HACETTEPE UNIVERSITY

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

BBM487 SOFTWARE ENGINEERING LAB
Group 14

LBLS Desktop
Software Design Description

20926046	RIDVAN ALTUNEL	Project Manager & Developer
21327618	AYŞE AKCAN	Architecht & Tester
21228731	MAHMUT TEKİNYER	Analyst & Technical Writer

Library Book Loan System
Software Design Description

1. Revision History

Version	*Date*	*Author*	*Change Description*
V1	11.04.2017	Rıdvan Altunel Mahmut Tekinyer Ayşe Akcan Ahmet Çalışkan	Prototype 1: Demo of a single use case
V2	25.04.2017	Rıdvan Altunel Mahmut Tekinyer Ayşe Akcan Ahmet Çalışkan	Prototype 2: Demo of 6 use cases (half of system)

2. INTRODUCTION

2.1 Purpose and Scope

The aim of this design document is; We have tried to eliminate the mistakes in the previous versions and tried to introduce a more systematic functional design. Partial changes to the interface designs have been made to go with a few additions. In other words, if it is necessary to explain in general terms, it is aimed to make a detailed design transition, to design data management and data storage forms, to introduce structural design, to introduce component design, to introduce component design and to introduce GUI by means of visuals.

2.2 Document Overview

In general this document is; An overview of the system, descriptions used in the system, design decisions, details of the design, behavior diagrams (sequence diagram), data models, user interfaces software requirements are defined and elaborated.

2.3 System Overview

2.4 Definitions, Acronyms, and Abbreviations

Term/Acronym	*Definition*
MVC	Model View Controller
SQL	Structured Query Language
IEEE	The Institute of Electrical and Electronics Engineers
Std	Standard

2.5 References

The following documents have been used for reference to this document:

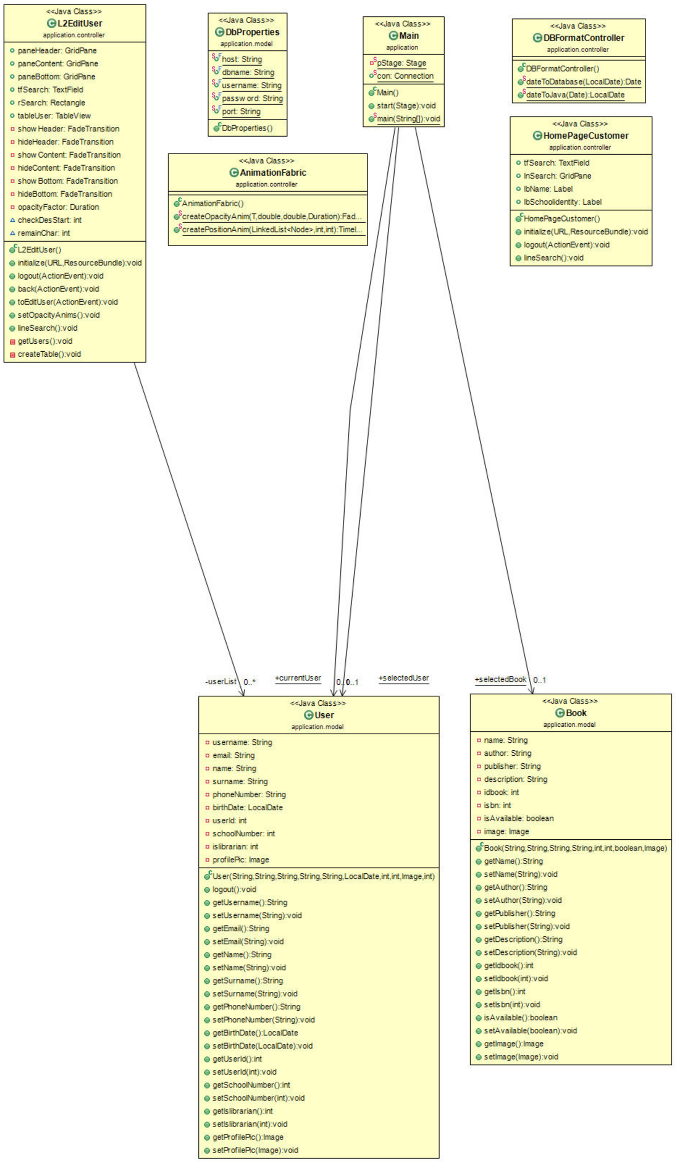
- Project Plan and Vision Document
- Software Requirement Document
- IEEE-1016 Standard Certificate
- Java Code Conventions

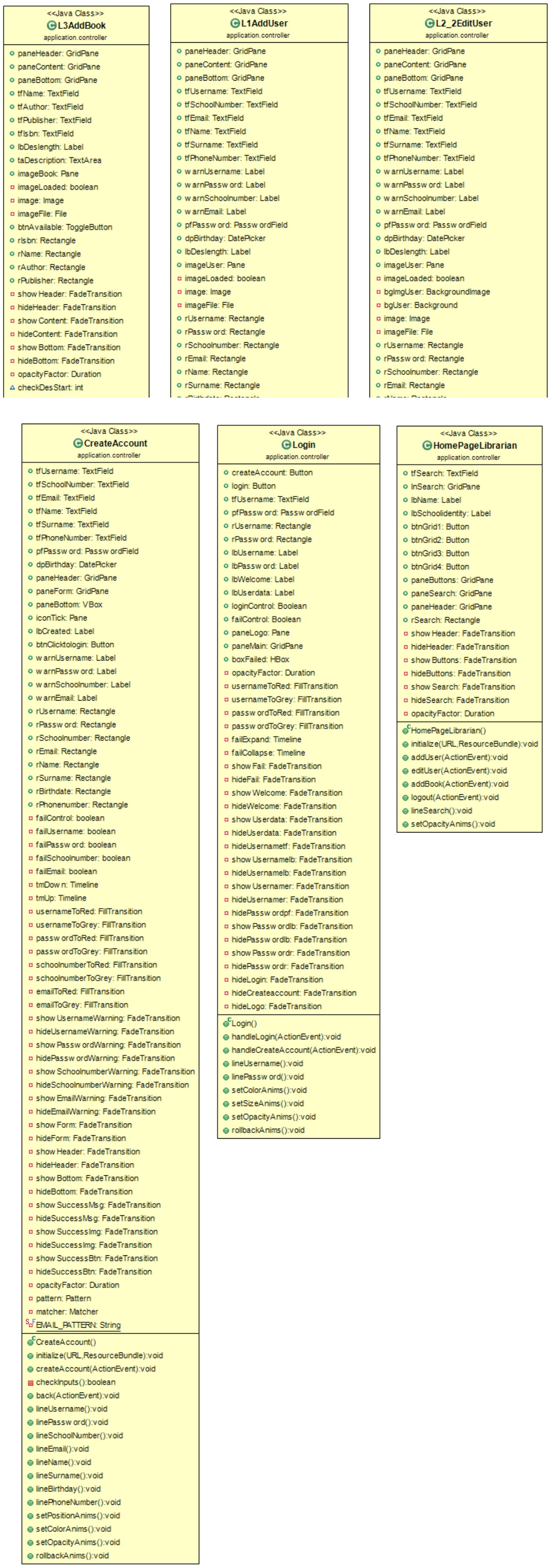
3. Design Constraints and Decisions

In the project, MVC architecture is used. In the project, we have our interfaces as view, not consol application. In the controller section, the code part that contains the usability of our use cases and which can connect to the interface and SQL database is used as the model. The general design of the project is shown below.

4. Design Details

4.1 Software Components





<<Java Class>>
L3AddBook
application.controller

○ paneHeader: GridPane
○ paneContent: GridPane
○ paneBottom: GridPane
○ tfName: TextField
○ tfAuthor: TextField
○ tfPublisher: TextField
○ tfIsbn: TextField
○ lbDeslength: Label
○ taDescription: TextArea
○ imageBook: Pane
□ imageLoaded: boolean
□ image: Image
□ imageFile: File
○ btnAvailable: ToggleButton
○ rIsbn: Rectangle
○ rName: Rectangle
○ rAuthor: Rectangle
○ rPublisher: Rectangle
□ showHeader: FadeTransition
□ hideHeader: FadeTransition
□ showContent: FadeTransition
□ hideContent: FadeTransition
□ showBottom: FadeTransition
□ hideBottom: FadeTransition
□ opacityFactor: Duration
△ checkDesStart: int

<<Java Class>>
L1AddUser
application.controller

○ paneHeader: GridPane
○ paneContent: GridPane
○ paneBottom: GridPane
○ tfUsername: TextField
○ tfSchoolNumber: TextField
○ tfEmail: TextField
○ tfName: TextField
○ tfSurname: TextField
○ tfPhoneNumber: TextField
○ warnUsername: Label
○ warnPassword: Label
○ warnSchoolnumber: Label
○ warnEmail: Label
○ pfPassword: PasswordField
○ dpBirthday: DatePicker
○ lbDeslength: Label
○ imageUser: Pane
□ imageLoaded: boolean
□ image: Image
□ imageFile: File
○ rUsername: Rectangle
○ rPassword: Rectangle
○ rSchoolnumber: Rectangle
○ rEmail: Rectangle
○ rName: Rectangle
○ rSurname: Rectangle

<<Java Class>>
L2_2EditUser
application.controller

○ paneHeader: GridPane
○ paneContent: GridPane
○ paneBottom: GridPane
○ tfUsername: TextField
○ tfSchoolNumber: TextField
○ tfEmail: TextField
○ tfName: TextField
○ tfSurname: TextField
○ tfPhoneNumber: TextField
○ warnUsername: Label
○ warnPassword: Label
○ warnSchoolnumber: Label
○ warnEmail: Label
○ pfPassword: PasswordField
○ dpBirthday: DatePicker
○ lbDeslength: Label
○ imageUser: Pane
□ imageLoaded: boolean
□ bgImgUser: BackgroundImage
□ bgUser: Background
□ image: Image
□ imageFile: File
○ rUsername: Rectangle
○ rPassword: Rectangle
○ rSchoolnumber: Rectangle
○ rEmail: Rectangle

<<Java Class>>
CreateAccount
application.controller

○ tfUsername: TextField
○ tfSchoolNumber: TextField
○ tfEmail: TextField
○ tfName: TextField
○ tfSurname: TextField
○ tfPhoneNumber: TextField
○ pfPassword: PasswordField
○ dpBirthday: DatePicker
○ paneHeader: GridPane
○ paneForm: GridPane
○ paneBottom: VBox
○ iconTick: Pane
○ lbCreated: Label
○ btnClicktologin: Button
○ warnUsername: Label
○ warnPassword: Label
○ warnSchoolnumber: Label
○ warnEmail: Label
○ rUsername: Rectangle
○ rPassword: Rectangle
○ rSchoolnumber: Rectangle
○ rEmail: Rectangle
○ rName: Rectangle
○ rSurname: Rectangle
○ rBirthdate: Rectangle
○ rPhonenumber: Rectangle
□ failControl: boolean
□ failUsername: boolean
□ failPassword: boolean
□ failSchoolnumber: boolean
□ failEmail: boolean
□ tmDown: Timeline
□ tmUp: Timeline
□ usernameToRed: FillTransition
□ usernameToGrey: FillTransition
□ passwordToRed: FillTransition
□ passwordToGrey: FillTransition
□ schoolnumberToRed: FillTransition
□ schoolnumberToGrey: FillTransition
□ emailToRed: FillTransition
□ emailToGrey: FillTransition
□ showUsernameWarning: FadeTransition
□ hideUsernameWarning: FadeTransition
□ showPasswordWarning: FadeTransition
□ hidePasswordWarning: FadeTransition
□ showSchoolnumberWarning: FadeTransition
□ hideSchoolnumberWarning: FadeTransition
□ showEmailWarning: FadeTransition
□ hideEmailWarning: FadeTransition
□ showForm: FadeTransition
□ hideForm: FadeTransition
□ showHeader: FadeTransition
□ hideHeader: FadeTransition
□ showBottom: FadeTransition
□ hideBottom: FadeTransition
□ showSuccessMsg: FadeTransition
□ hideSuccessMsg: FadeTransition
□ showSuccessImg: FadeTransition
□ hideSuccessImg: FadeTransition
□ showSuccessBtn: FadeTransition
□ hideSuccessBtn: FadeTransition
□ opacityFactor: Duration
□ pattern: Pattern
□ matcher: Matcher
□ EMAIL_PATTERN: String

● CreateAccount()
● initialize(URL,ResourceBundle):void
● createAccount(ActionEvent):void
■ checkInputs():boolean
● back(ActionEvent):void
● lineUsername():void
● linePassword():void
● lineSchoolNumber():void
● lineEmail():void
● lineName():void
● lineSurname():void
● lineBirthday():void
● linePhoneNumber():void
● setPositionAnims():void
● setColorAnims():void
● setOpacityAnims():void
● rollbackAnims():void

<<Java Class>>
Login
application.controller

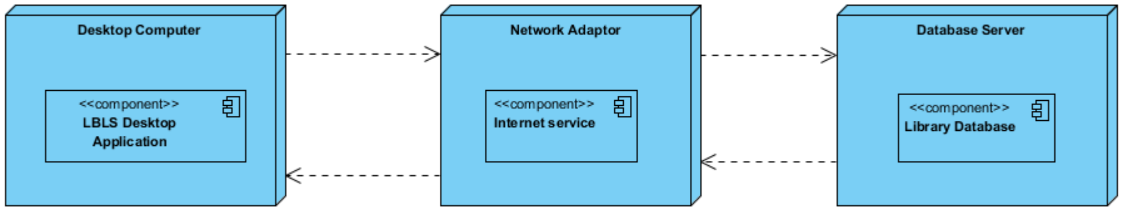
○ createAccount: Button
○ login: Button
○ tfUsername: TextField
○ pfPassword: PasswordField
○ rUsername: Rectangle
○ rPassword: Rectangle
○ lbUsername: Label
○ lbPassword: Label
○ lbWelcome: Label
○ lbUserdata: Label
○ loginControl: Boolean
○ failControl: Boolean
○ paneLogo: Pane
○ paneMain: GridPane
○ boxFailed: HBox
□ opacityFactor: Duration
□ usernameToRed: FillTransition
□ usernameToGrey: FillTransition
□ passwordToRed: FillTransition
□ passwordToGrey: FillTransition
□ failExpand: Timeline
□ failCollapse: Timeline
□ showFail: FadeTransition
□ hideFail: FadeTransition
□ showWelcome: FadeTransition
□ hideWelcome: FadeTransition
□ showUserdata: FadeTransition
□ hideUserdata: FadeTransition
□ hideUsernametf: FadeTransition
□ showUsernamelb: FadeTransition
□ hideUsernamelb: FadeTransition
□ showUsernamer: FadeTransition
□ hideUsernamer: FadeTransition
□ hidePasswordpf: FadeTransition
□ showPasswordlb: FadeTransition
□ hidePasswordlb: FadeTransition
□ showPasswordr: FadeTransition
□ hidePasswordr: FadeTransition
□ hideLogin: FadeTransition
□ hideCreateaccount: FadeTransition
□ hideLogo: FadeTransition

● Login()
● handleLogin(ActionEvent):void
● handleCreateAccount(ActionEvent):void
● lineUsername():void
● linePassword():void
● setColorAnims():void
● setSizeAnims():void
● setOpacityAnims():void
● rollbackAnims():void

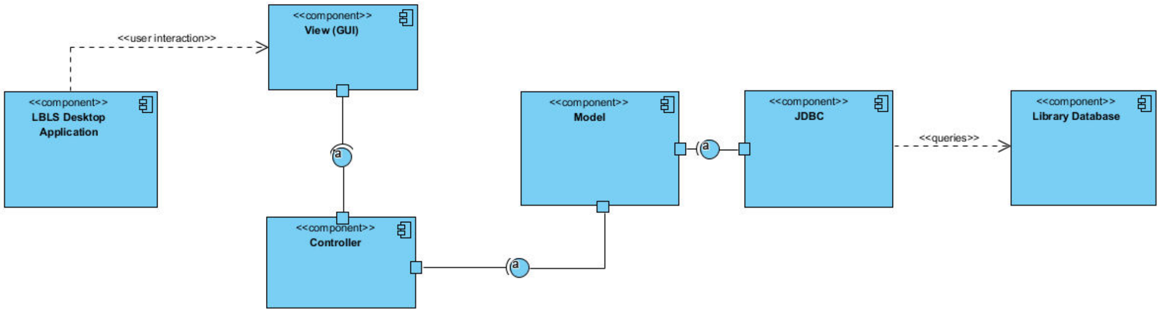
<<Java Class>>
HomePageLibrarian
application.controller

○ tfSearch: TextField
○ lnSearch: GridPane
○ lbName: Label
○ lbSchoolidentity: Label
○ btnGrid1: Button
○ btnGrid2: Button
○ btnGrid3: Button
○ btnGrid4: Button
○ paneButtons: GridPane
○ paneSearch: GridPane
○ paneHeader: GridPane
○ rSearch: Rectangle
□ showHeader: FadeTransition
□ hideHeader: FadeTransition
□ showButtons: FadeTransition
□ hideButtons: FadeTransition
□ showSearch: FadeTransition
□ hideSearch: FadeTransition
□ opacityFactor: Duration

● HomePageLibrarian()
● initialize(URL,ResourceBundle):void
● addUser(ActionEvent):void
● editUser(ActionEvent):void
● addBook(ActionEvent):void
● logout(ActionEvent):void
● lineSearch():void
● setOpacityAnims():void



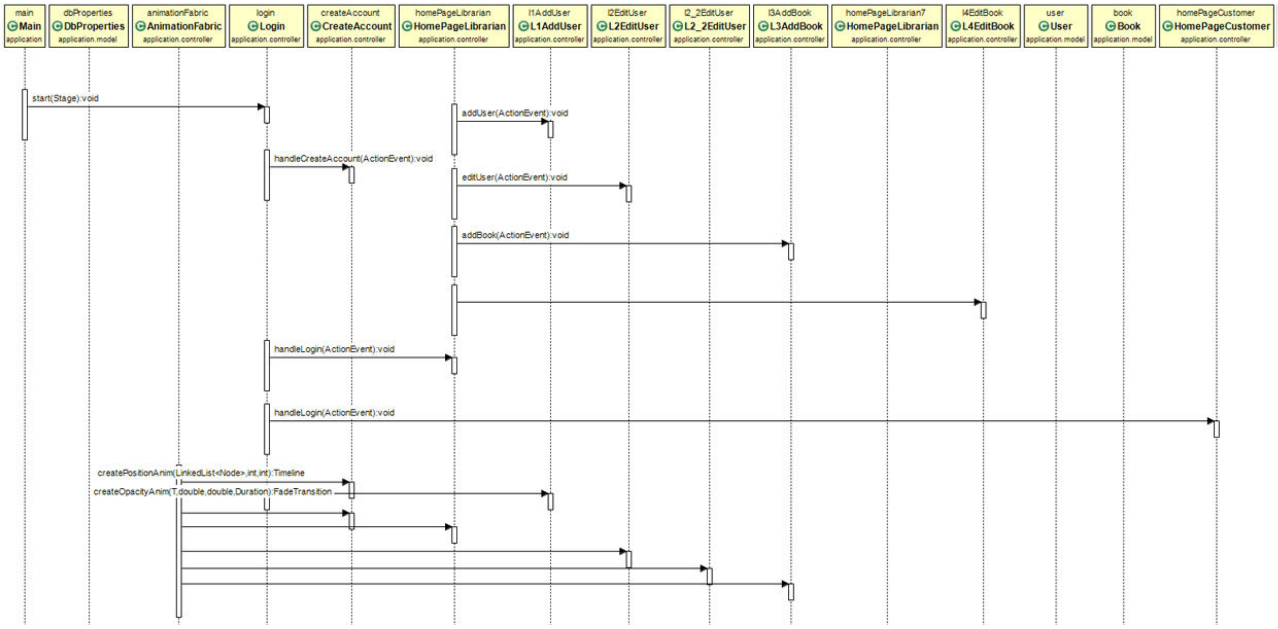
LBLS Desktop Deployment Diagram



LBLS Desktop Component Diagram

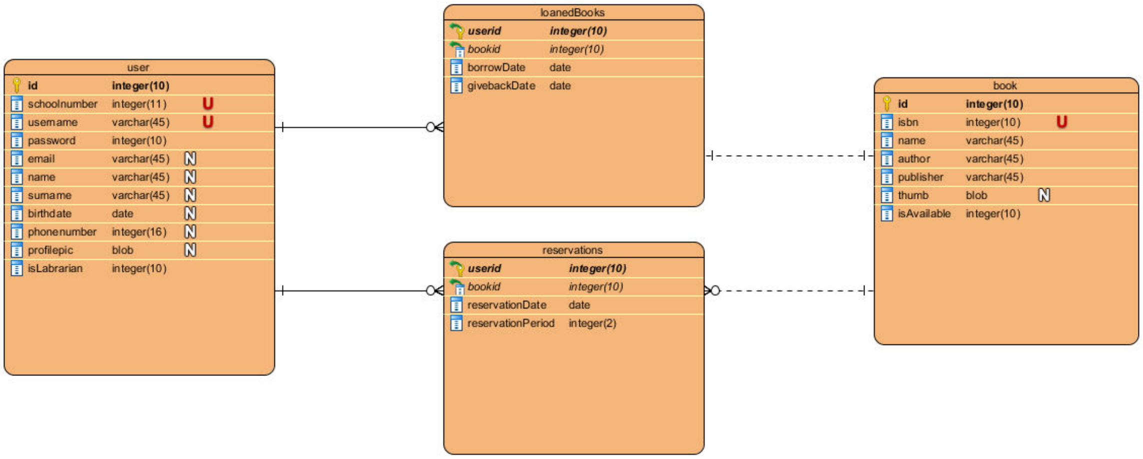
4.2 Software Behavior

Sequence diagrams are diagrams of sequence diagrams that are used to model the events of the message flow between consecutive objects or components in a system. A sequence diagram consists of objects, messages, and timelines. Below are the Sequence diagrams prepared in accordance with our system. The following pictures indicate which use case belongs.



4.3 Data Model (E-R Diagram)

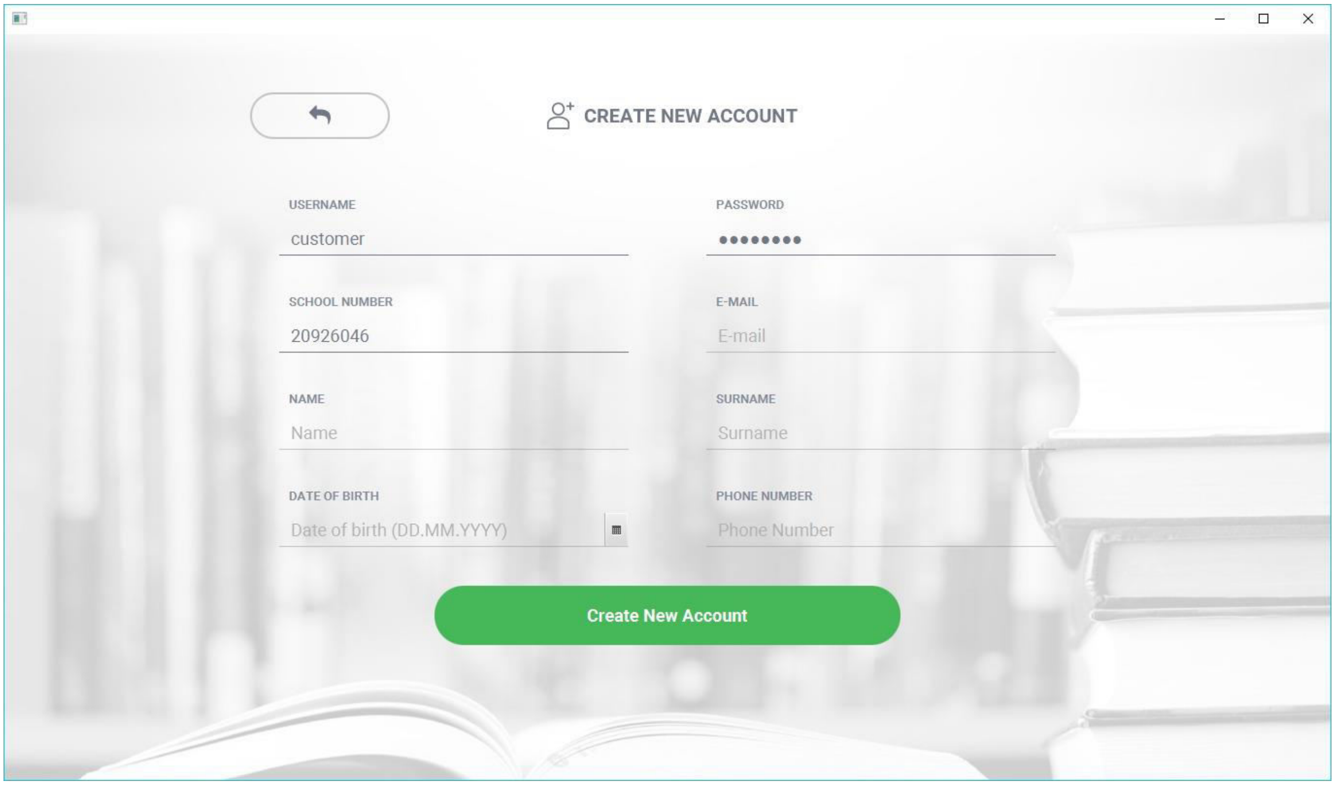
Relational data model is used as data model. Thanks to the relational data model used in database, we can manage our data more easily and search, insert, delete, update can be made easier thanks to database.
The tables are as follows.



LBLS Desktop ER Diagram

4.4 User Interface Design

User Interface Designs are available as an attachment in the "Architectural Notebook" document from previous documents.



CREATE NEW ACCOUNT

USERNAME
customer

PASSWORD
••••••••

SCHOOL NUMBER
20926046

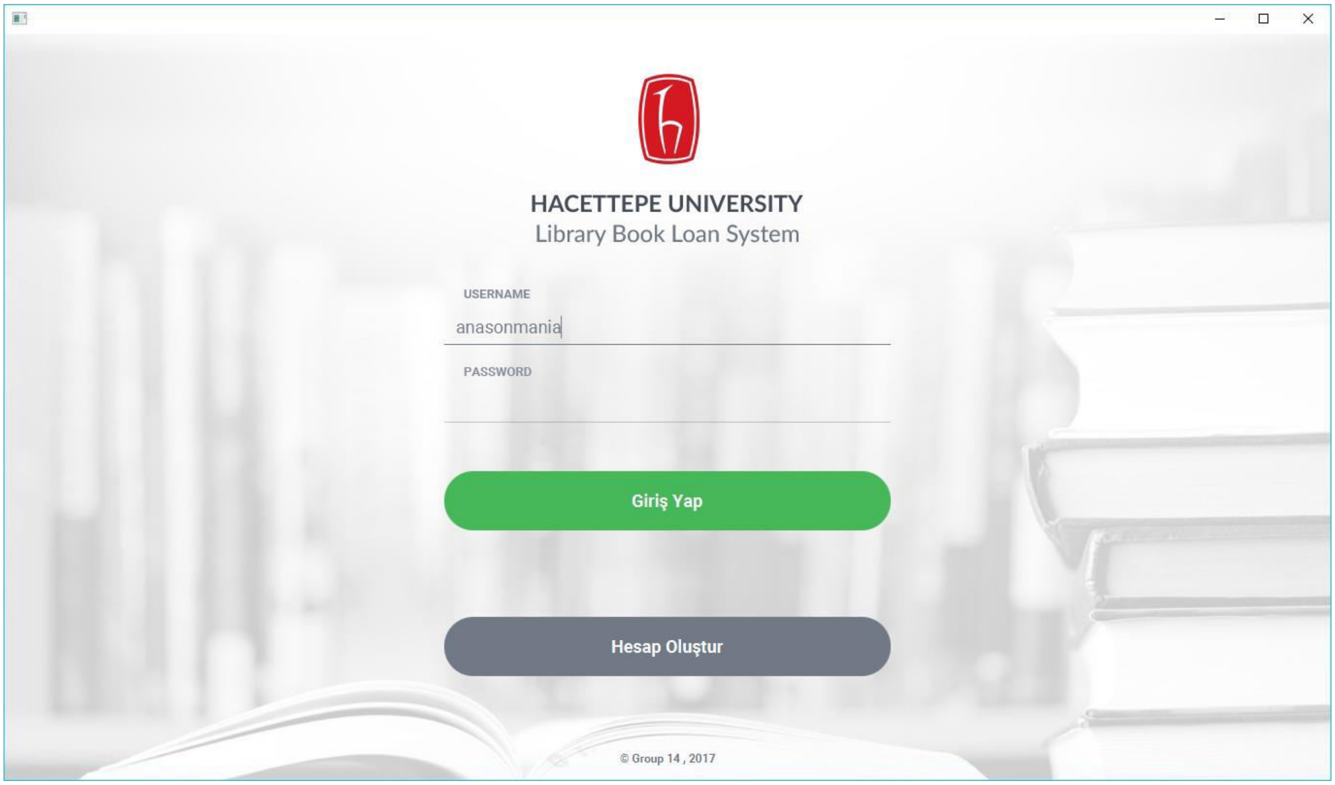
E-MAIL
E-mail

NAME
Name

SURNAME
Surname

DATE OF BIRTH
Date of birth (DD.MM.YYYY)

PHONE NUMBER
Phone Number



HACETTEPE UNIVERSITY
Library Book Loan System

USERNAME
anasonmania

PASSWORD

5. Requirements Traceability

Need	Priority	Features	Is it done?
Connecting a user to system	1	Login & Logout Feature	
Searching for books	1	Search book feature	
Adding a new book to system or editing	1	Manipulate Book Feature	
Adding a new user to the system or editing	1	Manipulate User Feature	
Borrowing a book	1	Self check-out feature	
Giving back a book	1	Self return feature	
Add name to book waiting list	2	Waiting list feature	
View books that has been borrow**ed**	2	View my boks feature	
Issue late fine or pay fine	1	Fine managing feature	
Notify of book being available	2	Notify feature	