

**Library Management System**

**Submitted by:**

**U1810149** – Juraeva Sabokhat

**U1810151** – Mardonova Mubina

**U1810171** – Isroilov Tukhtamurod

**U1810185** – Abduraupov Uktamjon

**U1810200** – Makhsudov Jasur

**U1810203** – Mukhamedov Bekzod

**Team’s GitHub URL\*:**

https://github.com/jasurmakhsudov/Library-management-system

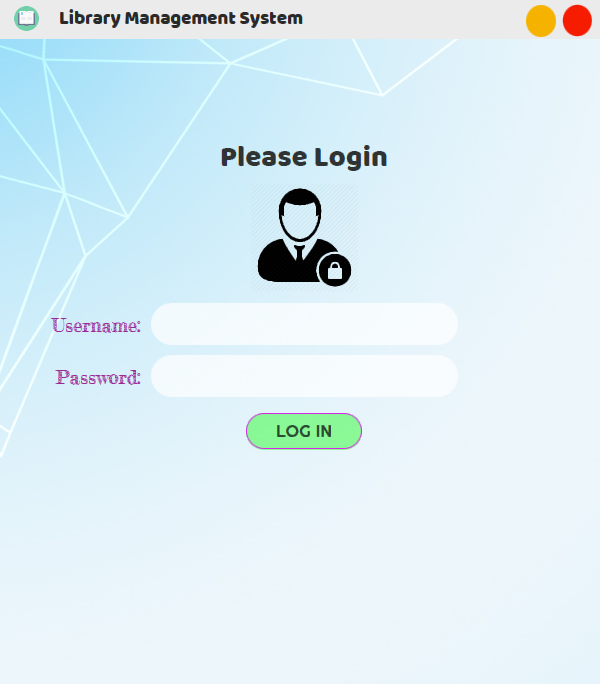
**Contents**

1. Description of Implemented Functionality………3
2. User Accounts…………………………………...10
3. UML Diagrams………………………………….11
4. Entity-Relationship Diagram……………………13

**Description of Implemented Functionality**

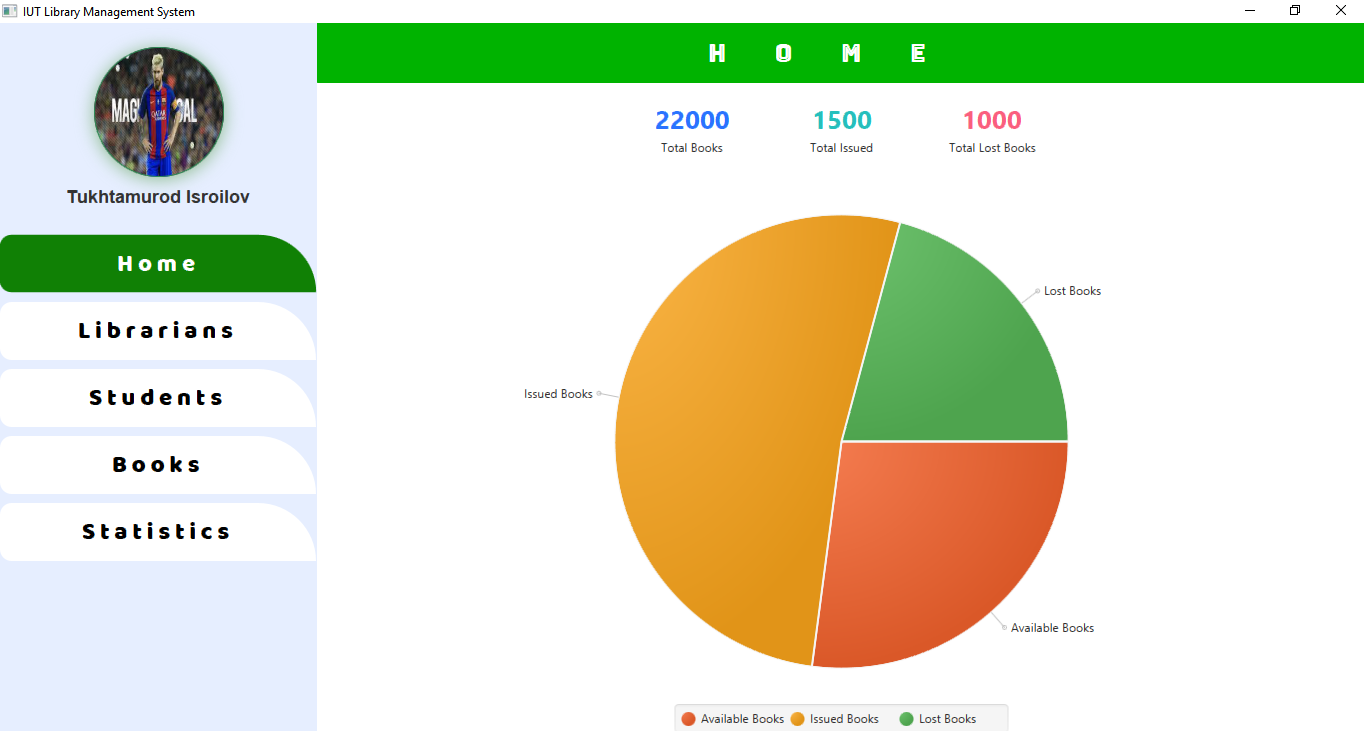
**Login Page**

We started our project with creating login page to our LMS. We can see that in our **loginPage** package we created 3 files which can support us for logging in to appropriate user interface. More precisely, in this pane we can log into 3 different types of users, namely Administrators, Librarians and Students by writing their unique ID and password. For implementing this functionality, we used IDs and passwords from our Database as primary keys. After typing them we can go into 3 different kinds of main windows according to their role, because every users’ actions are different from each other. If you want to know who was responsible to do this task, please go to the our team’s GitHub URL\*.



**Main Page**

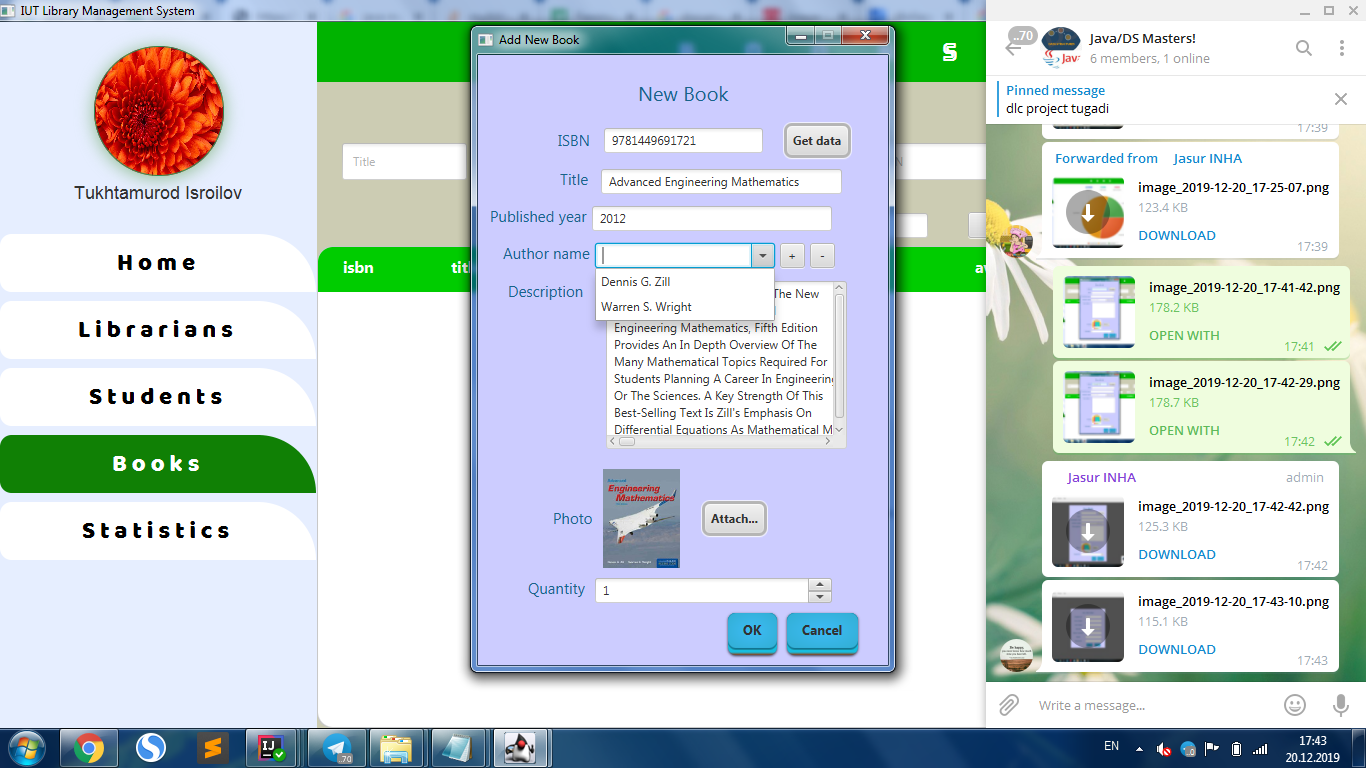
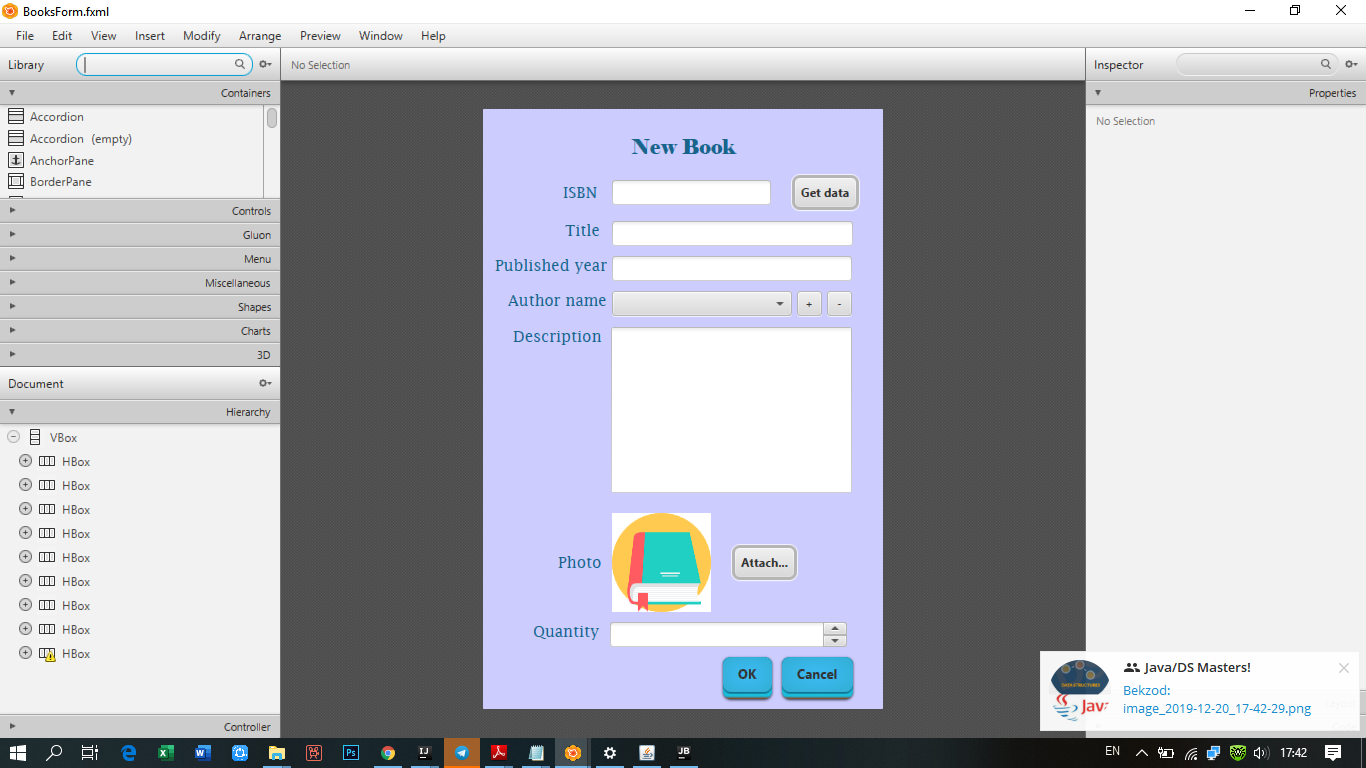
Consequently, we created **mainPage** package for showing all users’ functionality. For example, for administrators, librarians and students separately and they have different rights or functions. After logged in we go to main window where we can see ‘home’ by default for all of them. For example, if we log in as an admin we get the table with menu which contains “Librarian”, “Students”, “Books” to add new librarians, students and books respectively. If we logged in as librarian we get the menu which contains “Students”, “Books” to add new students and books respectively. Finally, if we logged in as a student librarian we get the menu which contains “Books” to see and choose or issue books. Also all of the menus for Admins, Librarians, Students have option “Statistics”. If you want to know who was responsible to do this task, please go to our team’s GitHub URL\*.

1. **As an admin:** 

From the menu by pushing **“Librarian”** button, admin goes to the new window and can add new librarians to the database by inserting their data. For this, admin has to fill the form where should be new librarian’s ID, Last Name, First Name, Phone Number and email as well. Also, admin can choose any librarian and delete from list.

By pushing **“Students”** button, admin goes to the new Student window where he/she can add and delete new, old students respectively. Student window requires to fill the table and if we want to add new student, we should insert student’s ID, Last Name, First Name, Faculty, Phone number and student’s email as well.

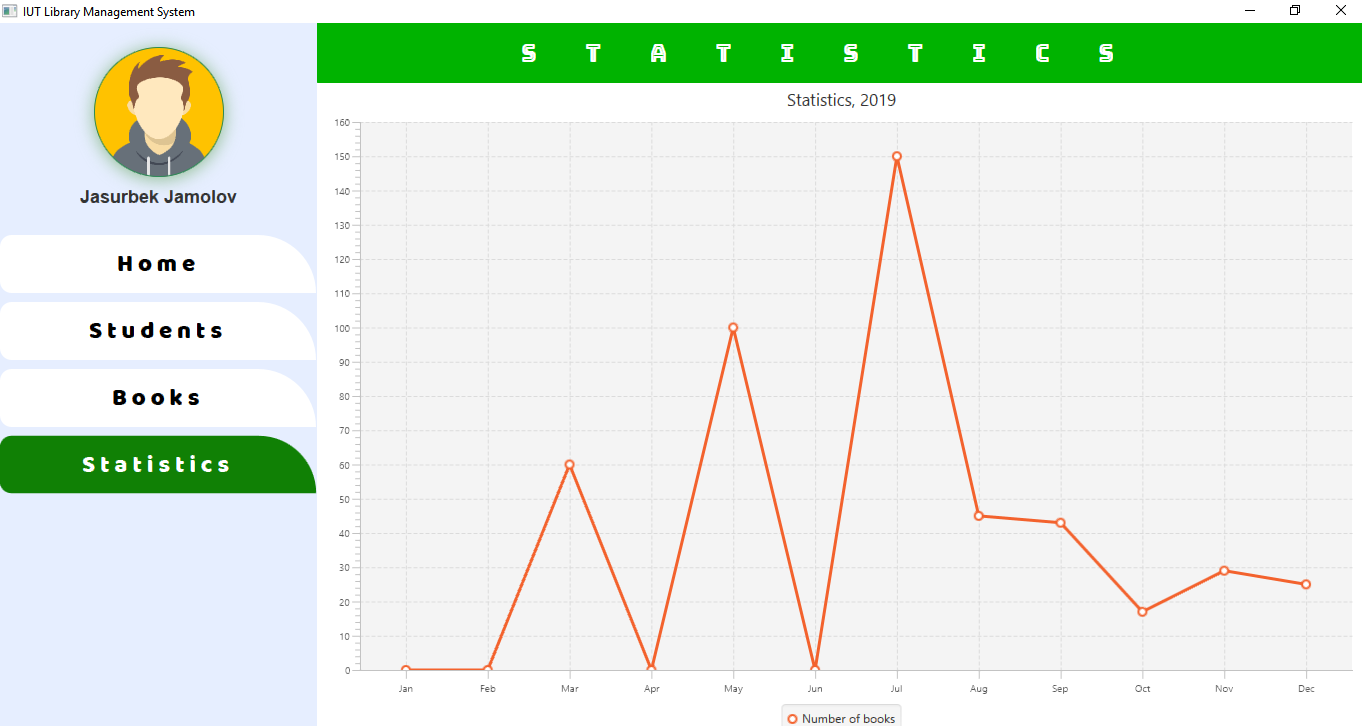
By pushing **“Books”** button, admin goes to the new Books window where all books’ information is concentrated. An admin can search the books, add new one, delete it and so on. In order to add new book, book table requires admin to fill the form where he/she should input all necessary data about that book. For example, Book’s ISBN, its name, author or authors, published year, photo and a brief description for book. The good thing is that if we want to add not just one, but several of the same book, we should just give a quantity and insert same type of several books at the same time.



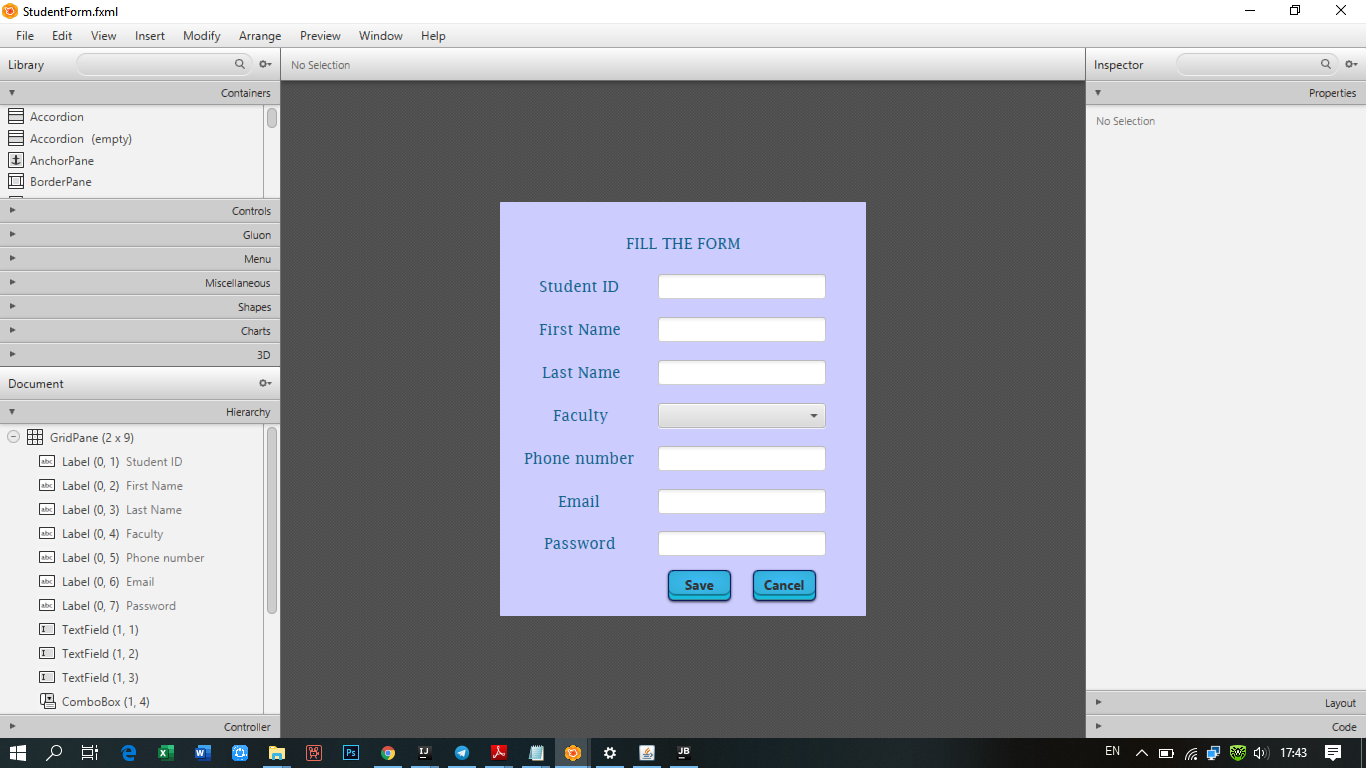
By pushing **“Statistics”** button, admin goes to the new window where all information about how many students issued the books, total number of books, total number of lost books and so on.

If you want to know who was responsible to do this task, please go to our team’s GitHub URL\*.

1. **As a librarian:**

****

From the menu by pushing **“Students”** button, librarian goes to the new Student window where he/she can add and delete new, old students respectively. Student window requires to fill the table and if we want to add new student, we should insert student’s ID, Last Name, First Name, Faculty, Phone number and student’s email as well.

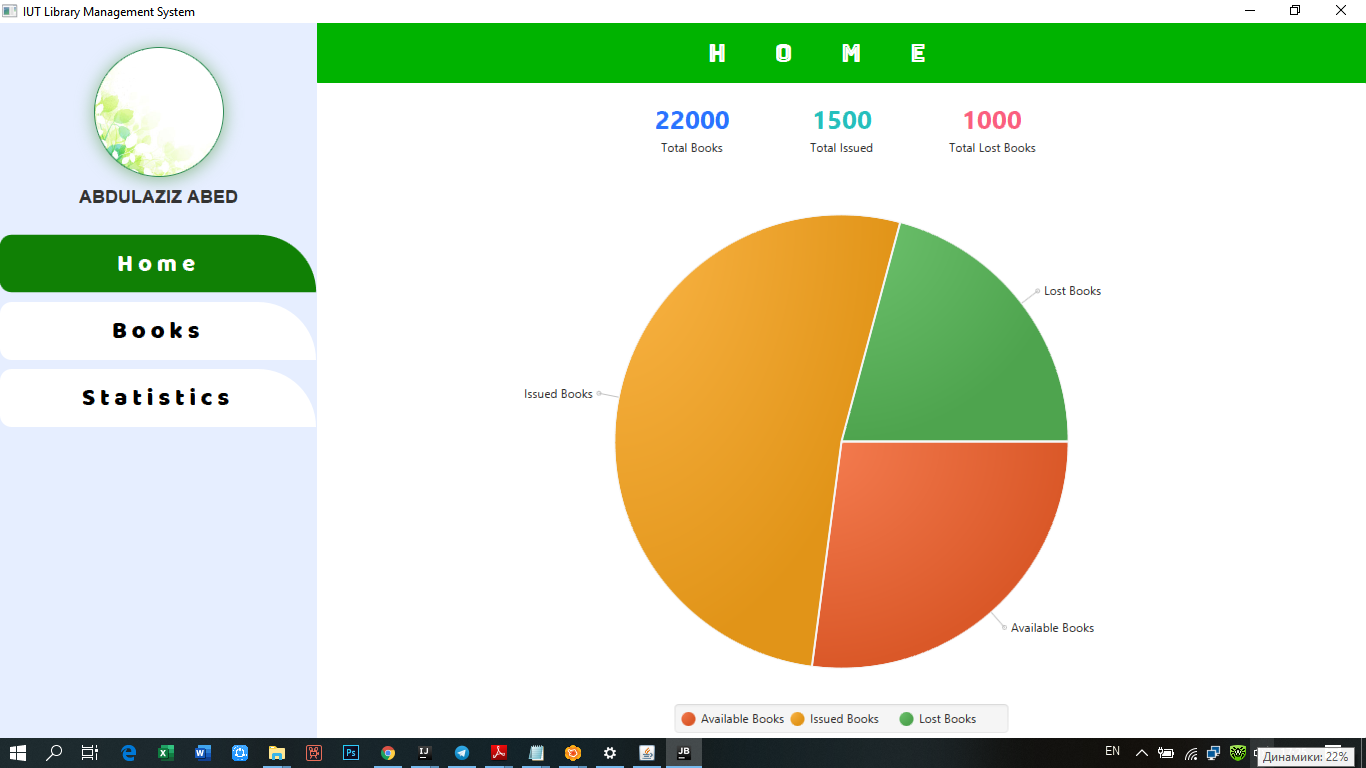


By pushing **“Books”** button, librarian goes to the new Books window where all books’ information is concentrated. A librarian can search the books, add new one, delete it and so on. In order to add new book, book table requires librarian to fill the form where he/she should input all necessary data about that book. For example, Book’s ISBN, its name, author or authors, published year, photo and a brief description for book. The good thing is that if we want to add not just one, but several of the same book, we should just give a quantity and insert same type of several books at the same time.

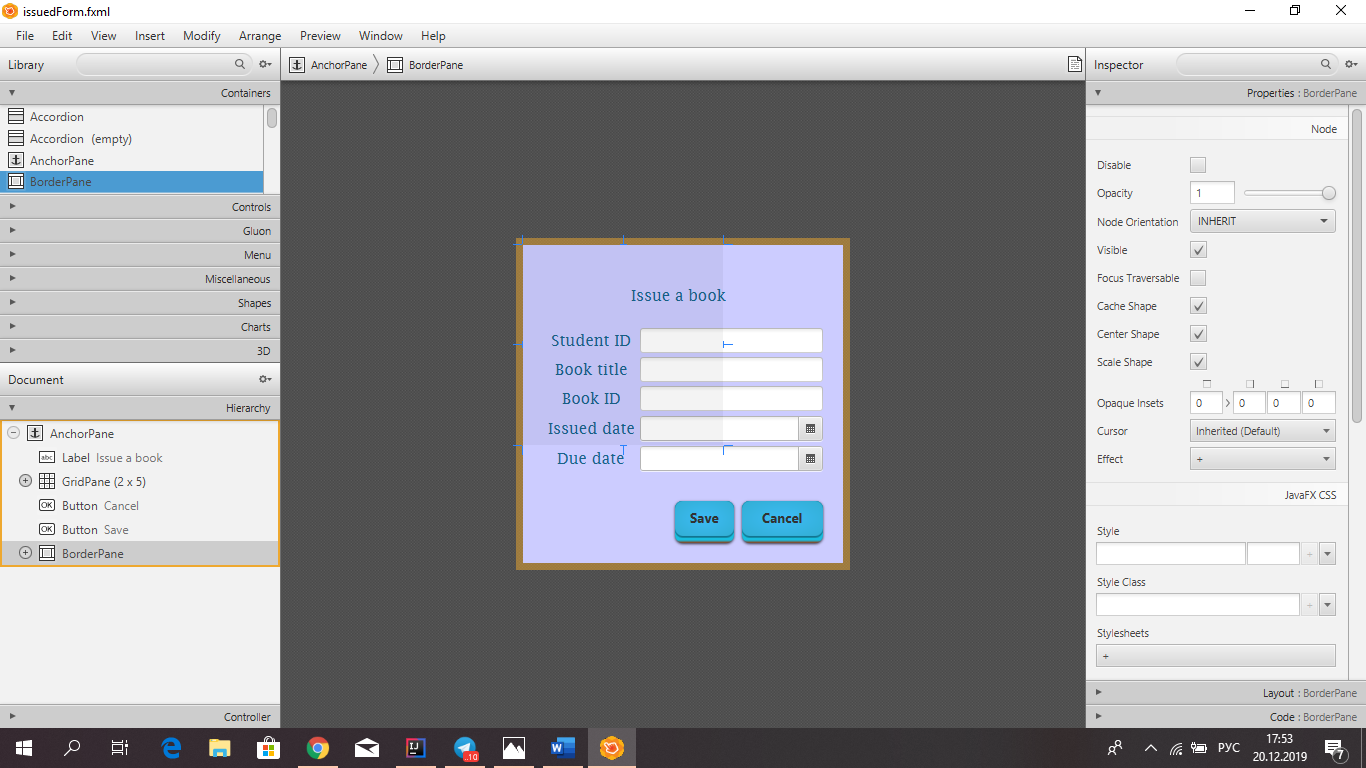
By pushing **“Statistics”** button, librarian goes to the new window where all information about how many students issued the books, number of students, what type of books are issued more and so on.

If you want to know who was responsible to do this task, please go to our team’s GitHub URL\*.

1. **As a Student:**

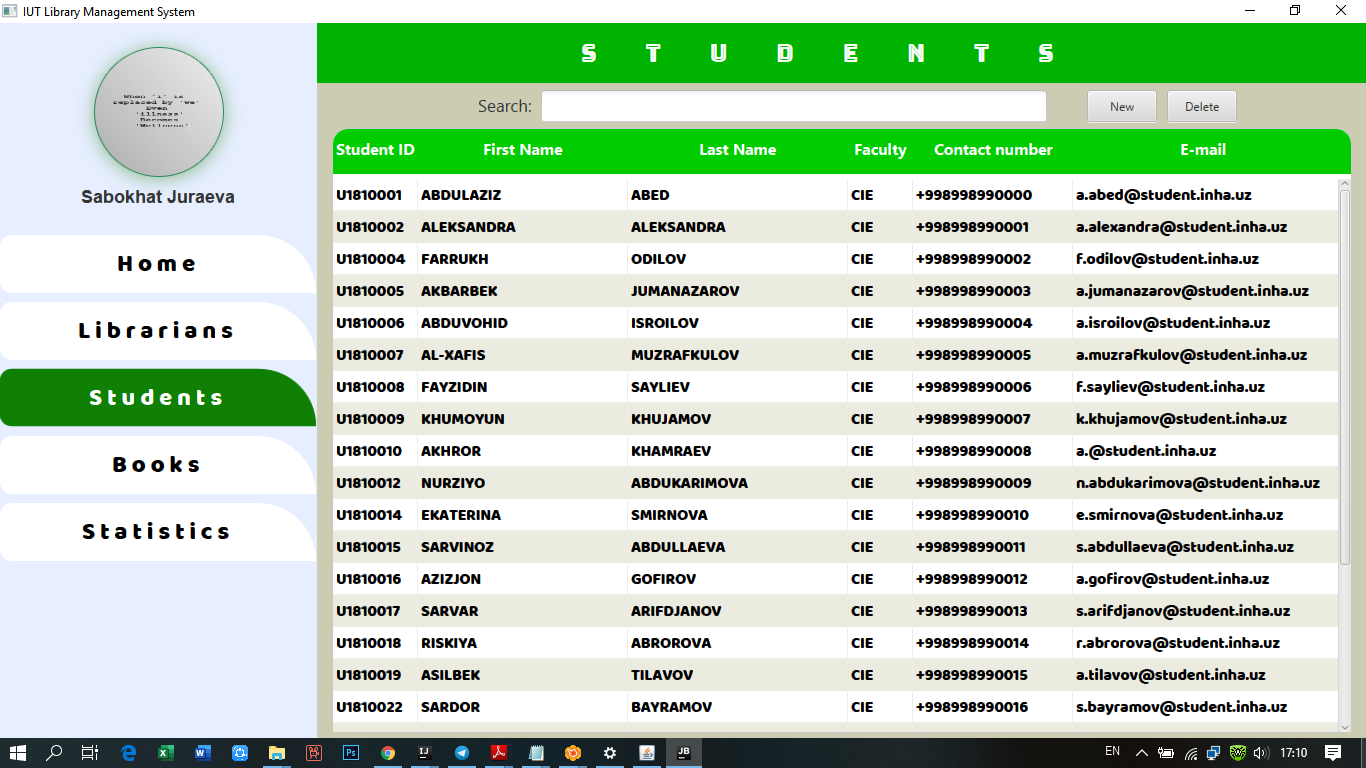


From the menu by pushing **“Books”** button, student goes to the new Books window where all books’ information is concentrated. A student can search the books, add new one, delete it and so on. In order to add new book, book table requires students to fill the form where he/she should input all necessary data about that book. For example, Book’s ISBN, its name, author or authors, published year, photo and a brief description for book. The good thing is that if we want to add not just one, but several of the same book, we should just give a quantity and insert same type of several books at the same time.



By pushing **“Statistics”** button, student goes to the new window where all information about how many students issued the books, number of students, what type of books are issued more and so on.

Next, we have created the package called **table** which includes the functions that read all information of students from database and display them in the form of table. For example, all students’ ID, First Name, Last Name, their Faculty, Contact Number and E-mail in separate columns respectively.



If you want to know who was responsible to do this task, please go to our team’s GitHub URL\*.

**Singleton Classes**

Also, we created **WindowLoader** singleton class whichfor loading each window. Inside the class we have 4 main functions, they are loadMainWindow() , loadAddNewUserWindow() and loadAddNewBookWindow(), loadLoginWindow() .

**DB Connector** singleton class for establishing connection between program and database.

**UserSession** singleton class in order to identify who is logging in.

If you want to know who was responsible to do this task, please go to our team’s GitHub URL\*.

**Challenges we came against and our solution for them**

* We had the challenges while we update and delete Student information from database. In URL we used relative path instead of absolute path. Fortunately, we solved this problem.
* We faced the challenge of keeping user information after login in. For this we got solution from the *stackoverflow* the link is here <https://stackoverflow.com/questions/46508098/how-to-keep-user-information-after-login-in-javafx-desktop-application> .

**User Accounts**

**Students:**

1. **Login: U1810001**

**Password: abd0000**

1. **Login: U1810002**

**Password: ale0001**

1. **Login: U1810004**

**Password: far0002**

**Librarians:**

1. **Login: U18012**

**Password: jaj6654**

1. **Login: U16102**

**Password: ult0502**

1. **Login: U17043**

**Password: zim1767**

**Administrators:**

1. **Login: U1810171**

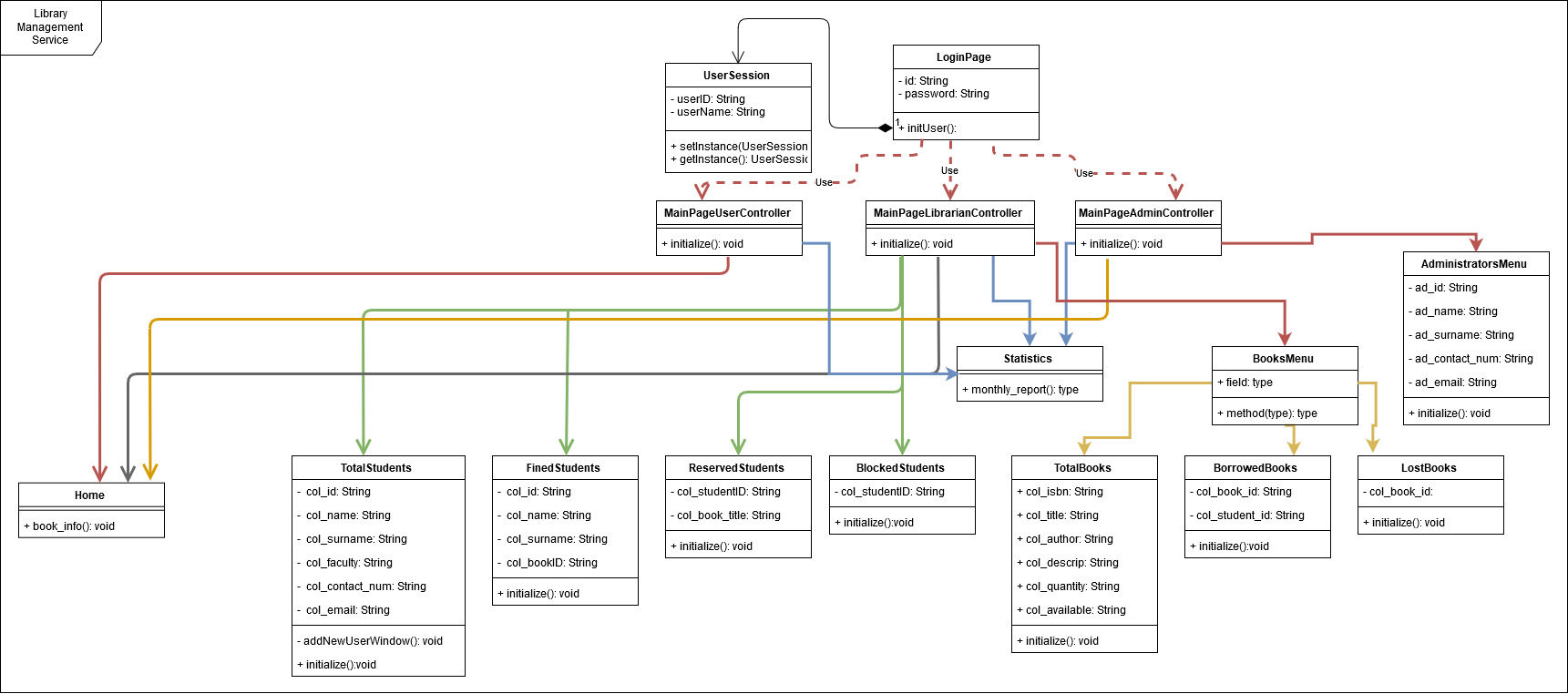
**Password: webhero171**

1. **Login: U1810151**

**Password: anibum801**

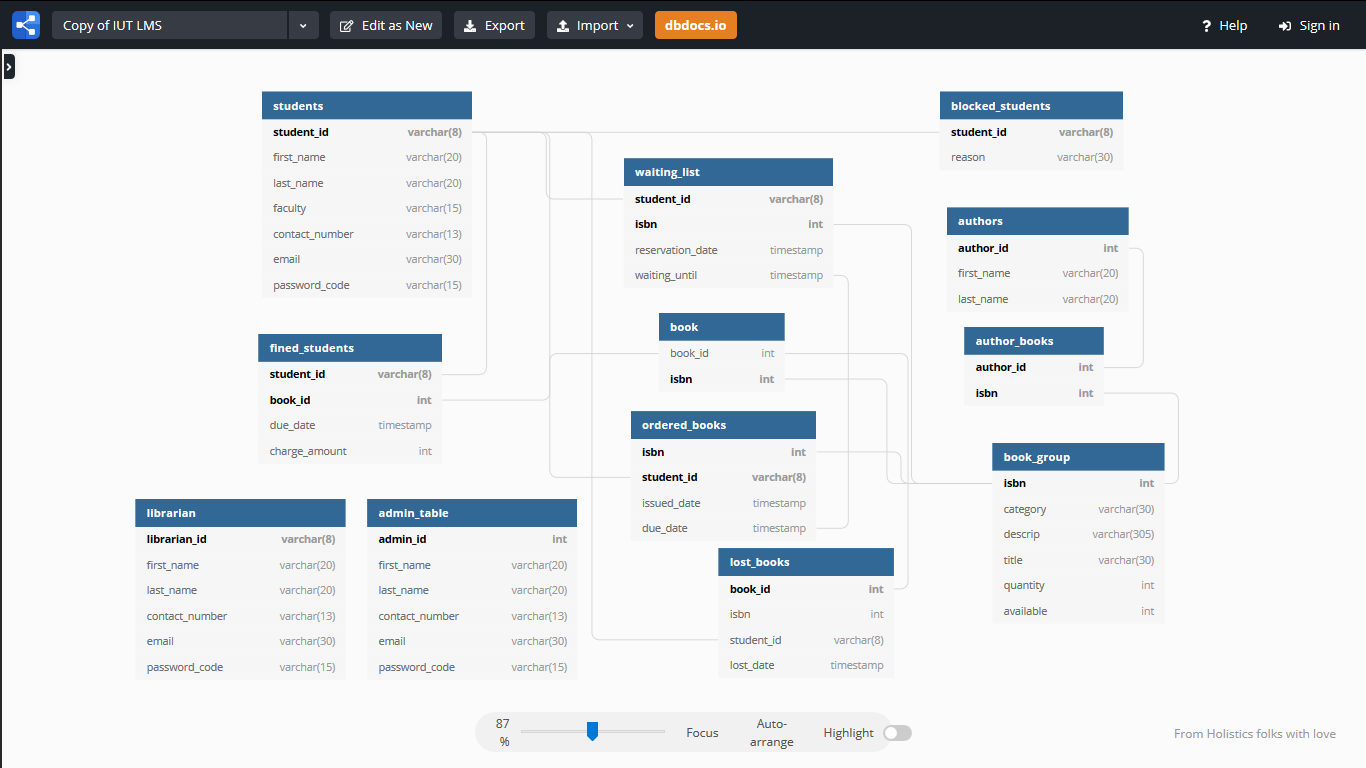
1. **Login: U1810149**

**Password: sabo0002**



**UML Diagram**

https://www.draw.io/?lightbox=1&target=blank&highlight=0000ff&edit=\_blank&layers=1&nav=1&title=Untitled%20Diagram.drawio#R7V1dc5u4Gv41meleJAOIz8s4ac%2FunHQnu%2BmZtuemI4Ns02LkA3IS99cf8SEbkDDYRpDOKr2oEQJjvc%2FzfumVuAJ369d%2FJXCz%2BogDFF0ZWvB6Be6vDMPUbIf%2Bl7XsihbDAlrRskzCoGjTDw1P4U9UNrJu2zBAaa0jwTgi4abe6OM4Rj6ptcEkwS%2F1bgsc1b91A5eIa3jyYcS3fg4DsipaXUs7tP%2BOwuWKfbOulWfm0P%2BxTPA2Lr%2FvygCL%2FK84vYbsXmX%2FdAUD%2FFJpAu%2BvwF2CMSk%2BrV%2FvUJQNLhu24roPLWf3z52gmPS54PGvH3f%2FXj8%2B%2FTf5vkrnf12%2FfMb314Zml09HdmxEUEAHqDzECVnhJY5h9P7QOst%2FNcruq9GjQ58HjDe0UaeN3xEhu1LacEswbVqRdVSeRa8h%2BVJenn3%2Bmn2%2Bcazy8P61cu5%2Bxw5ikuy%2BVHpmx1%2BrJw%2FX5UfswpQk%2BAe6wxFO8l8IbN9F88X%2BDJM7oC2LMIoqPQOI3IVP24tRyoamdfjLphRvEx8dGXO3RDlMlogc6QfcPUoo%2FRBeI%2Fqj6IUJiiAJn%2BsPAksaLPf9yktvkwTuKh02OIxJWrnzY9ZAO5SUtp0Sr3tCW1VY0Q%2FFHdlR5dEOTTn0ToBhOSTPMNqWP%2BsjDONHyr3bYB3Gd5jKE0cRSjiwpi%2FhOoJxhsoF7cVwm4EMRuEypp99KiF6JZg9o4SElPm35QmSoXXmr8IoeIA7vM2EkRJKa3Y0W%2BEk%2FElvCxlw6emElLA27FqPp%2BzKEnEJSmmfRwYOvdH0Eb7WOj7AlJQNPv2ZcJOG8%2F3PWFOchPEME4LXZadytOjPQa%2FH4cijh4nZbYiZ6eOXgxLU7bJtVVGArE0EuApITscAe4AKCDhpR2Eu6Tpt9QZtY5x3YuKP0IIIhL8OgyBXZukG%2BmG8%2FJSrrmv90PKQX5hrhbLl73IYQK4DCSSwEFMmkwjOUfSI05CEOLt%2FUvSdlXyjj2%2FNrqz7vCUhFNH0R8AwFxai4n9BGQT6SbadQry4mXjtftJ1ZQlXFwh3ll0V0wGjovqJ3v12BW6zLpjeqSl4OhyE0%2BSloE%2BXfUH8uphNXsxZE6bXLqLcbq8oYlAsEH1dxDM6rHfajZUJ27ijx%2FrheDL5A7Of%2FBlOBgeAoVmTOhoH5%2BJr5VSno%2FEr%2Bxm6MZajcZlqMFqN%2F0M4T2ASwpoDoHE4Uj7AeT6AafXwAVxrTB%2FAVD7ABTagoNJJToBIvNKcAEs5AXKdgA4AiLwAEQCkeQG63arr%2F5OiRKl5GWp%2Bn4F6O6EeH%2B8rNX8Cy%2B23Het5Ss1LVvPHATB9sMe79L%2FTWyuNPpBGvzYb2TtL924cfWKtbijn%2FRJSG6c772NqdaPNeZ9j%2FONbGC%2BwUuoy5T%2B5Utc9l5PqZFOFvWcKL0vgBfbctuw%2BCbzFYmH4AyfwAJsP70rg2S3mQu5MoWPWjZCpeVWQdfd3tKP9be9ofzkzkYbGB56j5qnPSlNfOB8%2BMcy1vjC3LoT5RfoP8POTM2r60o8o3irPdiDP1gV1zgOnZ67ClWX2QNvM5SJEWalR5vGQ3YYPbpTH0%2BXxAE2MhRM9XmkeD%2BDDWBXQDCZey5g2oAGghdj09iscvMtI%2FZvitzwA2D1Vuzx%2B8wmLvNwszMaB4ESZ9iFNu641%2FH%2FD7qngdU9WygrwSY3r7HuCb2Fp2p9IQumnyH86%2Bc23bdx1i5%2BEKmUfwzVS0pcp%2Fb62X6L0%2BUmqUvrpNlEAkA0Ax50aADYfzpcA8DPj6pNv8XatQCATBNQjmBwFfHRfogCtYRgp%2BUuVP5g8AuDrkTgxqwj%2FbPlaE89ZAkcgXlWJMiIC7Kk1POC9fBo2Exrih37KCVzF9ufF9pZ1ZtpeWjGKybt3SrOfwOu3vZzMbJuUWVPyrKLdtwRl4%2FJO5W8lgmDyihSTn5%2F5hHPduaWjSpR6H6zeUG%2FUG5qeQL97ItkzGUlYUSoK4O%2BMq1vartGRUSncS%2FlfyO4kIyDEgLToXRMl8aoQUIk82SAQZXJHBoEoj18FgUrojoADUUJ3XBwI1hHXcbCA%2FjYiO4UDmTgQ5nRHBgJf2FEHgkrwjwUGUYJ3VDCYXVhQaX75KBClgcdFgVqZJFO%2Bbk8fUN7mQrx8i1m84E%2F0ki0s%2FxzGAX5RuX6pMOidE5BG8577Syj5S0kH6JOreT4S%2FEC1eqDygUNvHNQjGygs5GTTgRJMgKiSU2UDh6T%2FMFNCEuM%2BvphDZQPHBcH0dZ0GPzOosoGj42Dy8k6za2Yg27Xgj3uFApkomL6%2Bk5WlqMBfinz71m%2FKCvwttavwxAwHUyt6i0%2Fw%2Fo1SlDyrsG%2F4jQTBuXEfm5sbXvzizF%2Fu6BXiV0b%2BQhVQMOwNx32CMjCGgXxvKhKSSPn7UkEwedxniVK%2FytEbSLx9wzlpfp5omY7y88YDgDu1krf4lTyzCPs%2FlJc3fHJfP9PLk7YFU8tKfeXkDch%2F5207eYLl%2Bsq%2BDybeqbdiEqzFVxP3o0l%2F8n2YBIvw83U8%2BfaKyq4PZNftc%2B36vrBDwmxdm2Ofz9in81iZ9YuY37Yx8Fsx67pwAwYGAJW4kY6AyRM3euseDRkC4JasMgEqCMiDwORz9bpgj4YDBAKU%2Bkm4URiQiYHpZ%2Bp10PbWoAwE%2F9vCmFoDtXZLLgom349JF27bsbcGzzCMijFVMJAIg77bNskLCEVbM6tsz1DyFa3XGTXdI1qOpaZzRkRA79BfGsP5%2BdoZThL8ggKV9Rn0hRramVkfeTU7tqhOv1KvoVZqXMr%2BlvfkvJmsj3BVdm1CT4FAMggmT%2FzYoryP8vEGEu%2FUFTt2W0pHzeiNIP3Jy3UcXsE%2F4JQo125I185rvh9x8kIdp70au%2BrZKbqfSHfnjZdgO6r6VqJ4p67OcVT17bT8nrxAx%2BXNebarzhNK04wxyqAPY9Cb66tY0WNnqsaQtYu6KzboWyr8ZsktxR54vaUDwTr8ya2y33f5x6qIKh20Xog5wseL%2FQFT1m4srvIHLjEHHeLt%2FUpkWdJt8wdSRP6gYwFjH72r2Ieab1DRAdkly8olZb%2BaZal07%2Fz4D9Up8nDWM4ckbYt3Fs%2FWsgjU7j7CJf86B%2BV0nOd0mI2t%2FUFfp0OTZT088TvZmjMCNYdjA9P0BSdtXZRyOFk5FOR7uz6GZygfQ554p%2FYxvLaXs2c5h8xDYN6CYvbgop%2B8HlhQHKJzckZxcJsVjNAjvMkHe0XWzDrTk6Vt1g1mrFnvIIRrHAefVmHcsOO6yRo%2BhNH%2BVsESMWeCimOFlziG0ftDaxeE5qXBnqHXkHwpgZB9%2Fpp9ptIuju5fK6fud%2BwgpiP65dAxO%2FxaPXe4LD9i1xXDlT37cbeADineJj7qQUU6LtRbP8Od3L9EqO5nFEcJiiAJn%2BuPKYJUed%2FHjEEV76WRMtEt7Uar%2FoH6HYtfW97kgFPuvpZ32n2LweHuSzEHd5VuJf%2Fbf05jjRaryG17TK7%2F8e7gaHf6oXjeA3X3UryEzfxUMFXgPfjcIHEA0xUKSlbWuH4eRepeAWiYB%2Bo5Llwf%2BT5nS%2BiZuWuZ1mk062RPUW7Fs0ekhi8kjcXqbxgMmk5ab5Y4dTwxQHXRYjhs8YsGhsUWU9k3uufU1LZnm8cVNz14RElIf2IWcjaVOdDBKVjNj5q36w3XGYMrD%2Fkp7ESxnHQcpNvgxrG8%2FZ%2FbUOrmjamZe53OIohTadDM2xtaPxqcah2spvq2O6yDeWF%2F73h%2FwznaX5I9YV86gj2xXXc8ls6HYGm3kzYe93gX6jx6ucZI9Go%2BcAddbPfC%2Fu7x%2Fk7zedwx6OWK9lTtw61qzFRnWoVcp9img%2FE1Pb1ifPUbzXIuMb5WcflFtB7QcOpeT%2B4a47mIuuHdOJrXZjn3b4A4lcvX2o3hVcIoQzPt%2Bp1dOeQ2PK%2F%2BPR227Rp4%2BrELZLGvR1ZzQPa5xhzYdgf7DM0akn1mw%2FWlcbV%2BDgMbhjWwkBuYAzPT6snMtlfjSGAmaFgF0LSGvYM3q%2BNGQ%2BU03LruAB05jWtdM45dIIt5Pd4RNw7z2oAtPxfYifO2naQk4NxsVjc3cwt9cc7dqKeFGQ5ZPaphRtLpZyLoclXbjayW5PEIyNqHsZcmiffJ6KE1aPN7jNPSvI3%2BshSoqCzoTSnQAeHa9l4EKa54PVsLmijrC1fQ3L6%2FGZBLV4Q9lvwNiBA7%2F%2BtShM5JmrASkxruxV5xMzdkZf8mcmFHDC5BI6FjnuvCNmuXuBsNlRlqMkfrCh7toxfI0sCiRZfy%2BBU43lzrSt0YulejiW5eTpMFsvMUaoXFnlYLKq9pVAm8qfM6fY3JiNRzGil%2BltU4PZGjNyHe05qcyj2HaTBm%2Fswu7gH72AWSuOc5PNU6K0cSvI2DfA4ig96hzwPOq8sy1n1HhOxKRsItweL5yWp6VFZJyUTTir0NGGgpazqNMafC027QwCjfVNA6CdDoDzp8eU8%2F2l8Omg1NUG%2F15tGseU7TBlhnZRZ5S2fPbatXrLFYLIzcNg3IANDXjrQtI%2BzNgIvK9gyWtvulQPNLqMC%2BAADmtAAwf0EAaHp9MvDGtMEgSsP2XTRf9EpQQOQuBlYavScVzUnMZrNwsctsci%2BlGSOgMjQ%2BYfEQzhOYD9BHGMMlWqP8MZ9Q8hz6glVJK7jJPm7X0YckXxA7e1mFBD1tYC6%2BlwRu6nBmTr%2Bu53UprODaHmi5z7XV2BDumo1sNdpwWIReLfh2bK0dBS2agx4mGJOqXOgvXn3EAcp6%2FB8%3D



Here you can see our DB Diagram of our Library Management System. For the sake of clarity, we will also attach a link to this DB Diagram, there you can get more precise information about our DB, how multiple instances and classes connected to each other, relationships between them (one-to-one, one-to-many or many-to-many), memory usage and many more data.

**The link to DB Diagram:**

https://dbdiagram.io/d/5de8d8e9edf08a25543ebd96

**Entity-Relationship Diagram**