Software Design Document

**<Library Management System>**

**Version 1.0**

**Printing Date: - 25/12/2021**

**Faculty of Computers and Information**

**Bioinformatics Department**

**TEAM < ONE >**

# Revision Page

## Overview

## This version of the <Librarian Management System> consist of some main sections which are:

* Adding students
* Adding new librarians
* Allow the librarians to login to the system
* Adding new books
* Borrowing and returning books

## Target Audience

1. The Librarians

## Project Team Members

1- Randa Sabry Zakaria 2- Randa Adel Abd EL-Aziz

3- Reham Refaat Zaki 4- Habeba Allah Adnan

5- Sara Ahmed Mohamed

## Version Control History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| Version 1.0 |  |  | 29/12/2021 |

# Table of Contents

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **Introduction** | |  | 1 | |
|  | 1.1 | Purpose |  | 1 | |
|  | 1.2 | Scope |  |  | |
|  | 1.3 | Definitions, Acronyms and Abbreviations | | | |
|  | 1.4 | Reference Materials | | | |
|  | 1.5 | System Overview | | | |
| **2** | **System Architectural Design** | | | | |
|  | 2.1 | Architectural Style and Rationale | | | |
|  | 2.2 | Component Model | | | |
|  | 2.3 | Use Case Diagram | | | |
| 3 | **Detailed Description of Modules** | | | | |
|  | 3.1 | Complete Package Diagram | | | |
|  | 3.2 | Modules Detailed Descriptions | | | |
|  |  | 3.2.1 | Module <Students> | | |
|  |  | 3.2.1.1 | | Package Diagram | |
|  |  | 3.2.1.2 | | Class Diagram | |
|  |  | 3.2.1.3 | | Sequence Diagrams | |
|  |  | 3.2.2 | Module <Librarians> | | |
|  |  | 3.2.2.1 | | Package Diagram | |
|  |  | 3.2.2.2 | | Class Diagram | |
|  |  | 3.2.2.3 | | Sequence Diagrams | |
|  |  |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |

|  |  |  |
| --- | --- | --- |
| 4 | **Data Design** | |
|  | 4.1 | Data Description |
|  | 4.2 | Data Dictionary |
| 5 | **User Interface Design** | |
|  | 5.1 | Overview of User Interface |
|  | 5.2 | Screen Images |
| 6 | **Requirements Matrix** | |
|  |  | |

# Introduction

Our app is a university library management system. We have more sections about

books, librarian and students.

## Purpose

## the purpose from the SSD document is that describe overall project and give general overview of the application , its classes , functions , its objects, how can we use it and who , the full description of the database and its content , ERD diagram, system software, designs and references that help us in all project work when anyone who know nothing about our project and read this document he will understand all the project.

## 1.2 Scope

* This system will run on Microsoft windows V.10, Visual studio 2019 and Microsoft SQL.
* we use local database to get from it book names and record on its librarian and student names. Issued books will deleted from it and added books will inserted on it.
* Our application will help university's students to get books easily.
* The Library Management System should store all information about librarians, students, books, and their access keys etc.

## Definitions

## The Definitions that are available to the Librarian are: -

* Add new librarians and their information to the database.
* Edit the information of existing librarians.
* Delete information of librarian from database.
* A librarian can issue a book to the student.
* Can view the different categories of books available in the library.
* Can view the List of books available in each category.
* Can view the quantity of available books from each type and the price of the book.
* Can take the book returned from students.
* Can print the report of the returned books.
* Add new books to the database.
* Edit the information of existing books.
* Delete information of book from database.
* Add new students and their information to the database.
* Edit the information of existing students.
* Delete information of student from database.

## References

<https://krazytech.com/projects>

<https://www.youtube.com/channel/UCh-Wyi7aWSkvGujc-5FKc4w>

<https://youtube.com/playlist?list=PLqPejUavRNTWrbmE7fTvBC1HH-sAmz7c3>

<https://www.codecademy.com/catalog/language/c-sharp?g_network=g&g_device=c&g_adid=518718870675&g_keyword=c%23%20tutorial&g_acctid=243-039-7011&g_adtype=search&g_adgroupid=102650142673&g_keywordid=kwd-301345599083&g_campaign=ROW+Language%3A+Basic+-+Exact&g_campaignid=10074200771&utm_id=t_kwd-301345599083:ag_102650142673:cp_10074200771:n_g:d_c&utm_term=c%23%20tutorial&utm_campaign=ROW%20Language%3A%20Basic%20-%20Exact&utm_source=google&utm_medium=paid-search&utm_content=518718870675&hsa_acc=2430397011&hsa_cam=10074200771&hsa_grp=102650142673&hsa_ad=518718870675&hsa_src=g&hsa_tgt=kwd-301345599083&hsa_kw=c%23%20tutorial&hsa_mt=e&hsa_net=adwords&hsa_ver=3&gclid=EAIaIQobChMIw-DlkLqN9AIVNBoGAB2uzQfFEAAYASAAEgKNcvD_BwE>

## Overview

* **Firstly**
  1. We spoke about the introduction of the project to describe what the

project does and how it will work.

* 1. We spoke about the purpose of the of the project and what is the

benefits from it.

* 1. We spoke about the scope that defines the classes and functions of the application and what is each function definition.
  2. We spoke about definitions that define what the system has that act as the librarian’s role in the project.
  3. We spoke about references that define what is sources that help us to initiate this app from scratch, courses to learn C#, tutorials to learn how to design, channels teach us how to code.
* **Secondly**

1. We spoke about the architectural style that are used while designing the Software and why we use this architectural.
2. We spoke about the architectural model that are used in our system and the relation between the components.
3. We spoke about use case diagram, we explained how the system will

exactly work between students and the user.

* **Thirdly**

1. We added subsystem diagram for our project.
2. In the Detailed Description we gave two examples for our modules,

which are <The Student Module> and <The Librarian Module>.

* **Fourthly**

a) We added our project's ERD.

b) We described the database sections and every table on it and its ERD

Content.

* **Fifthly**

a) We spoke about our GUI and what it is consisting of and give a

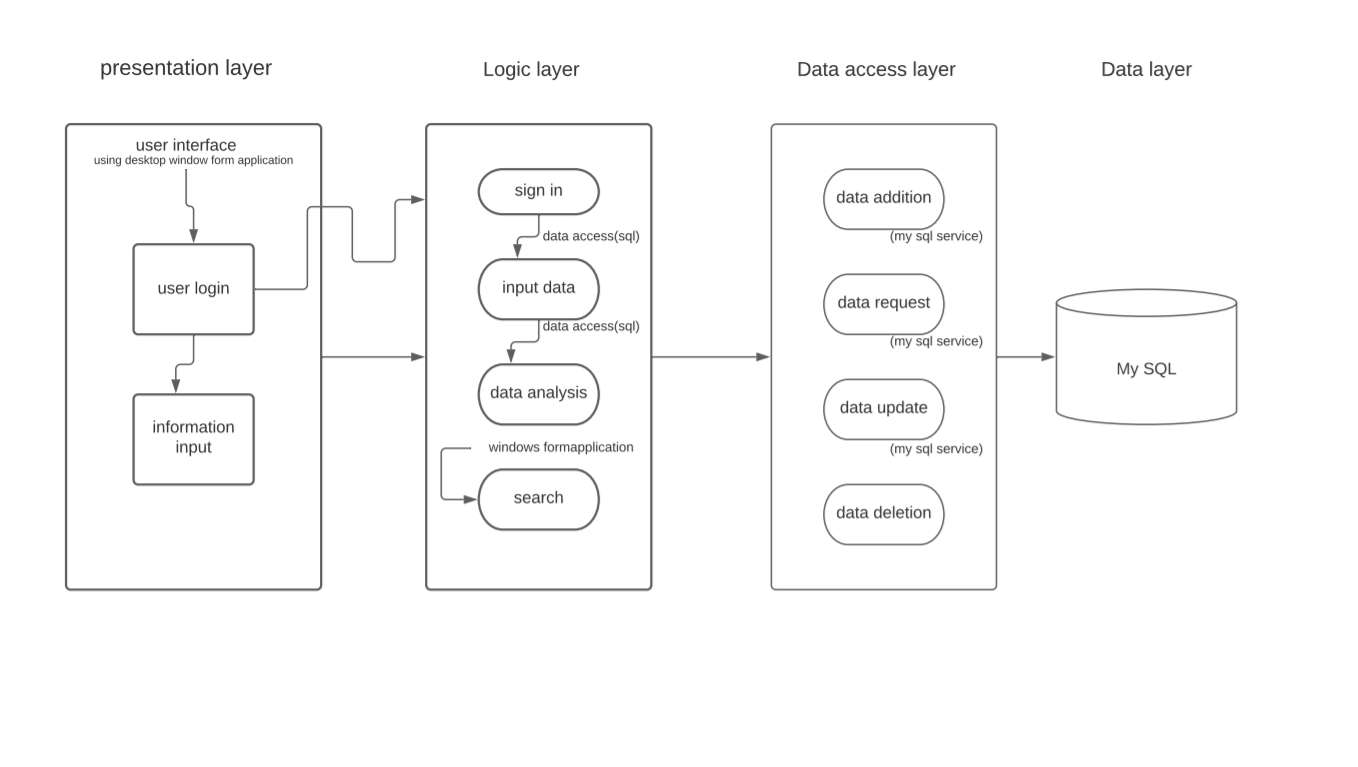
screenshots for each.

# System Architectural Design

## Architecture Style and Rationale

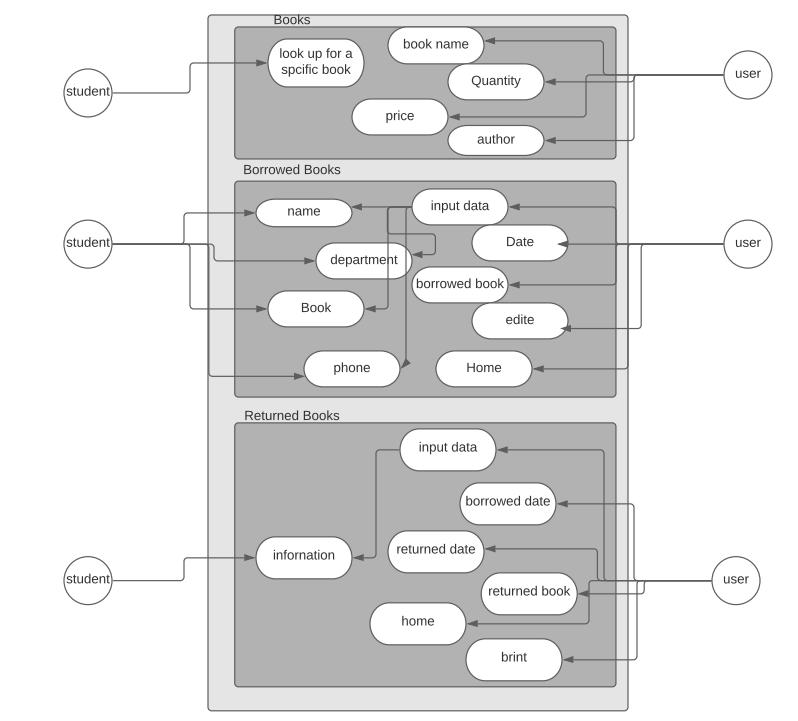
* 1. Data-centered architecture.
  2. Data-centered architecture helps integrity.
  3. Improved data visibility.
  4. Reduce operational risk (more controls on data access).
  5. Efficiency and cost reductions.

## Architecture Model

****

**Figure 2.1: Component Model of <Library Management System>**

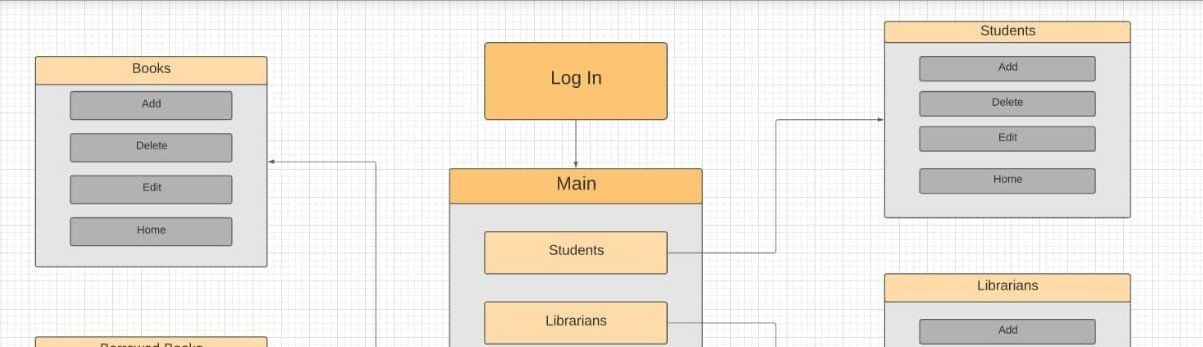
## Use Case Diagram

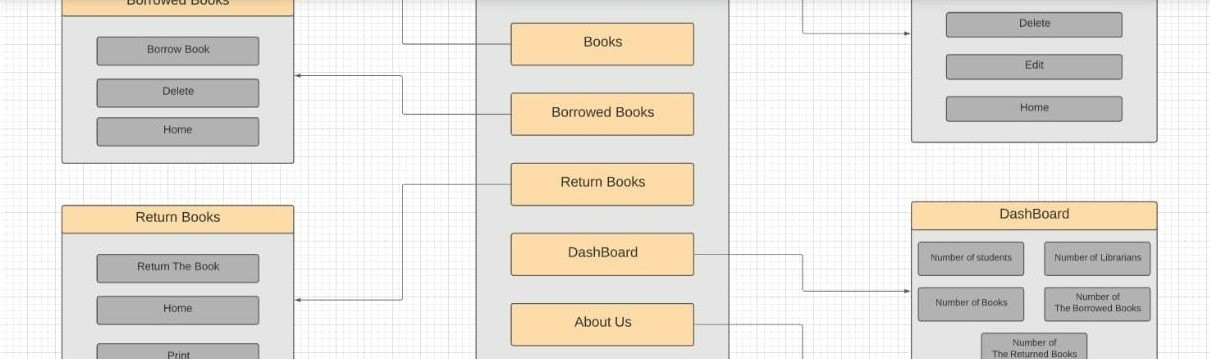
****

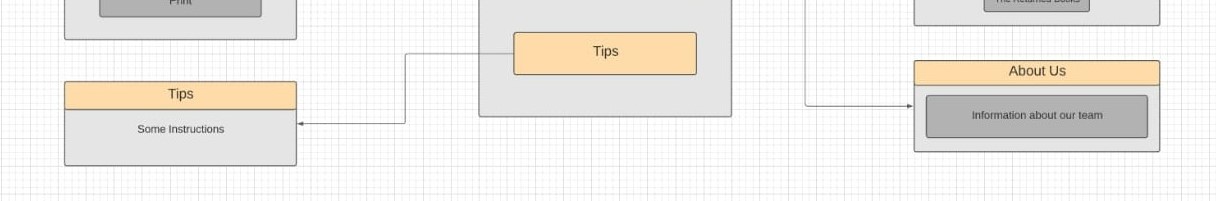
**Figure 2.2: Use Case Diagram of < Library Management System >**

# Detailed Description of Components

## Complete Package Diagram





****

**Figure 3.1: Subsystem of <Library Management System>**

## Detailed Description

## Module <Students>

* + - 1. **F.std: Package <** StudentsForm**>**

1. It includes the names of the student who entered the library or have borrowed

books from it.

1. It also includes their (ID, Phone Number**,** Department, Semester).
2. We can use this form to add or delete or edit the data of a student.

## Class Diagram

student

+ StudentForm ()

+ Exit (sender, e): void

+ Populate (): void

+ FromStudentToMainForm (sender, e): void

+ DeleteStudent (sender, e): void

+ StudentForm\_Load (sender, e): void

+ AddStudent (sender, e): void

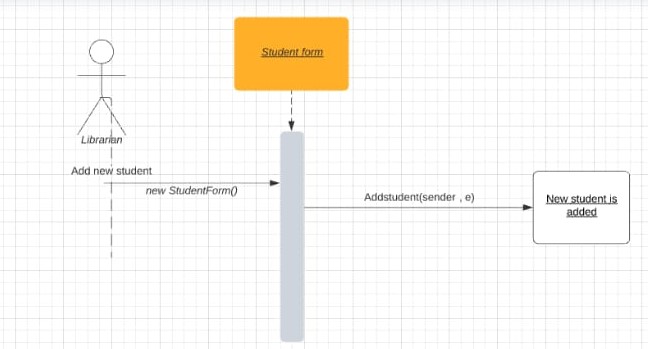
+ StudentDGV\_CellContentClick (sender, e): void

+ EditStudent (sender, e): void

**Figure 3.2: Class diagram<UML> for <Students Form Package>**

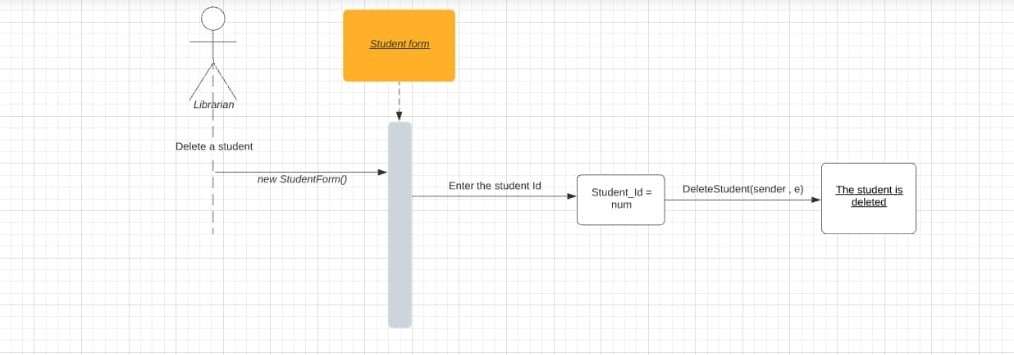
## Sequence Diagrams

1. SD001: Sequence diagram for Adding New student

****

**Figure 3.3: Sequence Diagram of <Adding New student scenario>**

b) SD002: Sequence diagram for Deleting a student Scenario



**Figure 3.4: Sequence Diagram of <** **Deleting a student Scenario >**

## Module <Librarians>

* + - 1. **F.lib: Package <** Librarians Form **>**

1. It includes the names of the librarians who work in the library.
2. It also includes their (ID, Phone Number**,** Password).
3. We can use this form to add or delete or edit the data of a librarian.

## Class Diagram

librarian

+ LibrarianForm ()

+ LibrarianForm\_Load (sender, e): void

+ Exit (sender, e): void

+ Populate (): void

+ FromLibrarianToMainForm (sender, e): void

+ DeleteLibrarian (sender, e): void

+ AddLibrarian (sender, e): void

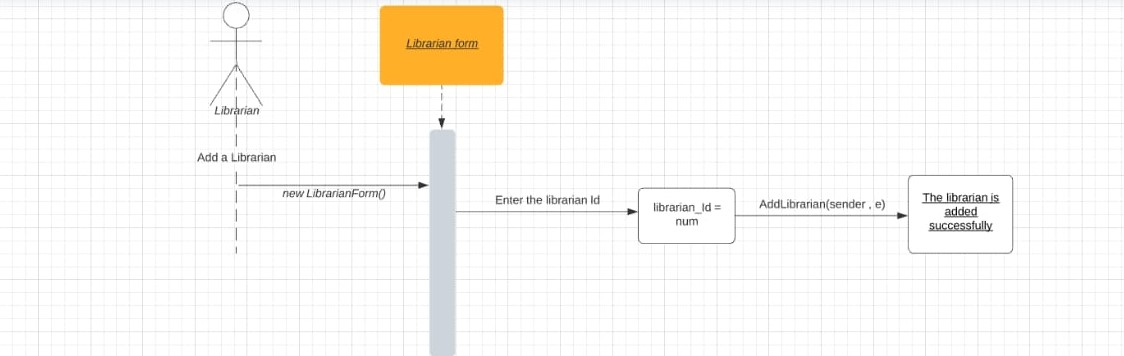
+ LibrarianDGV\_CellContentClick (sender, e): void

+ EditLibrarian (sender, e): void

**Figure 3.5: Class diagram<UML> for <Students Form Package>**

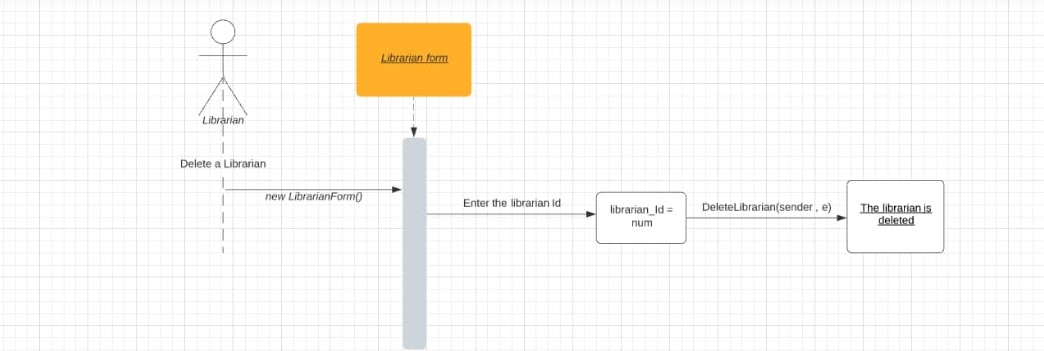
**3.2.2.3 Sequence Diagrams**

1. SD001: Sequence diagram for Adding New Librarian

****

**Figure 3.6: Sequence Diagram of <** **Adding New Librarian Scenario >**

b) SD002: Sequence diagram for Deleting a LibrarianScenario



**Figure 3.6: Sequence Diagram of <** **Deleting a Librarian Scenario >**

## 3.2.3 Module <Borrowed Books>

**3.2.3.1 F.lib: Package <** Borrowed Books Form **>**

1. It includes ids of the borrowing prosses.
2. We can use it to borrow a new book or delete a borrowed one
3. It also includes the (ID, Phone Number**,** Department, student id) of the student that makes the borrow.

## 3.2.3.2 Class Diagram

Borrow Book

+ BorrowedBooks ()

+ FillStudent (): void

+ FillBook (): void

+ Populate (): void

+ fetchstudent (): void

+ UpdateBook (): void

+ Borrowed\_Book\_Load (sender, e): void

+ FromBorrowedBooksToMainForm (sender, e): void

+ StdCd\_SelectionChangeCommitted (sender, e): void

+ BorrowBook (sender, e): void

+ Exit (sender, e): void

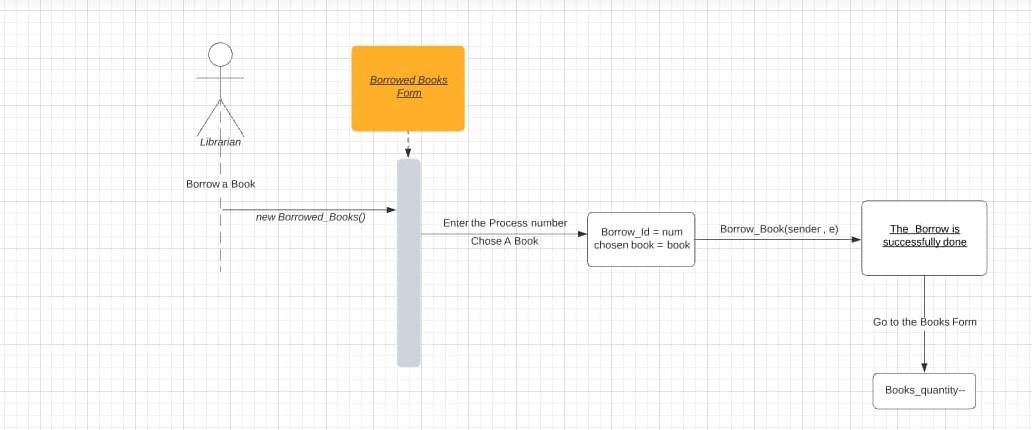
+ BorrowCancel (sender, e): void

+ BorrowedBookDGV\_CellContentClick (sender, e): void

**Figure 3.7: Class diagram<UML> for <** **Borrowed Books Form Package>**

# 3.2.3.3 Sequence Diagrams

a) SD001: Sequence diagram for Borrowing a Book



# Figure 3.8: Sequence Diagram of < Borrowing a Book Scenario >

## 3.2.4 Module <Returned Books>

**3.2.4.1 F.lib: Package <** Returned Books Form **>**

1. It includes ids of the returning prosses.
2. We can use it to return a borrowed book.
3. It also includes the (ID, Phone Number**,** Department, student id) of the student that makes the returning.

## 3.2.3.2 Class Diagram

Return Book

+ ReturnBookForm ()

+ Exit (sender, e): void

+ Populatereturn (): void

+ FillStudent (sender, e): void

+ FillBook (sender, e): void

+ ReturnBookFormLoad (sender, e): void

+ UpdateBook (): void

+ BorrowBookDGV\_CellContentClick (sender, e): void

+ BorrowCancele (): void

+ReturnBook (sender, e): void

+FormReturned\_BookToMainForm (sender, e): void

+pictureBox\_Click (sender, e): void

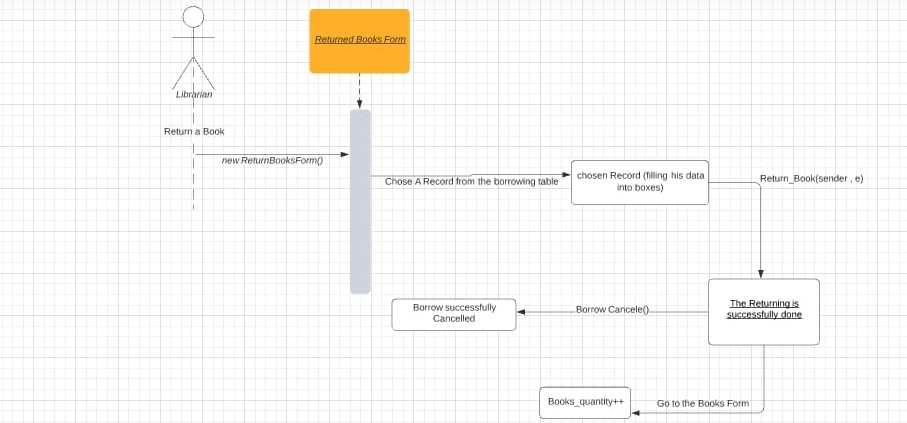
+printDocument1\_PrintPage (sender, e): void

+ Button2\_click (sender, e): void

**Figure 3.9: Class diagram<UML> for <** **Returned Books Form Package>**

# 3.2.3.3 Sequence Diagrams

a) SD001: Sequence diagram for Returning a Book



# Figure 3.10: Sequence Diagram of < Returning a Book Scenario >

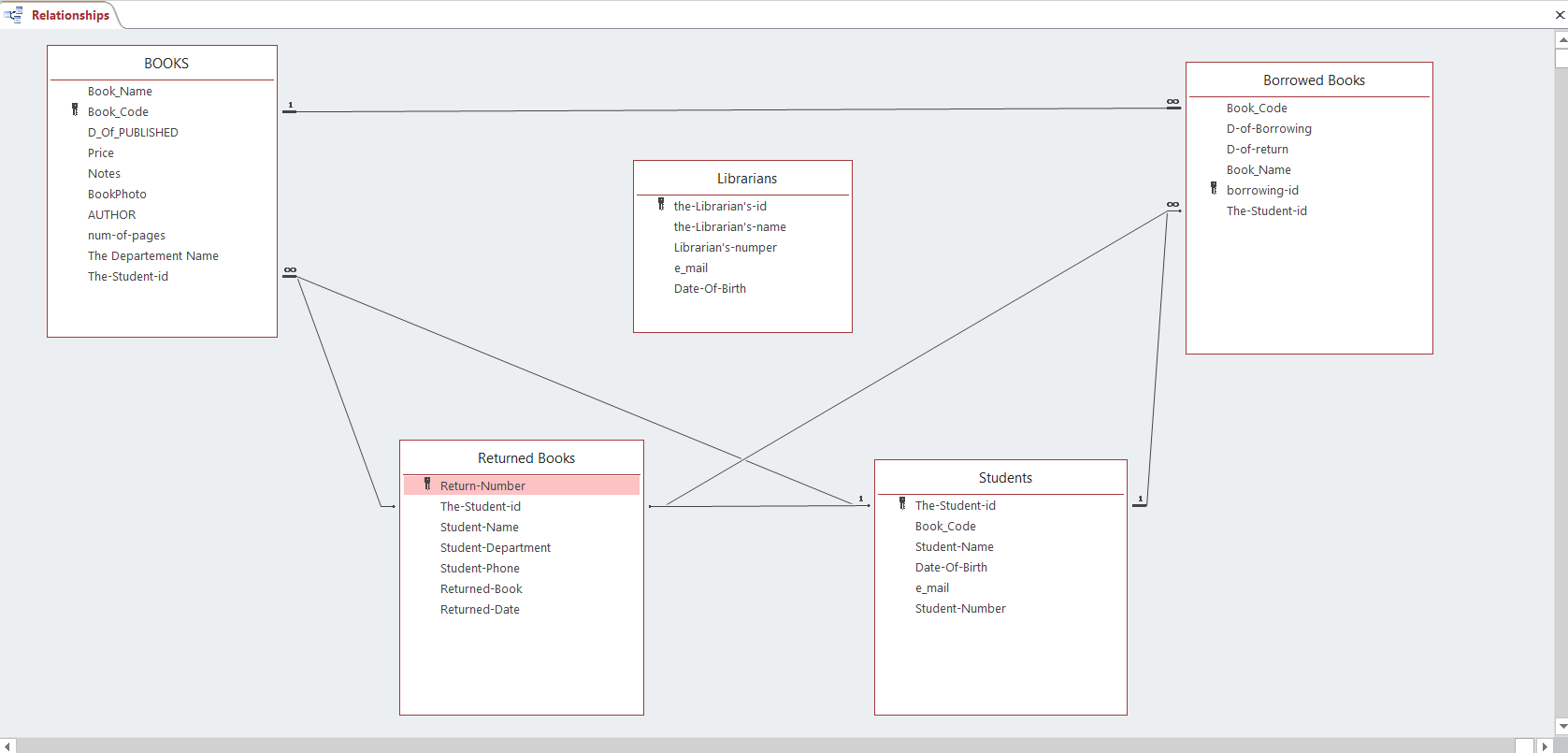
# 4. Data Design

## we describe the database sections and every table on it and its content

## Data Description

## 

**Figure 3.4: Entity Relationship Diagram (ERD) of < Library Management System>**

 **Another Entity Relationship Diagram (ERD) of < Library Management System>**

## Data Dictionary

## In this section we will speak about database and its organization:

## The first database about books and include

## Book Name

## Author

## Publisher

## Price

## Quantity

## The second database about issued books and include

## Issue Number (the number of issued)

## Student Id

## Student Name

## Student Department

## Student Phone

## Book Issue (the name of the book)

## Issue Date

## The third database about librarian and include

## librarian Id

## librarian name

## Librarian password

## Librarian phone

## The fourth database about returned books and include

## Returned Names

## Student Id

## Student Name

## Student Department

## Student Phone

## Book Returned

## Issue Date

## Returned Date

## The fifth database about students and include

## Student Id

## Student Name

## Student Department

## Student Phone

## Student Semester

# 5. User Interface Design

## we speak about our GUI and what is consisting of and give a screenshot for every one

## 5.1 Overview of User Interface

## We have three GUIs:

## The first one that start one where the librarian can log in on it by entering username and password.

## the second one which the librarian can access from it all the classes, sections and functions (librarians- students- books- borrow box- return box) and have a help section “TIPS”,

## “About us” we are defining ourselves and a window we can go to “Dashboard” (the third user interface).

## the dashboard act as statistics that collect all recorded data that are stored in these sections (librarians- students- books- borrow box- return box)

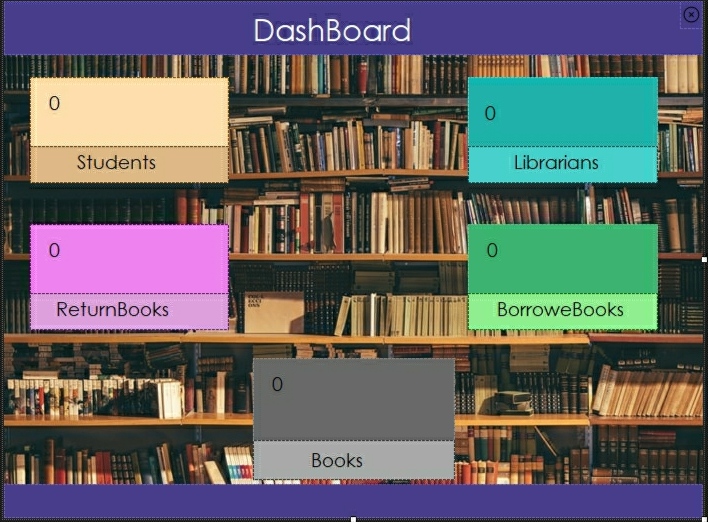
**5.2 Screen Images**

**1**-



**2**-



**3-**

**6- Requirements Matrix**

## *R1 ->* User Interfaces

## R1.1🡪 no one can access on it without sign in.

## R1.2🡪 we have standard buttons like delete, add, edit and back for each argument and

## some functions like students, books, and librarians to control on each element.

## R1.3🡪 If librarian need help, he can press “Tips”.

## R1.4🡪 If the librarian search on a book and this book is not here it will give him error

## message.

## R1.5🡪 When a new book or student or librarian is added, new book is issued or updated

## or returned it will give confirmation message.

## R1.6🡪 if someone enter a wrong username or password it will give warning message.

## *R2 ->* Software Interfaces

**R2.1**🡪 **This system will run on Microsoft windows V.10, Visual studio 2019 and**

**Microsoft SQL.**

**R2.2🡪 we use local database to get from it book names and record on its librarian and**

**student names.**

**R2.3🡪 Borrowed books will be deleted from it and added books will inserted on it.**

## *R3 ->* Communications Interfaces

**R3.1🡪 The librarian to login the system needs to enter username and password.**

**R3.2🡪 To protect this system the librarian must enter the password, or it will**

**give warning message.**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **R1.1** | **R1.2** | **R1.3** | **R1.4** | **R1.5** | **R1.6** | **R2.1** | **R2.2** | **R2.3** | **R3.1** | **R3.2** |
| Module 1 <Student> |  | **✓** |  |  | **✓** |  |  |  |  |  |  |
| Module 2 <Librarian> |  | **✓** |  |  | **✓** |  |  |  |  |  |  |
| Module 3 <Books> |  | **✓** |  | **🗶** | **✓** |  |  |  | **✓** |  |  |
| Module 4 <Borrowed  Books > |  | **✓** |  | **🗶** | **✓** |  |  |  | **✓** |  |  |
| Module 5 <Returned  Books> |  | **✓** |  |  | **✓** |  |  |  |  |  |  |
| Module 6 <Login> | **✓** |  |  |  |  | **✓** |  |  |  | **✓** | **✓** |
| Module 7 <Tips> |  |  | **✓** |  |  |  |  |  |  |  |  |
| The whole system and the database |  |  |  |  |  |  | **✓** | **✓** |  |  |  |