EX.NO:1 **PROBLEM STATEMENT PREPARATION FOR LIBRARY**

DATE : **MANAGEMENT SYSTEM**

Aim:

To prepare Problem Statement for Library Management System.

Problem Statement:

The Library management software for the purpose of monitoring and controlling the transactions in a library. A college library management is a project that manages and stores books information electronically according to students needs. The system helps both students and library manager to keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for colleges to keep a continuous check on the books issued and returned and even calculate fine. This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes. Thus this system reduces manual work to a great extent allows smooth flow of library activities by removing chances of errors in the detail.

Functionalities of Library Management System:

Admin login: Admin is the one who administers the system by adding or removing e-books into and from the system respectively.

User login: Students have to register themselves into the system to create an account. After registering successfully, they can then login into the system by entering 10 digit mobile number and their email id.

Add and Update Books: The admin can add books to the system by entering the details of the books and can even update the details.

Search option: Admin and Students can even search for books by entering the name of the book.

View Order: The admin can view order for the books.

Place order : The students can place order for the books and simultaneously the quantity of the book ordered will be decremented.

Calculate Fine: The student can view the issue and expiry date for the book issued and can even calculate fine.

Conclusion: The problem statement of Library Management System was written successfully by following the steps described above.

EX.NO:2 **PROBLEM STATEMENT PREPARATION FOR BANK MANAGEMENT**

DATE : **SYSTEM**

Aim:

To define Problem statement and Project scope of Bank Management System.

Information objective: Bank Management System would be able to maintain information and able to keep records of that particular event.

Problem Definition:

This system provides the basic services to manage bank accounts at a bank. Bank has many branches, each of which has an address and branch number. A client opens accounts at a branch. Each account is uniquely identified by an account number; it has a balance and a credit or overdraft limit. There are many types of accounts, including: A mortgage account, a chequing account, and a credit card account. It is also possible to have a joint account.

Although the basic type of services offered by a bank depends upon the type of bank and the country, services provided usually include: Taking deposits from their customers and issuing current or checking accounts and savings accounts to individuals and business. Extending loans to individuals and business, Cashing cheque. Facilitating money transactions such as wire transfer and cashiers cheque, Consumer & commercial financial advisory services, financial transaction can be performed through many different channels.

Motivation behind this project:

- To develop a system that will overlook the activities going transaction the particular bank without manual processing.
- All transaction will be updated automatically by using the information stored in record. The main motive behind this project is to develop a system which will able to handle the overall tasks going inside the institutions without much effort.

Overview of Bank Management System

There are various job positions within bank institutions. These positions are as follows:-

- Customer Service Position consists of a bank teller, bank marketing representative.
- Mid Management Position consists of internal auditor, data processing officer etc.
- Executive Level Position consists of loan officer, branch manager etc.

Bank Teller

- Bank tellers are the first line of advertising for a bank.
- Tellers must be friendly and have an attitude of service.
- Tellers are responsible for basic account transactions such as servicing savings and checking accounts and providing account details to customers.

Bank Marketing Representative

A bank marketing representative is an individual who markets banking products to customers.Products range from checking accounts and savings accounts to CD and special deposit accounts.The bank marketing representative is also responsible for understanding the strategic plan for marketing various products and implementing plans for the bank.

Internal Auditor

Audit is extremely important in banking environments.The internal audit position is considered a midlevel management position that ensures the bank is following compliance and regulatory laws concerning bank operations and procedures.The internal auditor is the individual responsible for creating audit programs for all areas of operations.Positions under the scope of internal auditor are Financial Auditor and Information Systems Auditor.Information Systems Auditor monitors data processing, data security and disaster recovery strategies for the bank.

Branch Manager

- Branch managers are assigned by midlevel or executive management to drive-through bank operation.
- Branch managers create branch work schedules for personnel, provide a budget to executive management concerning resources and ensure that bank policies are followed at the branch level.
- A branch manager can work from the main bank and have several branch banks assigned under her supervision.

Loan Officer

- Meets with applicants to obtain information for loan applications and to answer questions about the process.Analyze applicants' financial status, credit, and property evaluations to determine feasibility of granting loans.

- Explain to customers the different types of loans and credit options that are available, as well as the terms of those services.
- Obtain and compile copies of loan applicants' credit histories, corporate financial statements, and other financial information.
- Review and update credit and loan files.

Data Processing Officer

- The data processing officer (DPO) is responsible for the operation, maintenance and security of the bank information systems and offline terminals or devices not attached to the system.
- At the end of a daily processing cycle, the DPO ensures the general ledger account is balanced.
- The officer also ensures that daily transaction exception reports are created for managerial review.
- The DPO is responsible for creating security profiles for employees.

Scope of the project:

This project can be implemented in any bank by fulfilling basic requirements.

Conclusion: This Bank Management System will provide the transaction going inside the bank without manual processing. All information will be updated automatically by using the information stored in the system files.

EX.NO:3 SRS DOCUMENT PREPARATION FOR AUTOMATED BANKING SYSTEM

DATE :

AIM:

To Develop SRS Document for Automated Banking System.

PROBLEM STATEMENT:

To develop an automated banking system, which is required to perform the following functions:

• The customer logs into the system using card number and pin number. The system checks for validation.

• The system queries the customer for the type of account either fixed deposit or credit account. After getting the type of account the system shows the balance left.

• The system queries the customer for the transaction type either withdrawal or deposit and the required amount. The user enters the amount and the transaction if carries out

INTRODUCTION

1.Purpose

- The purpose of this SRS is to describe the requirements involved in developing an Automated Banding System (ABS).
- The intended audience is any person who wants
  - To create account.
  - To withdraw or deposit either in fixed deposit or credit account.

2.Scope

- The product is titled Automated Banking System (ABS).
- The product will perform the following tasks
  - Allow a new user to create an account, either fixed or credit account by entering the details and by depositing an initial amount.
  - Allow the existing user to enter his account details like card number, pin number and account type to view his balance.
  - Allow the existing user to deposit an amount by entering the amount to be deposited after the balance had been viewed.

- Allow the existing user to withdraw an amount by entering the amount to be withdrawn after the balance had been viewed.
- The primary benefits expected of the system are: user friendly, continuous connectivity without failure, fault tolerant and involves lesser manpower.
-

3.Definitions, Acronyms and Abbreviations

- ABS: Automated Banking System.

4.References

- IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.
- IEEE Software Requirements Specifications Template http://www.cas.master.ca/~carette/SE3M04/2003/fil es/srs_template.doc

5.Overview

- The SRS contains an analysis of the requirements necessary to help easy design.
- The overall description provides interface requirements for the Banking system, product perspective, hardware interfaces, software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.
- Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect banking system in any fashion.

THE OVERALL DESCRIPTION

1.Product perspective

2.Hardware interfaces

- Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system (suchas any RDBMS) running on high rpm hard-disk permitting complete data redundancy and backup systems to support the primary goal of reliability.
- The system must interface with the standard output device, keyboard and mouse to interact with this software.

3.Software interfaces

- Back End: Oracle
- Front End: Microsoft Visual Basic 6.0

4.Operations

- The user can create a new account.
- The existing user can access his account and view his balance by entering his details.
- The user can deposit and withdraw money from his account.

5.Product Functions

- Creating a New Account

  The user should provide his personal details to facilitate the bank clerk to create a new account. The user should provide:

  - Customer Name.
  - Customer address.
  - Required account type.
  - Pin Number.
  - Initial deposit.

- Operating with created account

  The user should be able to operate with his new account after:

  - Entering card number.
  - Entering pin number.
  - Entering the account type, transaction type and amount involved in the transaction.

6.User characteristics

- The intended users of this software need not have specific knowledge as to what is the internal operation of the system. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user
- The Product is absolutely user friendly, so the intended users can be the naïve users.

- The product does not expect the user to possess any technical background. Any person who knows to use the mouse and the keyboard can successfully use this product.

7..Constraints:

- At the time of creating the new account, each user gives a pin number and is provided with a unique card number that must be used for further transactions. Hence the user is required to remember or store these numbers carefully.
- At the time of creating the new account, the initial deposit should not be less than the specified amount.

Conclusion: This Bank Management System will provide the transaction going inside the bank without manual processing. All information will be updated automatically by using the information stored in the system files. The problem statement of Online Banking System was written successfully using the steps described above.

EX.NO:4 SRS DOCUMENT PREPARATION FOR FLIGHT MANAGEMENT SYSTEM

DATE :

AIM

The purpose of this document is to build an online system to manage flights and passengers to ease the flight management.

1.PROJECT SCOPE

The purpose of the online flight management system is to ease flight management and to create a convenient and easy-to-use application for passengers, trying to buy airline tickets. The system is based on a relational database with its flight management and reservation functions. We will have a database server supporting hundreds of major cities around the world as well as thousands of flights by various airline companies. Above all, we hope to provide a comfortable user experience along with the best pricing available.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

A distributed airline database system stores the following information.

- **Flight details:**
  It includes the originating flight terminal and destination terminal, along with the stops in between, the number of seats booked/available seats between two destinations etc.
- **Customer description:**
  It includes customer code, name, address and phone number. This information may be used for keeping the records of the customer for any emergency or for any other kind of information.
- **Reservation description:**
  It includes customer details, code number, flight number, date of booking, date of travel.

2.2 USER CLASS and CHARACTERISTICS

Users of the system should be able to retrieve flight information between two given cities with the given date/time of travel from the database. A route from city A to city B is a sequence of connecting flights from A to B such that: a) there are at most two connecting stops, excluding the starting city and destination city of the trip, b) the connecting time is between one to two hours. The system will support two types of user privileges, Customer, and Employee. Customers will

have access to customer functions, and the employees will have access to both customer and flight management functions. The customer should be able to do the following functions:

- Make a new reservation
  - Multi-city
  - Flexible Date/time
  - Confirmation
- Cancel an existing reservation
- View his itinerary

The Employee should have following management functionalities:

- CUSTOMER FUNCTIONS.
  - Get all customers who have seats reserved on a given flight.
  - Get all flights for a given airport.
  - View flight schedule.
  - Get all flights whose arrival and departure times are on time/delayed.
  - Calculate total sales for a given flight.
- ADMINISTRATIVE
  - Add/Delete a flight
  - Add a new airport
  - Update fare for flights.
  - Add a new flight leg instance.
  - Update departure/arrival times for flight leg instances.

Each flight has a limited number of available seats. There are a number of flights which depart from or arrive at different cities on different dates and time.

2.3 OPERATING ENVIRONMENT

Operating environment for the airline management system is as listed belowdistributed database

- client/server system
- Operating system: Windows.
- database: sql+ database
- platform: vb.net/Java/PHP

2.4 DESIGN and IMPLEMENTATION CONSTRAINTS

- The global schema, fragmentation schema, and allocation schema.
- SQL commands for above queries/applications

- How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
- Implement the database at least using a centralized database management system.

2.5 ASSUMPTION DEPENDENCIES

Let us assume that this is a distributed airline management system and it is used in the following application:

- A request for booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.
- Calculation of high fliers (most frequent fliers) and calculating appropriate reward points for these fliers.

Assuming both the transactions are single transactions, we have designed a distributed database that is geographically dispersed at four cities Delhi, Mumbai, Chennai, and Kolkatta as shown in fig. below.

3. SYSTEM FEATURES

3.1 DESCRIPTION and PRIORITY

The airline reservation system maintains information on flights, classes of seats, personal preferences, prices, and bookings. Of course, this project has a high priority because it is very difficult to travel across countries without prior reservations.

3.2 STIMULUS/RESPONSE SEQUENCES

- Search for Airline Flights for two Travel cities
- Displays a detailed list of available flights and make a “Reservation” or Book a ticket on a particular flight.
- Cancel an existing Reservation.

3.3 FUNCTIONAL REQUIREMENTS

Other system features include:

3.4 DISTRIBUTED DATABASE:

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as shown in below figure.

3.5 CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACES

- Front-end software: Vb.net version
- Back-end software: SQL+

4.2 HARDWARE INTERFACES

- Windows.
- A browser which supports CGI, HTML & Javascript.

4.3 SOFTWARE INTERFACES

Following are the software used for the flight management online application.

Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the flight records, passengers records we have chosen SQL+ database.
VB.Net	To implement the project we have chosen Vb.Net language for its more interactive support.

4.4 COMMUNICATION INTERFACES

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of airline database are as listed below.

A) E-R DIAGRAM

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ENTITIES:** Which specify distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
- **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

B) NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes.

5.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4 SOFTWARE QUALITY ATTRIBUTES

- **AVAILABILITY:** The flight should be available on the specified date and specified time as many customers are doing advance reservations.
- **CORRECTNESS:** The flight should reach start from correct start terminal and should reach the correct destination.
- **MAINTAINABILITY:** The administrators and flight in chargers should maintain correct schedules of flights.
- **USABILITY:** The flight schedules should satisfy a maximum number of customers needs.

Conclusion:

Thus the Software Requirment Specification document for Flight Management System was written successfully

EX.NO:5 **ENTITY RELATIONSHIP DIAGRAM FOR LIBRARY MANAGEMENT**

DATE: **SYSTEM**

AIM

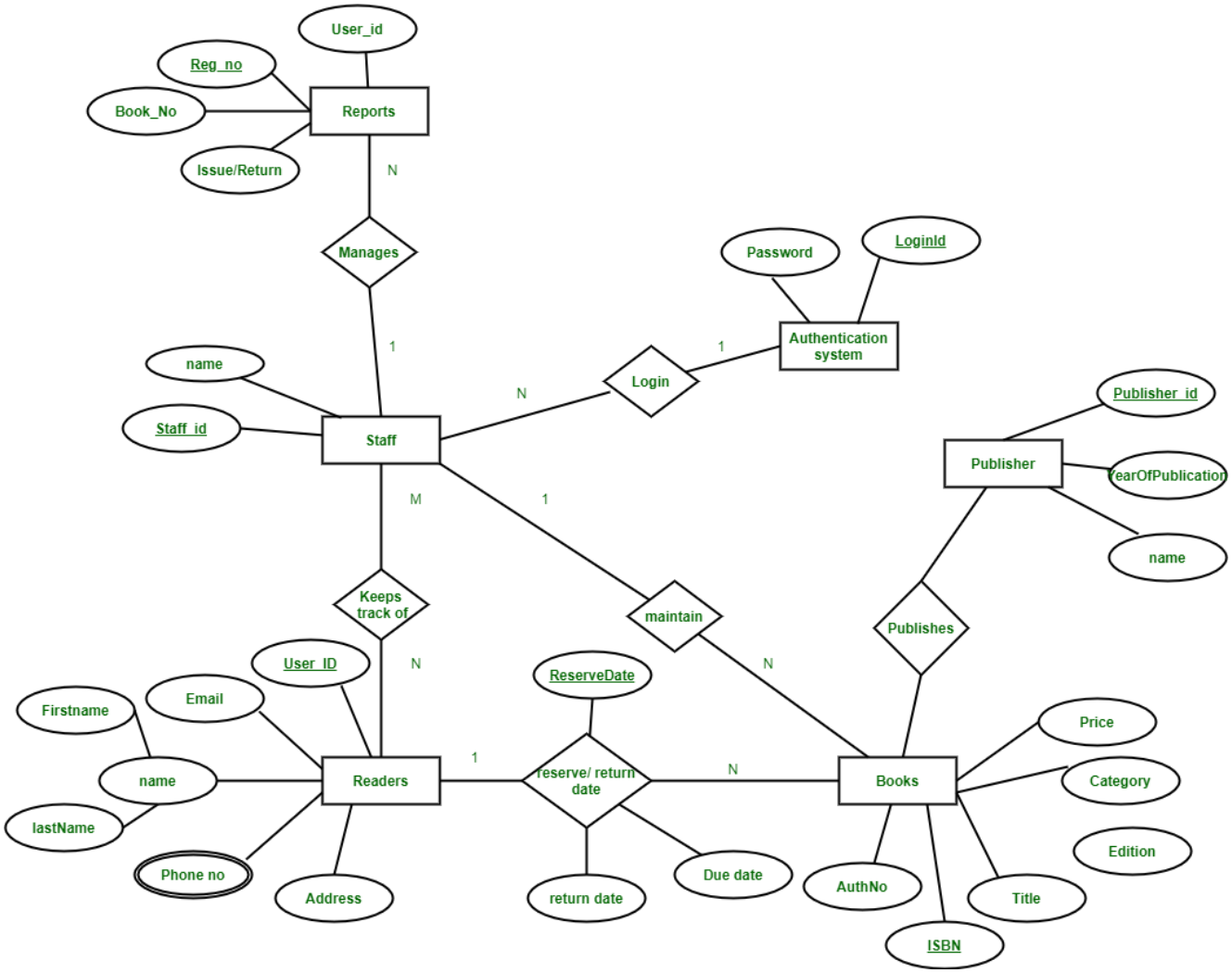
To design the Library Management System using Entity Relationship Diagram

PROCEDURE

ER Diagram is known as Entity-Relationship Diagram, it is used to analyze to the structure of the Database. It shows relationships between entities and their attributes. An ER Model provides a means of communication.

The Library Management System database keeps track of readers with the following considerations :

- The system keeps track of the staff with a single point authentication system comprising login Id and password.
- Staff maintains the book catalog with its ISBN, Book title, price(in INR), category(novel, general, story), edition, author Number and details.
- A publisher has publisher Id, Year when the book was published, and name of the book.
- Readers are registered with their user_id, email, name (first name, last name), Phone no (multiple entries allowed), communication address. The staff keeps track of readers.
- Readers can return/reserve books that stamps with issue date and return date. If not returned within the prescribed time period, it may have a due date too.
- Staff also generate reports that has readers id, registration no of report, book no and return/issue info.



ER Diagram of Library Management System

This Library ER diagram illustrates key information about the Library, including entities such as staff, readers, books, publishers, reports, and authentication system. It allows for understanding the relationships betweenEntities.

Entities and their Attributes:

- **Book Entity :** It has authno, isbn number, title, edition, category, price. ISBN is the Primary Key for Book Entity.
- **Reader Entity :** It has UserId, Email, address, phone no, name. Name is composite attribute of firstname and lastname. Phone no is multi valued attribute. UserId is the Primary Key for Readers entity.
- **Publisher Entity :** It has PublisherId, Year of publication, name. PublisherID is the Primary Key.

- **Authentication System Entity :** It has LoginId and password with LoginID as Primary Key.
- **Reports Entity :** It has UserId, Reg_no, Book_no, Issue/Return date. Reg_no is the Primary Key of reports entity.
- **Staff Entity :** It has name and staff_id with staff_id as Primary Key.
- **Reserve/Return Relationship Set :** It has three attributes: Reserve date, Due date, Return date.

Relationships between Entities

- A reader can reserve N books but one book can be reserved by only one reader. The relationship 1:N.
- A publisher can publish many books but a book is published by only one publisher. The relationship 1:N.
- Staff keeps track of readers. The relationship is M:N.
- Staff maintains multiple reports. The relationship 1:N.
- Staff maintains multiple Books. The relationship 1:N.
- Authentication system provides login to multiple staffs. The relation is 1:N.

Conclusion:

Thus the ER Diagram for Library management system is drawn.

EX.NO:6 **ENTITY RELATIONSHIP DIAGRAM FOR BANKING SYSTEM**

DATE:

AIM

To design the Automated Banking System using ER Diagram

PROCEDURE

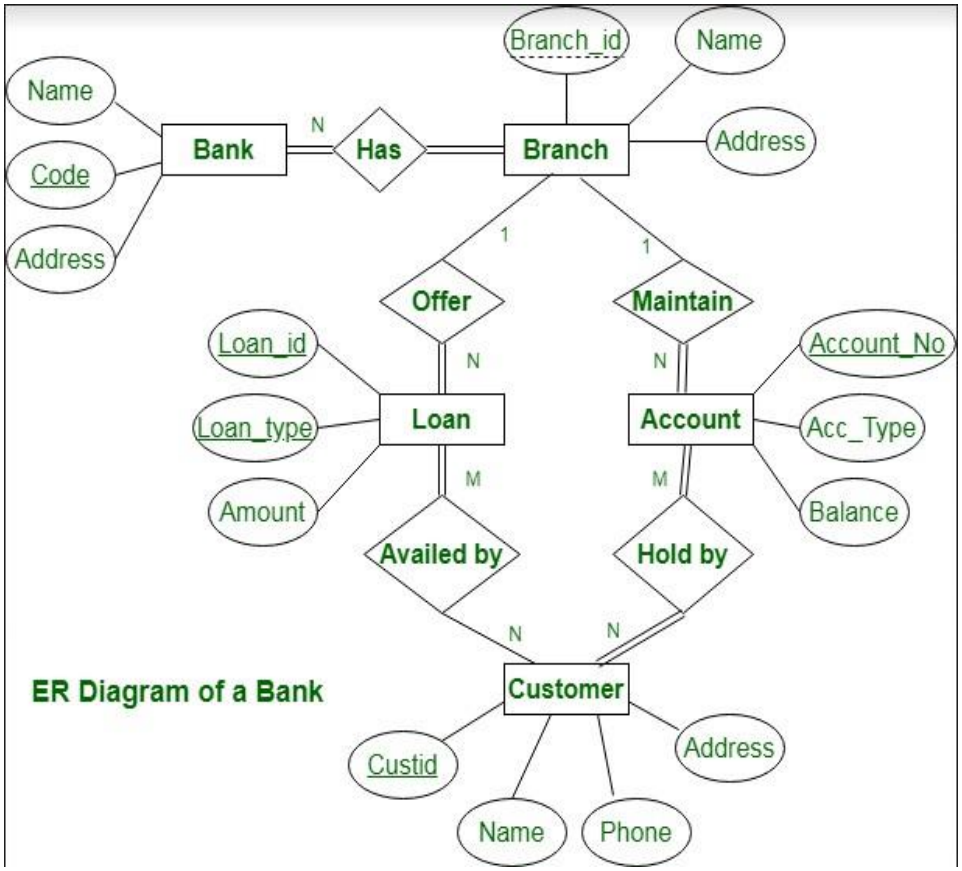
ER diagram is known as Entity-Relationship diagram. It is used to analyze to structure of the Database. It shows relationships between entities and their attributes. An ER model provides ameans of communication.

ER diagram of Bank has the following description :

Bank have Customer.

- Banks are identified by a name, code, address of main office.
- Banks have branches.
- Branches are identified by a branch_no., branch_name, address.
- Customers are identified by name, cust-id, phone number, address.
- Customer can have one or more accounts.
- Accounts are identified by account_no., acc_type, balance.
- Customer can avail loans.
- Loans are identified by loan_id, loan_type and amount.
- Account and loans are related to bank's branch.

ER Diagram of Bank Management System :



This bank ER diagram illustrates key information about bank, including entities such as branches, customers, accounts, and loans. It allows us to understand the relationships betweenentities.

Entities and their **Attributes** are :

- **Bank Entity :** Attributes of Bank Entity are Bank Name, Code and Address.
  Code is Primary Key for Bank Entity.
- **Customer Entity :** Attributes of Customer Entity are Customer_id, Name, Phone Number and Address.
  Customer_id is Primary Key for Customer Entity.

- **Branch Entity :** Attributes of Branch Entity are Branch_id, Name and Address.
  Branch_id is Primary Key for Branch Entity.
- **Account Entity :** Attributes of Account Entity are Account_number, Account_Type and Balance.
  Account_number is Primary Key for Account Entity.
- **Loan Entity :** Attributes of Loan Entity are Loan_id, Loan_Type and Amount.
  Loan_id is Primary Key for Loan Entity.

Relationships are :

- **Bank has Branches => 1 : N**
  One Bank can have many Branches but one Branch can not belong to many Banks, so therelationship between Bank and Branch is one to many relationship.

  Branch maintain Accounts => 1 : N
  One Branch can have many Accounts but one Account can not belong to many Branches, sothe relationship between Branch and Account is one to many relationship.

  Branch offer Loans => 1 : N
  One Branch can have many Loans but one Loan can not belong to many Branches, so therelationship between Branch and Loan is one to many relationship.

  Account held by Customers => M : N
  One Customer can have more than one Accounts and also One Account can be held by oneor more Customers, so the relationship between Account and Customers is many to many relationship.

  Loan availed by Customer => M : N
  (Assume loan can be jointly held by many Customers).
  One Customer can have more than one Loans and also One Loan can be availed by one ormore Customers, so the relationship between Loan and Customers is many to many relationship

Conclusion:

Thus the ER Diagram for bank system is drawn.

EX.NO:7 **DATAFLOW DIAGRAM FOR ONLINE BANKING SYSTEM**

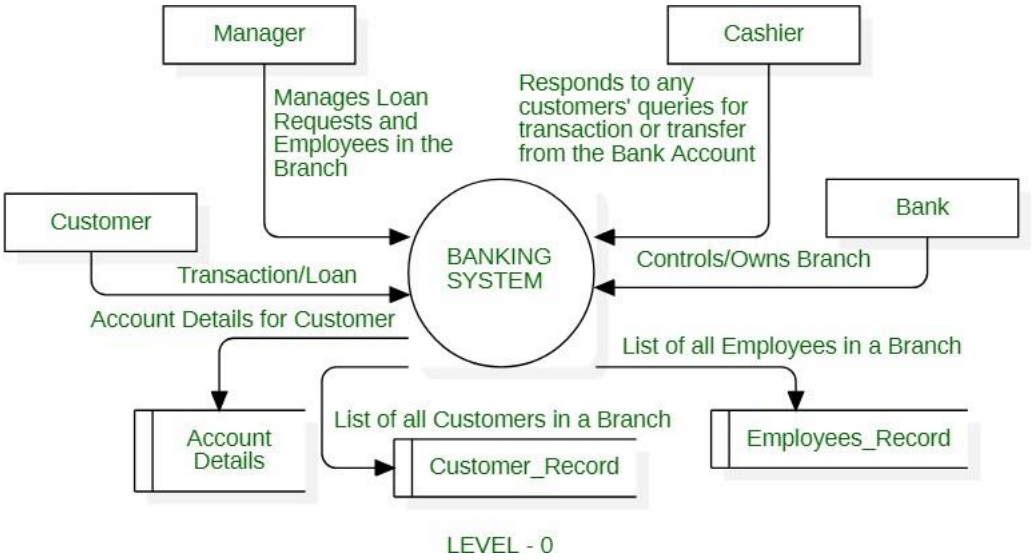
DATE:

AIM

To design the Automated Banking System using Dataflow Diagram

PROCEDURE

Data Flow Diagrams are used to represent the flow of data as well as the processes and functionsinvolved to store, manipulate, and distribute data among various components of the system andbetween the system and the environment of the system by a specific set of graphicalrepresentations. It also depicts the logical flow of information in a system and appropriatelydefines and determines the physical requirements for the construction of the system. The keyfeatures of a Data Flow Diagram involve simplicity of notation as well as generation of a clear overview of the manual and automated requirements of the system. The system is described as follows –



DFD — Online Banking System

Explanation :

1. **Transaction Request/Loan Request –**

   In this customer provides the specific details to initiate or support a transaction and the customer can apply for a loan through the system.

2. **Manager –**

   The manager can access and manage any request as well as input commands to change or update Employee_Record Database. A Manager manages the employees of the bank as well as either accepts or rejects any Loan Request and hence inputs details to update the corresponding databases as Customer_Record and Employee_Record Database.

3. **Cashier –**

   It responds actively to any Transaction Request and assists the Manager. The cashier can alsoresolve any query from the Customer and provide immediate help to the Customer. However, a Cashier can not access any database independently.

4. **Bank_Headquarter –**

   It acts as an entity that controls all the branches of the Bank. The entity can access all the databases, and it also controls the operations of the Manager. The Bank can modify or eliminate any policies as well as create new policies for the Online Banking System.

5. **Customer_Record –**

   It stores all the non-personal details of an individual who is a Customer of any specific branchof the Bank. As an individual initiates a request to open an account in the Bank, Customer_Records creates a new entry that is unique and specific to the Customer only. Thisrecord is accessible by Managers or Employees of the Banks as and when required.

6. **Employee_Record –**

   It stores the record of all the Employees at the Bank. The record can be accessed but not modified by the Customer of the Bank. The record can be modified by either the Manager of a Branch or by the Bank itself.

7. **Account_Details –**

   It stores the personal details of every individual Customer. It can be accessed/modified by the Customer only. The Bank can not modify the personal details of the Customer.

Conclusion:

Thus DFD diagram for Bank system is drawn.

EX.NO:8 **USECASE DIAGRAM FOR AUTOMATED BANKING SYSTEM**

DATE:

AIM

To design the Automated Banking System using Usecase Diagram

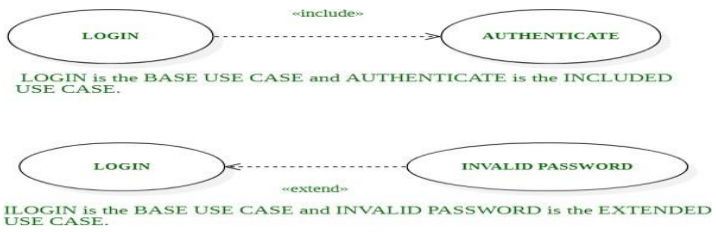
PROCEDURE

The Use Case Diagram is a UML Diagram where the each use-case specifies the behaviour expected from software from the perspective of end-user and relation as well as provides brief overview for different components concerning interaction between use-case, actors and systems . The Use-Case Diagram is used to prepare, present and understand functional requirements of the system. Use-Case Diagram specifies exact context of the software being developed. It does not specifies order in which actions must be performed. Each use-case represents function of system which is either process-automated or manual.

Difference between **<<include>>** and **<<extend>>** in **Use Case Diagram :**

- ***<<include>>*** extends **Base Use Case** and it specifies that an **Included Use Case** must run successfully to complete Base Use Case. The Base Use Case is incomplete in the absence of an Included Use Case. The Included Use Case can be Base Use Case itself or it might be shared by a number of distinct Base Use Cases.
- ***<<extend>>*** on the other end , is used to add an **Extended Use Case** which extends the **BaseUse Case.** Base Use Case can run successfully even without invoking/calling extended use case called Optional Use Case. The Base Use Case is complete in itself but under certain conditions it would require to refer to extension condition.

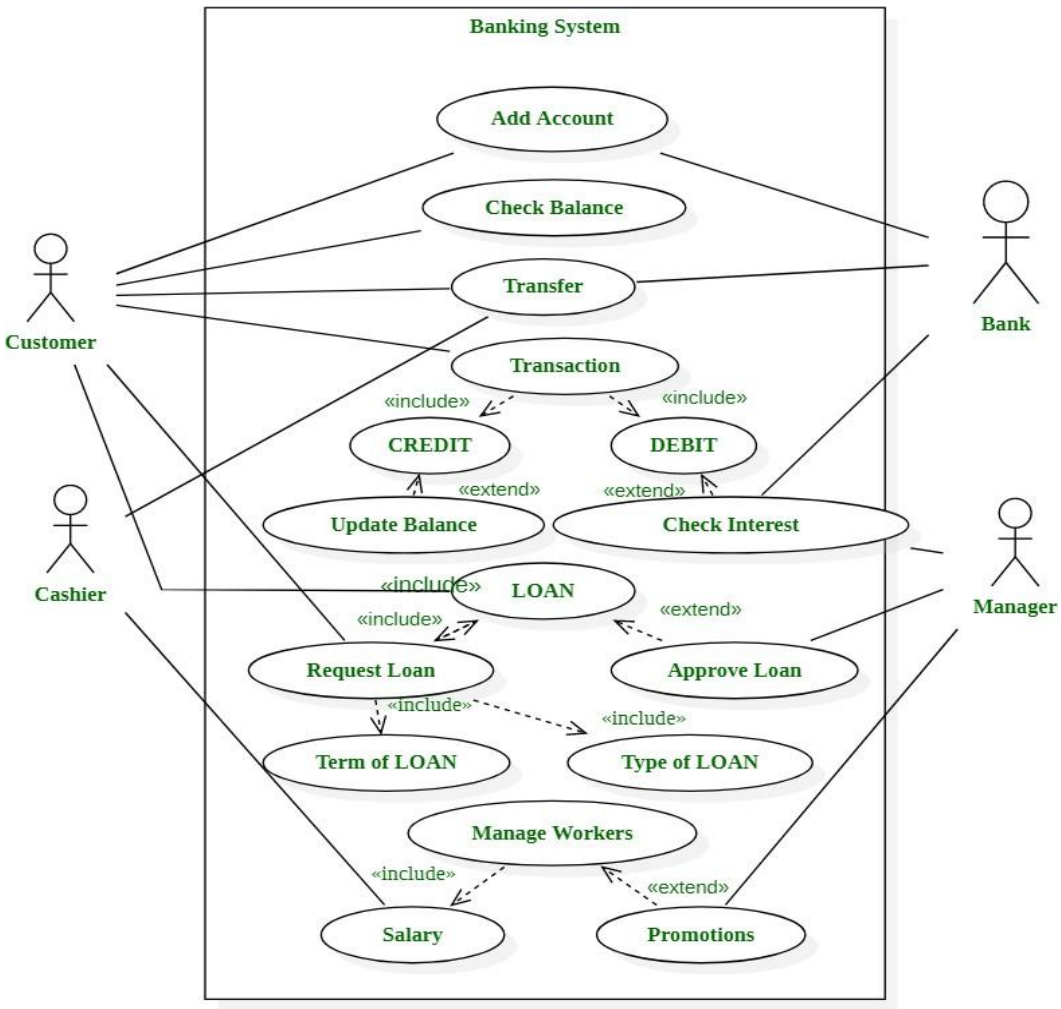
The representation of <<include>> and <<extend>> is as below :-



A user must provide appropriate details to securely Login. Software must check and verify thedetails at every attempt to Login. Here LOGIN is the Base Use Case and AUTHENTICATE isthe **Included Use Case**.

If a user enters appropriate details , user is allowed to Login. However if the details entered bythe user are incorrect, software must be able to catch and display problem to the user and allowthe user to re-enter details. LOGIN is hence a complete use case. However under certain situations it might use action corresponding to INVALID PASSWORD. Here LOGIN is the Base Use Case and INVALID PASSWORD is **Extended Use Case**.

The Use- Case Diagram for an Online Banking System is as follows :



Here, we will try to understand the design of a use case diagram for the Online Banking System**.**

Some possible scenarios of the system are explained as follows :

1. A Customer is required to create an account to avail services offered by Bank. Bank verifies detail and creates new account for each new customer. Each customer is an actor for the Use-Case Diagram and the functionality offered by Online Banking System to Add Account is Use-Case.
2. Each customer can check the balance in bank account and initiate request to transfer an account across distinct branches of Bank. Cashier is an employee at bank who supportsservice to the customer.
3. A customer can execute cash transactions where the customer must either add cash value tobank account or withdraw cash from account. Either of two or both that is credit as well as debit cash, might be executed to successfully execute one or multiple transactions.
4. After each successful transaction customer might or might not want to get details for action.Manager can check interest value for each account corresponding to transaction to ensure and authenticate details.
5. A customer can also request loan from bank where customer must add request for loan withthe appropriate details.
6. The type of loan in accordance with purpose or the need for loan and term or duration to payback the loan must be provided by customer.
7. The manager of each branch of bank has choice to either accept or approve loan to initiateprocess further or just reject request for loan based on terms and conditions.
8. The record for each employee of bank is maintained by bank and bank manages all employees of each branch of bank. The manager of each branch has choice to offer bonus toemployees. Note here that each employee is paid as part of management of staff but promotion or bonus might or might not be offered certainly to each employee.

Conclusion:

Thus the use case diagram for Automated bank system is drawn.

EX.NO:9 **TEST CASE FOR LIBRARY MANAGEMENT SYSTEM**

DATE:

AIM

To write the test case for Library Management System.

PROCEDURE

Library is the place with the huge collection of books. It is place from where the students and thefaculties issue the books for their reference purposes. But the maintenance of keeping the records of issuing and borrowing is difficult if you use a normal book as a registry. To make this task easier, the library management system will be very useful. The test cases for library management system is an application that explains the test cases for library management system. Software testing is a critical part that is involved in the overall development of the application. This will beone of the interesting projects that one can work on and implement in real time world.

Test Unit: Admin Component

- **Adding Login page**

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1.	If we are not enter the valid roll no or LOGIN ID.	LOGIN ID or Roll no	Please fill out this field.	SUCCESSFUL
2.	If we only put the valid roll no or LOGIN ID and after pressing the login button.	Your password	Please fill out this field.	SUCCESSFUL
3.	If we are putting the correct user name and wrong password and after selecting the valid login credential.	Select the user type	Kindly select the status	SUCCESSFUL

- **Adding Registration form:**

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1	If we are not selected any field.	Select Course	Kindly select the fields.	SUCCESSFUL
2	If we are not entered the name.	Name	Please enter the name.	SUCCESSFUL
3	If we are not entered the valid roll no.	Roll no	Please enter the valid roll no.	SUCCESSFUL
4	If we are not entered the valid password.	Enter Password	Please enter the valid password.	SUCCESSFUL
5	If the password is not same as previous password.	Confirm password	Please enter the valid password once again.	SUCCESSFUL
6	If we are not entered the valid emailid.	Email id	Kindly insert the emailid.	SUCCESSFUL
7	If we are not suppose to enter contact no.	Contact no.	Please enter the contact no.	SUCCESSFUL
8	If we are not suppose to add any image files on the choose File.	Choose file	Kindly upload Image.	SUCCESSFUL

Adding admin/branch

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1.	If we are not enter the branch name.	Branch name	Please fill out this field.	SUCCESSFUL
2.	If the branch description not to entered.	Branch description	Please fill out this field.	SUCCESSFUL
3.	If we not selected any status	Branch status	Kindly select the status	SUCCESSFUL

- **Admin/book:**

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1.	If you are not selected category.	Book category	Please fill out this field.	SUCCESSFUL
2.	If we are not enter the book name.	Book name	Please enter book name.	SUCCESSFUL
3	If we are not enter the book cost fields.	Book cost	Please enter book cost.	SUCCESSFUL
4	If we are not enter book author fields.	Book author	Please enter book author.	SUCCESSFUL
5	If we not selected any status.	Book status	Kindly select status.	SUCCESSFUL
6	If we are not suppose to add any image files on the choose File.	Book Image	Kindly upload Image.	SUCCESSFUL
7	If we are not enter book description.	Book description	Book description should not be empty.	SUCCESSFUL

Admin/Book category:

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1.	If we are not enter the valid book category.	Book category	Please fill out this fields.	SUCCESSFUL
2.	If we are not suppose the book description.	Book description	Book description should not be empty..	SUCCESSFUL
3.	If we are not selected status.	Book status	Kindly select status	SUCCESSFUL

- **Admin/Course:**

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1.	If you are not select the branch	Select branch.	Please enter the valid fields.	SUCCESSFUL
2.	If the course name not entered.	Course	Please enter the course.	SUCCESSFUL
3.	If the course note did not entered.	Course note	Please You should fill out course note.	SUCCESSFUL
4.	If we are not selected status.	Course status	Kindly select the status.	SUCCESSFUL

Admin/Librarian:

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1	If we are not entered the name.	Name	Please enter the name.	SUCCESSFUL
2	If we are not selected the type.	Type	Please select the type.	SUCCESSFUL
3	If we are not entered the valid login id	Login id	Enter the valid Login id.	SUCCESSFUL
4	If we are not entered the password field.	Password	Enter the valid password.	SUCCESSFUL
5	If we are not entered the same password.	Confirm password	Enter the valid password once again.	SUCCESSFUL
6	If we are not selected any status.	Status	Kindly select the status	SUCCESSFUL

- **Admin/Student:**

Serial No.	Condition To be Tested	Test Data	Expected Output	Remarks
1	If we are not selected any course.	Course	Please fill out the fields.	SUCCESSFUL
2	If we are not entered the student name.	Student name	Please enter the student name.	SUCCESSFUL
3	If we are not entered the valid roll no.	Roll no	Please enter the valid roll no.	SUCCESSFUL

4	If we are not entered the valid password.	Password	Please enter the valid password.	SUCCESSFUL
5	If the password is not same as previous password.	Confirm password	Please enter the valid password once again.	SUCCESSFUL
6	If we are not entered the valid emailid.	Email id	Kindly insert the emailid.	SUCCESSFUL
7	If we are not suppose to enter contact no.	Contact no.	Please enter the contact no.	SUCCESSFUL
8	If we are not selected status	Status	kindly select status.	SUCCESSFUL
9	If we are not suppose to choose the image file.	Student image	Kindly upload the image.	SUCCESSFUL

Conclusion:

Thus, the test cases for library management system are created.

EX.NO:10 **TEST CASE FOR TRAFFIC SIGNAL**

DATE:

AIM

To write the test case for Traffic Signal.

PROCEDURE

This procedure we have tried to cover maximum test scenarios for the traffic signal. We includedpositive and negative test cases for traffic signal.Before we write test cases for traffic signal,

let's first understand the lights.

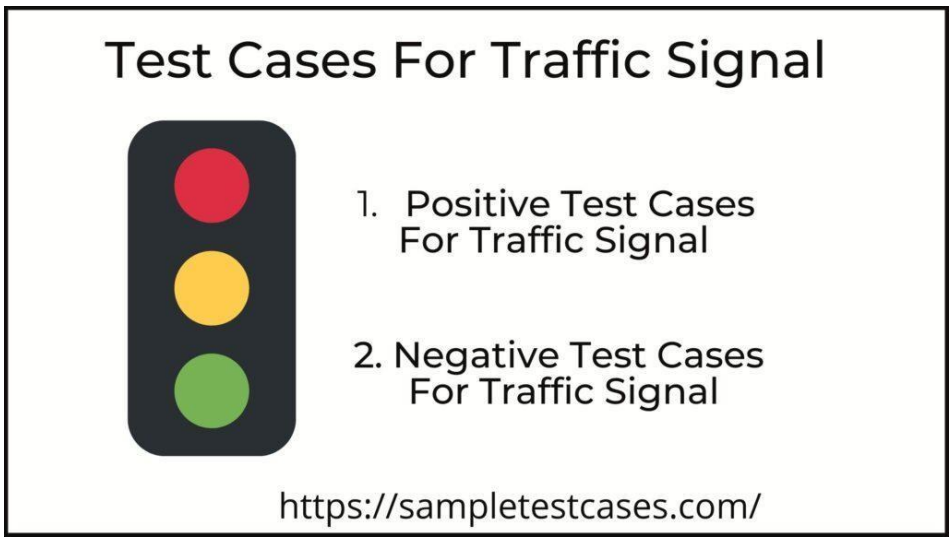
Red – stop and wait before the zebra crossing line.Green

– go.

Sr.No	Testcase_ID	Test Cases
1	Testcase_01	Verify that traffic signal should be have three lights Red, Yellow, and Green
2	Testcase_02	Verify that traffic signal lights should be turned on as per the traffic rule
3	Testcase_03	Verify the time of the Red light traffic signal
4	Testcase_04	Verify the time of the Yellow light traffic signal
5	Testcase_05	Verify the time of the Green Light traffic signal
6	Testcase_06	Verify that the traffic signal lights should be properly visible to the human or not
7	Testcase_07	Verify that power connection to the traffic signal
8	Testcase_08	Verify that if the RED light is On then the rest of the two lights should be OFF
9	Testcase_09	Verify that if the GREEN light is On then the rest of the two lights should be OFF

10	Testcase_10	Verify that if the YELLOW light is On then the rest of the two lights should be OFF
11	Testcase_11	Verify that traffic signal location as per requirement or not

Yellow – stop, unless – 1) you've already crossed the stop line after green light is off. 2) you're so close to it that pulling up might cause an accident. 3) You're in middle of the road and shouldmove to next side rather than going back causing traffic problem.



Positive Test Cases – Traffic Signal

Negative Test Cases – Traffic Signal

Sr.No	Testcase_ID	Test Cases
1	Negative_01	Check the traffic signal conditions in the heavy rain
2	Negative_02	Check that any lights are deemed or not
3	Negative_03	Check that if any chance to short circuit if rain water drops on traffic signal
4	Negative_04	Check that if traffic signal is dust free or not

Conclusion:

Thus the test cases on traffic signals are created.

EX.NO:11 **TESTING CALCULATOR**

DATE:

AIM

To write the test case for Calculator

PROCEDURE

Test cases for the calculator depends on the type – scientific, simple calculator, financial calculatoror some other specific calculator like programmable calculator.

For the sake of simplicity, we are choosing simple calculator for the test cases. You can come up with plenty of test cases in order to test calculator based on the mathamatical computation. However, we are here testing the calculator to see if it is fit for cosumer use. So we have to makesure that it fits for the most obvious tests. If those tests are passed then we can safely assume thatproduct is fit for use.

Basic Operational Tests

Write the test cases based on the following functions and scenarios.

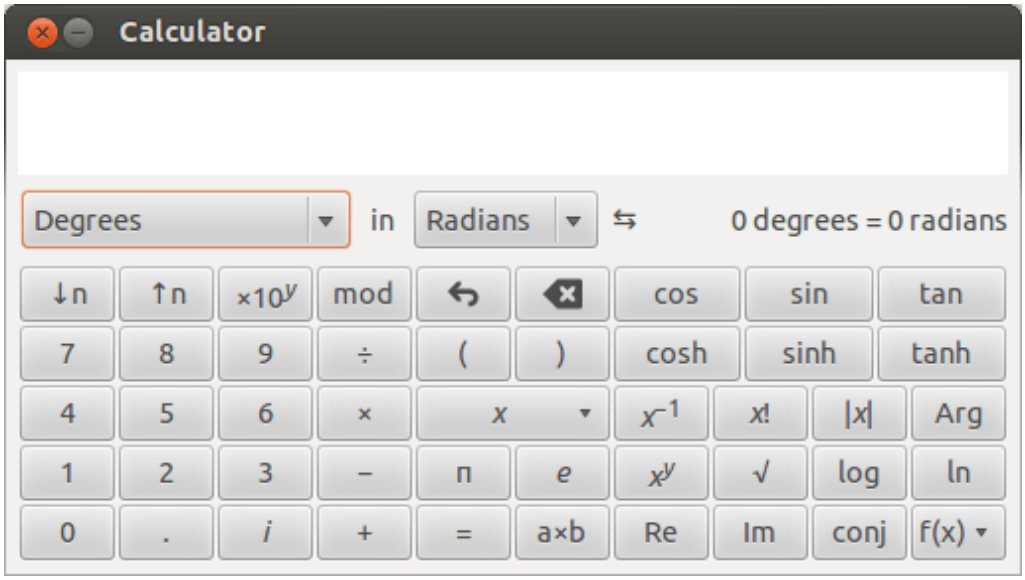
- Check the calculator if it starts by on button. If it is software based calculator then check if it starts via specific means like from searching for calculator in search bar and then executing application. Or by accessing menu item in the Windows.
- Check if the calculator window maximizes to certain window size.
- Check the if the calculator closes when the close button is pressed or if the exit menu isclicked from file > exit option.
- Check if the help document is accessed from Help > Documentation.
- Check if the calculator allows copy and paste functionality.
- Check if the calculator has any specific preferences.
- Check if all the numbers are working (0 to 9)
- Check if the arithmetic keys (+, -, *, %, /) are working.
- Check if the clear key is working.
- Check if the brackets keys are working.
- Check if the sum or equal key is working.
- Check if the square and square root key is working.

Functionality Test Cases

- Check the addition of two integer numbers.
- Check the addition of two negative numbers.
- Check the addition of one positive and one negative number.
- Check the subtraction of two integer numbers.
- Check the subtraction of two negative numbers.
- Check the subtraction of one negative and one positive number.
- Check the multiplication of two integer numbers.
- Check the multiplication of two negative numbers.
- Check the multiplication of one negative and one positive number.
- Check the division of two integer numbers.
- Check the division of two negative numbers.
- Check the division of one positive number and one integer number.
- Check the division of a number by zero.
- Check the division of a number by negative number.
- Check the division of zero by any number.
- Check if the functionality using BODMAS/BIDMAS works as expected.

Advanced Tests on Scientific Calculator

If your calculator has advanced features as shown in the screenshot.



You can add few more tests in the scientific calculator.

- Check if the sin, cos, tan and cos is operational using the keys.
- Check if the x-1, x!,|x|,x^y and f(x) is operational and works as expected.
- Check if the log key is operational and works as expected.
- Check if the natural logarithm key i operational and works as expected.
- Check if the factorial key is working as expected.
- Check if the real and imaginary component keys are working as expected.
- Check if the complex conjugate keys are working as expected.

Conclusion:

Thus the test cases on calculator is created successfully.

EX.NO:12 **TESTING MARKSHEET**

DATE:

AIM:

To validate the data in student marksheet by applying testing strategies.

PROCEDURE:

An individual report card of each student has to be displayed and printed at a keystroke accordingto any selected format. An important aid for teachers and students to judge their performance. Merit list printing by totals for a class by individual subject marks for a class. Student performancein a particular subject or all the subjects must be expressed. Performance of teachers of various classes can be easily compared.The system displays the list of all issues that are open, closed, in progress. If the user can get registered by clicking on the logon button and provide the required information asspecified. Each time the registered customer come on to the site he can makes use of theuser name and the password that is allocated to him.

Software requirements

Operating System: Window 2000, XPUser interface : Java, Servlets, JSPDatabase : My SQLDocumentation Tool : Ms Office

Hardware requirements

- Processo:Standard processor with a speed of 1.6 GHz or more
- RAM:256 MB RAM or more
- Hard Disk :20 GB or more
- Monitor :Standard color monitor
- Keyboard:Standard keyboard
- Mouse:Standard mouse

Scope

Existing System with Limitations

1. It is time consuming process as the user has to type the dbase commands. He has toremember all the commands which are difficult.
2. It is limited to a single system.

3. A user who wants only to have some information has to contact the administrator everytime.

Proposed System Features

- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- It can be accessed over the internet.
- Vast amount of data can be stored.
- There is no risk of data mismanagement at any level while the project development isunderprocess.
- Relationship between the administrator, owner/developer and subcontractor can bemaintained very easily.
- It provides high level of security using different protocols like https etc.

The Student Result Processing consists of 3 users or modules, they are:

- Administrator
- Student
- Staff

Testing Methodologies

- Black box Testing
- White box Testing
- Gray Box Testing

Testing:

- The process of executing a system with the intent of finding an error.
- Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
- Quality is defined as justification of the requirements. Defect is nothing but deviation fromthe requirements. Defect is nothing but bug.
- Testing --- The presence of bugs.
- Testing can demonstrate the presence of bugs, but not their absence

- Debugging and Testing are not the same thing!
- Testing is a systematic attempt to break a program or the AUT
- Debugging is the art or method of uncovering why the script /program did not execute properly.

Testing Methodologies

Black box Testing : is the testing process in which tester can perform testing on an applicationwithout having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

White box Testing: is the testing process in which tester can perform testing on anapplication withhaving internal structural knowledge. Usually The Developers are involved in white box testing.

Gray Box Testing: is the process in which the combination of black box and white box techniquesare used.

Positive Test Cases:

- The positive flow of the functionality must be considered

Valid inputs must be used for testing

- Must have the positive perception to verify whether the requirements are justified.

Example for Positve Test cases:

Example for GUI Test cases:

T.C. No	Description	Expected value	Actual value	Result
1	Check for all the features in the screen	The screen must contain all the features		
2	Check for the alignment of the objects as per the validations	The alignment should be in proper way		

Negative Test Cases:

- Must have negative perception.
- Invalid inputs must be used for test.

Example for Negative Test cases

T.C. No	Description	Expected value	Actual value	Result
1	Try to modify the information in date and time	Modification should not be allow		
2	Enter invalid data in to the student details form, click on Save	It should not accept invalid data, save should not allow		

Conclusion: Hence positive and negative and test cases are defined for marksheet testing

EX.NO:13 **TESTING USAGE OF TEXT**

DATE:

AIM:

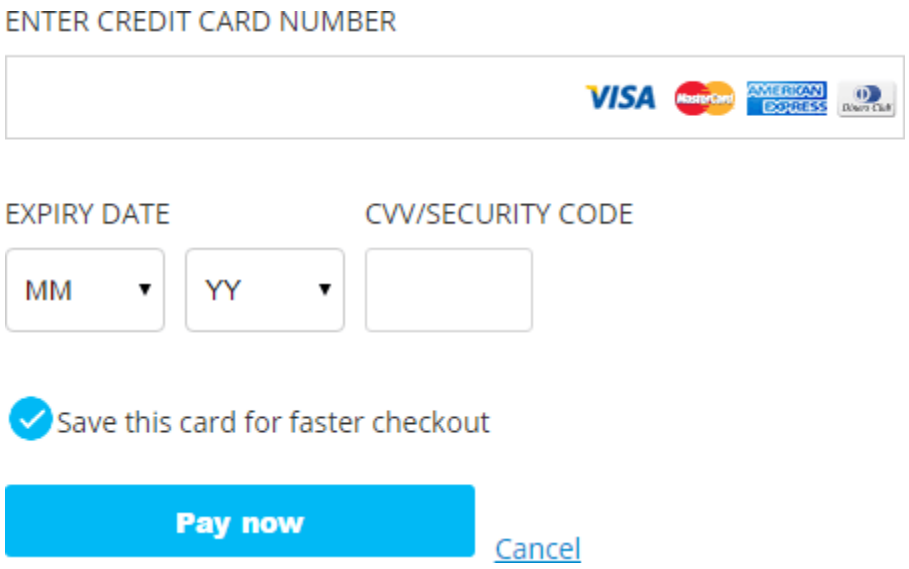
To write the test cases for Usage of text.

PROCEDURE:

Test cases to discuss the different types of text fields, use cases and tools to test the text fieldthen After that we'll check some of the sample test cases for the text field.

PROCEDURE :

Depending on the text field requirement in the form the input needs to be processed accordingly.



1. Name field needs to only accept the alphabet values.
2. Name field should not accept the numeric content.
3. Name field should not accept the symbols.
4. Card Number is a numeric field then it should only accept numbers.
5. Card Number field should not accept characters and symbols in the input field.
6. Forms with Credit Card number field only accept the specific numbers.
7. Card number field should not accept the input more than the field needs.

8. Credit and Debit card number field detects the card type based on the number.
9. Card number field should also detect the debit or credit card type.

1. Zip Code field should accept only the alphanumeric input from the users.
2. Address field should allow characters, numeric and symbol input.
3. Address field could be single or multi-line input box.
4. Address may or may not be a mandatory field.

If the text box is not specified for the input. In such case it can accept numeric, text or symbolsas it's input. The test in such case has to be on the basis of the type of data that needs to be processed. For example Search engine accepts this type of input. However in order to process such content, search engine has input processing system. The tests then have to be made with respect to the content processing specifications.

Text Fields Use Cases

The web form could be login form, contact form, registration form or any other data entry form. In case of document based forms, you may find such fields inside the PDF Forms. Mobile apps also have the forms embeded inside them. In case of apps you may notice the fieldslike username, password, phone number, contact address and name. Mobile apps also have custom text input for social status if used for social media. The SMS app or the messaging app also has the text field with specific limit. In case if you are testing desktop applications, you'll find the text field in apps like Word, Notepad, skype, Gtalk and few other apps. Each desktop app has multiple or single text field within itself.

Tools to Test Text Field

There are different types of tools that can be used to test the text fields. Depending on whatyou're testing, webapp or desktop app, you have to choose the tool accordingly.

Test Data

You can use the text generator tools like – *Dummy text generator* and *Blind text generator*. Most of the web designers also use *lorem ipsum* text generator as a placeholder text. Such dummy textgenerator is a good for test data.

Bug Magnet Chrome Add-on

This add-on is good for exploratory testing. It can be used to figure out the error handling, boundary-related errors, and security issues of your web-app. If your form has bunch of textfields then you should definitely choose this add-on to test.

Browser Developer Tools

Chrome, Firefox and Opera have a developer tools options installed by default. You can use thedefault developer tools to test the forms and other elements on the page. You can also check thefields for the different types of test data.

Test Cases for Text Field

Here are some of the test cases for the text field.

1. How much is the maximum length of text field?
2. How much is the minimum length of text field?
3. How much input is expect in the text field?
4. Does the text field allows input more than the textbox?
5. Does the text field allows input less then the specified textbox?
6. Does the textbox accepts numbers only?
7. Does the textbox accepts decimal numbers?
8. Does the textbox accepts formatted numbers?
9. Does the textfield accepts alphabets?
10. Does the textfield accepts uppercase alphabets?
11. Does the textfield accepts lowercase alphabets?

12. Does the textfield accepts mix of upper and lowercase alphabets?
13. Is there any specific character that field allows?
14. Is there any specific character that field doesn't allow?
15. Does the field accepts HTML characters?
16. Does the text field accepts javascript?
17. Does the text field immune to SQL injection?
18. Does the text field allows copy paste?
19. Does the text field allow drag and drop of text content?
20. Does the cursor appears while typing the content?
21. Does the text field allows spaces?
22. Does the text field processes content with only spaces?
23. Does the text field allows blank input?
24. How does the text field manages trailing spaces?
25. How does the text field manages leading spaces?

These are some of the test cases that you can use for text field. There are many types of textinputs so you have to test according to the specification.

Conclusion:

Thus the test cases on usage of text is created successfully.

EX.NO:14 **TESTING SORTING**

DATE:

AIM: To write the test cases for sorting columns in Ascending order

PROCEDURE:

This basic testcase is designed to verify that the column sorting feature works properly.

Steps/Description

1. Select Bookmarks|Manage Bookmarks.
2. Click on the "Name" column once.
3. Click on the "Name" column a second time.
4. Click on the "Name" column a third time.
5. Click on the "URL" column once, and a second time.
6. Click on the "Custom Keyword" column once, and a second time.
7. Click on the "Description" column once, and a second time.
8. Click on the "Last Visit" column once, and a second time.
9. Click on the "Added On" column once, and a second time.
10. Click on the "Last Modified" column once, and a second time.

Expected ResultsBy step

1. The bookmarks should sort into alphabetical order. (a-b-c-d-e)
2. The bookmarks should sort into reverse-alphabetical order. (e-d-c-b-a)
3. The bookmarks should sort into "natural" order (the way they were originally sorted). (a-d-e-b-c)
4. The bookmarks should sort into order by URL and then the reverse of that order. (d-a-b-c-e)
5. The bookmarks should sort into order by Custom Keyword and then the reverse of thatorder. (c-d-a-b-e)
6. The bookmarks should sort into order by Description and then the reverse of that order.(b-c-d-a-e)

7. The bookmarks should sort into order by Last Visited time and then the reverse of thatorder.
8. The bookmarks should sort into order by Added On date and then the reverse of thatorder.
9. The bookmarks should sort into order by Last Modified date and then the reverse of thatorder.
10. Separators should separate the sort as well, ie step one should result in a-b-c-d-e|aa-bb-cc-dd-ee, not a-aa-b-bb-c-cc-d-dd-e|.

Conclusion:

Thus the test cases on sorting is done successfully.

EX.NO:15 **TESTING LOGIN FORM**

DATE

AIM

To write the test case for creating Login form.

PROCEDURE

Following is the possible list of functional and non-functional test cases for a login page:

Functional Test Cases:

Sr. No.	Functional Test Cases	Type- Negative/ itive Test Case
1	Verify if a user will be able to login with a valid username and valid password.	Positive
2	Verify if a user cannot login with a valid username and an invalid password.	Negative
3	Verify the login page for both, when the field is blank and Submit button is clicked.	Negative
4	Verify the 'Forgot Password' functionality.	Positive
5	Verify the messages for invalid login.	Positive
6	Verify the 'Remember Me' functionality.	Positive
7	Verify if the data in password field is either visible as asterisk or bullet signs.	Positive
8	Verify if a user is able to login with a new password only after he/she has changed the password.	Positive
9	Verify if the login page allows to log in simultaneously with different credentials in a different browser.	Positive

Sr. No.	Functional Test Cases	Type- Negative/ itive Test Case
10	Verify if the 'Enter' key of the keyboard is working correctly on the login page.	Positive
	Other Test Cases	
11	Verify the time taken to log in with a valid username and password.	Performance & Positive Testing
12	Verify if the font, text color, and color coding of the Login page is as per the standard.	UI Testing & Positive Testing
13	Verify if there is a 'Cancel' button available to erase the entered text.	Usability Testing
14	Verify the login page and all its controls in different browsers	Browser Compatibility & Positive Testing.

Sr. No.	Security test cases	Type- Negative/ Positive Test Case
1	Verify if a user cannot enter the characters more than the specified range in each field (Username and Password).	Negative
2	Verify if a user cannot enter the characters more than the specified range in each field (Username and Password).	Positive
3	Verify the login page by pressing 'Back button' of the browser. It should not allow you to enter into the system once you log out.	Negative
4	Verify the timeout functionality of the login session.	Positive

Sr. No.	Functional Test Cases	Type- Negative/ itive Test Case
5	Verify if a user should not be allowed to log in with different credentials from the same browser at the same time.	Negative
6	Verify if a user should be able to login with the same credentials in different browsers at the same time.	Positive
7	Verify the Login page against SQL injection attack.	Negative
8	Verify the implementation of SSL certificate.	Positive

Conclusion:

Thus the test cases on login forms are created successfully.