)**

**frm.configure(bg="#fcde67")**

**frm.place(relx=0,rely=.2,relwidth=1,relheight=.8)**

**def back():**

**frm.destroy()**

**login\_screen()**

**def openacn\_db():**

**name=e\_name.get()**

**pwd=e\_pass.get()**

**email=e\_email.get()**

**mob=e\_mob.get()**

**acn\_type=cb\_type.get()**

**if(acn\_type=="Saving"):**

**bal=1000**

**else:**

**bal=10000**

**opendate=str(datetime.now().date())**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("insert into accounts(account\_name,account\_pass,account\_email,account\_mob,account\_type,account\_bal,account\_opendate) values(?,?,?,?,?,?,?)",(name,pwd,email,mob,acn\_type,bal,opendate))**

**con.commit()**

**con.close()**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select max(account\_no) from accounts")**

**tup=cur.fetchone()**

**con.close()**

**messagebox.showinfo("Success",f"Account Opened with ACN :{tup[0]}")**

**frm.destroy()**

**login\_screen()**

**btn\_back=Button(frm,text="back",command=back,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_back.place(relx=0,rely=0)**

**lbl\_name=Label(frm,text="Name",bg="#fcde67",font=('Arial',20,'bold'))**

**lbl\_name.place(relx=.3,rely=.05)**

**e\_name=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_name.place(relx=.42,rely=.05)**

**e\_name.focus()**

**lbl\_pass=Label(frm,text="Password",bg="#fcde67",font=('Arial',20,'bold'))**

**lbl\_pass.place(relx=.3,rely=.15)**

**e\_pass=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_pass.place(relx=.42,rely=.15)**

**lbl\_email=Label(frm,text="E-mail Id",bg="#fcde67",font=('Arial',20,'bold'))**

**lbl\_email.place(relx=.3,rely=.25)**

**e\_email=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_email.place(relx=.42,rely=.25)**

**lbl\_mob=Label(frm,text="Phone No.",bg="#fcde67",font=('Arial',20,'bold'))**

**lbl\_mob.place(relx=.3,rely=.35)**

**e\_mob=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_mob.place(relx=.42,rely=.35)**

**lbl\_type=Label(frm,text="Accn Type",bg="#fcde67",font=('Arial',20,'bold'))**

**lbl\_type.place(relx=.3,rely=.45)**

**cb\_type=Combobox(frm,values=['Saving','Current'],font=('Arial',20,'bold'))**

**cb\_type.current(0)**

**cb\_type.place(relx=.42,rely=.45)**

**btn\_open=Button(frm,text="Open",command=openacn\_db,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_open.place(relx=.45,rely=.55)**

**btn\_reset=Button(frm,text="Reset",font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_reset.place(relx=.55,rely=.55)**

**def forgotpass\_screen():**

**frm=Frame(win)**

**frm.configure(bg="#fcde67")**

**frm.place(relx=0,rely=.2,relwidth=1,relheight=.8)**

**def back():**

**frm.destroy()**

**login\_screen()**

**def get\_db():**

**acn=e\_acn.get()**

**email=e\_email.get()**

**mob=e\_mob.get()**

**if(acn=="" or email=="" or mob==""):**

**messagebox.showwarning("Validation","Please fill all fields")**

**return**

**else:**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select account\_pass from accounts where account\_no=? and account\_email=? and account\_mob=?",(acn,email,mob))**

**tup=cur.fetchone()**

**if(tup==None):**

**messagebox.showerror("Forgot","Invalid details")**

**else:**

**messagebox.showinfo("Forgot",f"Your Password is:{tup[0]}")**

**btn\_back=Button(frm,text="back",command=back,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_back.place(relx=0,rely=0)**

**lbl\_acn=Label(frm,text="ACN",bg='#fcde67',font=('Arial',20,'bold'))**

**lbl\_acn.place(relx=.3,rely=.2)**

**e\_acn=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_acn.place(relx=.45,rely=.2)**

**e\_acn.focus()**

**lbl\_email=Label(frm,text="Email",bg='#fcde67',font=('Arial',20,'bold'))**

**lbl\_email.place(relx=.3,rely=.35)**

**e\_email=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_email.place(relx=.45,rely=.35)**

**lbl\_mob=Label(frm,text="Mob",bg='#fcde67',font=('Arial',20,'bold'))**

**lbl\_mob.place(relx=.3,rely=.5)**

**e\_mob=Entry(frm,font=('Arial',20,'bold'),bd=5)**

**e\_mob.place(relx=.45,rely=.5)**

**btn\_get=Button(frm,text="Generate",command=get\_db,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_get.place(relx=.45,rely=.65)**

**btn\_reset=Button(frm,text="Reset",font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_reset.place(relx=.55,rely=.65)**

**def welcome\_screen():**

**frm=Frame(win)**

**frm.configure(bg="#fcde67")**

**frm.place(relx=0,rely=.2,relwidth=1,relheight=.8)**

**def logout():**

**option=messagebox.askyesno('confirmation','Do you want to logout?')**

**if(option==True):**

**frm.destroy()**

**login\_screen()**

**def checkbal():**

**ifrm=Frame(frm,highlightbackground="brown",highlightthickness=5,highlightcolor="brown")**

**ifrm.configure(bg='white')**

**ifrm.place(relx=.25,rely=.2,relwidth=.6,relheight=.6)**

**lbl\_page.configure(text="This is Check Balance Page")**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select account\_name,account\_bal,account\_opendate from accounts where account\_no=?",(tup[0],))**

**row=cur.fetchone()**

**lbl\_acn=Label(ifrm,text=f"Account Number\t{tup[0]}",bg='white',font=('Arial',20,'bold'),fg='green')**

**lbl\_acn.place(relx=.3,rely=.1)**

**lbl\_name=Label(ifrm,text=f"Holder Name\t{row[0]}",bg='white',font=('Arial',20,'bold'),fg='green')**

**lbl\_name.place(relx=.3,rely=.3)**

**lbl\_bal=Label(ifrm,text=f"Available Bal\t{row[1]}",bg='white',font=('Arial',20,'bold'),fg='green')**

**lbl\_bal.place(relx=.3,rely=.5)**

**lbl\_date=Label(ifrm,text=f"ACN Open Date\t{row[2]}",bg='white',font=('Arial',20,'bold'),fg='green')**

**lbl\_date.place(relx=.3,rely=.7)**

**def deposit():**

**ifrm=Frame(frm,highlightbackground="brown",highlightthickness=3,highlightcolor='brown')**

**ifrm.configure(bg='white')**

**ifrm.place(relx=.25,rely=.2,relwidth=.6,relheight=.6)**

**lbl\_page.configure(text="This is Deposit page")**

**def deposit\_db():**

**amt=float(e\_amt.get())**

**acn=tup[0]**

**txn\_type="CR."**

**dt=str(datetime.now())**

**if(amt<0):**

**messagebox.showerror("Deposit","-ve amount can not be deposited")**

**else:**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select account\_bal from accounts where account\_no=?",(acn,))**

**bal=cur.fetchone()[0]**

**con.close()**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("insert into txn values(?,?,?,?,?)",(acn,amt,bal+amt,dt,txn\_type))**

**cur.execute("update accounts set account\_bal=account\_bal+? where account\_no=?",(amt,acn))**

**con.commit()**

**con.close()**

**messagebox.showinfo("Deposit","Amount deposited")**

**lbl\_amt=Label(ifrm,text="Enter Amount",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_amt.place(relx=.2,rely=.2)**

**e\_amt=Entry(ifrm,font=('Arial',20,'bold'),bd=5)**

**e\_amt.place(relx=.45,rely=.2)**

**btn\_dep=Button(ifrm,text="deposit",command=deposit\_db,font=('Arial',20,'bold'),bd=5,bg="powder blue")**

**btn\_dep.place(relx=.5,rely=.5)**

**def withdraw():**

**ifrm=Frame(frm,highlightbackground="brown",highlightthickness=3,highlightcolor='brown')**

**ifrm.configure(bg='white')**

**ifrm.place(relx=.25,rely=.2,relwidth=.6,relheight=.6)**

**lbl\_page.configure(text="This is Withdraw page")**

**def withdraw\_db():**

**amt=float(e\_amt.get())**

**acn=tup[0]**

**txn\_type="DB."**

**dt=str(datetime.now())**

**if(amt<0):**

**messagebox.showerror("Withdraw","-ve amount can not be withdrawn")**

**else:**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select account\_bal from accounts where account\_no=?",(acn,))**

**bal=cur.fetchone()[0]**

**con.close()**

**if(bal>=amt):**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("insert into txn values(?,?,?,?,?)",(acn,amt,bal-amt,dt,txn\_type))**

**cur.execute("update accounts set account\_bal=account\_bal-? where account\_no=?",(amt,acn))**

**con.commit()**

**con.close()**

**messagebox.showinfo("Withdraw","Amount withdrawn")**

**else:**

**messagebox.showwarning("Withdraw","Insufficient bal")**

**lbl\_amt=Label(ifrm,text="Enter Amount",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_amt.place(relx=.2,rely=.2)**

**e\_amt=Entry(ifrm,font=('Arial',20,'bold'),bd=5)**

**e\_amt.place(relx=.45,rely=.2)**

**btn\_dep=Button(ifrm,text="withdraw",command=withdraw\_db,font=('Arial',20,'bold'),bd=5,bg="powder blue")**

**btn\_dep.place(relx=.5,rely=.5)**

**def transfer():**

**ifrm=Frame(frm,highlightbackground="brown",highlightthickness=3,highlightcolor='brown')**

**ifrm.configure(bg='white')**

**ifrm.place(relx=.25,rely=.2,relwidth=.6,relheight=.6)**

**lbl\_page.configure(text="This is transfer page")**

**def transfer\_db():**

**t\_acn=e\_to.get()**

**amt=float(e\_amt.get())**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select \* from accounts where account\_no=?",(t\_acn,))**

**row=cur.fetchone()**

**con.close()**

**if(row==None):**

**messagebox.showerror("Transfer","To account does not exist")**

**else:**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select account\_bal from accounts where account\_no=?",(tup[0],))**

**bal=cur.fetchone()[0]**

**con.close()**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select account\_bal from accounts where account\_no=?",(t\_acn,))**

**t\_bal=cur.fetchone()[0]**

**con.close()**

**if(bal>=amt):**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**dt=str(datetime.now())**

**cur.execute("update accounts set account\_bal=account\_bal-? where account\_no=?",(amt,tup[0]))**

**cur.execute("update accounts set account\_bal=account\_bal+? where account\_no=?",(amt,t\_acn))**

**cur.execute("insert into txn values(?,?,?,?,?)",(tup[0],amt,bal-amt,dt,"DB."))**

**cur.execute("insert into txn values(?,?,?,?,?)",(t\_acn,amt,t\_bal+amt,dt,"CR."))**

**con.commit()**

**con.close()**

**messagebox.showinfo("Transfer","Txn Done")**

**else:**

**messagebox.showwarning("Transfer","Insufficient bal")**

**lbl\_to=Label(ifrm,text="Enter To ACN",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_to.place(relx=.2,rely=.2)**

**e\_to=Entry(ifrm,font=('Arial',20,'bold'),bd=5)**

**e\_to.place(relx=.45,rely=.2)**

**lbl\_amt=Label(ifrm,text="Enter Amount",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_amt.place(relx=.2,rely=.4)**

**e\_amt=Entry(ifrm,font=('Arial',20,'bold'),bd=5)**

**e\_amt.place(relx=.45,rely=.4)**

**btn\_dep=Button(ifrm,text="transfer",command=transfer\_db,font=('Arial',20,'bold'),bd=5,bg="powder blue")**

**btn\_dep.place(relx=.5,rely=.6)**

**def update():**

**ifrm=Frame(frm,highlightbackground="brown",highlightthickness=3,highlightcolor='brown')**

**ifrm.configure(bg='white')**

**ifrm.place(relx=.25,rely=.2,relwidth=.6,relheight=.6)**

**lbl\_page.configure(text="This is update profile page")**

**def update\_profile():**

**name=e\_name.get()**

**pwd=e\_pass.get()**

**email=e\_email.get()**

**mob=e\_mob.get()**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("update accounts set account\_name=?,account\_pass=?,account\_email=?,account\_mob=? where account\_no=?",(name,pwd,email,mob,tup[0]))**

**con.commit()**

**con.close()**

**messagebox.showinfo("Update","Profile Updated")**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select \* from accounts where account\_no=?",(tup[0],))**

**row=cur.fetchone()**

**con.close()**

**lbl\_name=Label(ifrm,text="Name",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_name.place(relx=.05,rely=.2)**

**e\_name=Entry(ifrm,font=('Arial',15,'bold'),bd=5)**

**e\_name.place(relx=.15,rely=.2)**

**e\_name.insert(0,row[1])**

**lbl\_pass=Label(ifrm,text="Pass",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_pass.place(relx=.53,rely=.2)**

**e\_pass=Entry(ifrm,font=('Arial',15,'bold'),bd=5)**

**e\_pass.place(relx=.62,rely=.2)**

**e\_pass.insert(0,row[2])**

**lbl\_email=Label(ifrm,text="Email",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_email.place(relx=.05,rely=.4)**

**e\_email=Entry(ifrm,font=('Arial',15,'bold'),bd=5)**

**e\_email.place(relx=.15,rely=.4)**

**e\_email.insert(0,row[3])**

**lbl\_mob=Label(ifrm,text="Mob",bg='white',font=('Arial',15,'bold'),fg='red')**

**lbl\_mob.place(relx=.53,rely=.4)**

**e\_mob=Entry(ifrm,font=('Arial',15,'bold'),bd=5)**

**e\_mob.place(relx=.62,rely=.4)**

**e\_mob.insert(0,row[4])**

**btn\_dep=Button(ifrm,text="update",command=update\_profile,font=('Arial',15,'bold'),bd=5,bg="powder blue")**

**btn\_dep.place(relx=.35,rely=.6)**

**def txn\_history():**

**ifrm=Frame(frm,highlightbackground="brown",highlightthickness=3,highlightcolor='brown')**

**ifrm.configure(bg='white')**

**ifrm.place(relx=.25,rely=.2,relwidth=.6,relheight=.6)**

**lbl\_page.configure(text="This is txn history page")**

**tv=Treeview(ifrm)**

**tv.place(x=0,y=0,relheight=1,relwidth=1)**

**style = Style()**

**style.configure("Treeview.Heading", font=('Arial',10,'bold'),foreground='brown')**

**sb=Scrollbar(ifrm,orient='vertical',command=tv.yview)**

**sb.place(relx=.98,y=0,relheight=1)**

**tv['columns']=('Txn date','Txn amount','Txn type','Updated bal')**

**tv.column('Txn date',width=150,anchor='c')**

**tv.column('Txn amount',width=100,anchor='c')**

**tv.column('Txn type',width=100,anchor='c')**

**tv.column('Updated bal',width=100,anchor='c')**

**tv.heading('Txn date',text='Txn date')**

**tv.heading('Txn amount',text='Txn amount')**

**tv.heading('Txn type',text='Txn type')**

**tv.heading('Updated bal',text='Updated bal')**

**tv['show']='headings'**

**con=sqlite3.connect(database="bank.sqlite")**

**cur=con.cursor()**

**cur.execute("select txn\_date,txn\_amt,txn\_type,txn\_update\_bal from txn where txn\_account\_no=?",(tup[0],))**

**for row in cur:**

**tv.insert("","end",values=(row[0],row[1],row[2],row[3]))**

**btn\_logout=Button(frm,text="Logout",command=logout,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_logout.place(relx=.9,rely=0)**

**lbl\_wel=Label(frm,text=f"Welcome,{tup[1]}",bg='#fcde67',font=('Arial',20,'bold'),fg='green')**

**lbl\_wel.place(relx=0,rely=0)**

**btn\_checkbal=Button(frm,text="Check Balance",command=checkbal,width=14,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_checkbal.place(relx=0,rely=.1)**

**btn\_deposit=Button(frm,text="Deposit Amount",command=deposit,width=14,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_deposit.place(relx=0,rely=.25)**

**btn\_withdraw=Button(frm,text="Withdraw Amount",command=withdraw,width=14,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_withdraw.place(relx=0,rely=.4)**

**btn\_transfer=Button(frm,text="Transfer Amount",command=transfer,width=14,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_transfer.place(relx=0,rely=.55)**

**btn\_update=Button(frm,text="Update Profile",command=update,width=14,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_update.place(relx=0,rely=.7)**

**btn\_trahist=Button(frm,text="History",command=txn\_history,width=14,font=('Arial',20,'bold'),bd=5,bg="#5bccf6")**

**btn\_trahist.place(relx=0,rely=.85)**

**lbl\_page=Label(frm,text="This is Home Page",bg='#fcde67',font=('Arial',30,'bold','underline'),fg='blue')**

**lbl\_page.place(relx=.35,rely=0)**

**login\_screen()**

**win.mainloop()**

**9**. **CONCLUSION**

The project titled as “Banking Automation” is a window based application. This software is developed with scalability in mind. Additional modules can be easily added when necessary. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

**10. Scope for Future Enhancement**

In future we can able to add more modules in this project GP-Desk. In the proposed system the user is provided with a choice of data screen, which are similar in formats to the source documents. Data entry errors can be minimized through validity checks. After the verification only the data are placed in the permanent database. The software can be developed further to include a lot of modules because the proposed system is developed on the view of future, for example we should develop the system as a database independent using JDBC so we can connect it to any other database. Now the proposed system is based on PC and intranet but in the future if we need to convert it into internet then we need to change the front end only because we are developing this on the basis of OOP technology and most of the business logics are bounded in the class files and module like reusable components.

11.BIBLIOGRAPHY: -

The following books were referred during the analysis and execution phase of the project

PYTHON 3.11 (64 BIT)

My SQL