CS0411- SOFTWARE ENGINEERING LAB

LABORATORY MANUAL

IV YEAR CSE

VII SEMESTER

ACADEMIC YEAR: 2012-2013

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
SRM UNIVERSITY,
SRM NAGAR, KATTANKULATHUR-603203.
KANCHEEPURAM DISTRICT.

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
SRM UNIVERSITY,
SRM NAGAR, KATTANKULATHUR-603203.

SUBJECT : SOFTWARE ENGINEERING LAB**SUBJECT CODE** : CS0411

SEMESTER : VII **CLASS** : IV CSE

HOURS/WEEK : 6 HOURS

EX NO : 1
Date :

LIBRARY MANAGEMENT SYSTEM

1.0 PROBLEM DEFINITION

The library management system is software, which automates the job of a librarian.

1.1 The user can inquire about the availability of a book in which he can search by entering the author’s name or by entering the title of the book.

1.2 The user can borrow a book. He must provide the username and the card number, which is unique and confidential to each user. By confirming the authenticity of a user, the library management system provides information about the number of books already borrowed by the user and by referring to the database whether the user can borrow books or not. The library management system allows the user to enter the title and the author of the book and hence issues the book if it is available.

1.3 By entering the user details and the book details the user can return the borrowed book.

2.0 SYSTEM REQUIREMENT SPECIFICATION

2.1 INTRODUCTION

2.1.1 Purpose

2.1.1.1 The purpose of this SRS is to describe the requirements involved in developing a Library management system.

2.1.1.2 The intended audience is any person, who wants to inquire, borrow and return the books.

2.1.2 Scope

2.1.2.1 The product is titled Library Management System.

2.1.2.2 The product will perform the following tasks

2.1.2.2.1 Enquire about the availability of books.

2.1.2.2.2 Borrow books if available.

2.1.2.2.3 Return the borrowed books.

2.1.3 Definitions, Acronyms and Abbreviations

2.1.3.1 DDBMS – Database Management System.

2.1.4 References

2.1.4.1 IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.

2.1.5 Overview

2.1.5.1 The SRS contains an analysis of the requirements necessary to help easy design.

2.1.5.2 The overall description provides interface requirements for the Library Management System, product perspective, hardware interfaces, software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.

2.1.5.3 Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Library Management System in any fashion.

2.2 THE OVERALL DESCRIPTION

2.2.1 Product perspective

2.2.1.1 Hardware interfaces

2.2.1.1.1 Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system running on high rpm hard disk permitting complete data redundancy and back-up systems to support the primary goal of reliability.

2.2.1.1.2 The system must interface with the standard output devise, keyboard and mouse to interact with this software.

2.2.1.2 Software interfaces

2.2.1.2.1 Back End: MS-Access 2007

2.2.1.2.2 Front End: Microsoft Visual Basic 6.0

2.2.1.3 Memory Constraints

2.2.1.3.1 No specific constraints on memory.

2.2.1.4 Operations

2.2.1.4.1 The software allows three modes of operations

2.2.1.4.1.1 Enquire about the availability and status of books.

2.2.1.4.1.2 By extracting the username and password the software allows the user to borrow a maximum of three books.

2.2.1.4.1.3 By extracting the username and password the software allows the user to return the borrowed books.

2.2.2 Product Functions

2.2.2.1.1 Enquire about the availability and status of books.

2.2.2.1.2 Search the availability of book by entering the title of the book.

2.2.2.1.3 Search the availability of book by entering the author of the book.

2.2.2.1.4 The software validates the authentic user by extracting their user name and password.

2.2.2.1.5 After the validation of the user software allows the user to borrow a maximum of three books based on the number of books which where already borrowed.

2.2.2.1.6 After the validation of the user software allows the user to return the books, which where borrowed.

2.2.3 User characteristics

2.2.3.1 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

2.2.3.2 The Product is absolutely user friendly, so the intended users can be the naïve users.

2.2.3.3 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.

2.2.4 Constraints

2.2.4.1 The user has a unique username and password, there are no options to retrieve a password or username in case it is forgotten or lost hence the user is requited to remember or store the username and password.

2.3 SPECIFIC REQUIREMENTS

2.3.1 Logical Database Requirements

2.3.1.1 The system should contain databases that include all necessary information for the product to function according to the requirements. These include relations such as user details and book details.

2.3.1.2 The user details refer to the information such as name, card number, no. of books borrowed, the title and the name of the author of the books that were borrowed.

2.3.1.3 The book details refer to the information such as the title of the book, author availability status and the number of copies that is available.

2.4 FRONT – END DESCRIPTION

The library management system is automated library system where the user can search for the book by either entering the details of the book or the author's name. By entering the username and the password the software, by checking the number of books that are already borrowed enables us to borrow a maximum of three books. And by entering the username and password (card number), which is unique, the user can return the books.

2.5 BACK – END DESCRIPTION

The library management system consists of two tables. One contains the student details such as the name, card number that is the password, title and the author of the three books, which could be borrowed. The book details consist of the title of the book, number of copies, author and the availability status.

2.6 DATA STRUCTURES

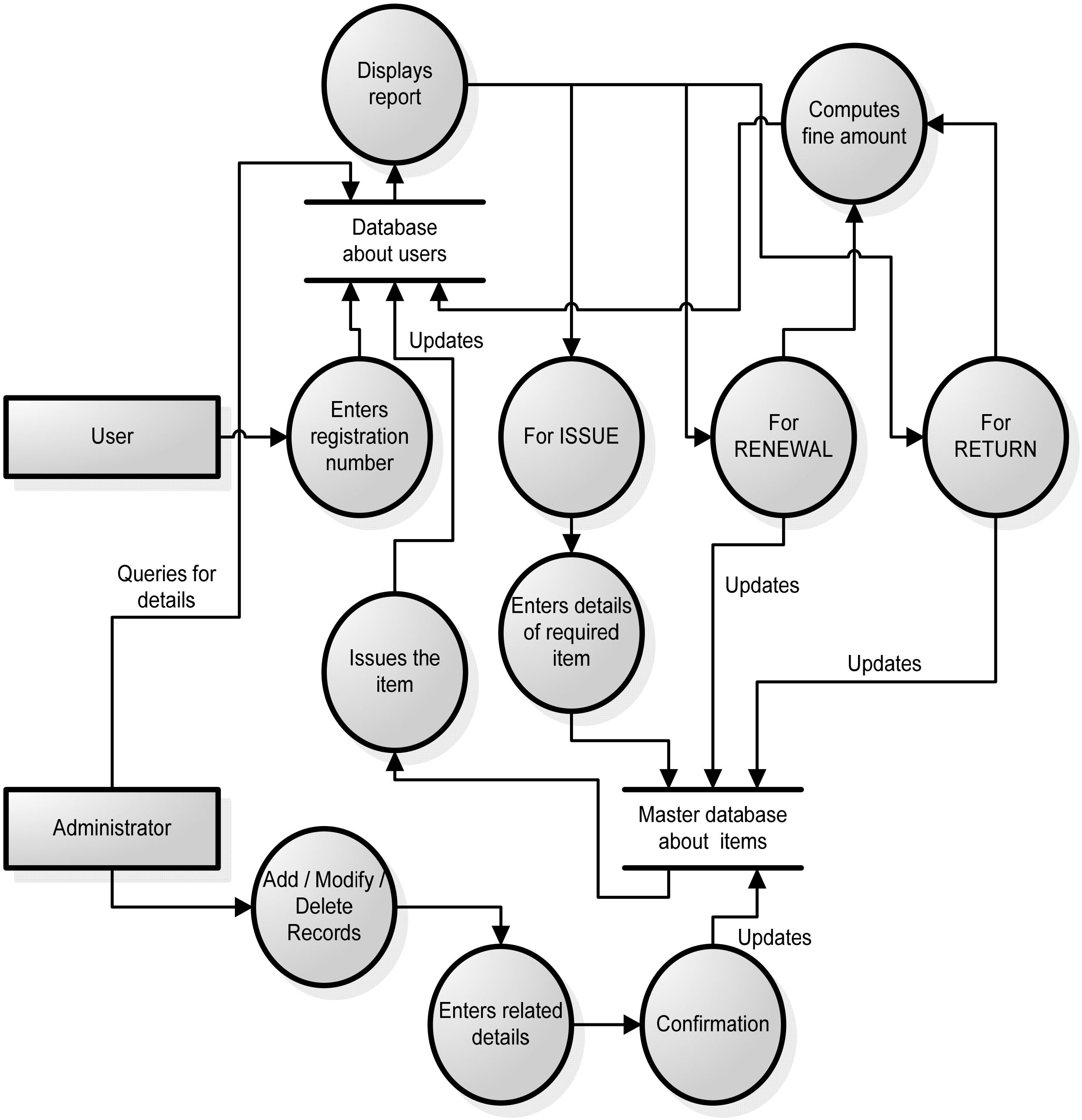
2.6.1. BOOK DETAILS

FIELD NAME	TYPE	CONSTRAINTS
REGISTER_NO	NUMBER	NOT NULL
BOOK_ID	NUMBER	NOT NULL
ISSUE_DATE	DATE/TIME	
RETURN_DATE	DATE/TIME	
BOOK_NAME	TEXT	

2.6.2. STUDENT DETAILS

FIELD NAME	TYPE	CONSTRAINTS
REGISTER_NO	NUMBER	NOT NULL
FNAME	TEXT	NOT NULL
LNAME	TEXT	
GENDER	TEXT	
DEPT	TEXT	
EMAIL	TEXT	
PASSWORD	TEXT	
NO_OF_BOOKS	NUMBER	

2.7 DATA FLOW DIAGRAM

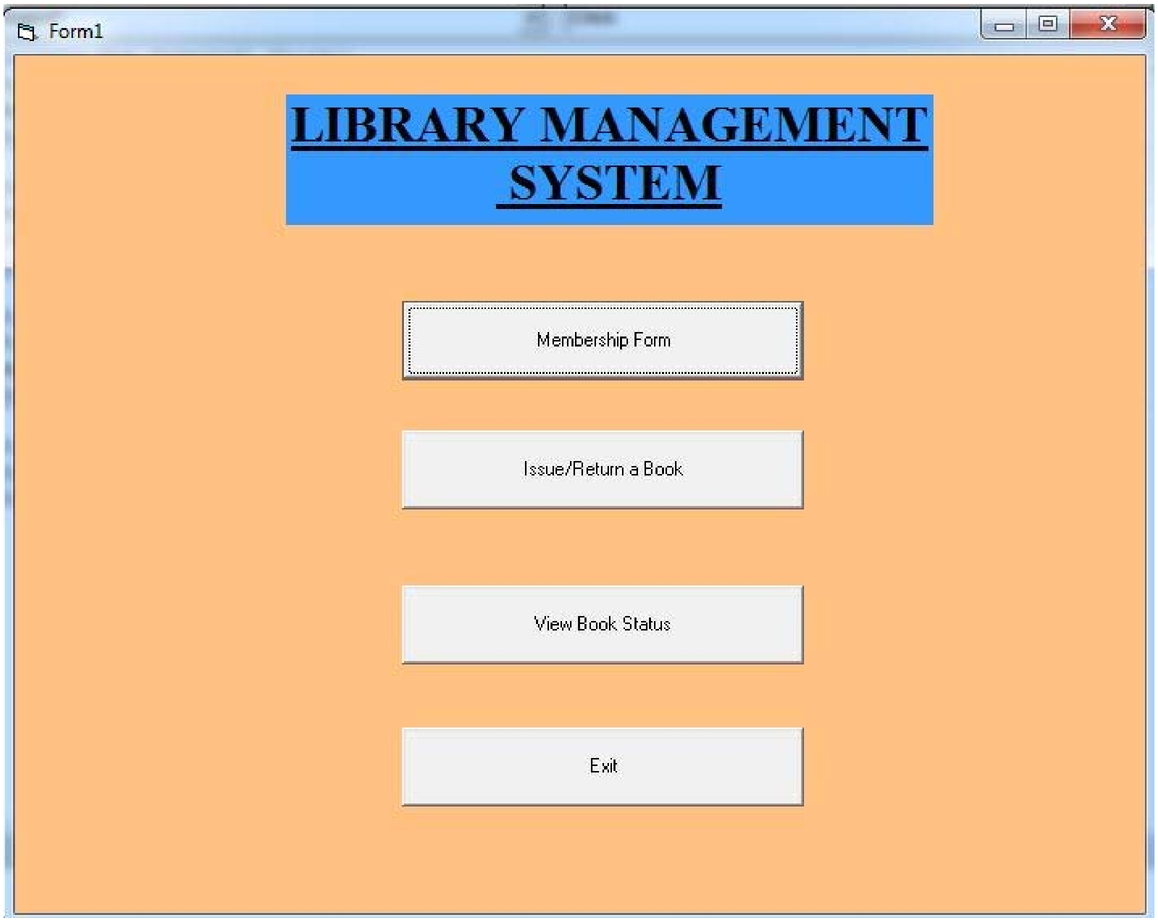


3.0 TESTING:

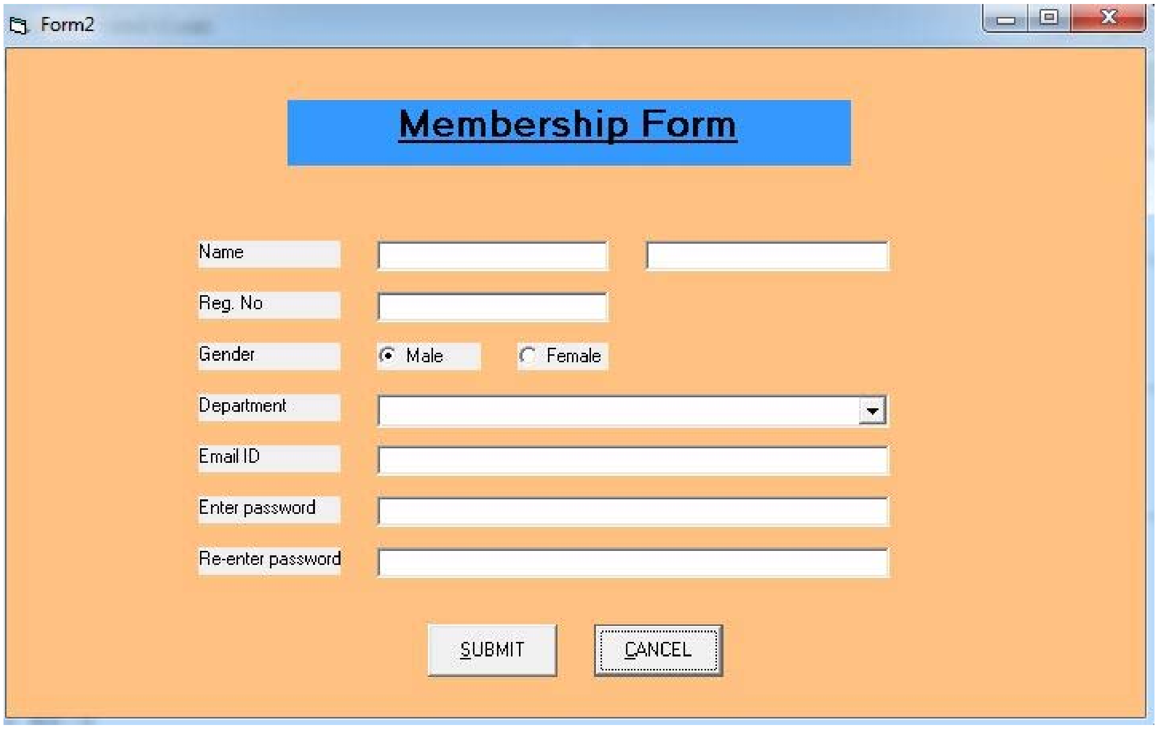
FORM NAME	**INPUT**	**EXPECTED OUTPUT**	**ACTUAL OUTPUT**	**STATUS**
MAIN MENU FORM	Menu Option	Show required Form	Required Form was displayed	Pass
MEMBERSHIP FORM	Member details are entered	Create new member account	New Account was created	Pass
LOGIN FORM	Member ID and password	If password is correct, login.	Member authenticated for future operations.	Pass
ISSUE FORM	Book ID	If books issued is less than three, issue the book.	Book issued	Pass
RETURN/REISSUE FORM	Book ID	Book returned/ reissued	Book returned/ reissued	Pass
BOOK ENQUIRY	Book Name	Book details are displayed	Book details are displayed	Pass

4.0 SAMPLE FORMS

MAIN MENU FORM

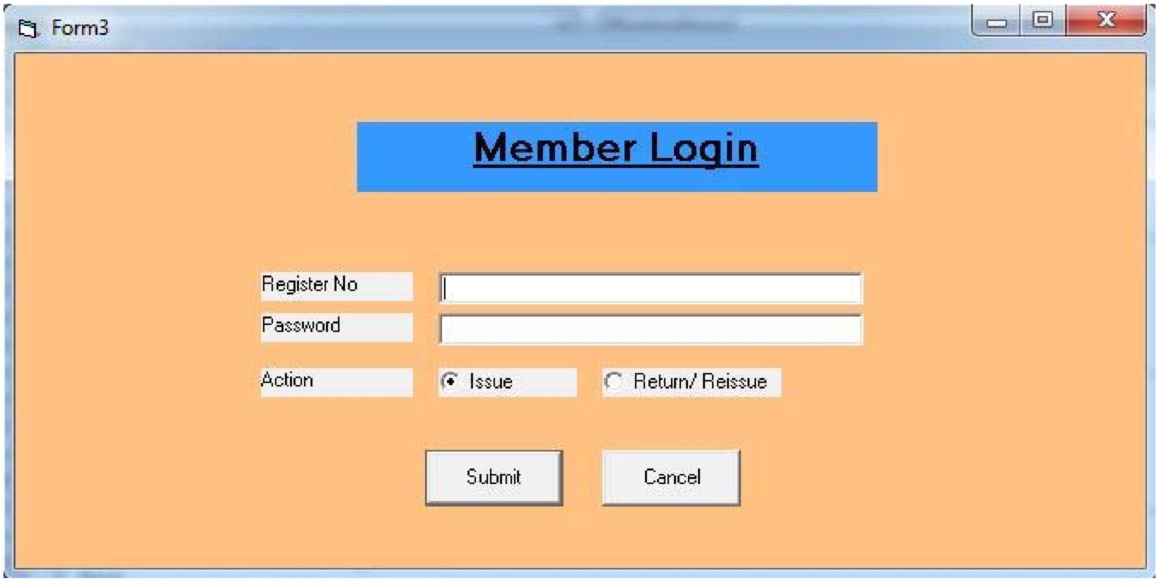


MEMBERSHIP FORM



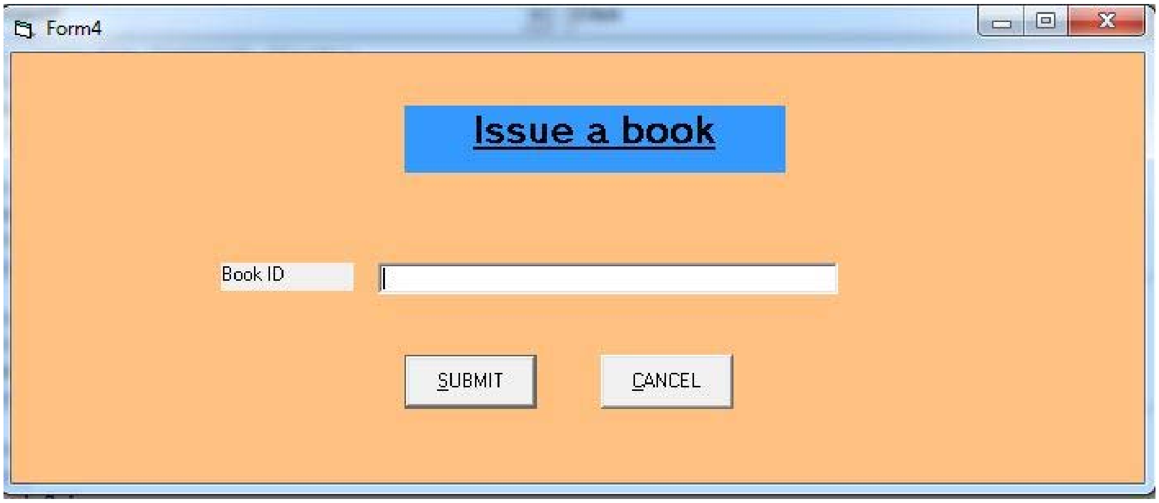
Gender ● Male ○ Female

LOGIN FORM

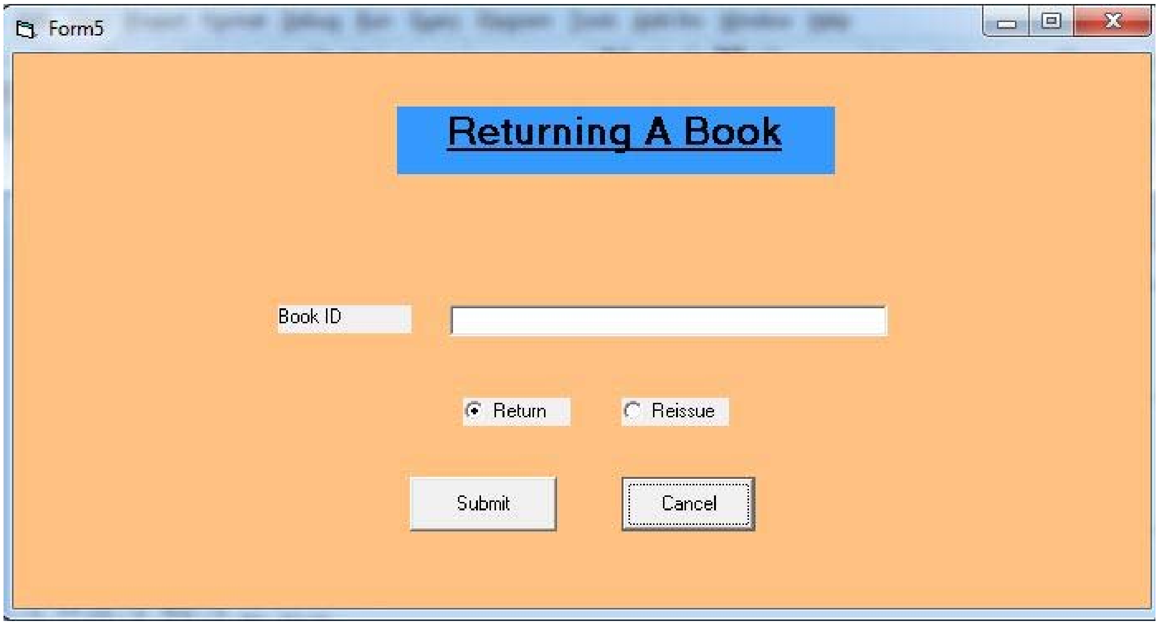


Action ● Issue ○ Return/ Reissue

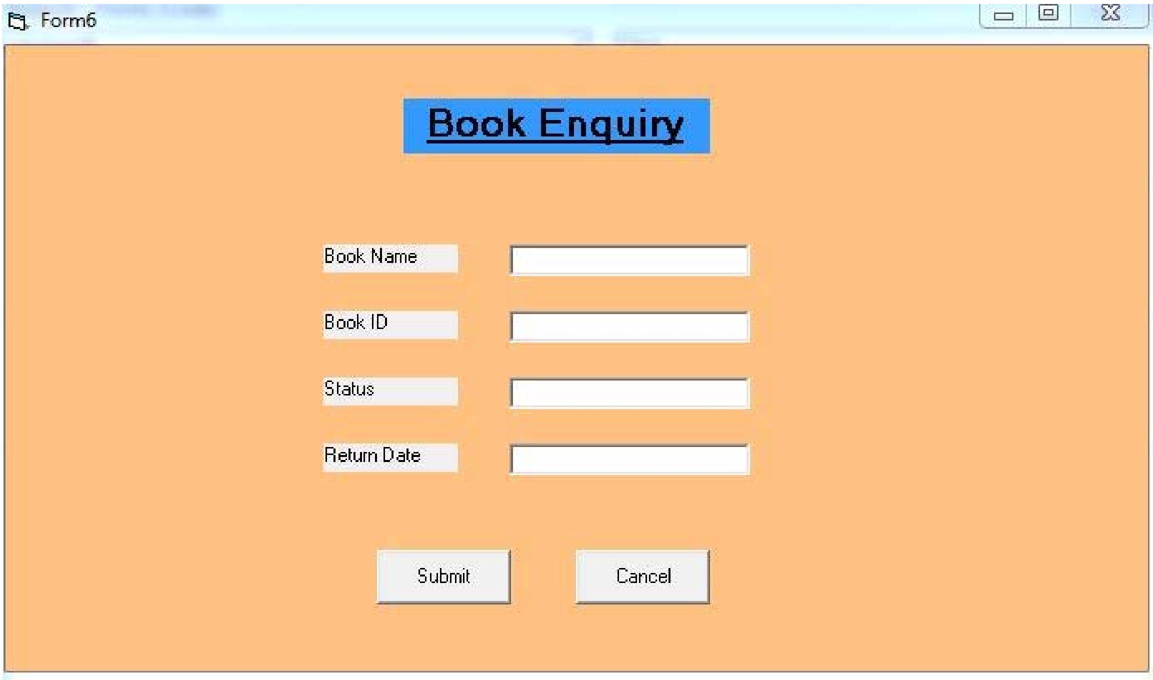
ISSUE FORM



RETURN/REISSUE FORM



BOOK ENQUIRY FORM



4.0 <u>RESULT:</u>

Thus the online Library System was implemented using the specified front end and back end tools.

EX NO : 2
Date :

AUTOMATED BANKING SYSTEM

1.0 PROBLEM DEFINITION

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

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

1.2 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.

1.3 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

2.0 SRS DOCUMENT FOR AUTOMATED BANKING SYSTEM

2.1 INTRODUCTION

2.1.1 Purpose

2.1.1.1 The purpose of this SRS is to describe the requirements involved in developing an Automated Banking System (ABS).

2.1.1.2 The intended audience is any person who wants

2.1.1.2.1 To create account.

2.1.1.2.2 To withdraw or deposit either in fixed deposit or credit account.

2.1.2 Scope

2.1.2.1 The product is titled Automated Banking System (ABS).

2.1.2.2 The product will perform the following tasks

2.1.2.2.1 Allow a new user to create an account, either fixed or credit account by entering the details and by depositing an initial amount.

2.1.2.2.2 Allow the existing user to enter his account details like card number, pin number and account type to view his balance.

2.1.2.2.3 Allow the existing user to deposit an amount by entering the amount to be deposited after the balance had been viewed.

2.1.2.2.4 Allow the existing user to withdraw an amount by entering the amount to be withdrawn after the balance had been viewed.

2.1.2.2.5 The primary benefits expected of the system are: user friendly, continuous connectivity without failure, fault tolerant and involves lesser manpower.

2.1.3 Definitions, Acronyms and Abbreviations

2.1.3.1 ABS: Automated Banking System.

2.1.4 References

2.1.4.1 IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.

2.1.4.2 IEEE Software Requirements Specifications Template http://www.cas.master.ca/~carette/SE3M04/2003/files/ srs_template.doc

2.1.5 Overview

2.1.54.1 The SRS contains an analysis of the requirements necessary to help easy design.

2.1.5.2 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.

2.1.5.3 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.

2.2 THE OVERALL DESCRIPTION

2.2.1 Product perspective

2.2.1.1 Hardware interfaces

2.2.1.1.1 Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system (such as any RDBMS) running on high rpm hard-disk permitting complete data redundancy and backup systems to support the primary goal of reliability.

2.2.1.1.2 The system must interface with the standard output device, keyboard and mouse to interact with this software.

2.2.1.2 Software interfaces

2.2.1.2.1 Back End: MS Access 2007

2.2.1.2.2 Front End: Microsoft Visual Basic 6.0

2.2.1.3 Operations

2.2.1.3.1 The user can create a new account.

2.2.1.3.2 The existing user can access his account and view his balance by entering his details.

2.2.1.3.2 The user can deposit and withdraw money from his account.

2.2.2 Product Functions

2.2.2.1 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:

2.2.2.1.1 Customer Name.

2.2.2.1.2 Customer address.
2.2.2.1.3 Required account type.
2.2.2.1.4 Pin Number.
2.2.2.1.5 Initial deposit.

2.2.2.2 Operating with created account

The user should be able to operate with his new account after:
2.2.2.2.1 Entering card number.
2.2.2.2.2 Entering pin number.
2.2.2.2.3 Entering the account type, transaction type and amount involved in the transaction.

2.2.3 User characteristics

2.2.3.1 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

2.2.3.2 The Product is absolutely user friendly, so the intended users can be the naïve users.

2.2.3.3 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.

2.2.4 Constraints:

2.2.4.1 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.

2.2.4.2 At the time of creating the new account, the initial deposit should not be less than the specified amount.

2.3 SPECIFIC REQUIREMENTS

2.3.1 Logical Database Requirements

2.3.1.1 The system should contain databases that include all the necessary information for the product to function according to the requirements. These include relations such as Customer Details and Account Details.

2.3.1.2 Customer details refer to the customer's name and address. Account details of the customer include the card number, account type, transaction type and the pin number given by the user to be used at the time of the transaction at the bank.

2.4 FRONT – END DESCRIPTION

The front end for the Automated Banking System (ABS) is designed using Microsoft Visual Basic 6.0. The front end contains a user-friendly interface. The first form contains a welcome screen that provides an option for the user to either crate a new account or to operate through an existing account. The "create account" module contains a provision to create a new account after collecting the customer name, address and other details. The card number and pin number of the user is obtained every time there is a transaction. The user is requested to select the required type of transaction and the amount involved in the transaction.

2.5 BACK – END DESCRIPTION

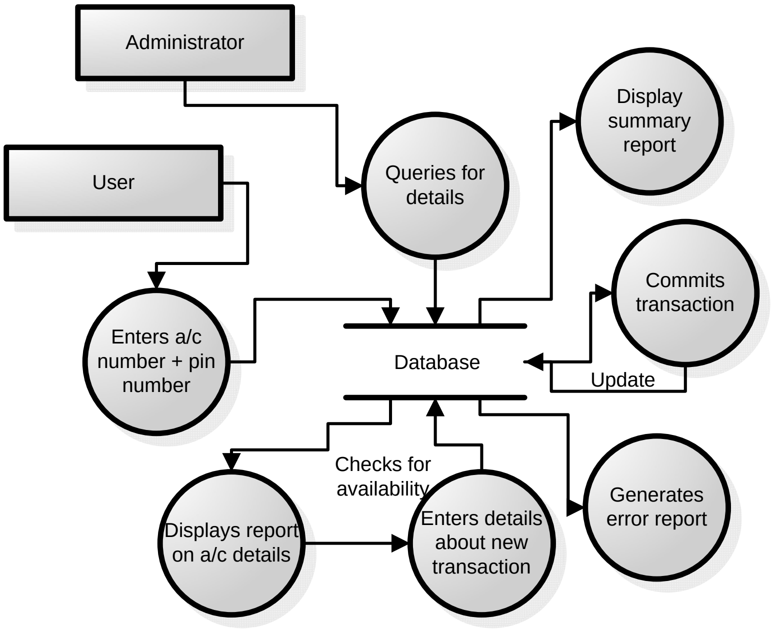
The Automated Banking System (ABS) database contains only one table. It correlates a unique card number, customer name, account type, pin number and the balance.

2.6 DATA STRUCTURES

2.6.1. ACCOUNT DETAILS

FIELD NAME	TYPE	CONSTRAINTS
NAME	TEXT	
ACC_NO	AUTONUMBER	NOT NULL
AGE	NUMBER	
GENDER	TEXT	
EMAIL	TEXT	
PHONE_NO	NUMBER	
PASSWORD	TEXT	
BALANCE	NUMBER	

2.7 DATA FLOW DIAGRAM:

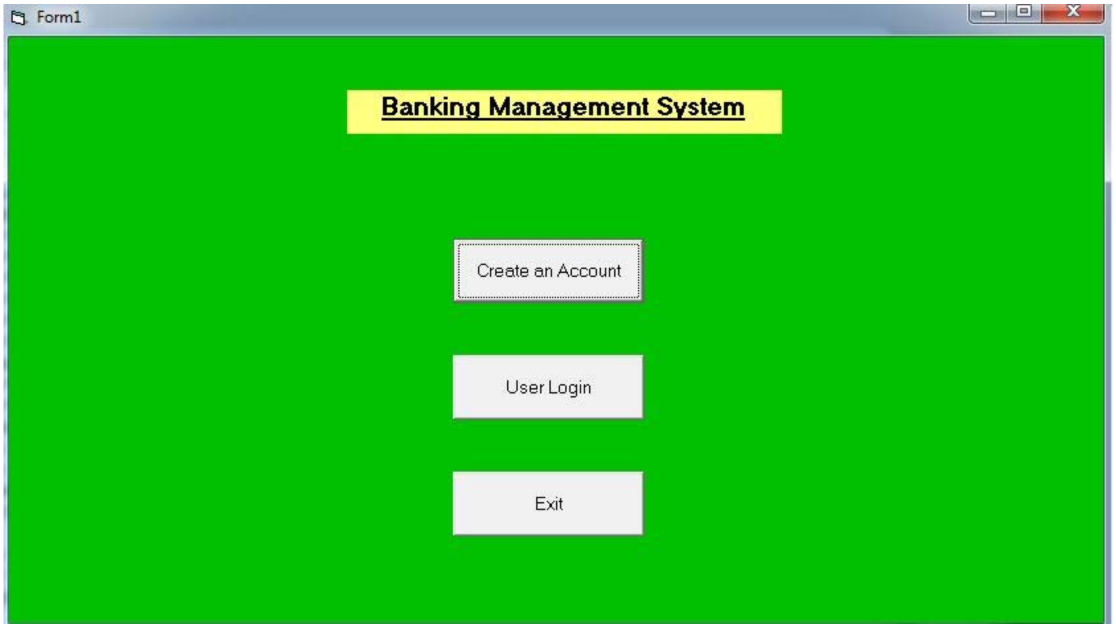


3.0 TESTING

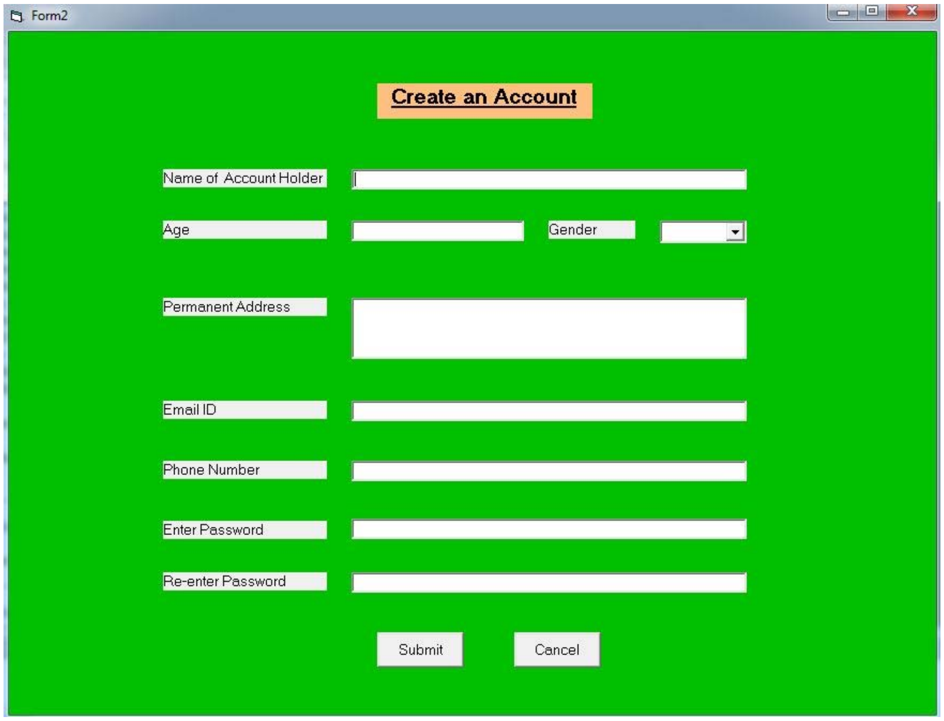
FORM NAME	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
MAIN MENU FORM	Menu Option	Required Form must be displayed	Required form was displayed	Pass
ACCOUNT CREATION FORM	User Details are entered	Create a new account	New account was created	Pass
LOGIN FORM	Account ID and password are entered	If passwords match, login	User permitted to perform further operations	Pass
WELCOME FORM	User can select deposit, withdraw or view balance	Show Deposit form or Withdraw form or view balance	Deposit form or withdraw form or balance is displayed	Pass
DEPOSIT FORM	Amount to be deposited is entered	Deposit amount	Amount was deposited	Pass
WITH DRAW FORM	Amount to be withdrawn is entered	Withdraw amount	Amount was withdrawn	Pass

4.0 SAMPLE FORMS

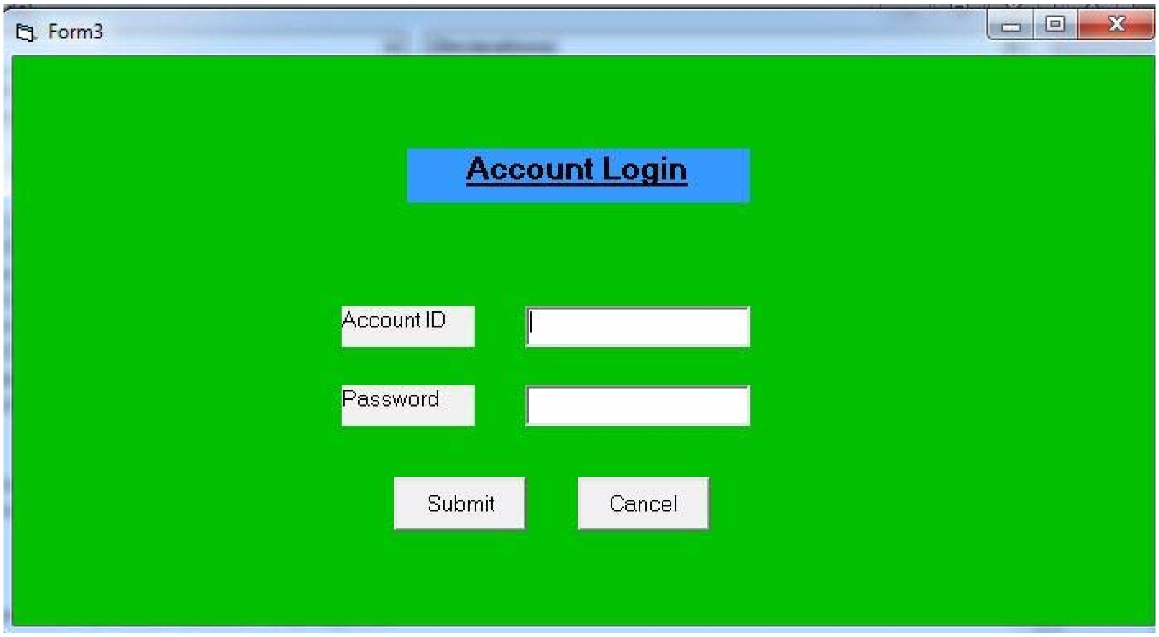
MAIN MENU FORM



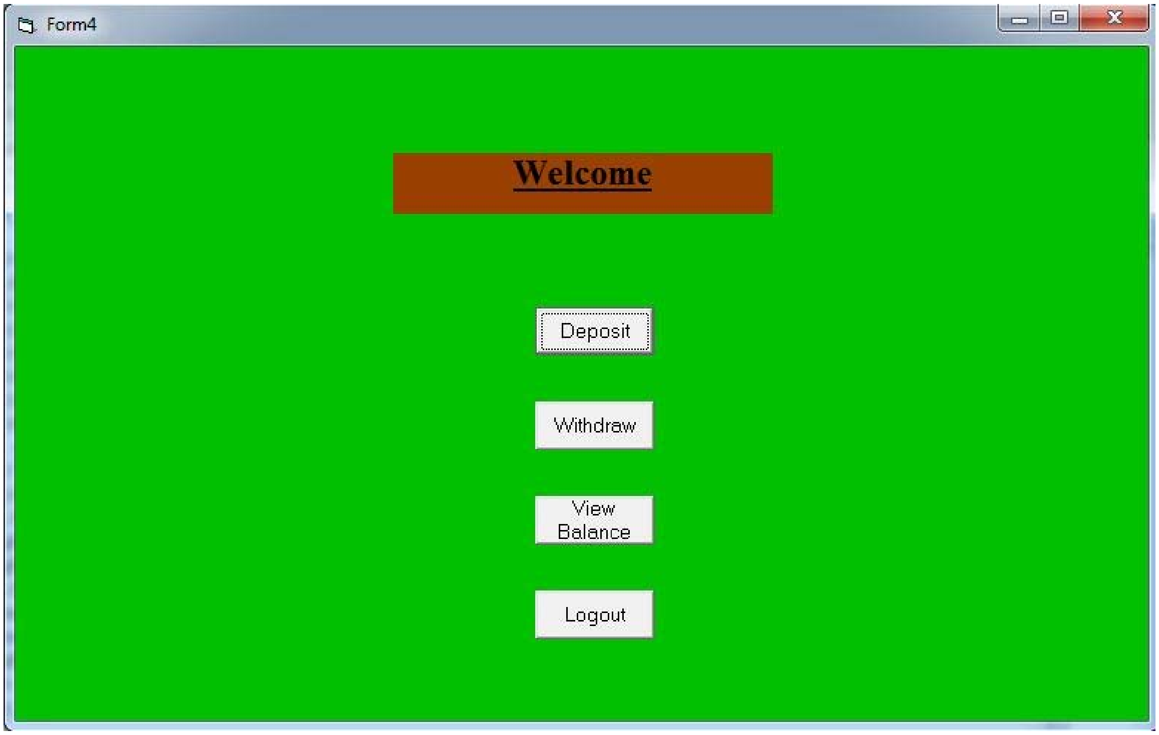
ACCOUNT CREATION FORM



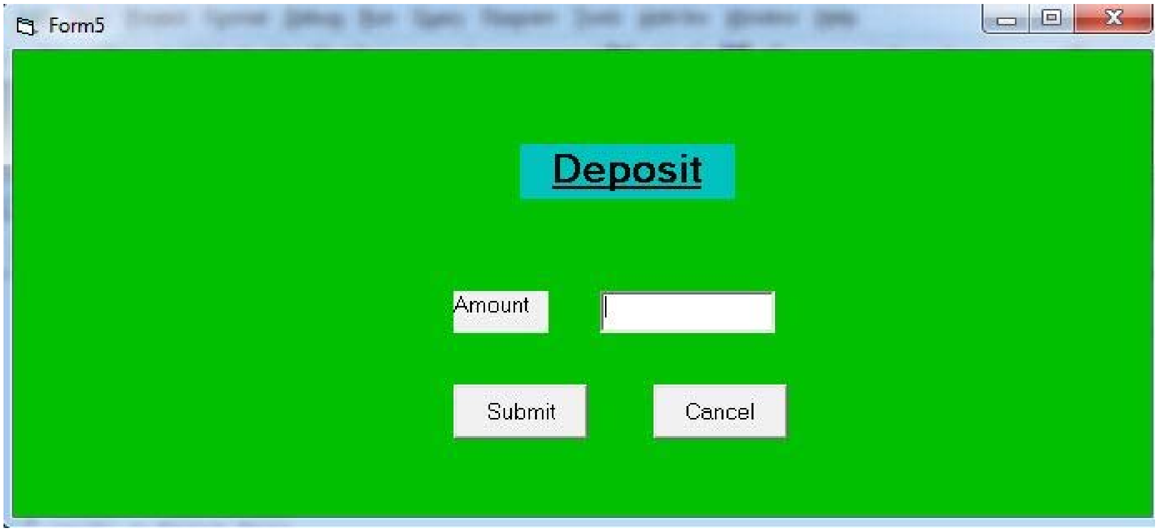
LOGIN FORM



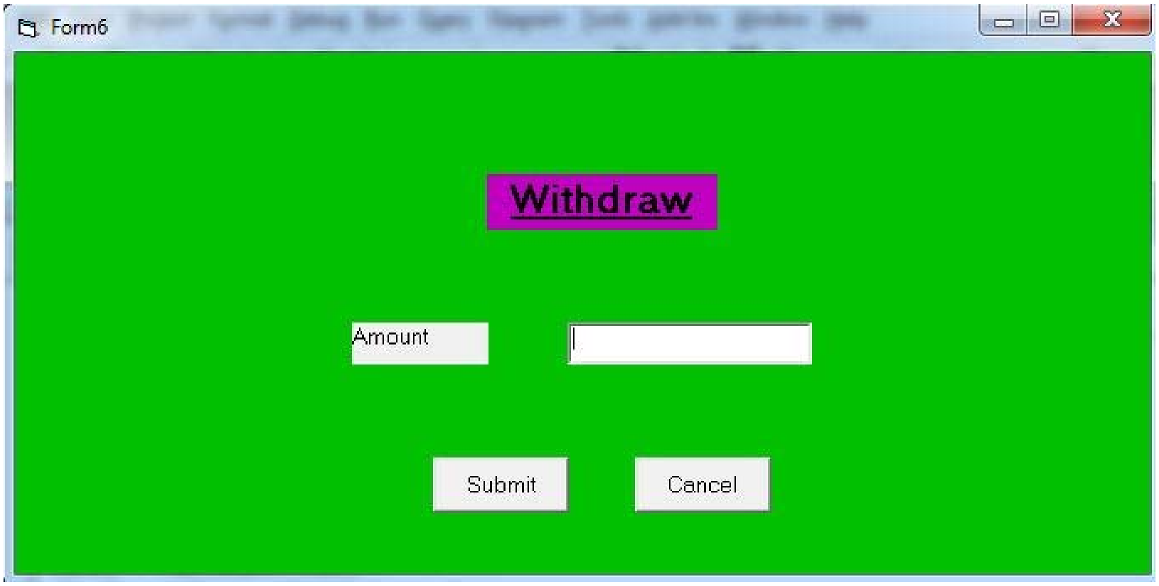
WELCOME FORM



DEPOSIT FORM



WITHDRAW FORM



5.0 RESULT

Thus the requirements involved in developing an Automated Banking System was completed successfully

EX NO : 3
Date :

AIRLINE RESERVATION SYSTEM

1.0 PROBLEM DEFINITION

Ticket reservation system for airlines has to be developed.

The system developed should contain the following features

1. The system should contain the following features
2. Search for information about the flight by means of flight number and destination
3. While displaying information about the flight it has to provide availability of seats.
4. While reserving tickets the system obtain following information from the user

Passenger Name, Sex, Age, Address.

Credit Card Number, Bank Name.

Flight number, Flight name, Date of Journey and number of tickets to be booked.

5. Based on the availability of tickets, the ticket has to be issued. The ticket issued should contain the following information –ticket number, flight no, flight name, date of journey, number of passengers, sex, age and departure time.
6. Cancellation of booked tickets should be available.

2.0 SRS DOCUMENT FOR AIRLINE RESERVATION SYSTEM

2.1 INTRODUCTION

2.1.1 Purpose

2.1.1.1 The purpose of this SRS is to describe the requirements involved in developing a Airline Reservation system (ARS).

2.1.1.2 The intended audience is any person who wants to reserve or cancel tickets or to check the availability of Airline tickets

2.1.2 Scope

2.1.2.1 The product is titled Airline Reservation system (ARS).

2.1.2.2 The product will perform the following tasks

2.1.2.2.1 The software that is being developed can be used to check the availability of the flight tickets for the specified flight, destination and date of journey

2.1.2.2.2 If the tickets are available to the users needs and specification, then the software provide a facility to book the tickets.

2.1.2.2.3 If the passengers wants to cancel the tickets, he can use the cancellation module of the Airline Reservation System.

2.1.3 Definitions, Acronyms and Abbreviations

2.1.3.1 ARS: Airline Reservation System.

2.1.4 References

2.1.4.1 IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.

2.1.5 Overview

2.1.5.1 The SRS contains an analysis of the requirements necessary to help easy design.

2.1.5.2 The overall description provides interface requirements for the Airline Reservation system, product perspective, hardware interfaces software interfaces,, communication interface, memory constraints, product functions, user characteristics and other constraints.

2.1.5.3 Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Airline Reservation system in any fashion.

2.2 THE OVERALL DESCRIPTION

2.2.1 Product perspective

2.2.1.1 Hardware interfaces

2.2.1.1.1 Hard disk: The database connectivity requires a hardware configuration with a fast database system running on high rpm hard-disk permitting complete data redundancy and back-up systems to support the primary goal of reliability.

2.2.1.1.2 The system must interface with the standard output device, keyboard and mouse to interact with this software.

2.2.1.2 Software interfaces

2.2.1.2.1 Back End: MS Access 2007

2.2.1.2.2 Front End: Microsoft Visual Basic 6.0

2.2.1.3 Operations

2.2.1.3.1 The user mode enables the end-users to do the end user operations like checking the availability, reserving and canceling of flight tickets.

2.2.2 Product Functions

2.2.2.1 Viewing Flight Details

The user must have the access up-to-date information about the flights including

2.2.2.1.1Flight number

2.2.2.1.2 Flight Name

2.2.2.1.3 Flight route(Start and Destination stations)

2.2.2.1.4 Flight timings

2.2.2.1.5 Seat availability.

2.2.2.2 Reserving Tickets

The user must be able to reserve tickets after selecting

2.2.2.2.1Flight number

2.2.2.2.2 Flight Route

2.2.2.3 Canceling Tickets

The user must be able to cancel tickets that he has earlier reserved by quoting the ticket number, credit card number and bank name.

2.2.3 User characteristics

2.2.3.1 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

2.2.3.2 The Product is absolutely user friendly, so the intended users can be the naïve users.

2.2.3.3 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.

2.2.4 Constraints

2.2.4.1 At the time of reservation, each user is provided a unique ticket number that must be used for further operation like cancellation. Hence the user is required to remember or store this number carefully.

2.3 SPECIFIC REQUIREMENTS

2.3.1 Logical Database Requirements

2.3.1.1 The system should contain databases that include all necessary information for the product to function according to the requirements. These include relations such as flight details, reservation details, and cancellation details.

2.3.1.2 The user details refer to the information such as flight number and name, start and destination stations, seat availability.

2.3.1.3 Reservation details refer to personal information that is obtained from the user

2.3.1.4 At the time of reservation, the passenger is provided a unique ticket no that could be used at the time of cancellation.

2.3.1.5 While displaying any information about the flight it has to provide the following information

Flight no and name

Availability of seats for the particular flight

The flight timing

The passenger personal details should be obtained for reserving the tickets.

2.4 FRONT – END DESCRIPTION

The front-end for the Airline Reservation system (ARS) is designed using Microsoft Visual Basic 6.0. The front-end contains a user- friendly interface. The first form contains a welcome screen that provides an option for the user to select one of the following

Enquiry

Reservation

Booking details

Cancellation

In the Enquiry form the user can get details of the flight by means of either flight name destination or date 0of journey. In the reservation form, there can book details by entering the personal details. The ticket is displayed with details about the flight name and number, number of passengers, ticket number, sex and age. The cancellation form helps the user to cancel a ticket, which he had booked earlier.

2.5 BACK – END DESCRIPTION

The Airline Reservation system consists of two tables. One contains the flight details such as the flight name, flight number, destination, date of journey and seats available in each class that is referred to during enquiry. The other table has the passenger details such as name, age, sex, credit card number, and bank name. This table is referred to at the time of reservation or cancellation.

2.6 DATA STRUCTURES

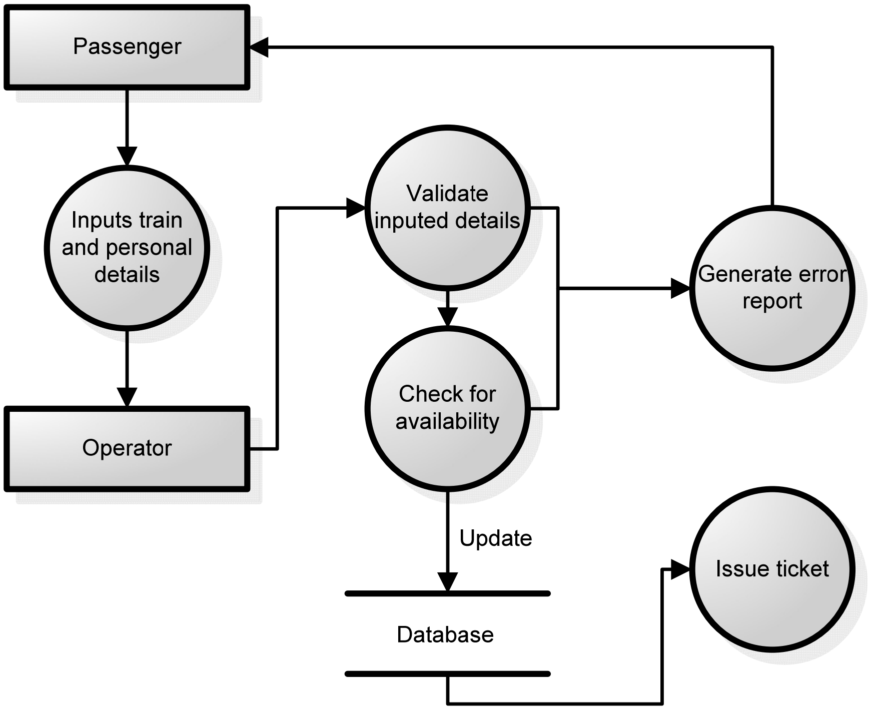
2.6.1. FLIGHT DETAILS

FIELD NAME	TYPE	CONSTRAINTS
ROUTE_NAME	TEXT	NOT NULL
FLIGHT_NO	NUMBER	NOT NULL
SEATS_AVAIL	NUMBER	
JOURNEY_DATE	DATE/TIME	
DEP_TIME	DATE/TIME	
ARR_TIME	DATE/TIME	
COST	NUMBER	

2.6.2. PASSENGER DETAILS

FIELD NAME	TYPE	CONSTRAINTS
TICKET_NO	AUTONUMBER	NOT NULL
NAME	TEXT	NOT NULL
GENDER	TEXT	
ADDRESS	TEXT	
CC_NO	NUMBER	NOT NULL
BANK_NAME	TEXT	
NO_OF_TICKETS	NUMBER	

2.7 DATA FLOW DIAGRAM

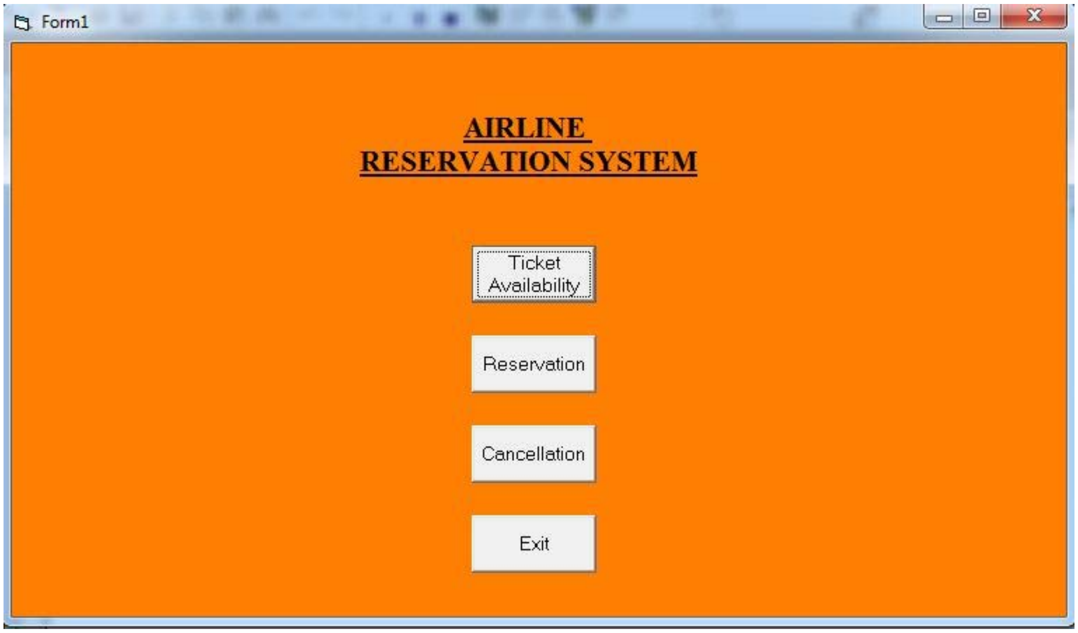


3.0 TESTING:

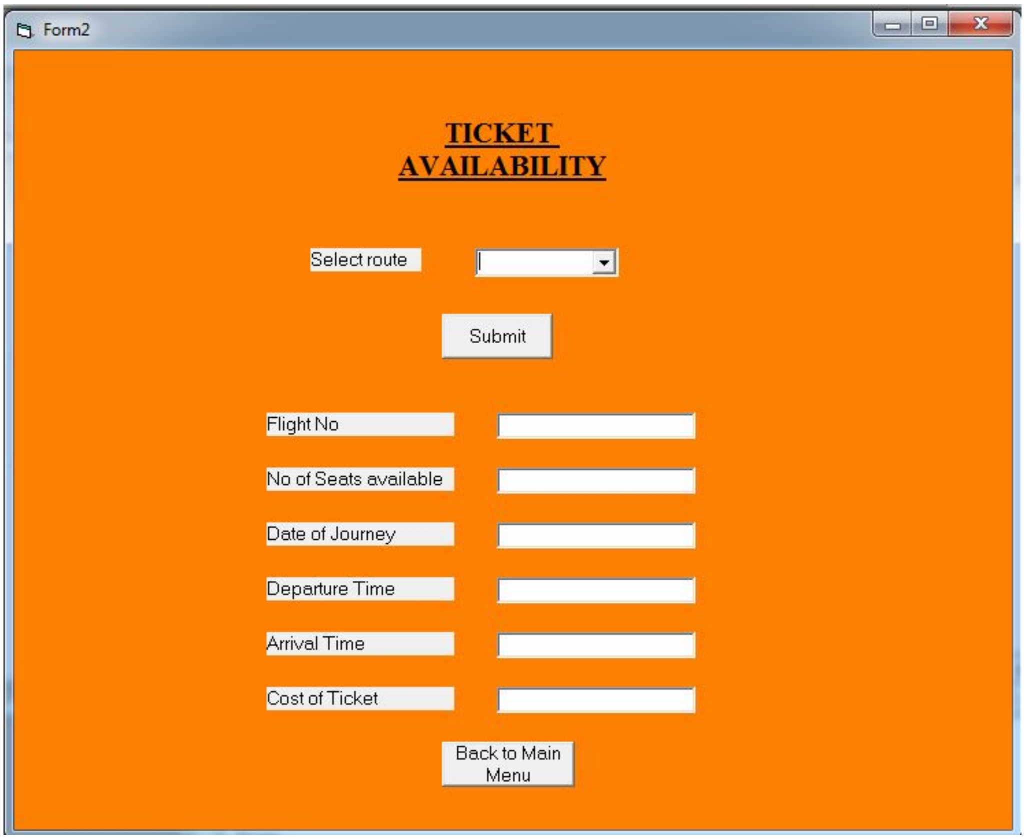
FORM NAME	**INPUT**	**EXPECTED OUTPUT**	**ACTUAL OUTPUT**	**STATUS**
MAIN MENU FORM	Menu Option	Required Form must be displayed	Required form was displayed.	Pass
TICKET AVAILABILITY FORM	Flight route or Flight name	Flight seats availability must be displayed	Flight seats availability are displayed.	Pass
RESERVATION FORM	Personal details were entered.	Ticket must be booked and database updated.	Ticket was booked and database was updated.	Pass
CANCELLATION FORM	Ticket number was entered.	Ticket must be cancelled and database must be updated.	Ticket was cancelled and database was updated..	Pass

4.0 SAMPLE FORMS

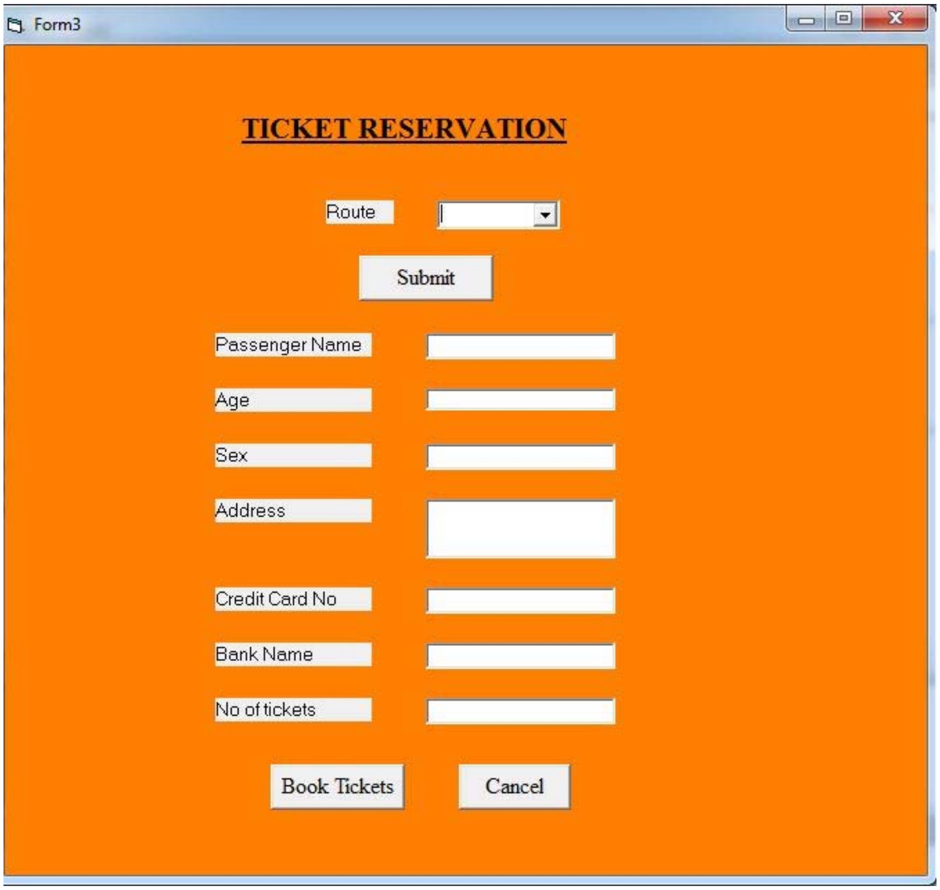
MAIN MENU FORM



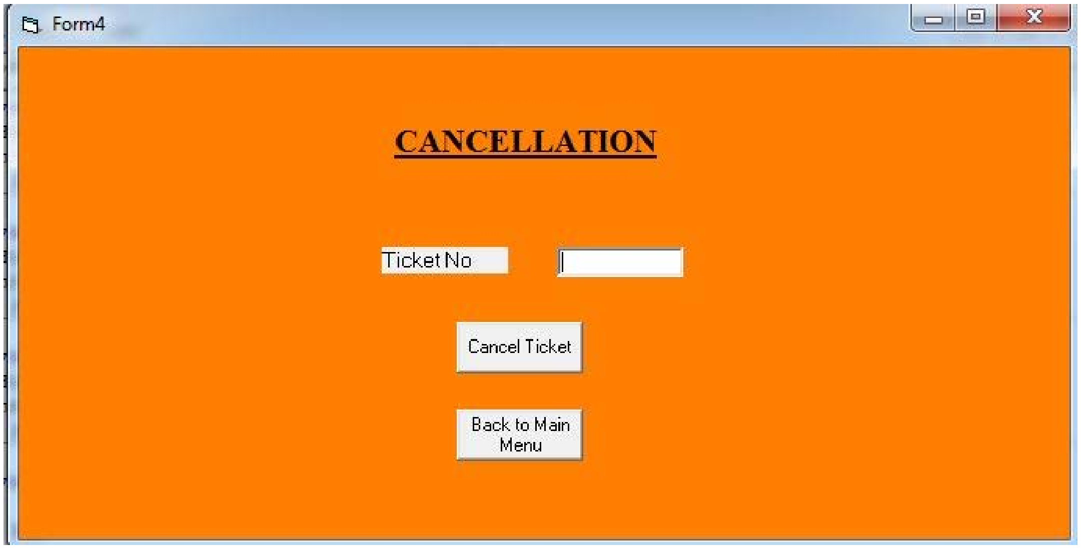
TICKET AVAILABILITY FORM



RESERVATION FORM



CANCELLATION FORM



5.0 RESULT

Thus the online Airline Reservations System was implemented using the specified front end and back end tools.

EX NO : 4
Date :

EMPLOYEE MANAGEMENT APPLICATION

1.0 PROBLEM DEFINITION

A payroll application is to be developed which is required to perform the following functions:

1.1 It must provide a user in employee mode with the details of an employee, which includes his name, department, date of joining and salary.

1.2 It must validate an user to enter in administrator mode using password. It must provide a user to enter in administrator mode to view or modify an employee's details using his employee ID. It must also allow the user to add a new employee and delete records of an existing employee.

2.0 SRS DOCUMENT EMPLOYEE MANAGEMENT APPLICATION

2.1 INTRODUCTION

2.1.1 Purpose

2.1.1.1 The purpose of this SRS is to describe the requirements involved in developing a system to manage employee records.

2.1.1.2 The intended audience is any person who wants

2.1.1.2.1 To check employee details (both employee and administrator mode).

2.1.1.2.2 To add new employee, modify any employee's details or delete records for the employee (only administer mode).

2.1.2 Scope

2.1.2.1 The product is titled Employee Management Application (EMA).

2.1.2.2 The product will perform the following tasks

2.1.2.2.1 Allow either an employee or an administrator to view employee details.

2.1.2.2.2 Allow the administrator to add a new employee with corresponding details.

2.1.2.2.3 Allow the administrator to modify the detail of an employee.

2.1.2.2.4 Allow the administrator to delete the records for an employee.

2.1.3 Definitions, Acronyms and Abbreviations

2.1.3.1 EMA: Employee Management Application.

2.1.4 References

2.1.4.1 IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.

2.1.5 Overview

2.1.5.1 The SRS contains an analysis of the requirements necessary to help easy design.

2.1.5.2 The overall description provides interface requirements for the Employee Management System, product perspective, hardware interfaces, software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.

2.1.5.3 Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Employee Management Application in any fashion.

2.2 THE OVERALL DESCRIPTION

2.2.1 Product perspective

2.2.1.1 Hardware interfaces

2.2.1.1.1 Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system (such as any RDBMS) running on high rpm hard-disk permitting complete data redundancy and back-up systems to support the primary goal of reliability.

2.2.1.1.2 The system must interface with the standard output device, keyboard and mouse to interact with this software.

2.2.1.2 Software interfaces

2.2.1.2.1 Back End: MS Access 2007

2.2.1.2.2 Front End: Microsoft Visual Basic 6.0

2.2.1.3 Memory Constraints

2.2.1.3.1 No specific constraints on memory.

2.2.1.4 Operations

2.2.1.4.1 The software allows two modes of operations

2.2.1.4.1.1 The administrator mode allows user to add a new employee, modify the existing details of an employee, view the details of an employee and also delete records for an existing employee.

2.2.2 Product Functions

2.2.2.1 Viewing the employee details

The user (both administrator and employee) must have the access to Up-to-date information about the employee including

2.2.2.1.1 Employee Id
2.2.2.1.2 Employee Name
2.2.2.1.3 Employee Department
2.2.2.1.4 Date of Joining
2.2.2.1.5 Salary

2.2.2.2 Adding a new employee

The user (only in administrator mode) must be able to add a new employee by supplying the following employee details.

2.2.2.2.1 Employee Name
2.2.2.2.2 Employee Department
2.2.2.2.3 Date of Joining
2.2.2.2.4 Salary

2.2.2.3 Modifying the details of an employee

The user (only in administrator mode) must be able to modify the following details of an existing employee.

2.2.2.3.1 Employee Name
2.2.2.3.2 Employee Department
2.2.2.3.3 Date of Joining
2.2.2.3.4 Salary

2.2.3 User characteristics

2.2.3.1 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

2.2.3.2 The Product is absolutely user friendly, so the intended users can be the naïve users.

2.2.3.3 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.

2.2.4 Constraints

2.2.4.1 At the time of adding a new employee, each employee must be assigned a unique ID number.

2.3 SPECIFIC REQUIREMENTS

2.3.1 Logical Database Requirements

2.3.1.1 There is only one database which contains all the necessary information about an employee which includes employee ID, employee name, department, date if joining and salary.

2.4 FRONT – END DESCRIPTION

The front end for the Employee Management Application (EMA) is designed using Microsoft Visual Basic 6.0. The front – end contains a user – friendly interface. It has a welcome screen that provides an option for the user to enter in employee mode or in administrator mode. In employee mode the user can enter the employee ID of the employee and view his details. The user has to validate him self using password to enter in administrator mode. In administrator mode, apart form viewing the details the user can also add a new employee by providing details or modify the existing details using the employee ID.

2.5 BACK – END DESCRIPTION

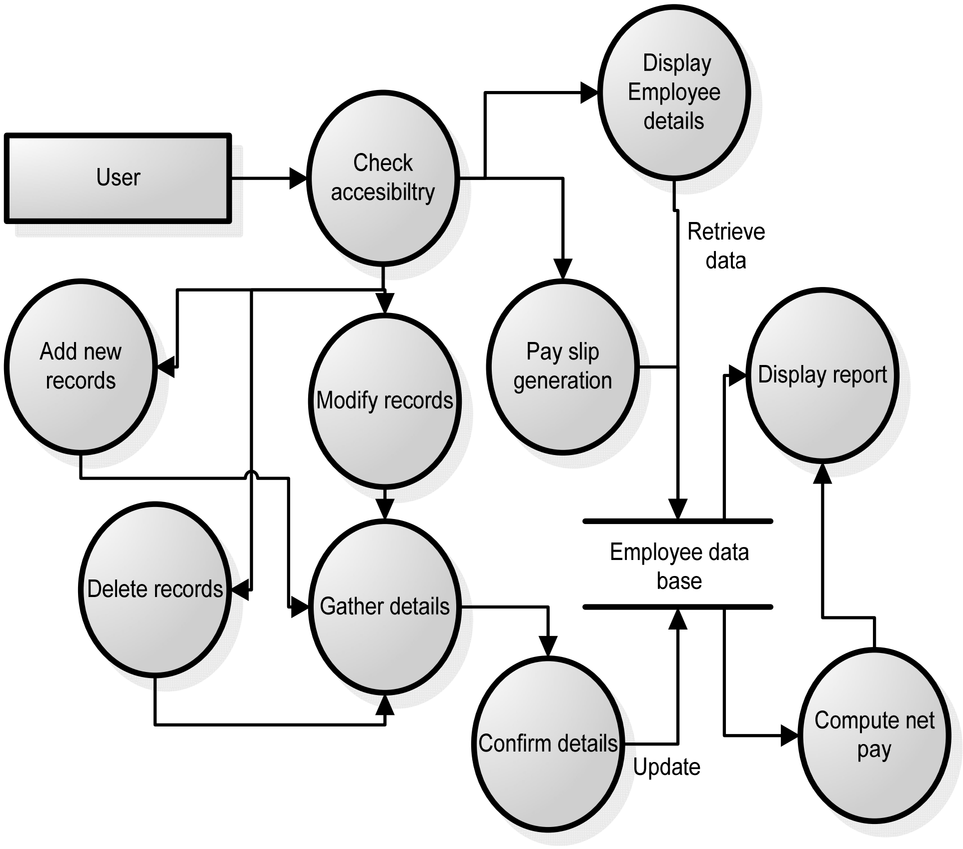
There is only one table. It correlates a unique employee ID with his name, department, date of joining and salary.

2.6 DATA STRUCTURES

2.6.1. EMPLOYEE DETAILS

FIELD NAME	TYPE	CONSTRAINTS
ID	AUTONUMBER	NOT NULL
NAME	TEXT	
DEPT	TEXT	
JOIN_DATE	DATE/TIME	
SALARY	NUMBER	

2.7 DATA FLOW DIAGRAM

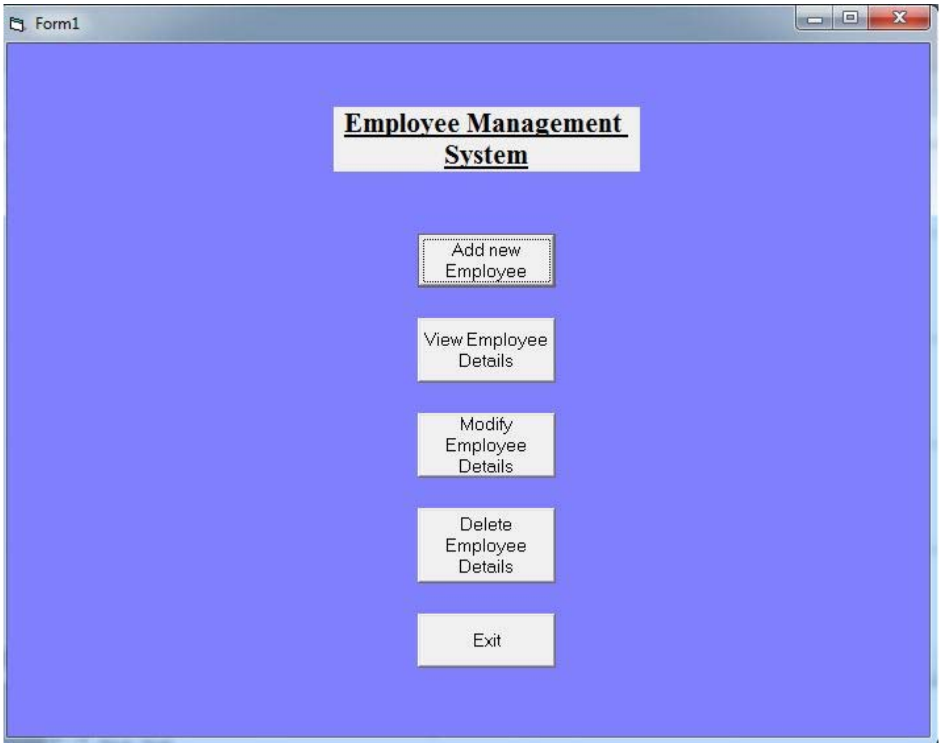


3.0   TESTING

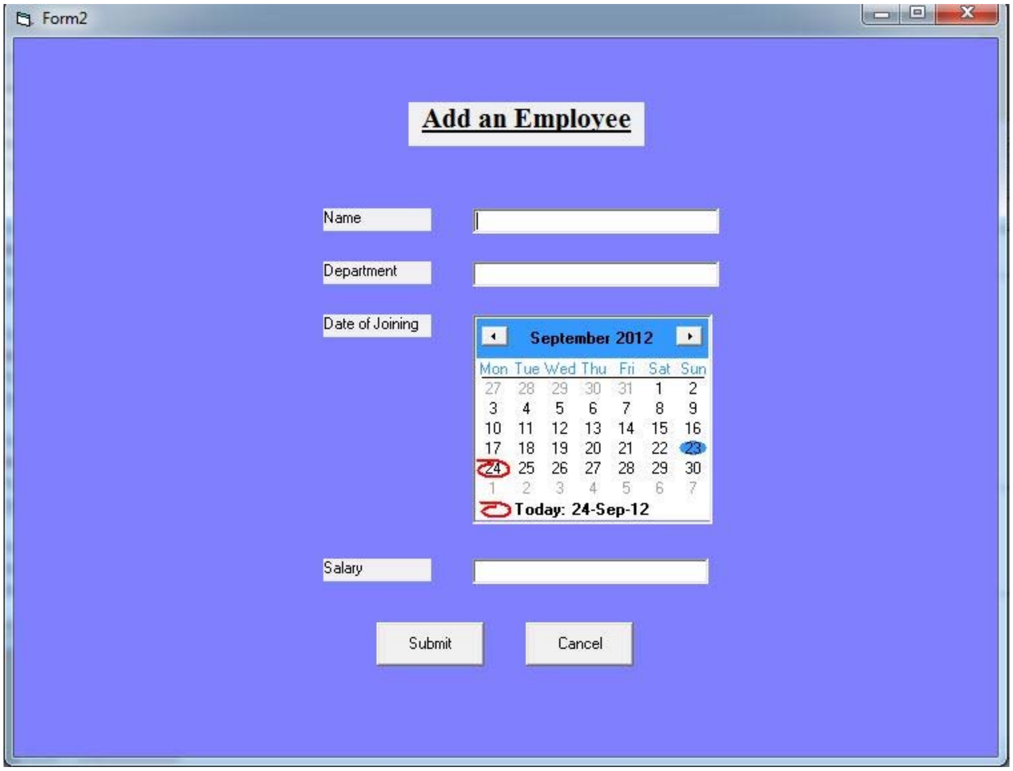
FORM NAME	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
MAIN MENU	Menu Option	Required Form must be displayed	Required Form was displayed	Pass
ADD NEW EMPLOYEE	Employee details	Add employee details to database	Employee details was updated to database	Pass
VIEW EMPLOYEE DETAILS	Employee ID	Show Employee Details	Employee details was displayed	Pass
MODIFY EMPLOYEE DETAILS	Employee ID	Modify employee details	Employee details was modified	Pass
DELETE EMPLOYEE DETAILS	Employee ID	Delete Employee Details	Employee details was deleted from database	Pass

4.0 SAMPLE FORMS

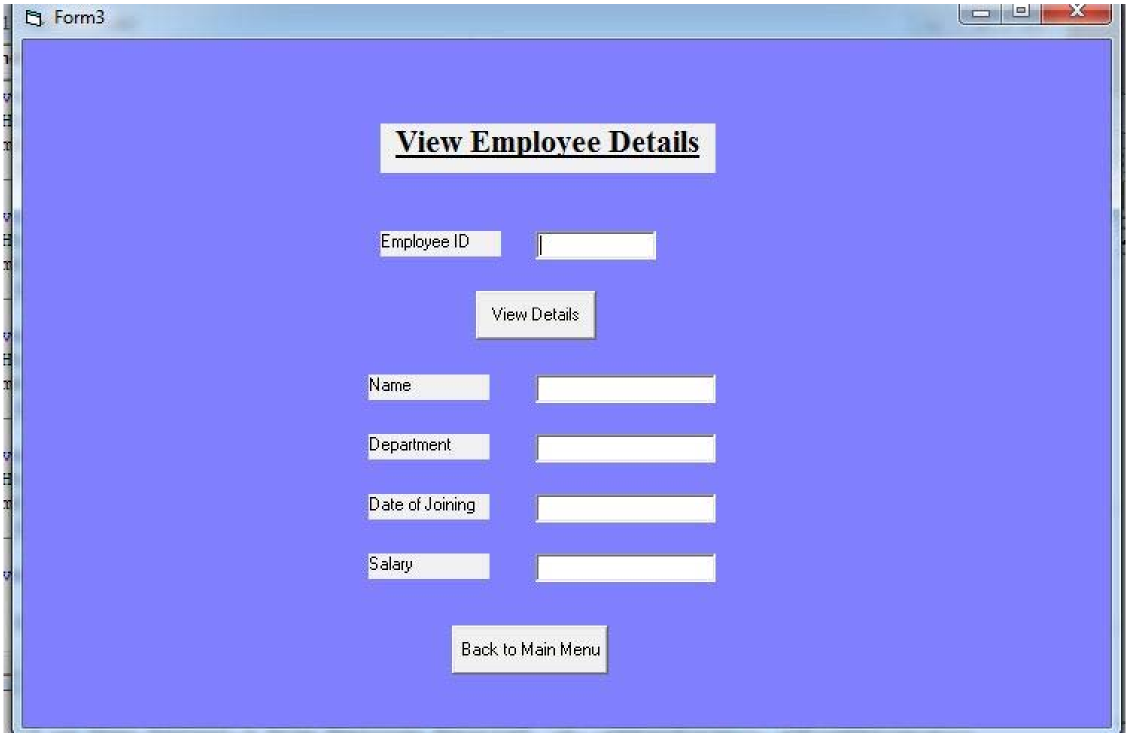
MAIN MENU



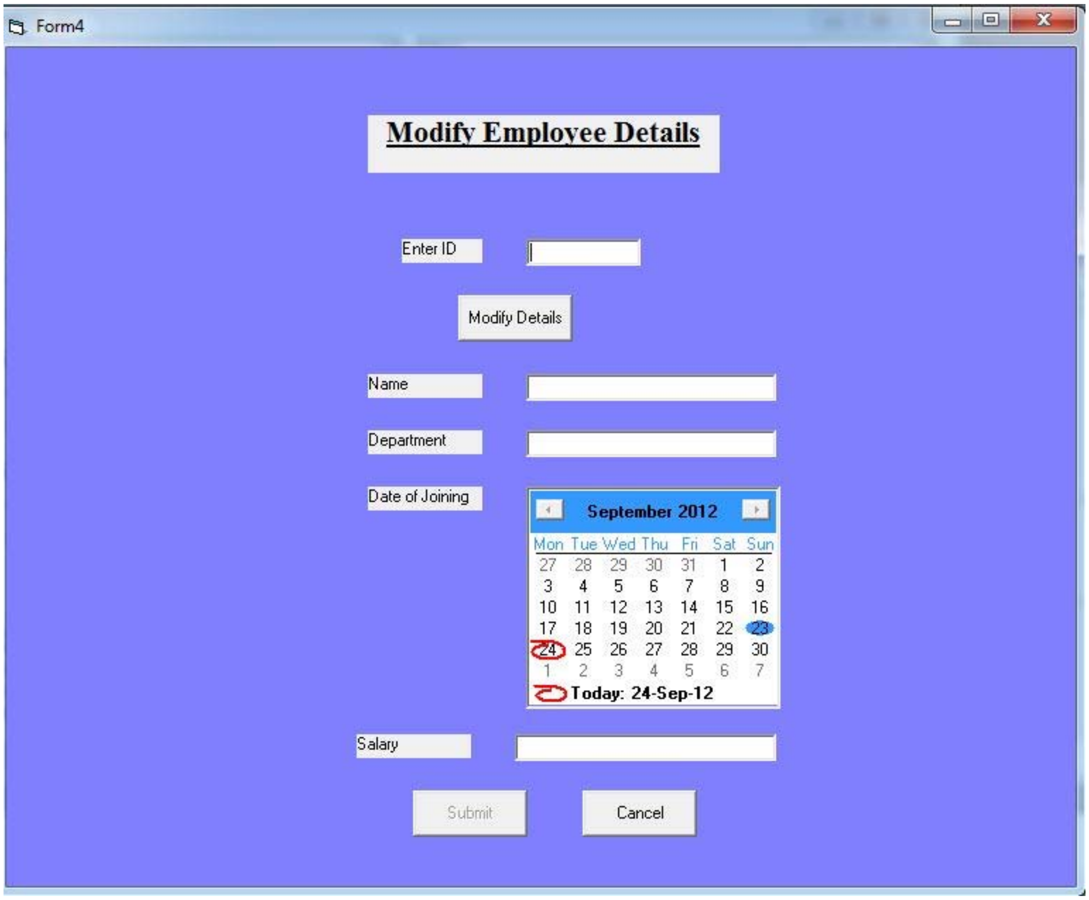
ADD NEW EMPLOYEE



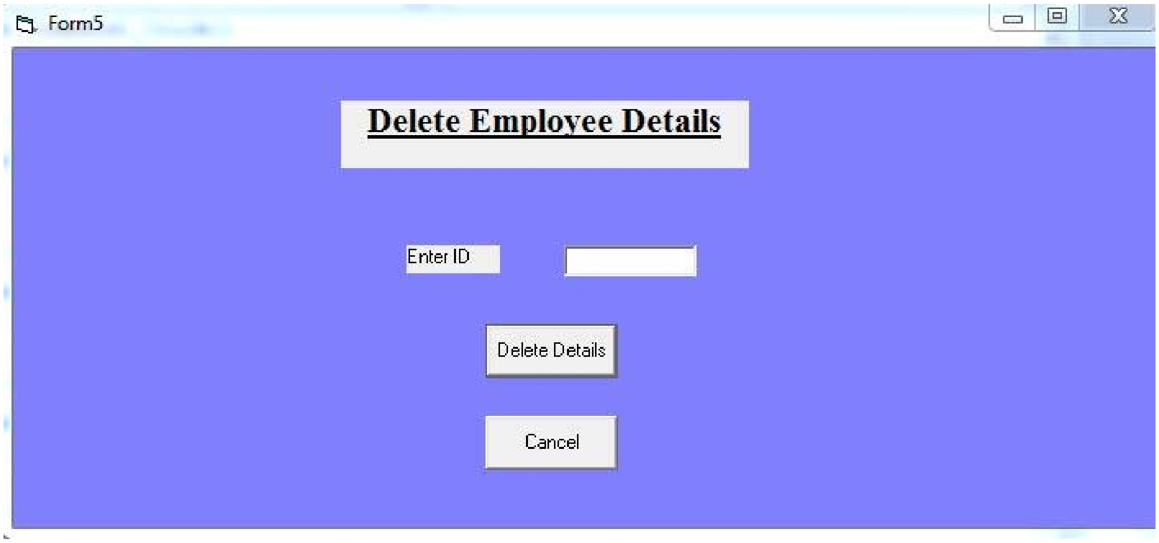
VIEW EMPLOYEE DETAILS



MODIFY EMPLOYEE DETAILS



DELETE EMPLOYEE DETAILS



5.0 RESULT

Thus the Employee Management System was implemented using the specified front end and back end tools.

EX NO :5
Date :

HOSPITAL MANAGEMENT APPLICATION

1.0 PROBLEM DEFINITION

A hospital application is to be developed which is required to perform the following functions:

1.1 It must provide a user in admin mode with the details of apatient, doctor.
1.2 It must provide a user in doctor mode who can modify the details of the illness and the treatment.

2.0 SRS DOCUMENT HOSPITAL MANAGEMENT APPLICATION

2.1 INTRODUCTION

2.1.1 Purpose

2.1.1.1 The purpose of this SRS is to describe the requirements involved in developing a system to manage hospital records.
2.1.1.2 The intended audience is any person who wants
2.1.1.2.1 To check patient and doctor details (both doctor and administrator mode).
2.1.1.2.2 To add new treatment details for any particular patient according to his illness. (onlydoctor mode).

2.1.2 Scope

2.1.2.1 The product is titled Hospital Management Application (HMA).
2.1.2.2 The product will perform the following tasks
2.1.2.2.1 Allow either an doctor or an administrator to view patient details.
2.1.2.2.2 Allow the administrator to add a new patient, doctor with corresponding details.
2.1.2.2.3 Allow the administrator to modify the detail of anpatient, doctor.
2.1.2.2.4 Allow the doctor to add the records for an ongoing treatment.

2.1.3 Definitions, Acronyms and Abbreviations

2.1.3.1 HMA: Hospital Management Application.

2.1.4 References

2.1.4.1 IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.

2.1.5 Overview

2.1.5.1 The SRS contains an analysis of the requirements necessary to help easy design.

2.1.5.2 The overall description provides interface requirements for the Hospital Management System, product perspective, hardware interfaces, software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.

2.1.5.3 Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Hospital Management Application in any fashion.

2.2 THE OVERALL DESCRIPTION

2.2.1 Product perspective

2.2.1.1 Hardware interfaces

2.2.1.1.1 Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system (such as any RDBMS) running on high rpm hard-disk permitting complete data redundancy and back-up systems to support the primary goal of reliability.

2.2.1.1.2 The system must interface with the standard output device, keyboard and mouse to interact with this software.

2.2.1.2 Software interfaces

2.2.1.2.1 Back End: MS Access

2.2.1.2.2 Front End: Microsoft Visual Basic 6.0

2.2.1.3 Memory Constraints

2.2.1.3.1 No specific constraints on memory.

2.2.1.4 Operations

2.2.1.4.1 The software allows two modes of operations

2.2.1.4.1.1 The administrator mode allows user to add a new patient, doctor, modify the existing details of a patient & doctor, view the details of an Patient & doctor.

2.2.2 Product Functions

2.2.2.1 Patient Details

The user (both administrator and doctor) must have the access to Up-to-date information about the patient including

2.2.2.1.1 Patient ID
2.2.2.1.2 Patient Name
2.2.2.1.3 Patient Age
2.2.2.1.4 Patient Address
2.2.2.1.5 Admit and Discharge Date

2.2.2.2 Doctor Details

The user (only in administrator mode) must be able to add and view by supplying the following doctor details.

2.2.2.2.1 Doctor ID
2.2.2.2.2 Doctor Name
2.2.2.2.3 Doctor Age

2.2.2.2.4 Doctor Address
2.2.2.2.5 Doctor Qualification

2.2.2.3 Illness Details

The user (only in doctor mode) must be able to modify the following details of an existing treatment.

2.2.2.3.1 Patient ID
2.2.2.3.2 Doctor ID
2.2.2.3.3 Illness
2.2.2.3.4 Medication

2.2.3 User characteristics

2.2.3.1 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

2.2.3.2 The Product is absolutely user friendly, so the intended users can be the naïve users.

2.2.3.3 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.

2.2.4 Constraints

2.2.4.1 At the time of adding a new patient and doctor, each must be assigned a unique ID number.

2.2.4.2 The ID numbers for the illness table must match the IDs in the respective tables

2.3 SPECIFIC REQUIREMENTS

2.3.1 Logical Database Requirements

2.3.1.1 There is one database which contains all the necessary information about a patient which includes patient ID, patient name, age, address, admit and discharge date

2.3.1.2. There is a database which contains all the necessary information about a doctor which includes doctor ID, doctor name, age, address, qualification.

2.3.1.3 There is a database which contains all the necessary information about a treatment which include Patient ID, doctor ID, illness, medication

2.4 FRONT – END DESCRIPTION

The front end for the Hospital Management Application (HMA) is designed using Microsoft Visual Basic 6.0. The front – end contains a user – friendly interface. It has a welcome screen that provides an option for the user to enter in doctor mode or in administrator mode. The user has to validate him self using password to enter in administrator mode. In administrator mode, apart form viewing the details the user can also add a new patient and doctor by providing details or modify the existing details using the patient ID and doctor ID.

2.5 BACK – END DESCRIPTION

There are 4 tables. The 1st one maintains login details for all the users. The 2nd table correlates a unique patient ID with his name, age, address, admit and discharge date.. The 3rd table correlates a unique doctor ID with his name, age, address and qualification. The 4th table correlates a doctor ID and patient ID with illness and medication

2.6 DATA STRUCTURES

2.6.1. LOGIN DETAILS

FIELD NAME	TYPE	CONSTRAINTS
ID	NUMBER	NOT NULL
PASSWORD	TEXT	NOT NULL

2.6.2. PATIENT DETAILS

FIELD NAME	TYPE	CONSTRAINTS
ID	NUMBER	NOT NULL
FNAME	TEXT	NOT NULL
LNAME	TEXT	
AGE	TEXT	
ADDRESS	TEXT	
INDATE	DATE/TIME	
OUTDATE	DATE/TIME	

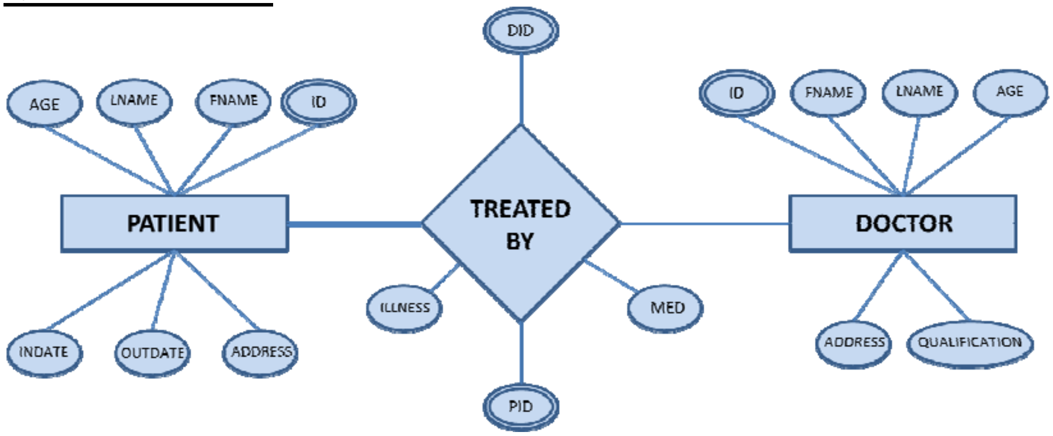
2.6.3. DOCTOR DETAILS

FIELD NAME	TYPE	CONSTRAINTS
ID	NUMBER	NOT NULL
FNAME	TEXT	NOT NULL
LNAME	TEXT	
AGE	NUMBER	
ADDRESS	TEXT	
QUALIFICATION	TEXT	

2.6.4. TREATMENT DETAILS

FIELD NAME	TYPE	CONSTRAINTS
PID	NUMBER	NOT NULL
DID	NUMBER	NOT NULL
ILLNESS	TEXT	
MEDICINE	TEXT	

2.7 E/R DIAGRAM

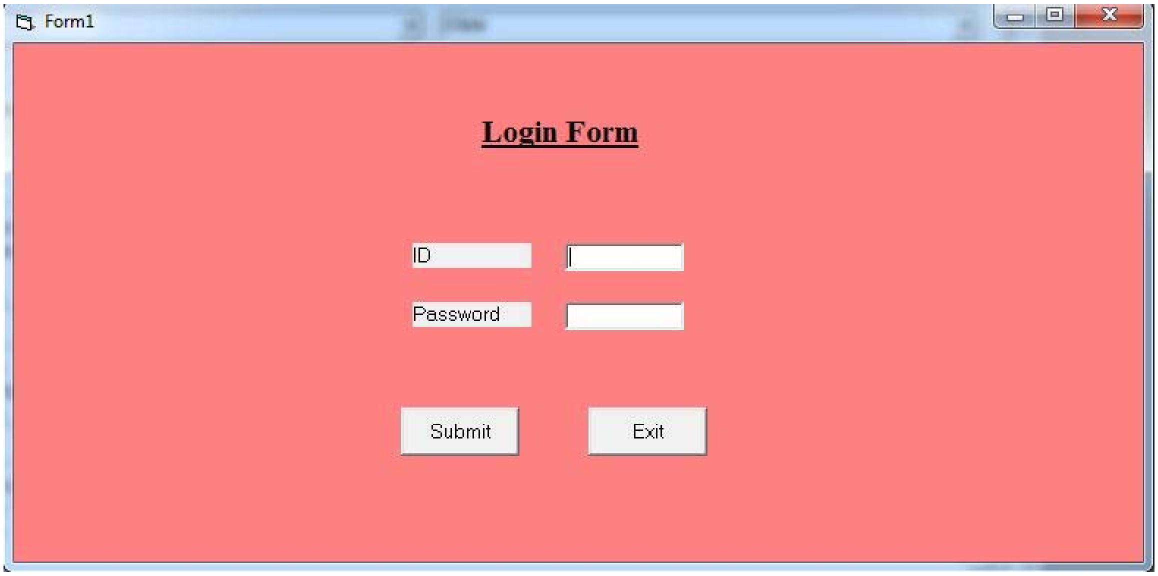


3.0 TESTING

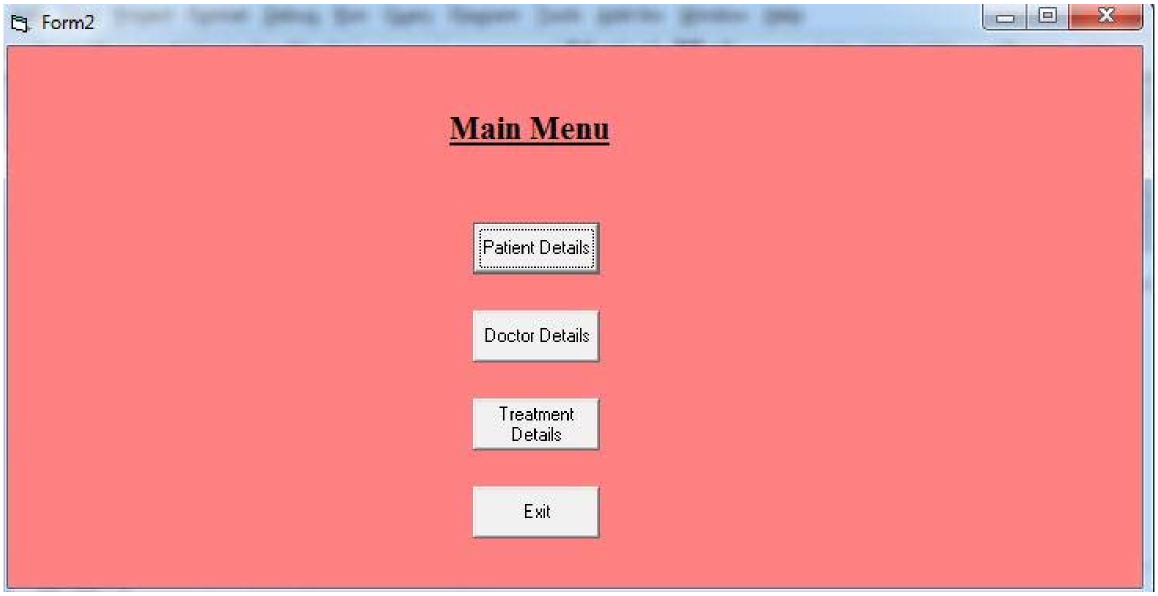
FORM NAME	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
LOGIN FORM	ID and Password	If correct password, main menu must be displayed	If correct password, main menu was displayed	Pass
MAIN MENU FORM	Menu Option	Required Form must be displayed	Required Form must be displayed	Pass
PATIENT DETAILS FORM	Patient ID or Patient details	Patient details must be displayed or updated to database.	Patient details was displayed or updated to database	Pass
DOCTOR DETAILS FORM	Doctor ID	Doctor details must be displayed	Doctor details was displayed	Pass
TREATMENT DETAILS FORM	Patient ID, Doctor ID, Illness and Medicines	Treatment details must be updated to database	Treatment details was updated to database	Pass

4.0 SAMPLE FORMS

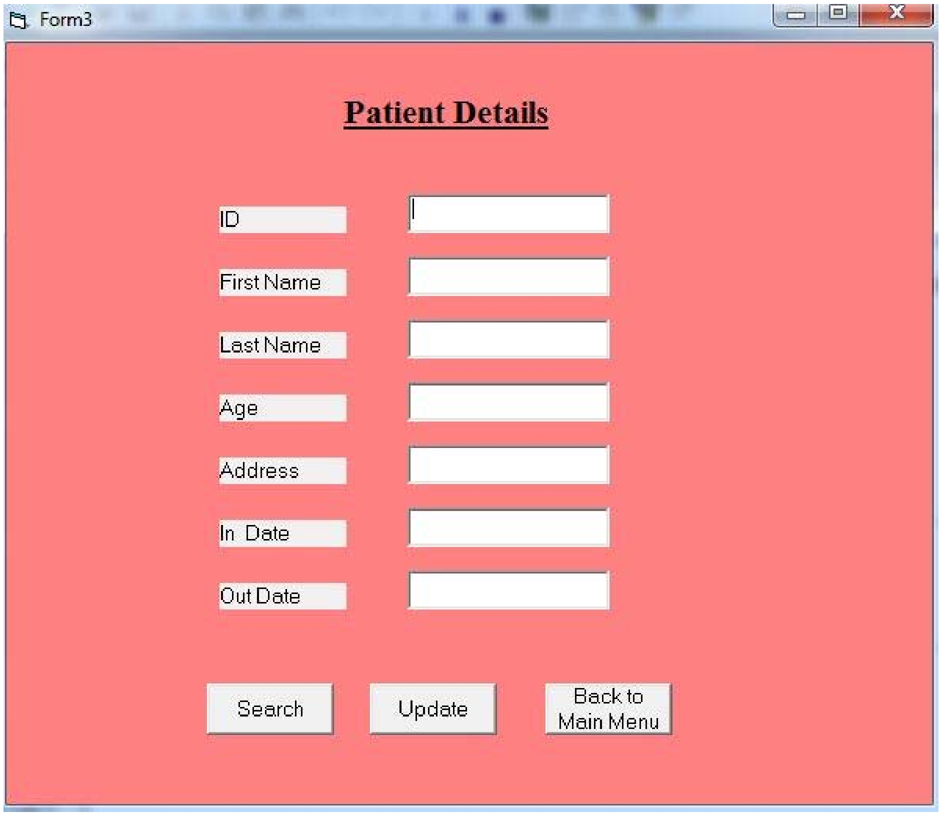
LOGIN FORM



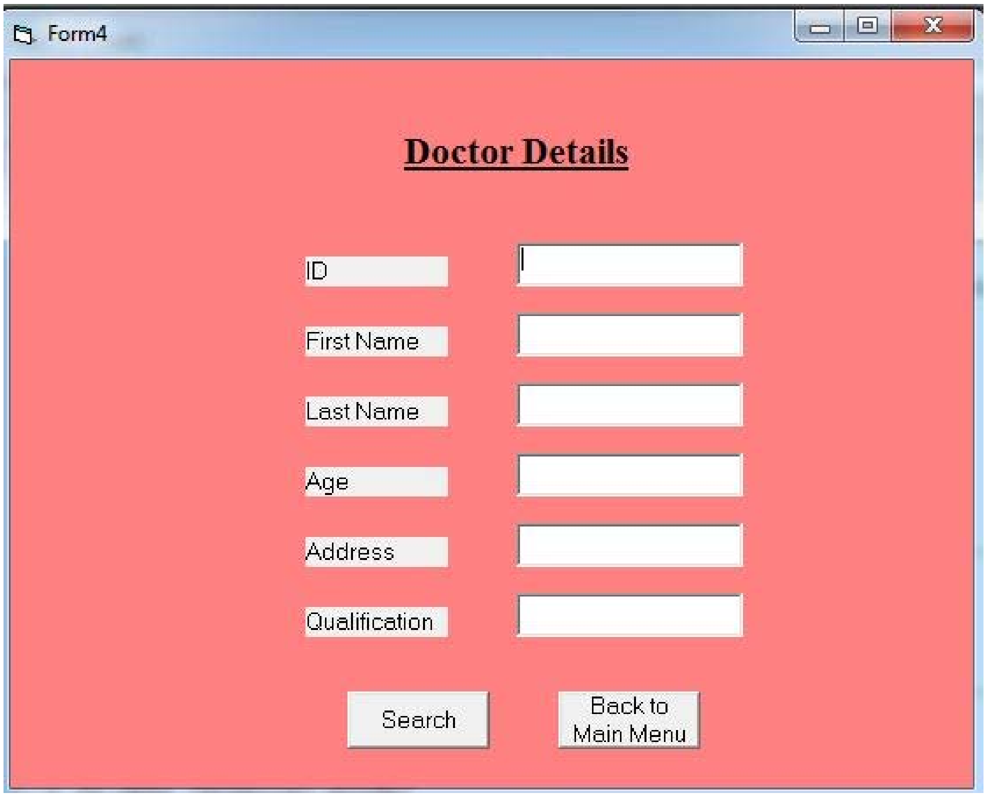
MAIN MENU FORM



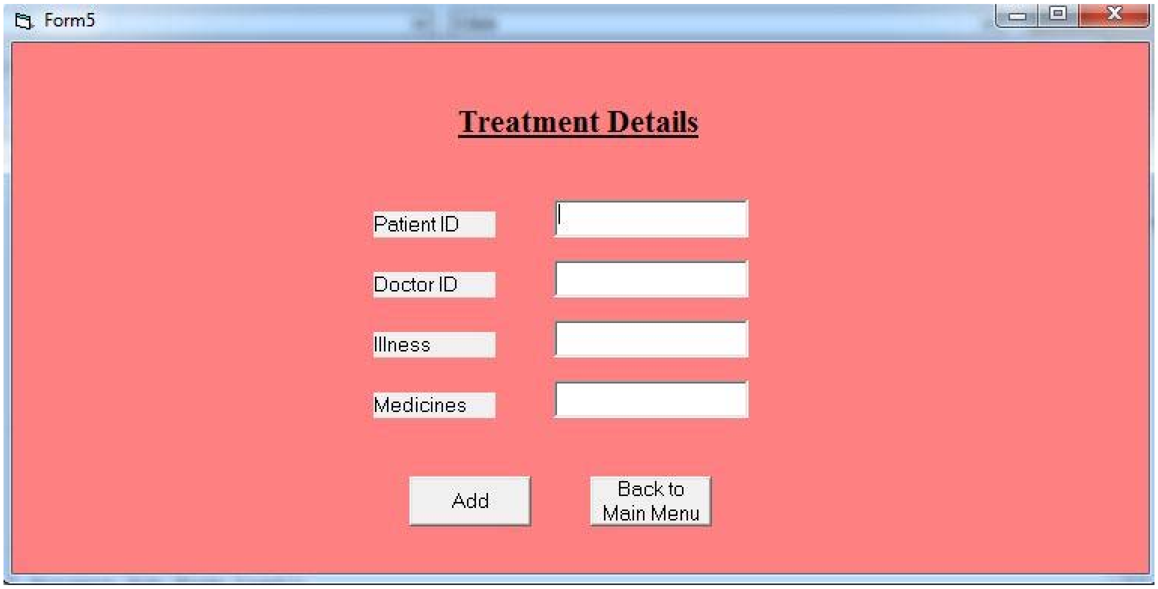
PATIENT DETAILS FORM



DOCTOR DETAILS FORM



TREATMENT DETAILS FORM



5.0 RESULT

Thus the Hospital Management System was implemented using the specified front end and back end tools.