Súkromná stredná odborná škola Dneperská 1,040 01 Košice

ReBooks

System manual

Authors: Vladimír Demko Košice 2016

Miloš Kočan

Matej Pliešovský

Tomáš Tomčík

Content

[1 Program description 3](#_Toc453279462)

[2 Analysis 4](#_Toc453279463)

[2.1 Process of development design of program 4](#_Toc453279464)

[2.2 Used developer languages 6](#_Toc453279465)

[2.2.1 C# language 6](#_Toc453279466)

[3. Process of Development Program and its components 7](#_Toc453279467)

[3.1 Database 7](#_Toc453279468)

[3.2 Program Interface 7](#_Toc453279469)

[3.3 Web Interface 7](#_Toc453279470)

[4. Diagrams 8](#_Toc453279471)

[4.1 Diagram of database 8](#_Toc453279472)

[4.2 Class diagram 9](#_Toc453279473)

[4.3 Flowchart of user 10](#_Toc453279474)

[4.4 Flowchart of administrator 11](#_Toc453279475)

[4.5 State diagram 12](#_Toc453279476)

[4.5.1 State diagram of user 12](#_Toc453279477)

[4.5.2 State diagram of book 12](#_Toc453279478)

[5 Forms, classes and methods 13](#_Toc453279479)

# 1 Program description

Rebooks program is a computer and web application of library's electronic system. Rebooks for the average user provide a web interface as an organized website. For the administrator, there is a computer application, in which the administrator can also switch to the user interface as a regular user. In addition to these interfaces, the application offers also installing the system on your computer. After installation it is not necessary to run it through Visual Studio, in which it was created.

The system administrator can edit the list of books confirm the registration of new users, confirm the booking of books or reject. He can modify user roles and give them permission administrator or an user. Or he can modify user profiles, edit book information, and add new ones.

Ordinary user can browse a list of books. Furthermore he can borrow books. The user can edit profile and your profile photo.

# 2 Analysis

ReBooks program implies requirements:

• Ability to sign in or sign up to an online database through application

• Modifications such as inserting and editing books, editing user

• Search the list of books by parameters

• Reservations of books

• Confirmation of registration

## 2.1 Process of development design of program

1. Definition of users
   1. Authentification (login, registration, logout)
   2. Role
   3. Administrator
2. Definition of books
   1. Categories
   2. Checkouts
3. Acts of books
   1. To lend
   2. Make a reservation
   3. refuse
4. Definition of book list
   1. Title
   2. Author
   3. Publisher
   4. Language
   5. Category
   6. ISBN
   7. Condition
   8. Add book
   9. Editing a book
   10. Search
   11. Description of book
5. User Interface
   1. View login window
   2. Add enter a name
   3. Add a password entry
   4. logo Applications
   5. Button to confirm login
   6. Button to register
   7. List of books
   8. Search filter
   9. **B**ook description
   10. Reserve books
   11. Edit profile
   12. Logout button
6. Administrator interface
   1. Adding books
   2. Adding book category
   3. Adding a book language
   4. List of books
   5. Book search
   6. Overview reservations
   7. Adding borrowing
   8. Overview loans
   9. Confirmation of registration of users
   10. Editing users
   11. Switching to the user mode
   12. Logout button
7. Development of diagrams
   1. Database diagram
   2. Flowchart of user interface
   3. Flowchart of administrator interface
   4. Class diagram
   5. State diagrams
8. Documentation
   1. Cover page
   2. Description of program
   3. Analysis
   4. Diagrams
   5. Database
   6. User manual
   7. System manual
9. Presentation of the project

## 2.2 Used developer languages

### 2.2.1 C# language

The Properties of the C # language:

* For the development of C# is used Visual Studio - the language in association with this environment is referred to as a Visual C#.
* Uses .NET framework.
* Compiled code is executed on the CLR (Common Language Runtime).
* C # is in many ways similar to Java.
* It is often seen as Microsoft's answer to Java.
* It has identified the need to transparently access the attributes and methods of objects.
* C # provides a mechanism labeled property - property.
* This mechanism allows separate internal representation of the attribute as it appears on the outside.

# 3. Process of Development Program and its components

## 3.1 Database

Our program uses a MySQL database to store and sort the data necessary for its operation.

The database is online on the server of the service provider WebSupport.sk.

Link: https://db55.websupport.sk/

The database consists of 12 tables, which are described in the sub-chapter 4.1 in a database diagram E-R.

## 3.2 Program Interface

The program is programmed in language C # in programming environment of the Visual Studio editor community. It consists of 17 forms, 18 classes. Each class has methods to interconnect methods or forms through variables. The forms, classes and methods are defined in Chapter 5.

The basis of the program is Connect2DB class. It includes methods for connecting to the online database which is describe in chapter 3.1. In the class Connect2DB, the main methods is method to obtain a user ID that is registered in the system. Using the user ID may be later in the program to get information from the database on his profile or borrow books or sort information about it. Data of the credentials can be obtained mainly trough user ID. They are to enter the system.

A second important method is the method to obtain the ID of the book. By ID book is included in the database. ID associates also features books.

## 3.3 Web Interface

The web interface is programmed in asp.net & C#, JavaScript and jQuery programming environment of the Visual Studio community and it is used Freamwork .NET 4.5.2. It consists of 8 +/- forms and one master page. The configuration file of the website and data to connect to the MySQL database resides in file Web.config Webpage use sessions which detect the user logon and logoff. For styling pages it is to use bootstrap freamwork, and CSS.

# 4. Diagrams

## 4.1 Diagram of database

In the database have entities:

• Users

• Books

• Rentals

• Book details

• User Credentials

• User Details

• Penalty

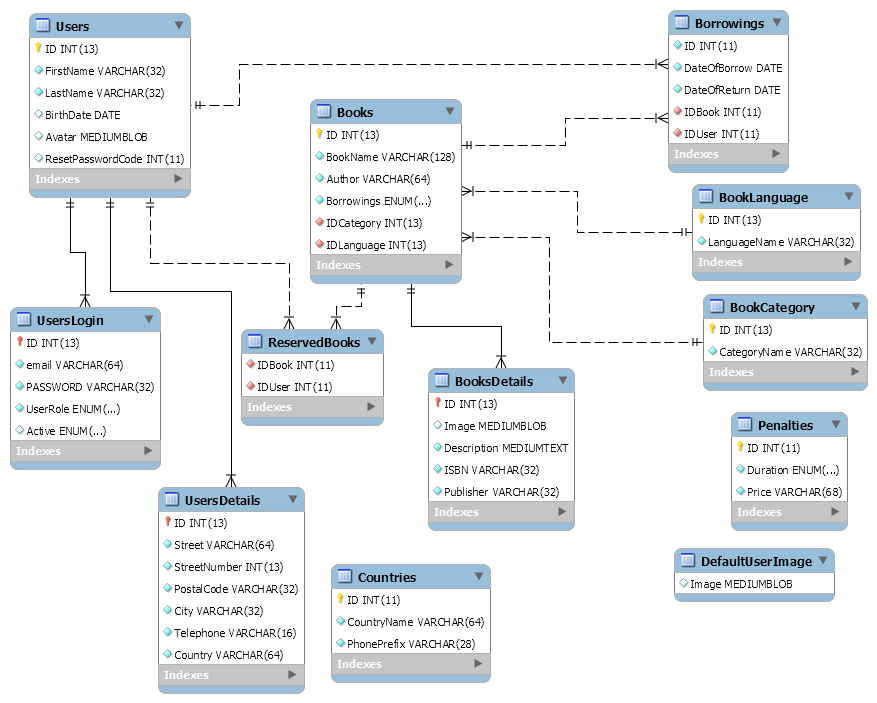
• Categories of books

• Reserved books

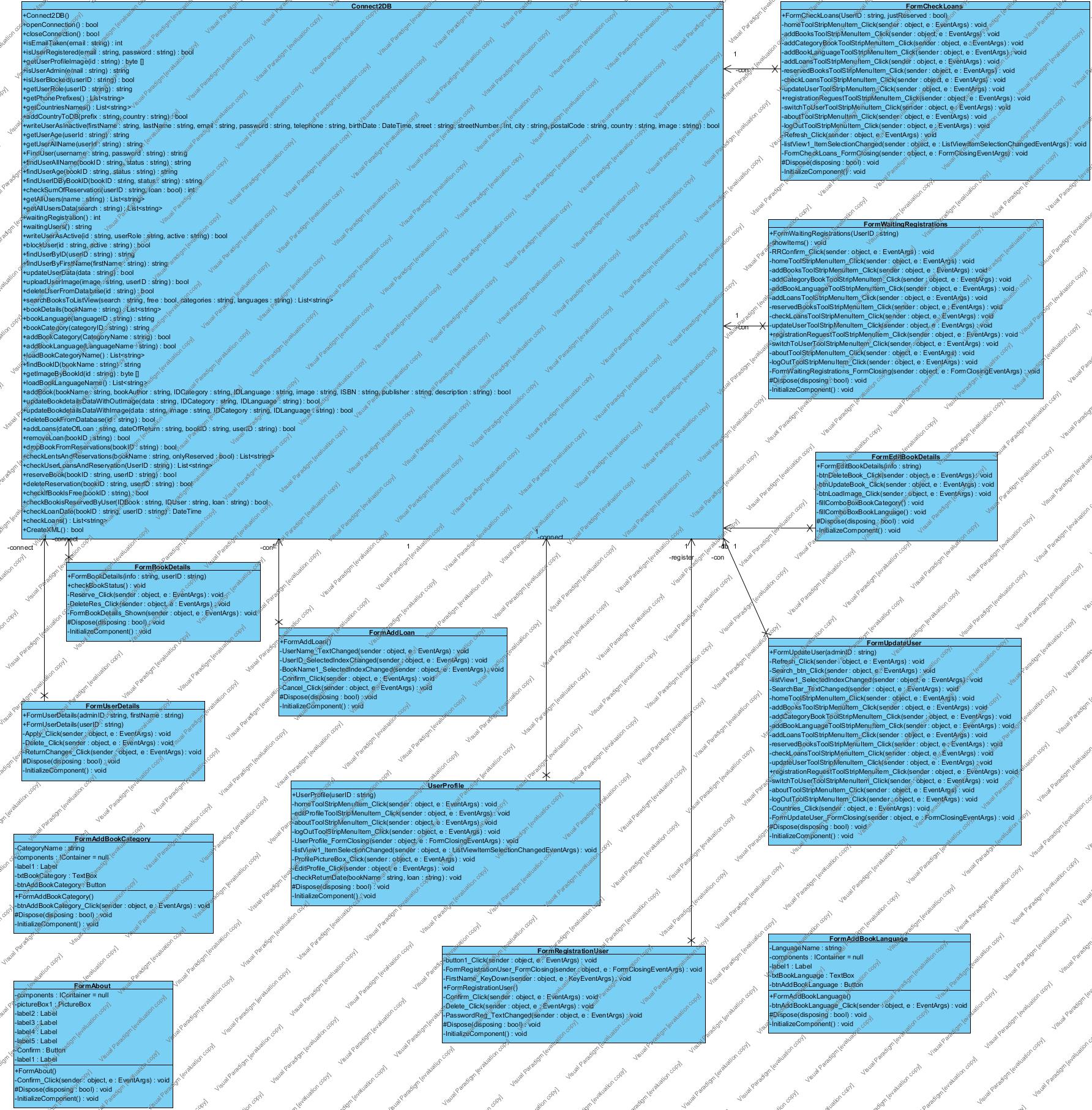
• Countries

• Book language

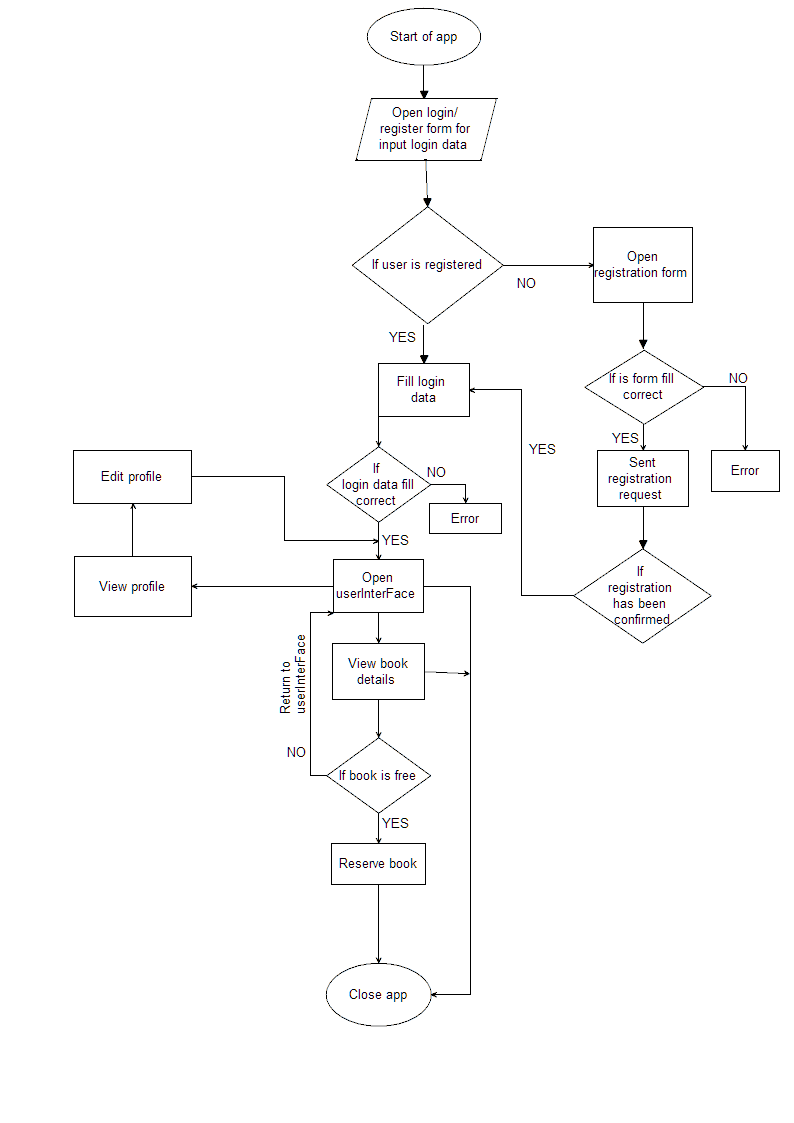
• Basic Image.



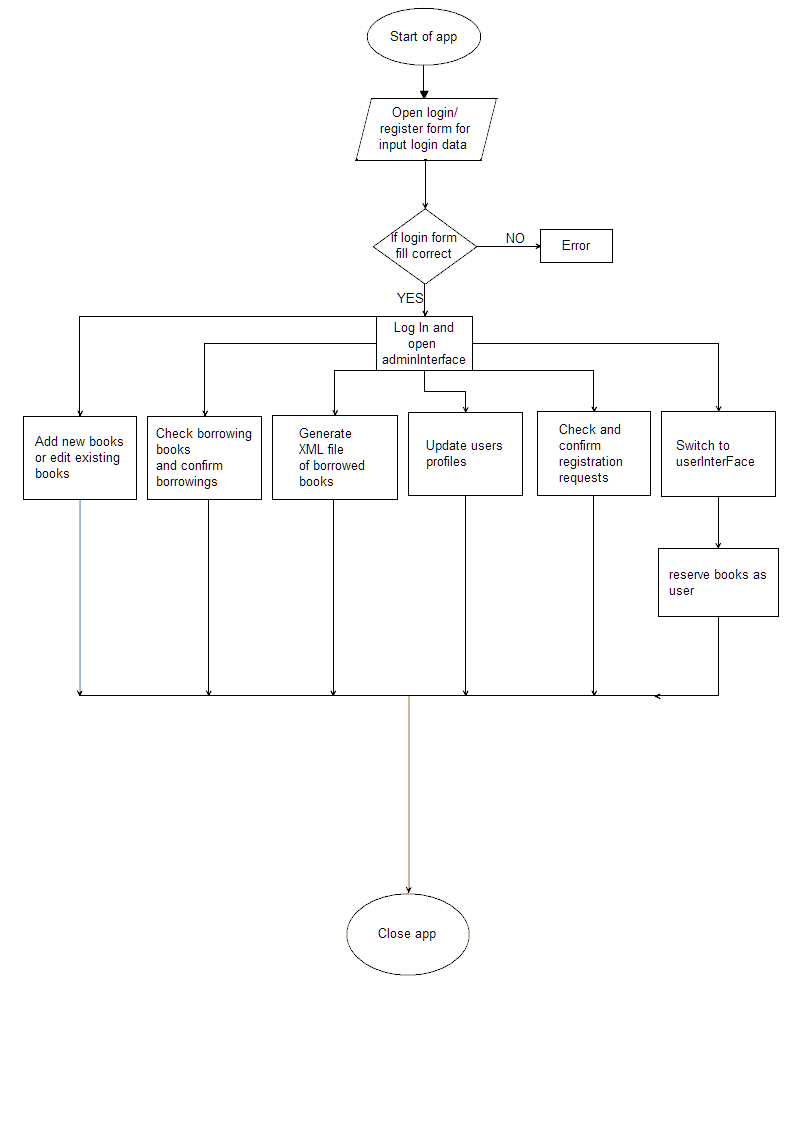
# 4.2 Class diagram



## **4.3 Flowchart of user**

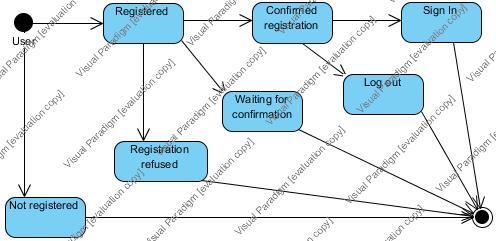


## 4.4 Flowchart of administrator

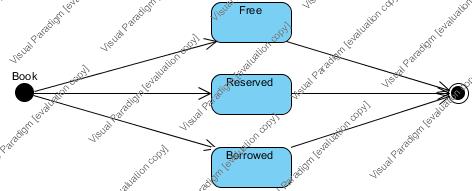
****

## **4.5 State diagram**

### **4.5.1 State diagram of user**



### 4.5.2 State diagram of book



# 5 Forms, classes and methods

Summary

|  |  |
| --- | --- |
| Name | Description |
| [FormEditBookDetails](#rV9qRKqGAqAAZBC6) | form for editing bookdetails |
| [FormAddLoan](#rV9qRKqGAqAAZBC8) | for for adding new borrowing of book |
| [FormAddBookLanguage](#rV9qRKqGAqAAZBC.) | form for adding new book Language |
| [FormAddBookCategory](#rV9qRKqGAqAAZBDA) | form for adding book category |
| [Connect2DB](#rV9qRKqGAqAAZBDE) | prototype of database connection |
| [UserProfile](#rV9qRKqGAqAAZBCo) | user profile form with user information and photo and list of borrowings |
| [FormAbout](#rV9qRKqGAqAAZBDC) | form about authors |
| [FormChangePassword](#rV9qRKqGAqAAZBC2) | form for changing password |
| [FormCheckLoans](#rV9qRKqGAqAAZBCy) | form for checking list of borrowings |
| [FormRegistrationUser](#rV9qRKqGAqAAZBCw) | form for register user to system, only send registration request |
| [FormChangePenalties](#rV9qRKqGAqAAZBC0) | form for changing of penalties |
| [FormUpdateUser](#rV9qRKqGAqAAZBCu) | form for updating user by admin |
| [FormUserDetails](#rV9qRKqGAqAAZBCs) | form with user details |
| [FormBookDetails](#rV9qRKqGAqAAZBC4) | form for showing details of book |
| [FormWaitingRegistrations](#rV9qRKqGAqAAZBCq) | form for validation of user registration |
| FormAdminInterFace | for administrator work interface |
| FormUserInterFace | for user work interface |

###### Details of Classes

Class - FormChangePenalties

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormChangePenalties | Constructor return string adminID |
| Confirm\_Click | Button which confirms panalty change  parameters: "sender", "e" |
| InitializeComponent | Required method for Designer support - does not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false |

Class - FormUserDetails

Operations

|  |  |
| --- | --- |
| Name | Description |
| Active\_KeyDown | If user types „Enter“ key, this method call confirmation |
| FormUserDetails | Constructor return string adminID, string firstName |
| FormUserDetails | Constructor return string userID |
| Timer1\_Tick | Set lasting of labels visibility |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false. |

Class - FormAddBookCategory

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormAddBookCategory | Constructor |
| BtnAddBookCategory\_Click | Button to confirm adding new book  Parameters: "sender", "e" |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false |

Class - FormBookDetails

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormBookDetails | Constructor return string info |
| CheckBookStatus | Check status of book |
| Reserve\_Click | If it is alloved(show), user can reserve book  Parameters: "sender", "e" |
| Timer1\_Tick | Set lasting of labels visibility |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false |

Class - FormAddBookLanguage

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormAddBookLanguage | Constructor |
| BtnAddBookLanguage\_Click | Confirm button for adding new language of book  Parameters: "sender" , "e" |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false |

Class - FormRegistrationUser

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormRegistrationUser | Empty constructor |
| Confirm\_Click | Button after click save the information to array, then, this information will be send to admin for confirm  Parameters: "sender" , "e" |
| Delete\_Click | Button after click remove all information and set the dateTimePicker to 1.1.2012  Parameters: "sender" , "e" |
| PasswordReg\_TextChanged | Set the looks of char at password textbox as \*  Parameters: "sender" , "e" |
| Timer1\_Tick | Set lasting of labels visibility |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false. |

Class - FormChangePassword

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormChangePassword | Constructor return string userID |
| Timer1\_Tick | Set lasting of labels visibility |
| OldPassword\_KeyDown | Call confirm function |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  <param name="disposing">true if managed resources should be disposed; otherwise, false.</param> |

Class - FormEditBookDetails

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormEditBookDetails | Constructor string info |
| BtnDeleteBook\_Click | Button for deleting book from DB  Parameters: "sender" , "e" |
| BtnUpdateBook\_Click | Button for updating of details of book  Parameters: "sender" , "e" |
| BtnLoadImage\_Click | Button for uploading image of book  Parameters: "sender", "e" |
| FillComboBoxBookCategory | This method will fill combobox with all categories saved in database |
| FillComboBoxBookLanguage | This method will fill combobox with all languages saved in database |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false |

Class - UserProfile

Operations

|  |  |
| --- | --- |
| Name | Description |
| UserProfile | Constructor return string userID |
| ListView1\_ItemSelectionChanged | Method for checking on to check which item of listview was clicked and after that sent information to form bookDetails and show them in form |
| ProfilePictureBox\_Click | Call formChangeUserProfilPhoto |
| ChangePassword\_Click | Call form ChangePassword |
| EditProfile\_Click | Call form UserDetails |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false. |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |

**Class - FormWaitingRegistrations**

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormWaitingRegistrations | Constructor string UserID |
| RRConfirm\_Click | confirming of registration  Parameters: "sender" ,"e" |
| Timer1\_Tick | Set lasting of labels visibility |
| FormWaitingRegistrations\_KeyDown | When enter is pressed call confirm |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false |

Class - FormAbout

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormAbout | Constructor |
| Confirm\_Click | Confirm button click  Parameters: "sender" , "e" |
| InitializeComponent | Required method for Designer support - do not modify  the contents of this method with the code editor. |
| Dispose | Clean up any resources being used.  Parameters: "disposing" > true if managed resources should be disposed; otherwise, false |

Class - FormUpdateUser

Operations

|  |  |
| --- | --- |
| Name | Description |
| FormUpdateUser | Constructor return string adminID |
| Refresh\_Click | Reload all data in form |
| Search\_btn\_Click | Call search function by set parameter |
| ListView1\_ItemSelectionChanged | Method for checking on which item of listview was clicked and after that sent information to form bookDetails and show them in form |
| Countries\_Click | Call form for add countries |
| SearchBar\_KeyDown | Call search function after enter is pressed |
| Dispose | Clean up any resources being used.  Parameters"disposing" > true if managed resources should be disposed; otherwise, false |
| InitializeComponent | Required method for Designer support - do not modify the contents of this method with the code editor. |

Class - Connect2DB

Operations

|  |  |
| --- | --- |
| Name | Description |
| Connect2DB | Constructor set the detail of connections |
| OpenConnection | Return open connection to DB |
| CloseConnection | Close DB connection |
| IsEmailTaken | Verification if user is already registered  Parameters: "email" |
| IsUserRegistered | Verification, if user is registered  Parameters: "email" , "password" |
| GetUserProfileImage | Read user´s profile image from db and load to profile page  Parameters: "id" |
| IsUserAdmin | Verification, if user is admin, or not  Parameters: "email" |
| IsUserBlocked | Verification if is user blocked  Parameters: "userID" |
| GetUserRole | Verification if user is admin or not according his id  Parameters: "userID" |
| GetPhonePrefixes | Adding phone prefixes to combobox |
| GetCountriesNames | Adding countries to combobox |
| AddCountryToDB | Adding prefix to DB  Parameters: "prefix" ,"country" |
| GetDefaultImage | Get field of image |
| IsCodeTaken | Parameters: "code" |
| SetPasswordCode | If you forgot password and choose forgotten password function, system send you your reset password which was generate by registration |
| WriteUserAsInactive | After users registration, his data will be saved to database, and admin must confirm, or refuse his request  Parameters: "firstName" , "lastName" , "email" , "password", "telephone" , "birthDate" , "street" , "streetNumber" , "city" , "postalCode" , "country" , "image" , "code" |
| GetUserAge | Method, that found user age for his profile  Parameters: "userId" |
| GetUserAllName | Method, that found user first and last name for his profile  Parameters: "userId" |
| FindUser | Method, that found user id for next works  Parameters: "username" , "password" |
| FindUserAllName | This find user all name thanks entity ReservedBooks  Parameters: "bookID" , "status" |
| FindUserAge | This find user age thanks entity ReservedBooks  Parameters: "bookID" , "status" |
| FindUserIDByBookID | Find user id by book id in entity ReservedBooks  Parameters: "bookID" , "status" |
| CheckSumOfReservation | Automatically count how many reservations has user and if he has some number, he can't borrow more books  Parameters: "userID" , "loan" |
| GetAllUsers | Find all user ID according their name  Parameters: "name" |
| GetAllUsersData | Find all users IDs  Parameters: "search" |
| WaitingRegistration | If there are some registration requests, this method find out how many there are |
| WaitingUsers | If there are some registration requests, this method find out details |
| WriteUserAsActive | If admin confirm user registration  Parameters: "id" , "userRole" , "active" |
| BlockUser | Admin can block user  Parameters: "id" , "active" |
| FindUserByID | Find details of one user by id  Parameters: "userID" |
| FindUserByFirstName | Find details of one user by his first name  Parameters: "firstName" |
| UpdateUserData | Update user info  Parameters: "data" |
| GetOldPassword | Get old password of user  Parameters: "userID" |
| ChangeUserPassword | Set new password  Parameters: "userID" , "password" |
| UploadUserImage | Enable upload user's profile image from his pc  Parameters: "image" , "userID" |
| DeleteUserFromDatabase | For deleting users from database, if users delete his account  Parameters: "id" |
| SearchBooksToListView | Searching bookname, author, status, category and language of books according search parameters  Parameters: "search" , "free, "categories" , "languages" |
| BookDetails | Find details about book according its name  Parameters: "bookName" |
| BookLanguage | Find book language according its id  Parameters: "languageID" |
| BookCategory | Find book category according its id  Parameters: "categoryID" |
| AddBookCategory | Add Book Category To database  Parameters: "CategoryName" |
| AddBookLanguage | Add Book Language To database  Parameters: "LanguageName" |
| LoadBookCategoryName | Load CategoryName from database |
| CategoriesForFilter | Load categories for Filter |
| LanguagesForFilter | Load languages for Filter |
| FindBookID | Find book ID according it's name  Parameters: "bookName" |
| GetImageByBookId | Get bookimage from database and show it in bookdetailsform  Parameters: "id" |
| LoadBookLanguageName | Load LanguageName from database |
| AddBook | Add book to database  Parameters: "bookName" ,"bookAuthor" , "IDCategory" , "IDLanguage" ,"image" ,"ISBN" , "publisher" , "description" |
| UpdateBookdetailsDataWithOutImage | Update book details without by admin  Parameters: "data" ,"IDCategory" , "IDLanguage" |
| UpdateBookdetailsDataWithImage | Update bookdetails with image  Parameters: "data" , "image" , "IDCategory" , "IDLanguage" |
| DeleteBookFromDatabase | Delete book from DB , Parameters: "id" |
| AddBorrowing | Add borrowings into database  Parameters: "dateOfBorrow" , "dateOfReturn" , "bookID", "userID" |
| RemoveBorrowing | Deleting borrowings  Parameters: "bookID" |
| DropBookFromReservations | Deleting book from entity ReservedBooks after succesful lent  Parameters: "bookID" |
| CheckBorrowingsAndReservations | Return list of lent and reservationd details  Parameters: "bookName" , "onlyReserved" |
| CheckUserBorrowingsAndReservation | Show all borrowings in admin part for checking borrowings  Parameters: "UserID" |
| ReserveBook | Reserve book by User  Parameters: "bookID " , "userID" |
| DeleteReservation | Delete reservation of book  Parameters: "bookID" , "userID" |
| CheckIfBookIsFree | Check state of book, if it is free  Parameters: "bookID" |
| CheckBookisReservedByUser | Check if is book reserve  Parameters: "IDBook" , "IDUser" , "loan" |
| CheckBorrowingDate | Finds date of return one loan, for setting penalty  Parameters: "bookID" , "userID" |
| CheckBorrowings | Finds and returns all loans to check, if there are any loans longer than 2 months to block these users |
| CreateXML | Generate XML file |
| GetPenalty | Get penalties to user if is some  Parameters: "caseOfPenalty" |
| GetPenalties | Displej penalties from DB |
| UpdatePenalties | Update penalties  Parameters: "penalties" |