Technical Manual - Library Management System

V.V.Satya Narayana Pandey Vineet Kumar
Devendra Bendkhale Namaba Jamir Srikar

20th Febraury 2014

Contents

project name is - lbms

1 tables in the software

Database name is - project

1. Table for students - studenttable
    - Student Name - name
    - webmail id -mailId
    - password in encrypted form -password
    - roll no. is primary key. This is because primary key has to be a number always. -rollNo
    - fine imposed -a float value - fine
    - remark for the student - remark
2. Table of faculty - facultytable
    - Faculty Name - name
    - webmail id - mailId
    - faculty id is primary key. - a integer value- facId

      This is created because webmail id is varchar and can't be a primary key.
    - password in encrypted form - password
    - fine imposed - a float value - fine
    - remark for the faculty - remark
3. Table for Librarian - librariantable
    - Librarian name - name
    - webmail id is primary key - mailId
    - password in encrypted form - password
4. Table for Admin - admintable
    - Admin name - name

- webmail id is primary key - mailId
- password in encrypted form - password

5. Feedback Table - feedbacktable
   - id is primary key, auto incremented.
   - webmail id - mailId
   - feed back - feedback
   - date of feed back - date
6. Table of issued books - issuedbooks
   - Account no. of the book issued is primary key - accNo
   - webmail id of person issued the book - mailId
   - Date of issue - issueDate
   - Category of the person issued(Faculty or student) - category
7. Table for requests - requeststable
   - title of the book - title
   - title id of the book - titleId
   - number of books requested - number
   - date of request - dateOfRequest
   - webmail id of person requesting book. - mailId
   - importance of book. - importance
   - status of the request - a varchar - status

     status = "rejected" if admin or librarian rejected the request.
     status = "partial" if admin or librarian accepted only few books.
     status = "accepted" if admin or librarian accepted request.
     status = "pending" if admin or librarian did n't give any responce yet.
8. Table for books to be ordered - orderstable

- title of the book - title
- title id of the book is primary key - titleId
- number of books to be ordered - number

9. Table of books present in library - allbooks
    - Account no. as primary key - accNo
    - rack no. - rackNo
    - title of the book - title
    - Author of the book - author
    - Publisher of the book - publisher
    - Subject - subject
    - issued status. It is a boolean value. - status
    - title id - titleId
10. fine per book - finetable
    - id as primary key - id
    - fine per book per day - fine

        consist of two rows
        First row consists of fine per book per day for faculty - id = 1
        Second row consists of fine per book per day for students - id = 2
11. request limit of users - requestlimits
    - id as primary key - id
    - request limit - limit

        Consist of two rows
        First row consist of request limit of faculty - id = 1
        Second row consist of request limit of students - id = 2

2 Forms used in software

- A form for home page. - home

  This consist of search button, combo box for search type, a data grid view for showing books , login form at side of it.

- Forms of student

  – A home page form for student - home_stud

    This shows the title, account no., date of issue, author name of all books issued. This form also shows fine of the user at the top of the form. This form contains a search button.

  – A form containing search options - search_stud

    This page contains options for search. It contains a text box for typing query. A combo box for selecting search type, a search button, a data grid view for showing books. For each book, a link has to be given to another form showing the book details. For each book, a request link is to be present.

  – A form showing book data. - bookData_stud

    This page is shown when user clicks on a book. It contains all the details of book.

  – A form showing requests made by student - requests_stud

    This page contain all the requests made by student. For each request, it also contain an option to remove request.

  → All forms contain logout option and link to all other forms.

- Forms of faculty
  Forms of student are repeated for faculty

  Changes made are
  – on clicking request button, another option have to appear asking number of books requested.
  – Fine has to be calculated in different manner.
  – Number of requests that can be made are different.

- Forms of admin
  - Home page of admin - home_adm
  - A page showing requests sorted according to date - viewRequests_adm

    It shows list of requests sorted according to date. For each request, book name, person requested, no. of books requested appears. Admin can accept equal or lesser number of books.
  - A page showing all book orders. - viewOrders_adm It shows list of books to be ordered.
  - A page for viewing feed back. - viewFeedback_adm

    On this page, admin can view feedback given by students and faculty. There will be a text box for admin to note developments after viewing feedback.
  - A page for viewing developments noted upto now.
  - A page for adding and removing students, faculty and librarian. - addPeople_adm

    This is helpful at beginning of the academic year for adding new students and deleting old students.
- Forms of Librarian
  - Home page for librarian - home_lib
  - A page for issuing books - issue_lib
  - A page for renewing books - renew_lib
  - A page for returning books - return_lib
  - A page for accepting requests - requests_lib
  - A page for adding a book to database - add_book_lib
  - A page for removing a book from database - remove_book_lib

    This is useful when a student or faculty loses a book.

3 Librarian

1. home_lib

   Components included : Groupbox for viewing account information which includes:

   - Combobox (name = category)

     Combobox (name = category) - specifies the category of the user whose account is to be viewed.

   - Text Box (name = rollNo)

     specifies the roll number of the user whose account is to be viewed.

   - Button (name = viewAcc)

     onclick it validates the inputs, connects to the databases 'studenttable' and 'issuedbooks' and fetches information about the user whose account is to be viewed.

   Functions :

   - private: System::Void viewAcc_Click(System::Objectŝender, System::EventArgsê)

     As the name suggests this function is invoked when 'viewAcc' button is clicked. It validates the input and then first it uses the given roll number to fetch corresponding webmail Id (as mailId is the primary key in issuedbooks table), then using the fetched mailId it fetches the books issued by the on the card of the user and displays it in a Data Grid View.

2. add_book_lib

   Components included :

   - Groupbox :

     A section for adding books to the database which includes:

- TextBox (name = accNo) :

  It contains the account number of the book to be added. Since account number is the primary key for allbooks table its value is automatically assigned as one greater than the largest account number present in the database. This textBox is Readonly in nature i.e its value cannot be changed by the librarian (this avoids any clashes in the value of primary key)

- TextBox (name = rackNo) :

  This contains the rack number where the book will be kept in the library.

- TextBox (name = title) :

  This contains the title of the book to be added.

- TextBox (name = author) :

  This contains the author's name of the book being added.

- TextBox (name = publisher) :

  This contains the publisher's name of the book being added.

- TextBox (name = titleId) :

  It contains the title Id of the book being added. If another copy of the book is present in the database a query is executed and same titleId is assigned to this book. If not then a unique title id is assigned to the book.

- Combobox (name = stat) :

  It shows the status of the book being added in the library (whether the book is issued or not issued.)

- Button (name = addbook_button) :

  It invokes a function when clicked which adds the book in the library database.

Functions :

- private: System::Void add_book_lib_Load(System::Objectŝender, System::EventArgsê)

It specifies the onload functions when the form is loaded. A connection is established with the database using the variable MySqlConnectionĉonDatabase and then the command "select * from project.allbooks order by accNo desc limit 1;" is executed The value of 'accNo' TextBox is assigned here

- private: System::Void addbook_button_Click(System::Objectŝender, System::EventArgsê)

  This functions add the book in the 'addbooks' database. First it establishes a connection with the database and then an insert query is executed which inserts the input values in the database.

- private: System::Void title_TextChanged(System::Objectŝender, System::EventArgsê)

  This function tracks the value of 'title' TextBox and if a book with the same title is present in the database then its titleId value is copied in the 'titleId' TextBox and the textBox is made readonly. If not then a unique value is assigned to the 'titleId' TextBox by taking the largest value of titleId in allbooks table and incrementing it.

3. remove_book_lib:

   Components :

   - TextBox (name = accNo) :

     It contains the account number of the book to be deleted. Since account number is the primary key for allbooks table, using its value we can get every other information about a book. Every other TextBox other than this is ReadOnly TextBox.

   - TextBox (name = rackNo) :

     It contains the rack number in which the book to be deleted is present.

   - TextBox (name = title) :

     It contains the title of the book to be deleted.

   - TextBox (name = author) :

It contains the author's name of the book to be deleted.

- TextBox (name = publisher) :

  It contains the publisher's name of the book to be deleted.

- TextBox (name = titleId):

  It contains the title Id of the book to be deleted.

- Button (name = del_book):

  It invokes a function which deletes the row which given primary key from the database.

Functions :

- private: System::Void accNo_TextChanged(System::Objectŝender, System::EventArgsê)

  It tracks the value of 'accNo' TextBox and checks whether a book is persent in the database with the given account number or not.
  If a book is persent with the given account number all the other details gets filled automatically else every other TextBox remains empty.
  "select * from project.allbooks where accNo = '"+this-¿accNo-¿Text+"';" This query is executed to get the information.

- private: System::Void del_book_Click(System::Objectŝender, System::EventArgsê)

  As the name indicates it is invoked when the del_book button is clicked. It validates the input and executes a query which uses the account number and deletes the corresponding row from allbooks table and also executes a query which deletes the row corresponding to the account number in issuedbooks table (if any).

```
**Form Books_Issue**

It contains a TextBox for bookid, booktitle,student/fac mail_id,category(facult
Date of issue :Comes as a current date.
```

Functionality:
1.If you enter a valid book id then corresponding book title will appear automa
2.If book is already issued then librarian cannot issue it again as a label7 wi
3.If book is being issued for user who has crossed his issuing limit then messa
4.After book is issued corresponding book and its details will go to issued bo
5.Also no. of books issued of corresponding person increases by one.

Form return_lib

It contains a TextBox for bookid, booktitle,student/fac mail_id,category(facult
Date of issue :Comes as a issued date.
Date of return :Comes as a current date.

1.If you enter a valid book id then corresponding book title, mail_id of the bo
2.If book is already issued then librarian cannot return not issued or invalid
3.After returning fine will be recalculated and no. of books issued of the corr

Form Renew books
It contains a TextBox for bookid, booktitle,student/fac mail_id,category(facult
Date of issue :Comes as a issued date.
Date of renew :Comes as a current date.

1.If you enter a valid book id then corresponding book title ,mail_id of the bo
2.If book is already issued then librarian cannot return not issued or invalid
3.After returning fine will be recalculated and no. of books issued of the corr

Form Search

It contains a comboBox to denote by what field can we search and text Box which
It contains DatagridView which displays the books table according to query.

Form Accept
It displays requests by the students in DataGridview and loads them on button c

4 Student and Faculty

Same forms are used for student and faculty since there is no big difference between those 2 users except for fee calculation and limits for requests etc.,

In order to distinguish between faculty and student, a userId is passed to the form. userId is 1 for faculty and userId is 2 for students. The same is used in tables which decides limits.
The forms for faculty and students are as follows :

1. home_stud.h
   This form contains 4 tabs.

   - Home tab :
     This tab shows all the books issued by the user. fine imposed on the user etc., The contents of the page are:
     - A datagrid view named dataGrid :
       This dataGrid is loaded when the form loads. That is the loading of this dataGrid happens in form load event.
   - Seach Tab :
   - Feedback Tab :
   - Requests Tab :

2. bookData_stud.h :
   This form is common for home page, faculty and students page. This page shows book title, author, publication, number of books present in library and number of books currently available. It also shows rack numbers, account numbers, issue status of all the books present in library.

   A global variable named userId is passed to this form. userId is 1 for faculty and userId is 2 for students. userId is 0 when the user didn't login and view book results.
   This form contains :

- Labels for showing details of the book.
- A data grid named data_dataGrid which contain rack number, account number, etc., for all books.
- A text box for showing number of books requested.
- A request button for accepting request from the user.

  In the click event of the button, following details of the user are checked.
  1) Have the user logged in? This is checked just by comparing userId.
  2) Have the user reached his request limits. This is done by retrieving request limits of the users and number of books requested by the user. (The pending requests) and then comparing them.
  3) Is the book present in library? This is done by comparing issue status of all books of same kind in allbooks database.
  4) Have the user already requested the book. This is done by comparing all the requests of user.

  This dataGrid is loaded in form_load event.