

Money Tree Banking System
Requirement Specification
for
DOT NET
Training

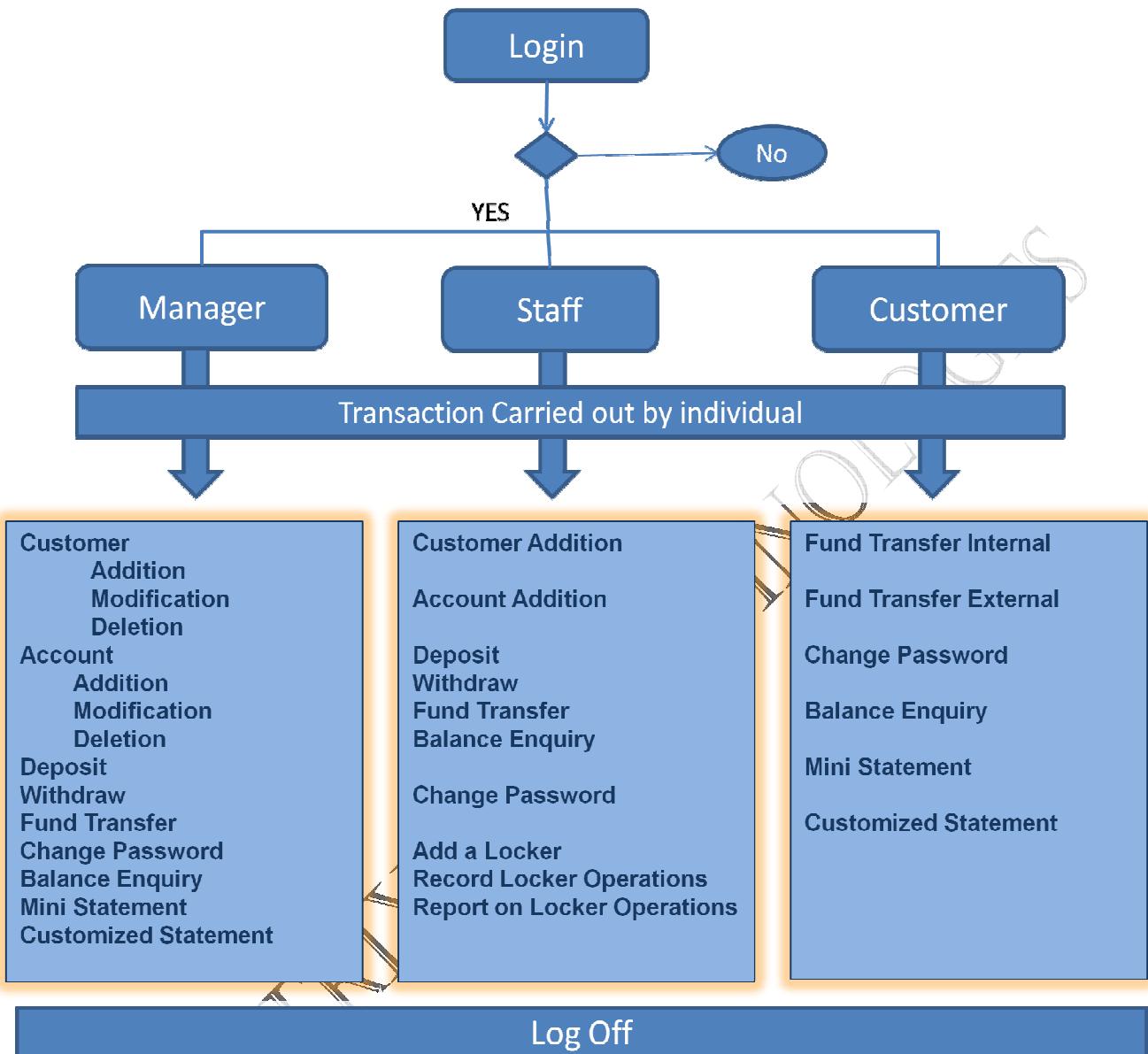
SUNMATRIXKL

LOGIES

Table of Contents

Sl. No.	Contents
1	Background
2	Business Requirements
3	Best Practices
4	Forms Creation

SUNMATRIXKENT TECHNOLOGIES



Introduction

This document contains the high level design of the project that has to be executed in order to complete the course Dot Net.

This project specification is designed to utilize all the features covered in training.

Business Requirements

The following are the business requirements for this project.

Sl.	Requirement	Windows Forms	ASP.NET
1	Login Screen	✓	✓
2	Add, Delete and Modify Customer Details	✓	✗
3	Add, Delete Account Details	✓	✗
4	Deposit	✓	✓
5	Withdrawal	✓	✓
6	Fund Transfer Internal	✓	✓
7	Balance Enquiry	✓	✓
8	Change Password	✓	✓
9	Fund Transfer External	✗	✓
10	Locker Addition, Modification, Deletion	✓	✗
11	Locker Assignment to Customer	✓	✗
12	Locker Transaction	✓	✗
13	Customized Statement	✓	✓

Best Practices

- Use the prefixes and guidelines recommended. The prefix should be in lower case and the name that follows it will start in upper case..A name may consist of different parts; e.g. txtCustName. Start each part of a name with an upper case character.

- Programmer defined names should be functionally meaningful, and should indicate the purpose of the variable/control/function in question
- Initialize the **maxlength** property of TextBoxes which expect string input from the user. While this may be set at design time, it is advisable to set it at runtime (in Constructor) by using the length of the variable which will eventually hold the input.
- Set the position of the form to be displayed.
- Variables
 - All variables MUST be declared before use.
 - Place all variable declaration statements at the beginning of a function, subroutine or an event procedure even though C# allows you to do otherwise. All variable declarations should be accompanied by a short, useful description.
Int intCounter // Scratch Pad variable for use in the for loop.
 - Initialize variable explicitly
 - Use Static variable & methods only if necessary
- Anticipate and handle runtime errors gracefully. A meaningful message must be displayed to the user, and the application should be terminated after proper cleanup
Error handling must be taken care of in all event procedures also where errors are expected.
- Statement at the same nesting level must be at the same indentation level.
- A Switch Case should always have a Default statement to handle unforeseen conditions.
- A menu structure with more than three levels is not recommended.
- Comments must be added whenever it is not easy for another programmer to understand the operation of a segment of code, or if unusual, tricky or unconventional techniques have been used in the source.
Comments must be succinct, concise, and grammatically correct. Comments should be clear to the reader. They should be used to add semantic information beyond what is expressible with the programming language.

Forms Creation

Note: User Interface design for Web Forms is not included as most of the screen is same as that of windows forms. You should implement it appropriately

Create Class Library (DLL) for following functionalities

- Login Validation
- Deposit :- A staff member should be able to update an account with an amount that is deposited to the account.

- Withdrawal :- A staff member should be able to update an account with an amount that is withdrawn from an account.
- Balance Enquiry : A staff member should be allowed to see the balance of any amount that is registered with the bank.

This module is used to create forms for entering data. To get more information about database and tables, refer to the Database Design.

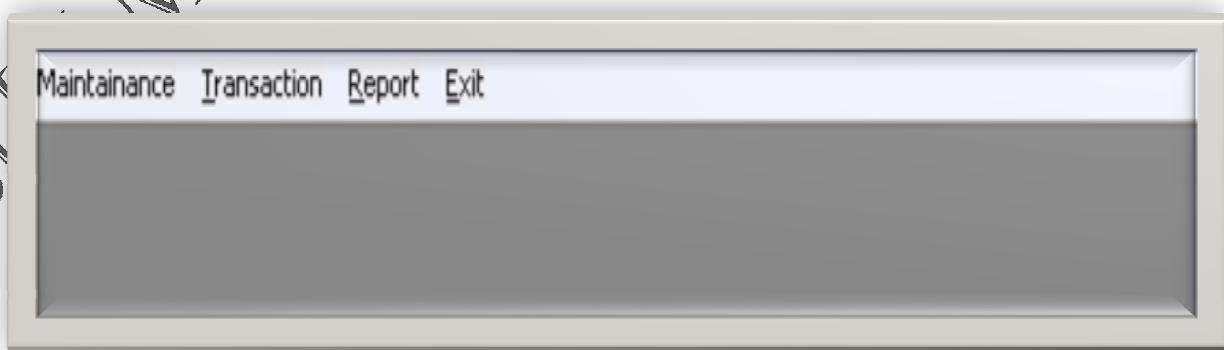
a) **Login Screen**

Create the Login Screen as shown in the below:

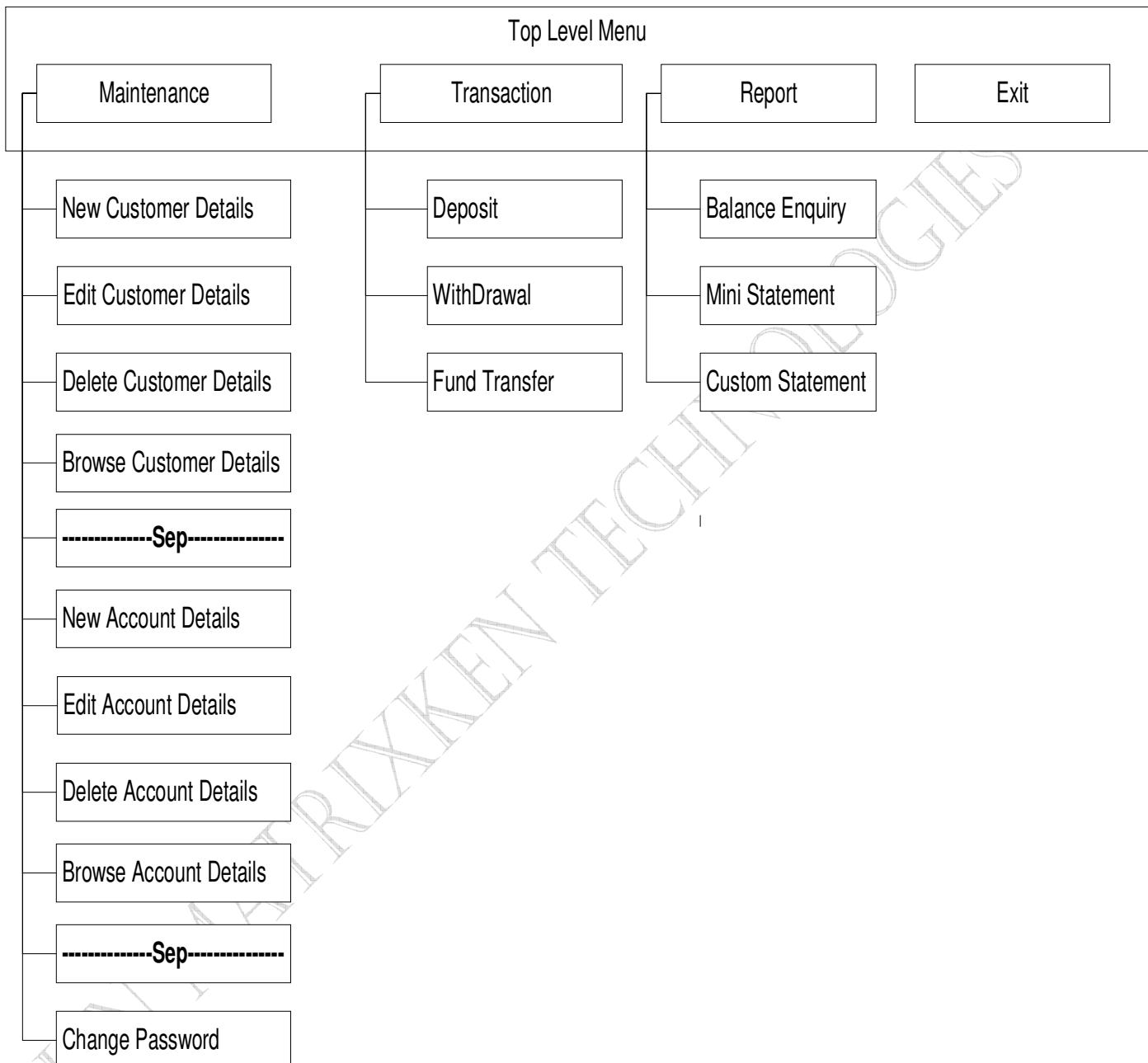
The functionalities of the above screen are as follows:

- Either User ID or Password fields should not be left blank.
- When the OK button is clicked, the User ID and the Password must be checked against the table **Login**. Depending on the role of the User ID, corresponding menu options in the MDI form must be enabled or disabled.
- Call the Class Library DLL functionality for validation.
- Only 3 invalid login credentials are allowed.
- When the Cancel button is pressed, the application should close.

b) **Create the following MDI form**



Create the following sub menus for the top level menus and choose appropriate ALT + Keys.



Note: - Date field is a Date Picker control

c) **Add a Customer**

On Click of Add Customer Menu the following screen should be displayed

The screenshot shows a Windows-style dialog box titled "New Customer Details". The form is organized into two columns. The left column contains fields for Customer ID*, User ID*, Name*, Gender (with radio buttons for Male and Female), Date Of Birth (a date picker showing "3 / 9 /2004"), and Address (a text area). The right column contains fields for City, State, Pin, Telephone, Fax, and EMail. At the bottom of the form are two buttons: "Save" and "Cancel".

The functionalities of the above screen are as follows:

- On the form load
 - Customer ID must be generated automatically and displayed in the Customer ID text field. Use the stored procedure created in the Oracle Project.
 - Customer ID field must be locked.
- User ID must be unique and has to be checked for uniqueness with the back end if not unique force the user to enter unique value with hints.
- When the Cancel button is clicked, then that record must not be saved and the form should close.
- All the fields marked * are compulsory fields.
- When Save button is clicked :
 - All the data fields on the form must be validated and appropriate messages has to displayed.
 - Record entered must be saved.
 - **Customer Details form** has to be closed and **New Account form** has to be displayed

d) Add Account Screen

The screenshot shows a window titled "New Account Details". It contains the following fields:

- Account Number *: A text input field.
- Date Of Opening: A dropdown menu showing "8/15/2005".
- Customer ID *: A text input field.
- Current Amount *: A text input field.
- Account Type *: A dropdown menu.
- Status:
 - Functional
 - Closed
- Buttons: "Save" and "Cancel".

The functionalities of the above screen are as follows:

- On Form load
 - The Account Number must be automatically generated, displayed in the Account Number field and should be locked (use stored procedure created in Oracle)
 - Customer ID value has to fetched from the previous screen, displayed in the Customer ID field and should be locked
 - By default Date Of Opening must be the current date
 - By default status should be functional and closed should be disabled.
- The Account Type value has to be populated with SB, CA, RD or FD.
- When Save button is clicked:
 - All the data fields on the form must be validated and appropriate messages has to displayed.
 - Current Amount should not be less than Rs 500.
 - Both the Customer Details and Account Details has to be committed to the database.
 - Close the form.
- When Cancel button is clicked
 - Pop up a message stating that Customer details entered in the previous form would be lost. Get the confirmation before proceeding
 - **Yes**:- Both the Customer Details and Account Details has to be rolled back. (ie to say even customer details saved in the previous screen should not be committed to the database). Close the form.
 - **No** :- Nothing to implement.

e) **Edit Customer Details**

On Click of the Edit Customer Menu option following screen has to be displayed

The screenshot shows a Windows-style dialog box titled "Edit Customer Details". At the top left is a text input field labeled "Customer ID *". To its right is a "Fetch" button. Below this is a large rectangular form area containing various input fields. On the left side of the form, there are fields for "User ID *", "Name *", "Gender" (with radio buttons for "Male" and "Female"), "Date Of Birth" (a dropdown menu showing "8/14/2005"), and "Address" (a text area with scroll bars). On the right side of the form, there are fields for "City", "State", "Pin", "Telephone", "Fax", and "EMail". At the bottom of the form area are two buttons: "Save" and "Close".

Functionality of the above screen is as follows

- When the form loads, all the controls except Customer text field, Fetch button and Close button must be disabled.
- Fetch Customer ID
 - Customer ID entered in the Customer ID text field has to be validated with the back end table on the click of Fetch Button.
 - Customer ID :
 - Found: then populate the Fields with respective values, Enable all the controls except Customer Id and Fetch Button.
 - Not Found: then pop up a message to say the Customer ID does not exists and set the focus to Customer ID text field.
- On Click of Save button save the modified data and pop up a message intimating data updation.
- On Click of Close button close the form.

f) **Delete Customer Details**

On click of the Delete Menu option following screen has to be displayed

The screenshot shows a window titled "Delete Customer Details". At the top left is a text input field labeled "Customer ID *". To its right is a "Fetch" button. Below this is a large rectangular panel containing various input fields: "User ID *", "Name *", "Gender" (with "Male" and "Female" radio buttons), "Date Of Birth" (a dropdown menu showing "8/14/2005"), "Address" (a scrollable text area), "City", "State", "Pin", "Telephone", "Fax", and "EMail". At the bottom of the window are two buttons: "Delete" and "Close".

Functionality of the above screen is as follows

- When the form loads, all the controls except Customer text field, Fetch button and Close button must be disabled.
- Fetch Customer ID
 - Customer ID entered in the Customer ID text field has to be validated with the back end table on the click of Fetch Button.
 - Customer ID :
 - Found: then populate the Fields with respective values, Enable all the controls except Customer Id and Fetch Button.
 - Not Found: then pop up a message to say the Customer ID does not exists and set the focus to Customer ID text field.
- On Click of Save button delete the record from the database and pop up a message intimating deletion.
- On Click of Close button close the form.

g) Browse Customer Details

On click of the Browse Customers Menu option following screen has to be displayed

The screenshot shows a window titled "Browse Customer Details". It contains several input fields and buttons. On the left, there are fields for "Customer ID *", "User ID *", "Name *", and "Gender" (with radio buttons for Male and Female). Below these are fields for "Date Of Birth" (set to 8/14/2005) and "Address" (with scroll bars). On the right, there are fields for "City", "State", "Pin", "Telephone", "Fax", and "EMail". At the bottom of the window are five buttons: "First Record", "Prev", "Next", "Last Record", and "Close".

Functionality of the above screen is as follows

On form load first record from the customer details has to be displayed in the respective fields. All the controls except the buttons has to be locked.

Buttons

First Record : Record pointer has to be moved to first record and corresponding details has to be displayed in respective fields

Prev : Previous Record from the current record details has to be displayed in respective fields

Next : Next Record from the current record details has to be displayed in respective fields

Last Record : Record pointer has to be moved to last record and corresponding details has to be displayed in respective fields

Close : Close the form.

h) Deposit Screen

On Click of Deposit Menu option the following screen has to be displayed.



The image shows a graphical user interface for a 'Deposit Screen'. The title 'Deposit Screen' is centered at the top. Below it is a vertical list of input fields and controls:

- Account Number**: An input field with a 'Fetch' button to its right.
- Customer ID**: An input field.
- Customer Name**: An input field.
- Account Type**: An input field.
- Date**: A date picker showing '8/15/2005'.
- Amount**: An input field.
- Buttons**: Two buttons at the bottom: 'Deposit' and 'Cancel'.

The functionalities of the above screen are as follows:

- When the form loads, All the controls except Account Number text field, Fetch button and Cancel button must be disabled.
- Fetch Account Number
 - Account Number entered in the Account Number text field has to be validated with the back end table on the click of Fetch Button.
 - Account Number :
 - Functional Account Number, then populate the Customer Id and Customer Name Field with respective values, set the Date to Current Date. Enable all the controls except Account Number, Customer Id, Customer Name, Account Type text fields and Fetch Button.
 - Closed Account Number, then pop up a message to say the Account number is closed and set the focus to Account Number text field.
 - Not a Valid Account Number then pop up a Message to say Account Number is not valid and set the focus to the Account Number text field.

- On Click of Deposit Button
 - Check for Amount entered to be a positive numeric value greater than 0.
 - Commit the data by calling the corresponding stored procedure created in Oracle to the data base and pop a message "Amount Credited" before closing the form. (Both the transaction and Account Table has to be Updated)
- On Click of Cancel Button
 - Close the form without saving the data

i) **Withdrawal**

On Click of WithDrawal Menu option the following screen has to be displayed.

WithDrawal Screen

Account Number **Fetch**

Customer ID

Customer Name

Account Type

Date

Amount

WithDraw **Cancel**

The functionalities of the above screen are as follows:

- When the form loads, All the controls except Account Number text field, Fetch button and Cancel button must be disabled.
- Fetch Account Number
 - Account Number entered in the Account Number text field has to be validated with the back end table on the click of Fetch Button.
 - Account Number :

- Functional Account Number, then populate the Customer Id and Customer Name Field with respective values, set the Date to Current Date. Enable all the controls except Account Number, Customer Id, Customer Name, Account Type text fields and Fetch Button.
- Closed Account Number, then pop up a message to say the Account number is closed and set the focus to Account Number text field.
- Not a Valid Account Number then pop up a Message to say Account Number is not valid and set the focus to the Account Number text field.
- On Click of WithDraw Button
 - Check for Amount entered to be a positive numeric value greater than 0.
 - Check for the available balance to be greater than minimum balance of Rs. 500 after deduction of the withdrawal amount. If it is less than the minimum balance pop up a message with the information of maximum amount that could be withdrawn and set the focus to the amount text field.
 - Commit the data by calling the corresponding stored procedure created in Oracle to the data base and pop a message "Amount Debited" before closing the form. (Both the transaction and Account Table has to be Updated).
- On Click of Cancel Button
 - Close the form without saving the data

j) **Fund Transfer**

On Click of Fund Transfer Menu option the following screen has to be displayed.



The image shows a window titled "Fund Transfer Screen". It contains four input fields: "From Account Number", "To Account Number", "Date" (set to 8/15/2005), and "Amount". Below the fields are two buttons: "Transfer" and "Cancel".

Fund Transfer Screen	
From Account Number	<input type="text"/>
To Account Number	<input type="text"/>
Date	8 /15/2005
Amount	<input type="text"/>
<input type="button" value="Transfer"/>	<input type="button" value="Cancel"/>

The functionalities of the above screen are as follows:

- **On Click of Transfer Button**
 - Validate for data in the account fields
 - By default date should be current date
 - Amount should be a Positive Numeric value greater than 0.
 - Call the corresponding stored procedure created in Oracle, trap the return value and proceed accordingly.
- On Click of Cancel Button
 - Close the form

k) **Balance Enquiry**

On Click of Balance Enquiry Menu option the following screen has to be displayed.

The screenshot shows a window titled "Balance Enquiry". It contains several input fields and buttons. At the top left is a label "Account Number" next to a text input field. To its right is a "Fetch" button. Below this row are four more input fields arranged horizontally: "Customer ID", "Customer Name", "Account Type", and "Amount". Each of these has its own separate text input field. At the bottom center of the window is an "Exit" button.

The functionalities of the above screen are as follows:

- Customer ID, Customer Name, Account Type and Amount text fields has to be locked.
- Fetch Account Number
 - Account Number entered in the Account Number text field has to be validated with the back end table on the click of Fetch Button.
 - Account Number :
 - Functional Account Number, then populate the Customer Id, Customer Name, Account Type and Amount Field with respective values,
 - Closed Account Number, then pop up a message to say the Account number is closed and set the focus to Account Number text field.
 - Not a Valid Account Number then pop up a Message to say Account Number is not valid and set the focus to the Account Number text field.
- When Exit button is clicked, the form should close.

I) **Change Password**

On Click of Change Password Menu option the following screen has to be displayed.

The dialog box has a title bar 'Change Password'. Inside, there are four text input fields:

- User ID
- Old Password
- New Password
- Confirm Password

At the bottom are two buttons:

- OK
- Cancel

The functionalities of the above screen are as follows:

- Once the Ok Button is clicked
 - Check for the exactness of New Password and Confirm Password
 - Check for the validity of User Id and Password by calling the stored procedure created in Oracle and implement changes accordingly.
- On Click of Cancel Button close the form.