Library Management System

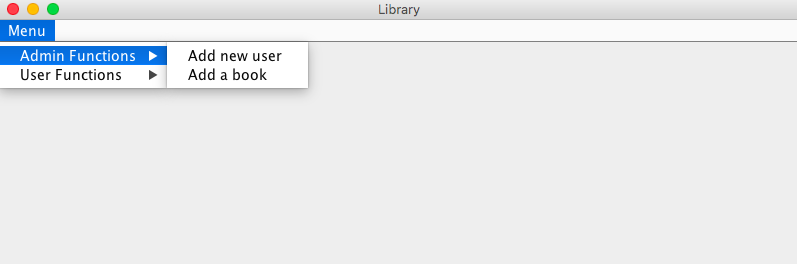
Design Document V1.0

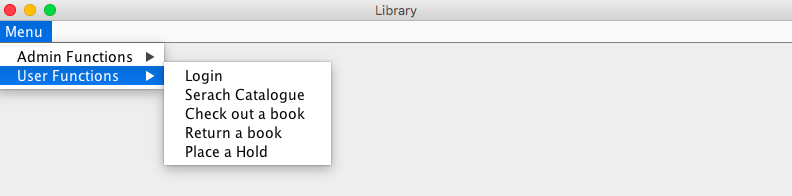
2/27/18

1.0 Introduction

This is a comprehensive design document to implement a Library Management system. It will provide user interfaces, UML diagrams, RDBMs design, An overall architectural picture and finally break down of the components that needs to be built.

2.0 UI interface Design





3.0 UML Design

3.1 Class Diagrams

|  |
| --- |
| **Book** |
| * id: String * title: String * author: String * dueDate: Date * borrowdBy: User * holds : List |
| + book ( id: String, title: String, author: String) : Book  + issue ( user: User) : Boolean  + returnBook ( ) : User  + renewBook ( user: User) : boolean  + placeHold ( hold: Hold) : void  + removeHold ( userId: String) : Boolean  + hasHold ( ) : Boolean  + getBorrower ( ) : User  + getId ( ) : String  + getTitle ( ) : String  + getAuthor ( ) : String  + getDueDate ( ) : Calendar |

|  |
| --- |
| **Book** |
| * id: String * title: String * author: String * dueDate: Date * borrowdBy: User * holds : List |
| + book ( id: String, title: String, author: String) : Book  + issue ( user: User) : Boolean  + returnBook ( ) : User  + renewBook ( user: User) : boolean  + placeHold ( hold: Hold) : void  + removeHold ( userId: String) : Boolean  + hasHold ( ) : Boolean  + getBorrower ( ) : User  + getId ( ) : String  + getTitle ( ) : String  + getAuthor ( ) : String  + getDueDate ( ) : Calendar |

|  |
| --- |
| **Hold** |
| - book: Book  - user: User  - date: Date |
| + getBook ( ) : Book  + getUser ( ) : User  + getDate ( ) : Calendar  + hold ( user: User, book: Book, date: Calendar) : Hold |

|  |
| --- |
| **Library** |
| - users: UserList  - books: BookList |
| + getBooks ( userId: String) : Iterator  + addBook (title: String, author: String, id: String) : Book  + searchUser ( userId: String) : User  + addUser (name: String, address: String, phone: String) : User  + issueBook (bookId: String, userId: String) : Book  + renewBook (bookId: String, userId: String) : Book  + returnBook (bookId: String) : int  + removeBook ( bookId: String) : int  + placeHold (bookId: String, userId: String, duration: int) : int  + removeHold ( bookId: String, userId: String) : int  + processHold ( bookId: String) : User  + getTransaction ( userId: String, date: Calendar) : Iterator |

|  |
| --- |
| **User** |
| - name: String  - address : String  - phone: String  - booksOnHold: List |
| + user ( name: String, address: String, phone: String) : User  + getName ( ) : String  + setName ( name: String) : void  + getId ( ) : String  + setId ( id: String) : void  + getAddress ( ) : String  + setAddress (address: String) : void  + getPhone ( ) : String  + setPhone ( phone: String) : void  + issue ( book: Book) : boolean  + returnBook (book: Book) : Boolean  + renew ( book: Book) : Boolean  + placeHold ( hold: Hold) : void  + removeHold ( bookId: String) : Boolean |

|  |
| --- |
| **UserList** |
| - users : Live |
| + search ( userId: String) : User  + getUsers ( ) : Iterator  + insertUser ( user: User) : boolean |

3.2 Sequence Diagrams

4.0 Database design

|  |  |  |
| --- | --- | --- |
| Table Name | Book | |
| Column | DataType | Null |
| Id | String | Not Null |
| Title | String | Not Null |
| Author | String | Not Null |
| Library\_id | String | Not Null |
| Borrowd\_User\_Id | String | Null |
| Due\_date | Date | Null |
|  |  |  |
| Table Name | User | |
| Column | DataType | Null |
| Id | String | Not Null |
| Name | String | Not Null |
| Address | String | Not Null |
| Phone | String | Null |
|  |  |  |
| Table Name | Hold | |
| Column | DataType | Null |
| Book\_Id | String | Not Null |
| User\_Id | String | Not Null |
| Date | Date | Not Null |
|  |  |  |
| Table Name | Library | |
| Column | DataType | Null |
| Id | String | Not Null |
| Name | String | Not Null |
| Address | String | Not Null |
| Phone | String | Null |
|  |  |  |

5.0 Components design