# **INTRODUCTION**

Loan management system is a system that can help one perform all the complex calculations related to banking loan activities. Basically, it is a very challenging task to provide all the facilities like calculation of EMI and generation of EMI chart as well as storing all the data related to customers. An organization has to enter the data of a customer as well as store it in an efficient way.

In finance, a loan is the lending of money from one individual, organization or entity to another individual organization or entity. Loan could be provided by an organization or individual to a person at an interest rate. In a loan, the borrower initially receives or borrows an amount of money which is known as Principal from the lender, and is responsible to repay an equal amount of money to the lender at a later time. Acting as a provider of loans is one of the principal tasks financial institutions such as banks and credit card companies.  
It provides a very reliable and convenient form for every loan and EMI related transaction and their related details. It is also a very user friendly and understandable to customers as well as bankers .

Hence, LOAN MANAGEMENT SYSTEM is a very efficient process to handle all loan related transaction in a very accurate and convenient way. It is very helpful for those banking staffs who are in the charge of loan management. It can be used by any of bankers for customers to manage the loan related activities.

1.1 PURPOSE

A loan management system provides us a suitable platform to deal with EMI generation if any loan is given to someone .It even helps us manage the loans by providing us different features like storing details in database and provides us a very fast access to the data and helps us manage transactions related to loans.

1.2 OVERVIEW AND SCOPE

Day by day people are moving towards a process that offers the simplest form of approach of solving a problem.This project is related to store transaction info which will be used by the staffs of a bank for handling their customers who have loans or want to have loan.

Operator need to use the registration form to create account for the new customers, and EMI details and the loan detail forms are present for dealing with the old customers or those who are already registered . This application has a user friendly interface so that the banking staffs can easily use the software and its features. It also incorporates steps for getting the transaction details on every transaction and next EMI date and remaining amount details that helps the operators to use the application efficiently.

There are various kind of loans availabie in banks like

PERSONAL LOANS

EDUCATION LOANS

BUSINESS LOANS

VEHICLE LOANS

HOUSING LOAN

MORTAGE LOAN

Problems faced before this system:-  
  
1. Separated and Isolated Data: To make a decision, a user might need data from two separate files. First, the files were evaluated by analysts and programmers to determine the specific data required from each file and the relationships between the data and then applications could be written in a programming language to process and extract the needed data. Imagine the work involved if data from several files was needed  
  
2. Duplication of data: Often the same information is stored in more than one file. Uncontrolled duplication of data is not required for several reasons, such as: • Duplication is wasteful. It costs time and money to enter the data more than once • It takes up additional storage space, again with associated costs. • Duplication can lead to loss of data integrity; in other words the data is no longer consistent. For example – Customer and transaction details would not be consistent.

3. Data Dependence: In file processing systems, files and records were described by specific physical formats that were coded into the application program by programmers. If the format of a certain record was changed, the code in each file containing that format must be updated. Furthermore, instructions for data storage and access were written into the application's code. Therefore, .changes in storage structure or access methods could greatly affect the processing or results of an application. In other words, in file based approach application programs are data dependent. It means that, with the change in the physical representation (how the data is physically represented in disk) or access technique (how it is physically accessed) of data, application programs are also affected and needs modification. In other words application programs are dependent on the how the data is physically stored and accessed. If for example, if the physical format of the master/transaction file is changed, by making the modification in the delimiter of the field or record, it necessitates that the application programs which depend on it must be modified.  
  
4. Difficulty in representing data from the user's view:  
  
To create useful applications for the user, often data from various files must be combined. In file processing it was difficult to determine relationships between isolated data in order to meet user requirements.

5. Data Inflexibility: Program-data interdependency and data isolation, limited the flexibility of file processing systems in providing users with ad-hoc information requests

6. Data Security. The security of data is low in file based system because, the data is maintained in the flat file(s) is easily accessible. For Example: Consider the Banking System. The Customer Transaction file has details about the total available balance of all customers. A Customer wants information about his account balance. In a file system it is difficult to give the Customer access to only his data in the· file. Thus enforcing security constraints for the entire file or for certain data items are difficult.  
  
7. Transactional Problems. The File based system approach does not satisfy transaction properties like Atomicity, Consistency, Isolation and Durability properties commonly known as ACID properties. For example: Suppose, in a banking system, a transaction that transfers Rs. 1000 from account A to account B with initial values' of A and B being Rs. 5000 and Rs. 10000 respectively. If a system crash occurred after the withdrawal of Rs. 1000 from account A, but before depositing of amount in account B, it will result an inconsistent state of the system. It means that the transactions should not execute partially but wholly. This concept is known as Atomicity of a transaction (either 0% or 100% of transaction). It is difficult to achieve this property in a file based system.  
  
8. Concurrency problems. When multiple users access the same piece of data at same interval of time then it is called as concurrency of the system. When two or more users read the data simultaneously there is problem, but when they like to update a file simultaneously, it may result in a problem.  
  
SYSTEM ANALYSIS

Introduction System analysis is the systematic analysis of a proposed system and the identification of the requirements that it should meet. It is a process of gathering and interpreting facts, diagnosing problem, defining the problem and recognizing causes. System Analysis is the starting point for system design. The term is most commonly used in the context of commercial programming, where software developers are often classed as either system analyst or programmers. The systems analysts are responsible for identifying requirements (i.e. systems analysis) and producing a design. The programmers are then responsible for implementing it. The System Analysis is performed in three different steps-  
  
Problem Analysis: Loan Department maintains all loan related information and transaction. Which comes from different branches. At present loan department section use a paper-based system to store and analyse these information. Some software like Ms. Word, Ms Excel, Ms Access etc. used in a limited basis. To manage all these information a paper based-manual system is not efficient. In BSB there are several departments and each department has their own data to process. Therefore huge data collection is going all the day and managing all these information in a paper based system is very difficult, time consuming and enormous. So the challenge was to build software to overcome these problems. In the paper-based system other problems they face are: ·  
\*Extra manpower, so more expenses is needed in every step to of these. Documents are not secured. ·

\* Calculations are complicated and error can occur easily.  
\*Documents are very large.  
\*Stake of files that is very difficult to maintain  
\*Extra overhead works

**Requirement analysis:-**  
  
In order to understand the existing system and identify the current problems with the system, a requirement analysis has to be implemented. This part plays a fundamental role in System analysis phase. Requirement Analysis relies on fact-finding techniques. Which includes interview, questioners, record review and observation.  
  
Purpose of Loan Management System activity: To handle all loan related transaction in a very accurate and convenient way. It is very helpful for those banking staffs who are in the charge of loan management, it provides a very reliable and convenient form for every loan and EMI related transaction and their related details. It is also a very user friendly and understandable form of storing customer’s details & transaction information.

The system starts when client submits a project for a loan. This project is verified in the Loan Department. The result of the verification is send to the management. Then the management decides whether the project is accepted or denied in a board meeting. When the project is accepted the detail information of the project is to collect. Like- project id, Name, Address etc. and send them to Not Due section in the loan Department. This section is responsible for giving money for the project and maintains the related information and transaction. A new account number and loan number is to open for that project loan in the not due section. Which contain sanction amount, instalment amount, interest during construction period amount etc. After that the loan sanction amount is given to the project by more than one instalment. This process is called disbursement process.

When the 1st amount of taka is given to the project, it is called the first disbursement. Disbursement balance report is prepared in this step. This information is also kept in the not Due section. When the disbursement process completed this is the time to start taking return from client with interest. That’s why repayment schedule created to receive the money. Repayment schedule contains instalment amount, date, instalment cycle, total balance. Over Due Section collect instalment amount from the client. This section also collects other charge from the client. After collecting the taka from the client this section update the current loan balance and also send the information to Not Due section. Loan recovery balance report is prepared here. If anybody fail to give the instalment for a long time than this section takes necessary steps, like- Re-phase/Write-off etc. In case of re-phase the old loan account is rearranged by the management and create a new loan account. The entire balance is then transferred from the old loan to new loan. Finally all the information is send to the CAD (Central Accounting Department).  
  
STEPS PERFORMED PLACE :

All the Loan Management works are performed in the Loan Department. But the department is classified in two parts – Not Due Section and Over Due Section. Not Due Section is responsible to give money to the project and preparing the Repayment Schedule. Over Due Section receives money from the client and takes necessary steps for loan defaulter.   
  
**PERFORMERS:**

1. The client or project applies for the loan.

2. The Inspector verifies the project.

3. The management accepts or denies the project.

4. General Manager, Manager, DGM and AGM perform rests of the processes.  
  
DURATION: Bank gives loan for long-term as well as for short time interval also. It mainly depends on the loan sanction amount. If the sanction amount is large than it takes long time.  
  
VOLUME OF LOAN: As these activities are performed for long term loan, so volume of transaction is great. Each loan goes for year after year for a particular project.  
  
DURATION: Project submission, verification, loan sanction this steps are performed only once for a particular loan. · Duration of Loan Disbursement, repayment activities are based on sanction amount. · Re-phase is done once or twice for a particular loan.  
  
PROBLEMS AND ITS EFFECTS: The main problem they face is that for manual system, each process becomes very time consuming. Then in the paper-based system other problems they face are: · Huge documents, stake of files that is very difficult to maintain. · Extra overhead works, · Extra manpower so more cost, are needed in every step to manage all of these. ·

Important documents are unsecured.  
  
The reason for these problem is that the Bank’s activities is expanding day by day but they still use the paper base manual system to manage all the transaction and information.  
  
**INFORMATION GATHERING**

System development Life cycle consist of two major steps of System analysis and design. It includes set of activities that analysts, designers and users carry out to develop and implement an information system. So, here we followed the different stages of system development lifecycle to develop Billing Software efficiently. The systems development life cycle consists of the following activities Preliminary Investigation, Determination Requirements, Design of System,Development of System ,System Testing,Implementation  
  
Preliminary Investigation: For this, the need arises to understand the viewpoint of two important entities… Top management and users. In order to gather information, I interviewed the Bank department and asked the following questions:  
  
**INTERVIEW:**

• How the present system works? ? • Who will be authenticated to access data and his/her access rights? • What specific facilities they want from new system, those are currently not in scope of existing system? • What all drawbacks are in the present system? • What is their vision about the new system? • How will data flow in the systemTo find more about present system’s working mechanism such as the ways of getting inputs and providing outputs, I interviewed the current users of the system by asking following question: • Are they comfortable with the present system? • What flaws exists in the current system? • Do they feel the necessity of new system? • What will be their requirements from new system? • Are they satisfied with their role in new system?  
  
After carrying out these interviews, I drew conclusion about the Top Management’s requirements and whether users are in support of the new system. This activity is consisting of three parts:-2.1.1. Request Clarification As previously stated, the requests are made from employees and users in the organization, which are not clearly stated, therefore a System investigation is being considered. The project request must be examined to determine precisely what the user’s wants. So taking in to consideration of all these tasks; the corresponding development came in to existence:-  
  
**FEASIBILTY STUDY**:

Studies to determine the advantages or disadvantages, practicability, or capability of accomplishing a projected plan, study, or project is known as Feasibility Study. If the proposed system is not feasible then there is total waste of both time and money. That’s why I had to study the feasibility of the proposed system. The outcome of my studies are·  
Technical Feasibility  
Economic Feasibility  
Operational Feasibility

**TECHNICAL FEASIBILITY**Technical Feasibility concerned with the possibility of whether the proposed solution can be implemented with the available hardware, software and technical resources. The system that I have designed is for network environment. So many user can use this application but the data storage will in the server pc only. Loan Management System is one of the modules of Bank Management System. Some modules like- Project Information System, Payroll system, Employee Management System had already exist there. So the necessary hardware, software and technical resources for my proposed solution were available there.  
  
So my System is technically feasible for this organization because it can be easily installed in the present hardware and software environment.  
  
**ECONOMIC FEASIBILITY**  
Economic feasibility determines whether the benefits of a proposed solution out-weight the cost. Implementation of my proposed system will ensure the best possible performance. This increase in performance will optimize the overall productivity and throughput of the mentioned resources. The Loan Management System is economically feasible in this concern.  
  
**OPERATIONAL FEASIBILITY**Operation Feasibility determines whether the proposed solution is desirable within the existing managerial and organizational framework. The Loan Management System is easy to use. The user does not need any costly training to operate this system. However the new users must have computer operation knowledge especially on Windows platform. As is common for any new software at the beginning things may appear a little unfamiliar. But the system is gradually used it will be found to be quiet easy to operate. And the user will become skilled in it automatically. It will take a few days to learn the operation of this new system. The output of this system will be useful for the relevant department and for the management for taking decisions.  
  
**REQUEST APPROVAL**

:Request Approval Not all the requested projects are desirable or feasible. However, those that are both feasible and desirable should be put into scheduled. After the project is approved, its cost, priority, completion time and personnel requirements are estimated and used to determine where to add it to any project list. The Top Management reviewed the above factors and gave their ascent. After the consent to request approval, the design was taken up. Means studying the adjacent system in order to collect the data  
  
Determination of System Requirement Determination of system requirement is regarding the way it captures data, processes the data, produces the output. I used following techniques for identifying system requirements: • Reviewing organization documents • Onsite observations • Conducting interviews  
  
Reviewing Organization Documents I first learnt about the organization involved in, or affected by the project, I got to know how the department works and the employees were directly involved/associated with the application. Annual manuals and reports were of great help to me.  
  
Onsite Observations It is a process of recognizing and observing people, objects and their occurrence to obtain the information. The major objective of the Onsite Observation is to get as close as possible to real system being studied. Here, I observed the activities of the system directly. I saw the office environment, workload on the system, and workload on the users. The physical layout of the current system along with the location & movement of staff was analyzed. In this way, the information about the present workflow, objects and people was gathered. This helped me to understand various procedures & processes, which were to be developed in the new system.  
  
Conducting Interviews Written documents and onsite observation just tell that how the system should operate. They do not include enough details to allow a decision to be made about the merits of system proposal, nor do they present user views about the current system. I conducted interviews of the staff, which were directly involved with the application. Also the regular users of the application were interviewed. Based on their viewpoints, crystal clear system requirements were jolted down. These interviews were of great help.

**DESIGN SPECIFICATION TOOLS**

After completing my requirement analysis I started to develop my proposed system. The next step of software development is System Design. The system design was divided into two phases:1. Logical design 2. Physical design  
  
Logical Design: A logical data flow diagram shows the flow of data through a transaction processing system without regard to the time period when the data flows or the processing procedures occur. Here I designed the software logically, using process modeling by Data Flow Diagram (DFD) and Data Modeling by ER-D technique.  
  
Process Modeling: A process modeling is a simplified representation of some aspect of real world process, which is used to help us analyze the real world. A model is a real time representation of the real world element, group of elements and the relationship among them. As MIS is very complex it is very useful to develop process model to understand complex problems.  
  
Data Flow Diagram (DFD) DFD is one of the most powerful and useful techniques available to the system analyst. It graphically represents the information system. Which means it shows all the incoming data, processing of data and finally output destination of the information.  
  
History of Data Flow Diagrams Data flow diagrams became popular in the 1970s in software development. They were first described in a classic text about Structured Design written by Larry Constantine and Ed Yourdon. Yourdon & Coad's Object Oriented Analysis and Design (OOA/OOD) was a way of visualizing software systems before UML diagrams.  
  
Data Flow Diagrams Notations There are essentially two different types of notations for data flow diagrams (Yourdon & Coad or Gane & Sarson) defining different visual representations for processes, data stores, data flow and external entities. Yourdon and Coad type data flow diagrams are usually used for system analysis and design, while Gane and Sarson type DFDs are more common for visualizing information systems. Visually, the biggest difference between the two ways of drawing data flow diagrams is how processes look. In the Yourdon and Coad way, processes are depicted as circles, while in the Gane and Sarson diagram the processes are squares with rounded corners. Process Notations. A process transforms incoming data flow into outgoing data flow.  
  
Data store Notations. Data stores are repositories of data in the system. They are sometimes also referred to as files.  
  
Dataflow Notations. Dataflows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.  
  
External Entity Notations. External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.  
  
Context Diagrams and DFD Layers and Levels Context Diagram. A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process 0") that generalizes the function of the entire system in relationship to external entities. DFD Layers. Draw data flow diagrams can be made in several nested layers. A single process node on a high level diagram can be expanded to show a more detailed data flow diagram. Draw the context diagram first, followed by various layers of data flow diagrams. DFD Levels. The first level DFD shows the main processes within the system. Each of these processes can be broken into further processes until you reach pseudo code.  
  
**DFD OF BANK ACCOUNT SYSTEM:-**  
ENTITY RELATIONSHIP DIAGRAM: An entity relationship diagram model, also called entity-relationship diagram, is a graphical representation of entities and their relationship to each other, typically used in computing in regard to organization of data within databases or information systems. An entity is a piece of data an object or concept about which data is stored.  
  
History of Entity Relationship Diagrams Peter Chen developed ERDs in 1976. Since then Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.  
  
Structure of an Entity Relationship Diagram with Common ERD Notations:An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD: •  
  
Entities, which are represented by rectangles. An entity is an object or concept about which you want to store information.  
  
•A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.

•Actions, which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.  
  
•Attributes, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.  
  
•A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.  
  
•A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.  
  
•Connecting lines, solid lines that connect attributes to show the relationships of entities in the diagram.  
  
RISK ANALYSIS:Some risks which occur usually along with some options for managing them.Logistics or product quality problems that block tests: These can be made moderate by careful planning, good defect triage and management, and robust test design. Test items that won’t install in the test environment: These can be mitigated through smoke (or acceptance) testing prior to starting test phases or as part of a nightly build or continuous integration. Having a defined uninstall process is a good contingency plan. Excessive change to the product that invalidates test results or requires updates to test cases, expected results and environments: These can be mitigated through good change-control processes, robust test design and light weight test documentation. When severe incidents occur, transference of the risk by escalation to management is often in order. Insufficient or unrealistic test environments that yield misleading results: One option is to transfer the risks to management by explaining the limits on test results obtained in limited environments. Mitigation – sometimes complete alleviation – can be achieved by outsourcing tests such as performance tests that are particularly sensitive to proper test environments.  
  
Let us also go through some additional risks and perhaps ways to manage them:   
  
Organizational issues such as shortages of people, skills or training, problems with communicating and responding to test results, bad expectations of what testing can achieve and complexity of the project team or organization. Supplier issues such as problems with underlying platforms or hardware, failure to consider testing issues in the contract or failure to properly respond to the issues when they arise. Technical issues related to ambiguous, conflicting or unprioritized requirements, an excessively large number of requirements given other project constraints, high system complexity and quality problems with the design, the code or the tests.  
  
DECISION TABLE   
  
Decision tables provide a systematic way of stating complex business rules, which is useful for developers as well as for testers. Decision tables can be used in test design whether or not they are used in specifications, as they help testers explore the effects of combinations of different inputs and other software states that must correctly implement business rules. It helps the developers to do a better job can also lead to better relationships with them. Testing combinations can be a challenge, as the number of combinations can often be huge. Testing all combinations may be impractical if not impossible. We have to be satisfied with testing just a small subset of combinations but making the choice of which combinations to test and which to leave out is also important. If you do not have a systematic way of selecting combinations, an arbitrary subset will be used and this may well result in an ineffective test effort.  
  
Decisions concerning credits granting are one of the most crucial issues in an everyday banks’ policy. Well-allocated credits may become one of the biggest sources of profits for any financial organizations. On the other hand, this kind of bank’s activity is connected with high risk as big amount of bad decisions may even cause bankruptcy. The key problem consists of distinguishing good (that surely repay) and bad (that likely default) credit applicants. Credit risk evaluation is an important and interesting management science problem in financial analysis. The main idea in credit risk evaluation investigations consists of building classification rules that properly assess bank customers as good or bad Let’s take another example. If you are a new customer and you want to open a credit card account then there are three conditions first you will get a 15% discount on all your purchases today, second if you are an existing customer and you hold a loyalty card, you get a 10% discount and third if you have a coupon, you can get 20% off today (but it can’t  
  
be used with the ‘new customer’ discount). Discount amounts are added, if applicable.  
  
conditions  
New customer  
Loyalty card  
Coupons  
Actions  
Discount  
  
We are applying this technique thoroughly, we would have one test for each column or rule of our decision table. The advantage of doing this is that we may test a combination of things that otherwise we might not have tested and that could find a defect. However, if we have a lot of combinations, it may not be possible or sensible to test every combination. If we are time-constrained, we may not have time to test all combinations. Don’t just assume that all combinations need to be tested. It is always better to prioritize and test the most important combinations. Having the full table helps us to decide which  
  
combinations we should test and which not to test this time. In the example above all the conditions are binary, i.e. they have only two possible values: True or False (or we can say Yes or No).  
  
Testing & Debugging Testing is a process of executing the program with the intent of finding errors and it establishes confidence that the program does what it is supposed to do. Testing can be performed in two ways:-  
  
Unit Testing: It is testing of individual module. Before initiating unit testing, it must be ensured that the code is peer previewed.  
  
Integration Testing: Integration testing is performed after all the software units are combined together. The objective here is to test the software interfaces. Project team conducts the integration testing. Before entering integration testing, it may be ensured that code review and unit testing have been performed on the individual software modules.  
  
I conducted the unit testing for finding errors and the results of the unit testing are recorded and action initiated for rectification of the errors/defects.  
  
Validation:

Almost every field in the database which is sensitive i.e. responsible for some important changes are validated. Basically we validated the field as such is  
  
numeric and the length in case of the primary keys and also used the Data base to complete most of the details once entered at the data entry time,  
  
FEATURES:

1. Customer account is created and maintained with all the proper customer details and loan related information. 2) All the equally monthly instalment related transaction information is showed in the customer receive form. 3) The entire customer related information for those customers who have missed their EMI are shown in the defaulter list. 4) Very easy to handle and operate. 5) As primary key is present so data duplication is blocked.

CONCLUSION  
  
Today loan management system has become the necessity for the banking staffs who handles the customers and their loan related transaction details. It provides a very easy and timesaving process for the banking staff to record the entire loan related customer details and to give proper information regarding their payable amount and instalment paying date. All the banks and loan management companies need this software for convenient transaction and loan giving process.

APPENDIX:DFD-

Data Flow Diagram ERD-Entity Relationship diagram SRS- System Requirement Specifications

GLOSSARY

1. DFD -It is the graphically representation of the information system, which means it shows all the incoming data, processing of data and finally output destination of the information

2. Er dIt- It is a graphical representation of entities and their relationship to each other, typically used in computing in regard to organization of data within databases or information systems.

3. Srs Software Requirement specification establish the basis for an agreement between customers and contractors on what software product is to do as well as what is not expected to do.