Final SOFTWARE DESIGN SPECIFICATION DOCUMENT

School Book Tracking Application

Chetan Sharma→ A20391333

Bin Sun →A20376746

Rachna Bafna → A20342050

Thomas Latrofa → A20367335

Chen Gong →A20379606

SOFTWARE DESIGN SPECIFICATION DOCUMENT

Project Title:

School

Book Tracking Application

1. INTRODUCTION

2. This section summarizes development requirements of the system we are going to build a prototype for. In order to demonstrate our work, we need requirement specification and on top of it analyze the requirements. With current age of digitization, the book keeping systems also need to be evolved. We will be addressing the digitized books along with traditional bookkeeping systems. There is also part where research papers are accessed by the students like IEEE papers. Our system will keep track of the ebooks, hardcover books as well as allow students to search for research papers, magazines etc. We also make sure the system distinguishes between the type of book and based on that fine is calculated. The digital copy of ebooks can be accessed through shared location and will not have any fine associated with it. Same is the case for research papers and digital magazines. In case paper packed or hardcover books students often prefer to borrow books at the beginning of the term. Some return books in a few weeks, while others wait till the end of term. Sometimes books are lost without

accountability or returned late without consequences. Students who are waiting on return of a book suffer in cases where earlier students failed to return the books. Books are also returned in unusable conditions and it is accounted as loss to the library. A simple note on book's condition on return can help to hold students responsible for the damage.

1.1 Goals and Objectives

Our goal is to come up with a platform to track books efficiently for digital as well as physical copies of the books. The significance of the project is determined by how well the students can interact with the system and the way in which books are being tracked.

The application maintains a repository of school books for physical as well as digital versions. The system provides ability to create user account for each student. The application accepts valid username and password to access the system. The students should be able to create/view/modify the user account and user profile. Students should be able to borrow the books. Students can search the books by title/author/ISBN, students can search the research papers and read digital magazines The system will also have administrators who will enter the new books and are able to update the inventory of the system. Admin user can edit the quantity of books, add new books, delete books, add new research papers and magazine links in the system. Admin user can also take the backup of the system. System will also have teacher as a type of user. Teachers can add recommendations for their courses in the system. Teachers can borrow more number of books as compared to students. System should respond with appropriate message or book details based on search result. System computes the fine for each student and should charge fine to user account if the book is not returned on time or returns book in damaged condition. Students should be asked to settle/view their fine. System also has ability to add new books, edit and delete existing books.

In earlier section we discussed the functional requirements. We also need to specify non-functional requirements like security, user friendliness and performance. Apart from of tacking books, the system provide security to access the system. All users need to log in to the system by username and password before they use the system. The system shall respond to all requests within 5 seconds. In terms of user friendliness the system should fit different platforms on different screen based on the operating system from which the user is making request to access the system.

1.2Statement of Scope

2 REQ	Description	Rank	Tested
1	Just in mind	Desirable	NO
2	Respond within 5 seconds	Desirable	NO
3	Secure to log in	Desirable	NO
4	User friendly	Desirable	NO

5	reliability	Desirable	NO
6	Compatibility	Desirable	NO
7	Fit different platforms and on different screen sizes	Essential	NO
8	Students info: name	Essential	NO
9	Students info: user name and password	Essential	NO
10	Book info: ISBN number	Essential	NO
11	Book info: title	Essential	NO
12	Book info: author	Essential	NO
13	Students:borrow the books	Essential	NO
14	Students:view/modify the profile	Essential	NO
15	Students: create account if they have not registered	Essential	NO
16	Students: search the book by title/author	Essential	NO
17	Students:view/pay the fine	Essential	NO
18	Students:pay the bill if they do not return the books on time.	Essential	NO
19	A fine will be created if the book is not return on time.	Essential	NO
20	return not found page if the book is not exist	Essential	NO
21	Admin: Can add the book	Essential	No
22	Admin: Can delete books	Essential	No
23	Admin: Update the book inventory	Essential	No

1.3 Major Constraints

The major constraints while developing this system are: the response time for each request should be less than or equal to 5 seconds, the user interface should be user-friendly and interactive and it should respond within the required constraints described in the project manual.

1.4 Requirements

Functional

1) Every user should have an account to log in to system: Students shall use the valid username and password log in to the system to be able to access the system.

2) Users can create a new account to this system: if the account is invalid or he/she dose not have account, users can create a new account to log in the system.

3) Users can view their profile: Users can view all his/her personal information, such as first name, last name, date of birth, email, phone, gender, fine details and book details.

4) Users can modify their profile: Users can update all his/her personal information, such as first name, last name, date of birth,email, phone, gender.

5) Users (student) can search the books by title: Users can type the name of the books, The system will display the searched book.

6) Users can search the books by author: Users can type the author of the books, The system will display the searched book.

7) If not existing book is searched, a not found page will be shown: If the the book the user searched is not available, the page will show not found.

8) Users (student) can borrow the book. Users (student) can borrow the book from this system.

9) If the book is not returned on time, a fine will be create to the user account: If the the borrowing time is exceed the return date, the system will create a pay fine button automatically to let the user pay the price.

10) Users can view their fine: User can can click the view fine button to view their price how much money they will pay.

11) Users can pay their fine: Users can click the pay fine button to finish the process of borrowing book.

12) Administrator can add new books

13) Administrator can delete outdated books

14) Administrator can update the inventory, licenses for ebooks, magazines and research papers

Non-functional

1) Response time should be within SLA: the response time should be less than 5 seconds, ensure the operation efficiency of the system

2)This system should have a high security. The security is important, it can prevent the bug invade to the system also DOS attacks as well as limit access to the authorized users only.

3) This system should be highly reliable: Highly reliable emphasizes the dependability in the life cycle of the product. It can shows ability of the system to function under the stated condition as well as under peak load.

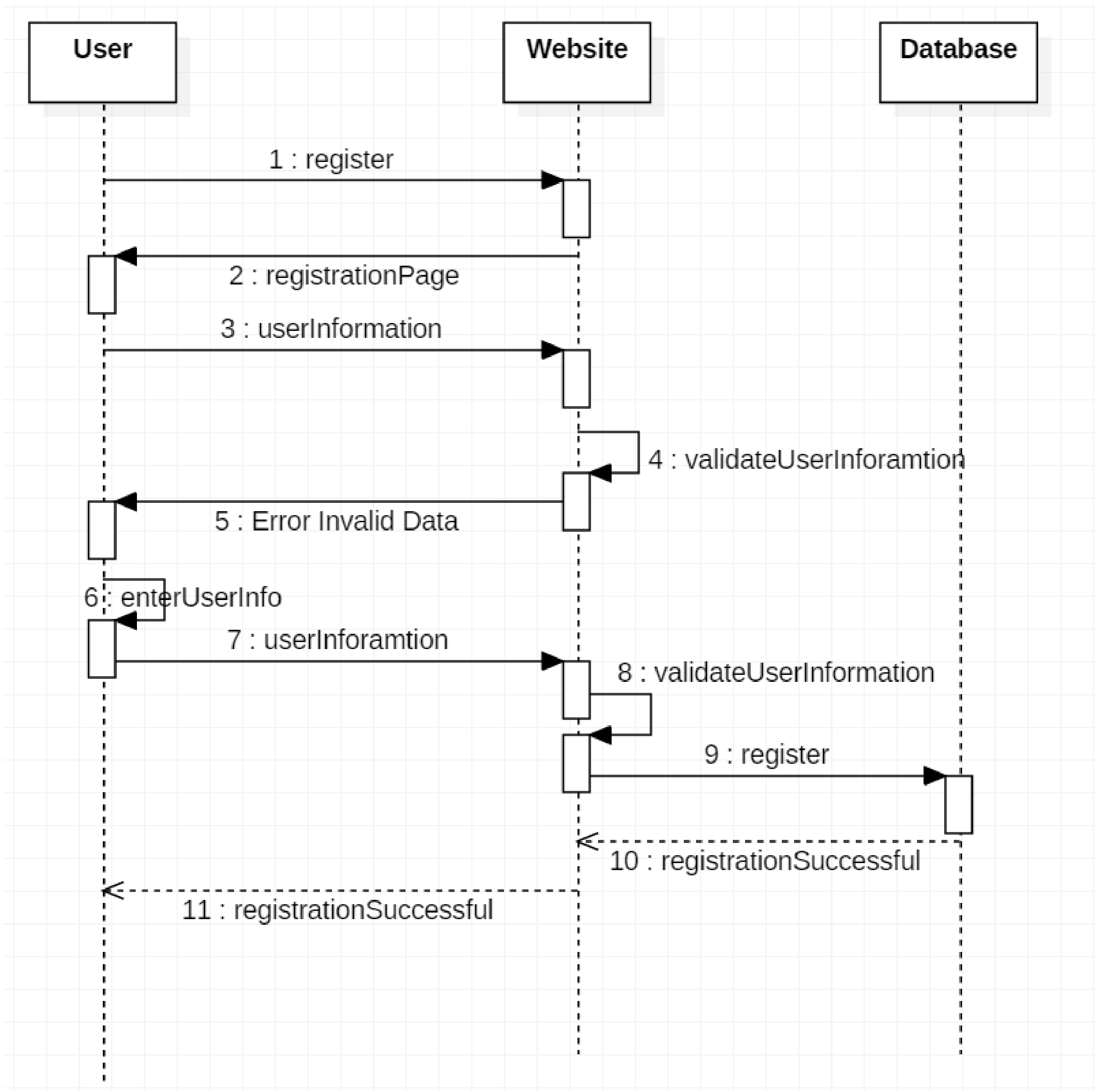
4) This system should be user friendly and intuitive. It means that anything can make it easier for novice to use the system. It does not have so many complicated steps to operate.

5) This system should have high compatibility: the system can operate on many different operating system, such as IOS, Android, windows.

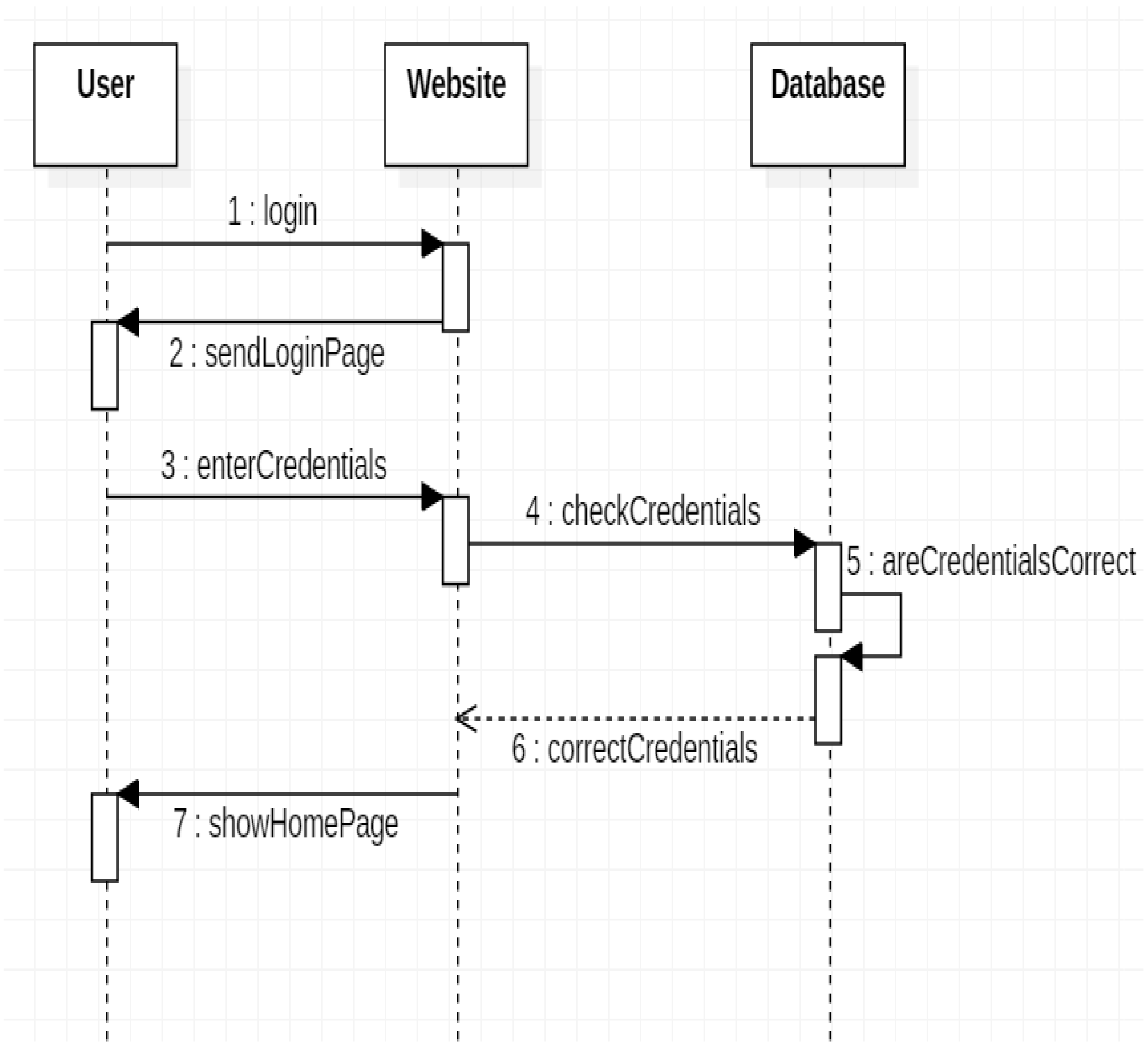
6) This system should fit different platforms on different screen, the system can display on different devices such as PC screen, smartphone and ipad and different ranges of screen resolutions

1.5 System Sequence Diagram

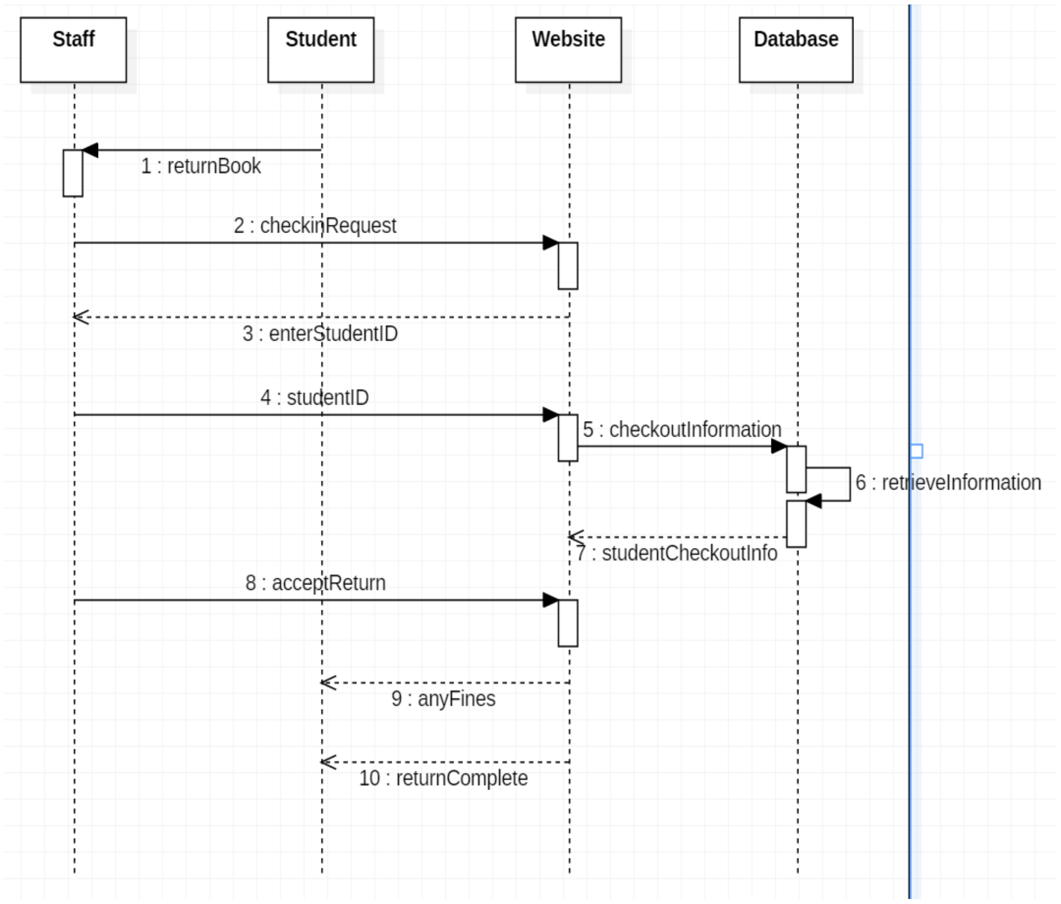
1) Register :


User
Website
Database
1 : register
2 : registrationPage
3 : userInformation
4 : validateUserInforamtion
5 : Error Invalid Data
6 : enterUserInfo
7 : userInforamtion
8 : validateUserInformation
9 : register
10 : registrationSuccessful
11 : registrationSuccessful

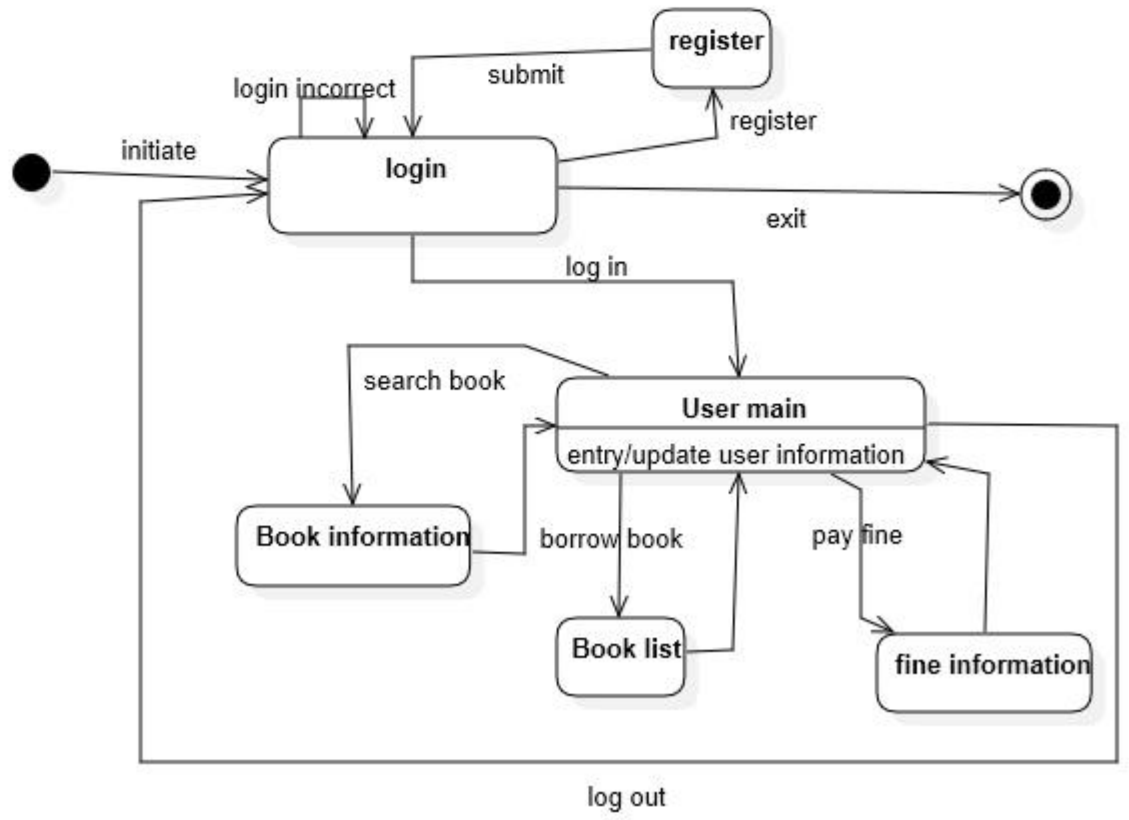
2) Login:


User
Website
Database
1 : login
2 : sendLoginPage
3 : enterCredentials
4 : checkCredentials
5 : areCredentialsCorrect
6 : correctCredentials
7 : showHomePage

3) Pay fine:


Staff
Student
Website
Database
1 : returnBook
2 : checkinRequest
3 : enterStudentID
4 : studentID
5 : checkoutInformation
6 : retrieveInformation
7 : studentCheckoutInfo
8 : acceptReturn
9 : anyFines
10 : returnComplete

1.5 State Transition Diagram



3. DATA DESIGN

The book tracking application has three layers:

a. User-Interface layer

The user interface layer is primarily made up of forms. A user will fill out a form and click on "SUBMIT". Before the "SUBMIT" button is clicked, data is temporarily stored and later will be used by functions of the business logic layer.

b. Business logic layer

The business logic layer receives the data submitted using the forms. The functions in this layer, prepare a query and send the query to the database. Upon return of query results, it sends the results back to the user-interface layer. The results will be embedded in HTML to neatly display the output of the query.

2.1 Internal Software Data Structure

The procedures in the business-logic layer have local data structures to store temporary data. Once the data is sent to the database as a query or the results are returned to the user-interface layer, these local variables in the procedures are discarded. Basically, once a form is submitted in the user-interface layer, the business-logic layer queries the database and returns the results to the user-interface layer. The business logic-layer is only temporarily holding data to complete an insert, update or delete request.

2.2 Global Data Structure

The tables stored in the database are the only global data available to the book-tracking application. The other data structures are local to each layer of the application. Data structures in the procedures of the business-logic are temporary and discarded after each request. After a connection is established with the database, insert, update, and delete can be performed.

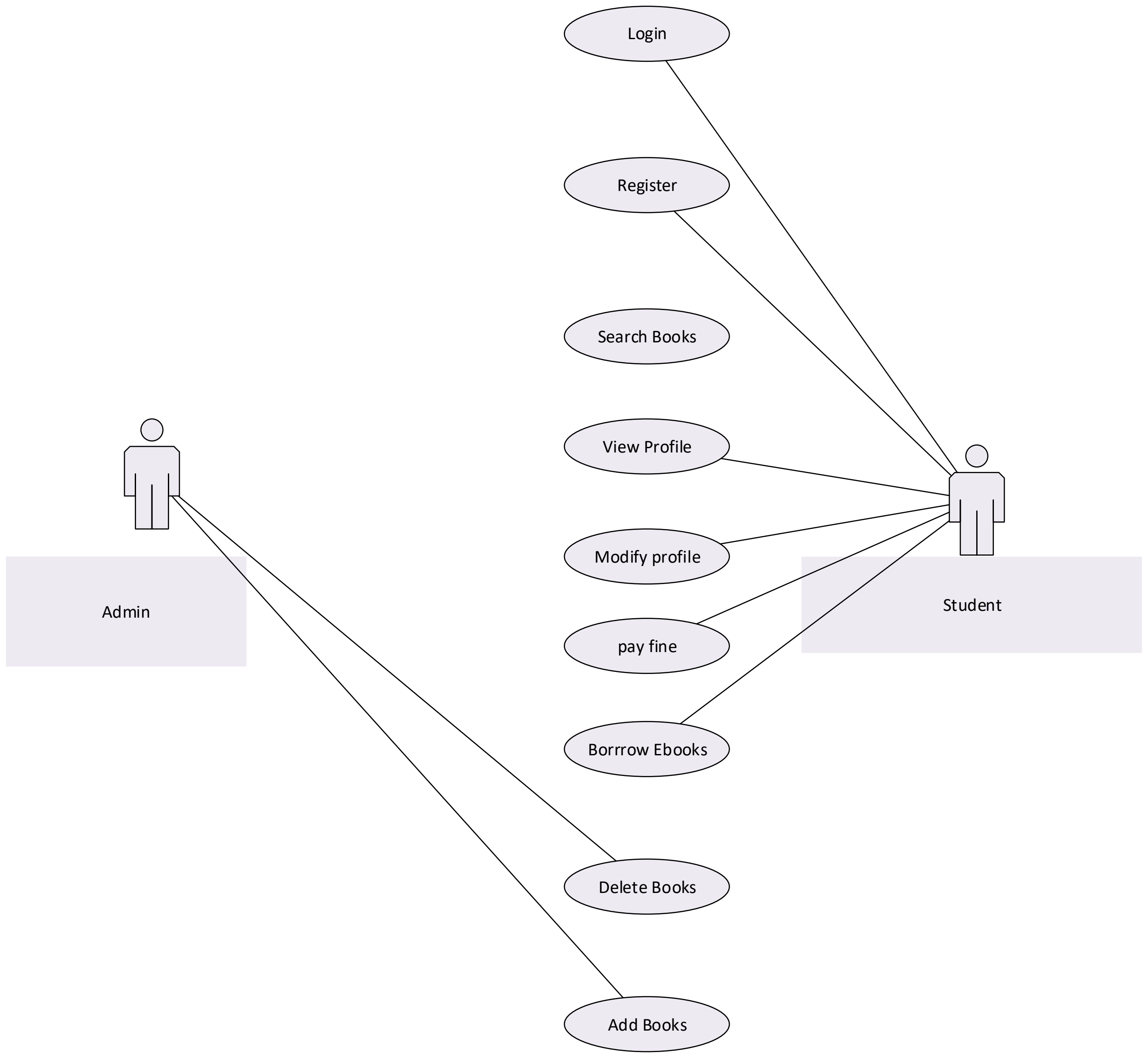
2.3 Temporary Data Structure

The only temporary data structures that can be identified are in following situations:

1)When user is filling out a form and has not yet clicked “SUBMIT”. The data is stored in the browser but if the browser is closed, the data will be lost.

2)Before “COMMIT” is executed on the database and changes become persistent. If “COMMIT” is not executed, any update will be lost.

2.4 Use Case Diagram



2.5 Use case description:

2.5.1 “Student log in” use case description

Use Case Name:	Student log in
Actor	Student
Brief Description:	Student login account
Precondition	User enter login interface successfully
Main Course:	Main course: 1) Users in the login interface to fill out the account number, password, then click on the login button; 2) The system is driven by the event of the button, converted to "homepage" interface.
Alternate course	1: If student forgets their password. 1) The student can click the "Forgot Password" button

2.5.2 “Student Registration” use case description

Use Case Name:	Student Registration
Actor	Student
Brief Description:	Student register account information
Main Course:	Main course: 1） The user clicks the "Register" button; 2） The system go to "Register" interface; 3） The user in the "Register" interface shall input user information, especially by using all information. When finished, click the "Register" button;

	4）The system is driven by the "Register" button event, adds the user registration information to the database, and then displays the prompt window "Register" successful". .

2.5.3 “Student Searches Books” use case description

Use Case Name:	Student Search Books
Actor	Student
Brief Description:	Student can successfully be able to search books clicking on search button.
Precondition	User enter login interface successfully
Main Course:	Main course: 1)User can search books by : “Book Title” , by “ISBN” , or by Course ID. 2) The system will display the searched book.
Alternate course	If book is not available , then display book not in the system .

2.5.4 “Student View Profile” use case description

Use Case Name:	Student View Profile
Actor	Student
Brief Description:	Student will be able to see his/her information on clicking view profile.
Precondition	User enter login interface successfully
Main Course:	Main course: 1)Student should click on view profile button . 2) The system will display the Student

	information.

2.5.5 “Student Modify Profile” use case description

Use Case Name:	Student Modify Profile
Actor	Student
Brief Description:	1)Student will be able to modify information
Precondition	User enter login interface successfully
Main Course:	Main course: 1)Student should click on modify profile button . 2) The system will display the Student information. 3) Student can revise and update the information.

2.5.6 “Student Pay Fine” use case description

Use Case Name:	Student Pay Fine
Actor	Student
Brief Description:	1)Student will be able to pay fine .
Precondition	User enter login interface successfully
Main Course:	Main course: 1)Student should click on pay fine button . 2) The system will display the fine to be paid 3) Student should click the pay button and system will show successful payement.

2.6 Data Model

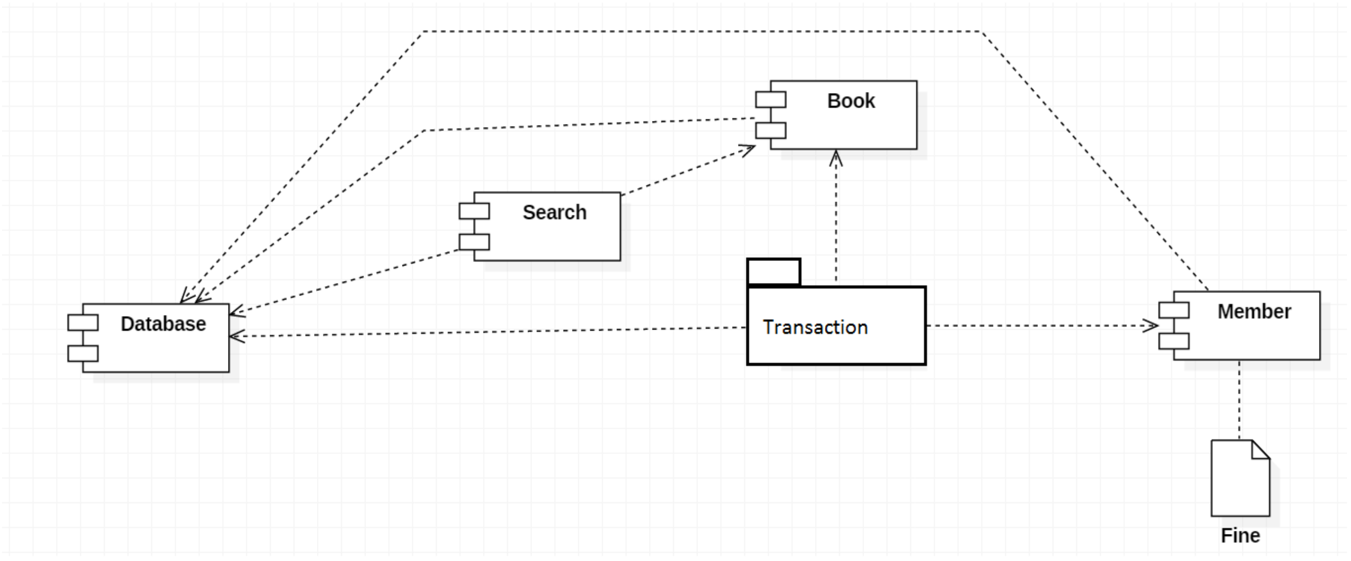
Schema Name	Attribute	Type	Constraint	Note
Student	First Name	String(30)	Not Null	
	Last Name	String(30)	Not Null	
	Date Of Birth	DATE		
	Address	String(50)	Not Null	
	Email	String(50)	Unique	Account
	Phone	Number		
	Gender	String(10)	Not Null	
	Username	String(10)	Primary Key	
	Password	String(10)	Not Null	
Course	Course ID	String(30	Primary Key	
	Couse Name	String(100)		
Book	Book Title	String(100)		
	ISBN	String(30	Primary Key	
	Couse ID	String(30)		

4. ARCHITECTURAL AND COMPONENT-LEVEL DESIGN

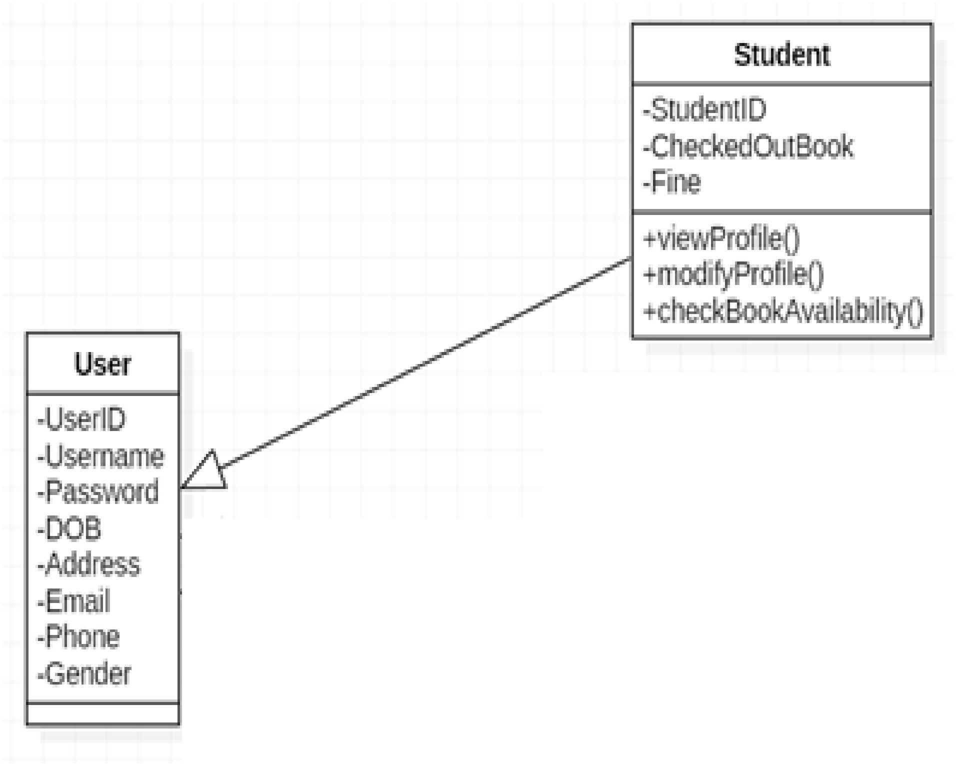
3.1 System Architecture

The system has one actor who will use the system that is the Student

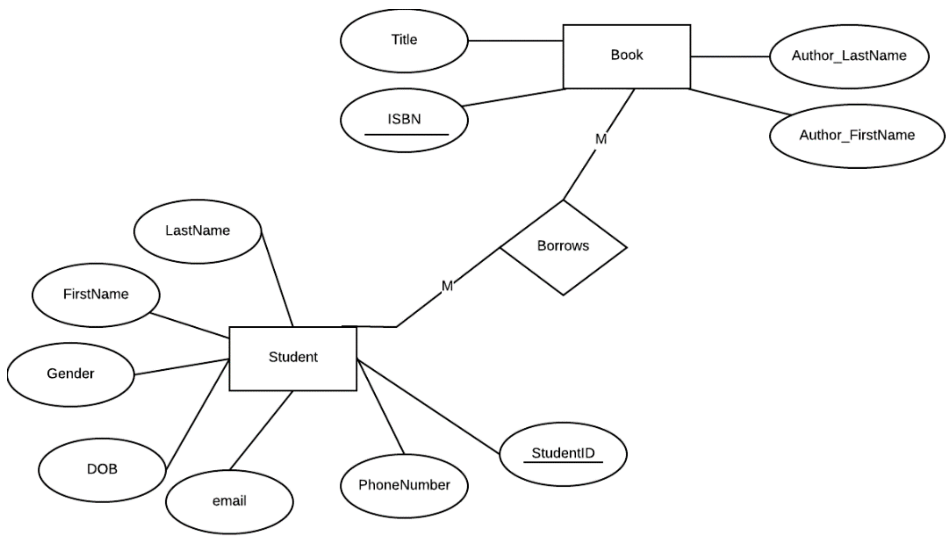
3.1.1 System Architecture Diagram



3.1.2 Design Class hierarchy for component Member


Student
-StudentID
-CheckedOutBook
-Fine
+viewProfile()
+modifyProfile()
+checkBookAvailability()
User
-UserID
-Username
-Password
-DOB
-Address
-Email
-Phone
-Gender

3.1.3 ER Diagram :



4.1 Exception Handling

Registration

Description: correct data input (Username already exist)

Input: invalid UserID (already exited in the database)

Output: account not created, UserID error message prompted(Exception Raised)

Try

{

// Do this

}

Catch Exception {

System.out.print(“ Username already exist”)}

Searching book

Description: correct title input

Input: valid title

Output: return the book

Description: multiple books share same title

Input: the shared title

Output: all the books has this title is returned

Description: correct author input

Input: valid author

Output: return the book

Description: an author has multiple books

Input: the author

Output: all the books written by this author are returned

Description: book not found

Input: information not exist in the database

Output: a book not found message is prompted

```
Try
{
// Do this
}
Catch Exception {
System.out.print( "  book not found message")}
```

Borrowing the book

Description: book available

Input: An available book is selected.

Output: the book is transferred to the user account

Description: book not available

Input: The book is not available

Output: The order will be dropped.(Exception Raised)

```
Try {
// Do this
```

}

Catch Exception {

System.out.print(“ Book not available”)}

Paying fine

Description: paying fine

Input: valid selection

Output: fine paid

Description: not correct payment

Input: invalid selection

Output: no changes made to the user account(Exception raised)

Try {

// Do this

}

Catch Exception {

System.out.print(“ not correct payment method”)}

Course search

Description: correct cousreID input

Input: valid courseID

Output: return the course

Description: incorrect title input

Input: invalid courseID

Output: a course not found message prompted(Exception raised)

Try {

// Do this

}

Catch Exception {

System.out.print(“ course not found message”)}

4.2 Expected software response

Students use case test

Title	**Example**	**Result**	**Procedure**
Sign in 1	• User: xxxx • Password:xxxxxx • Input:xxxxxx	Success	Sign in on the user console page
Sign in 2	• User: xxxx • Password:xxxxxx • Input:yyyyyy	fail hints: "wrong password or username"or "no data"	Sign in on the user console page
Sign up 1	input "Username, phone No, , password, email ,date of birth, First name ,last name	Success	Hit the 'Sign up' button
Sign up 2	fail to input input "Username, phone No, , password, email ,date of birth, First name ,last name"	Password will be mailed to email id , user need to enter username and emailid	Hit the 'Sign up' button
Search for Books	The search book page opens	Success	Hit the Search book by name , isbn and course Id
Borro w ebook	Book will be borrowed by student	success	Hit " book borrow button"
Updat e inform ation	Update information	Success	Hit the 'Update information '' button
Pay fine	Pay fine	Success	Hit the "Pay fine button"
Pay fine	fail to pay fine	Payment info is wrong	Hit the "Pay fine button"

Use cases is a technique that helps to identify test cases that exercise the whole system on a transaction by transaction basis from start to finish. Some of the examples would be:

4.3 Expected Software Response

The purpose of system testing is to find the program error and correct the error in time. Each test case produces a corresponding "testing result". If it does not match the "expected result", it indicates there is an erroring the program and needs to be corrected. Examples:

- Checking if the display information is correct
- Checking if the response time is less than 5 seconds
- Checking if data entry is corrected or not

4.4 Recommendations on how this project could be improved to be a more effective learning experience.

We can add several feature to the app that can make the library management more effective for students as well as for administrator.

Following feature can be added:

● Online Fine Payment and integration with Payment Gateways

● Students can request the books that are not available in the library

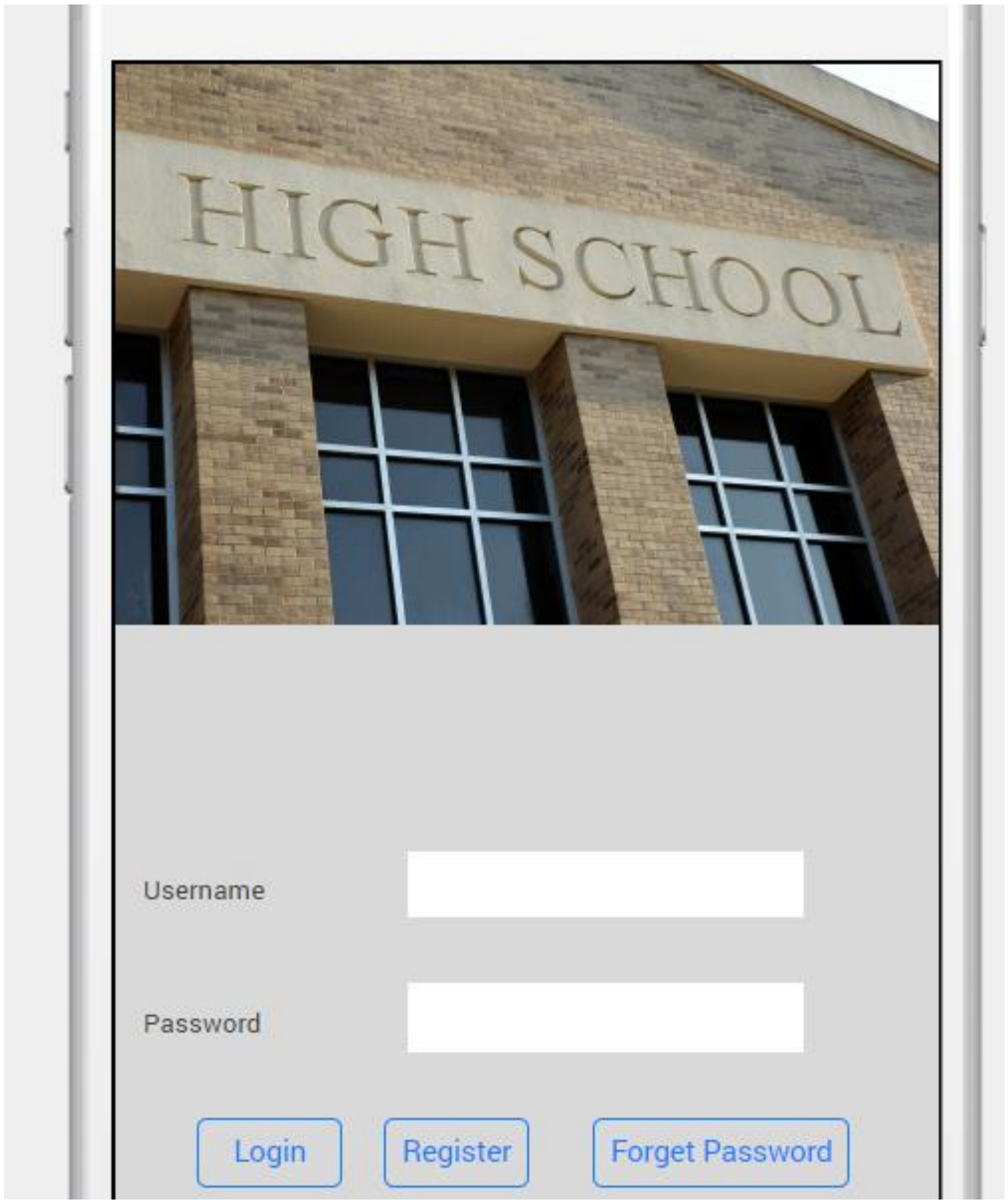
● An automated request order is created and sent to the preferred vendor based on approval from administrator and Teachers teaching the class

● Teachers can also create a new book request and is directly placed to preferred vendor

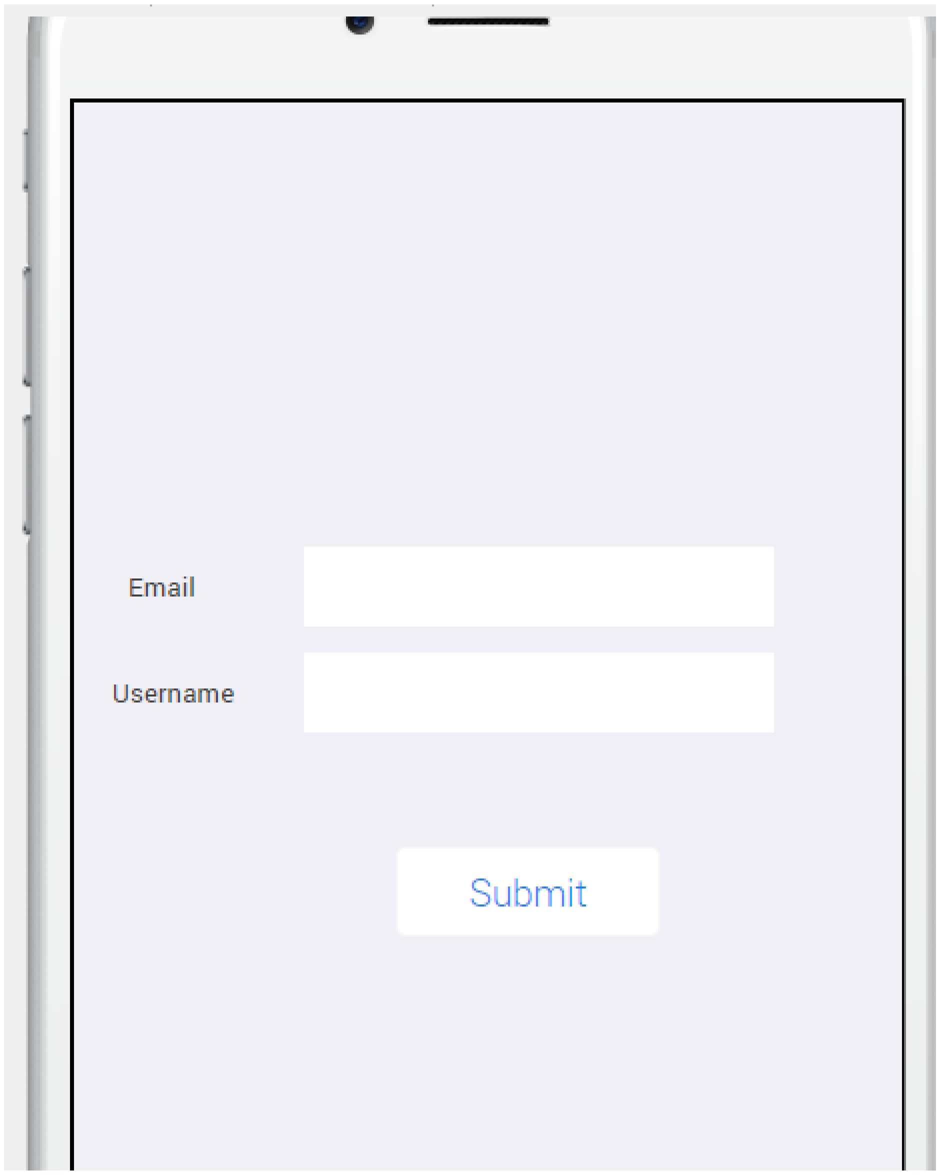
We can also try and follow agile process during this phase and have weekly calls, retrospection to better track the progress.

4.5 Prototype Screenshots

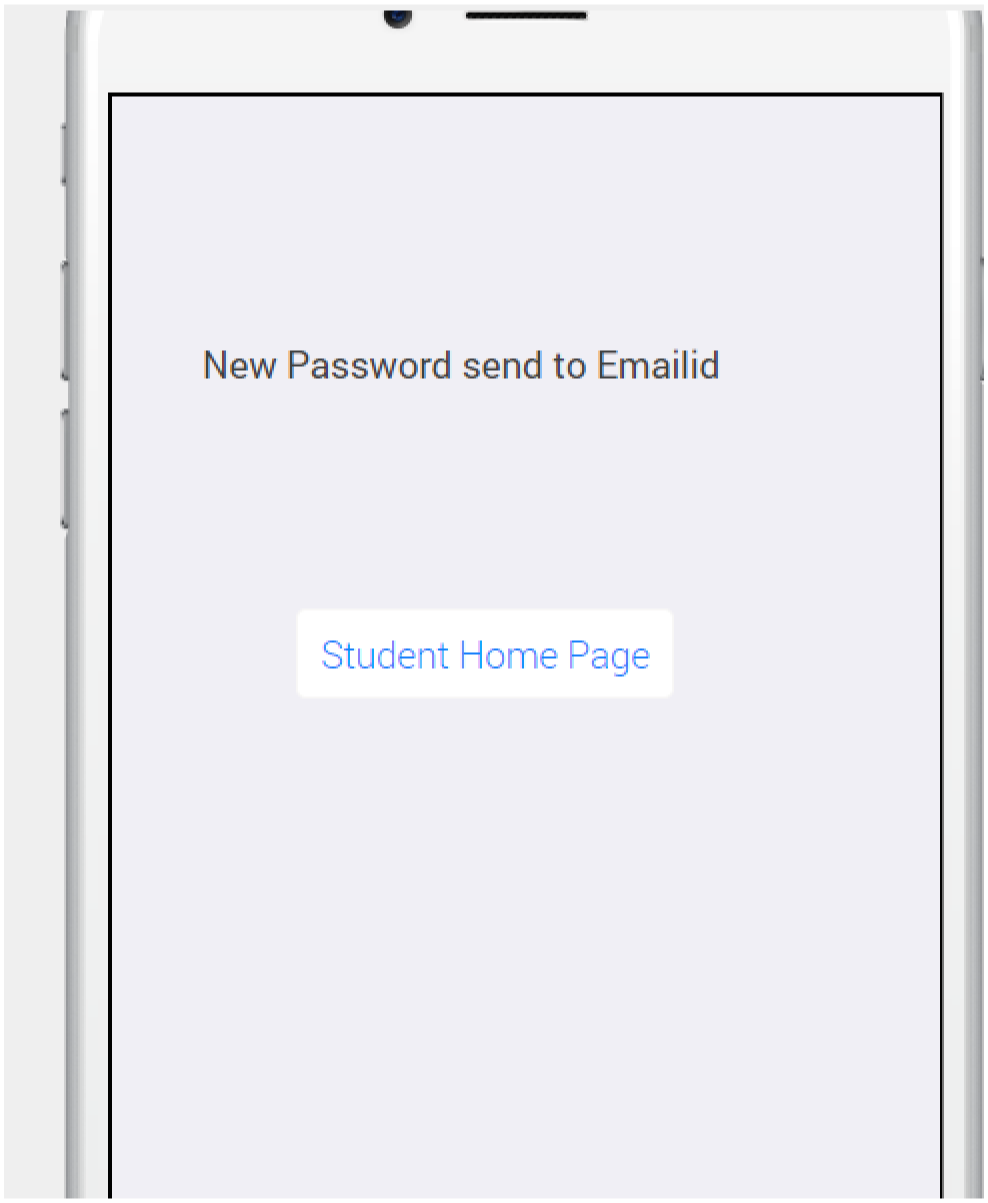
1)Login: Student can login by providing username and password .



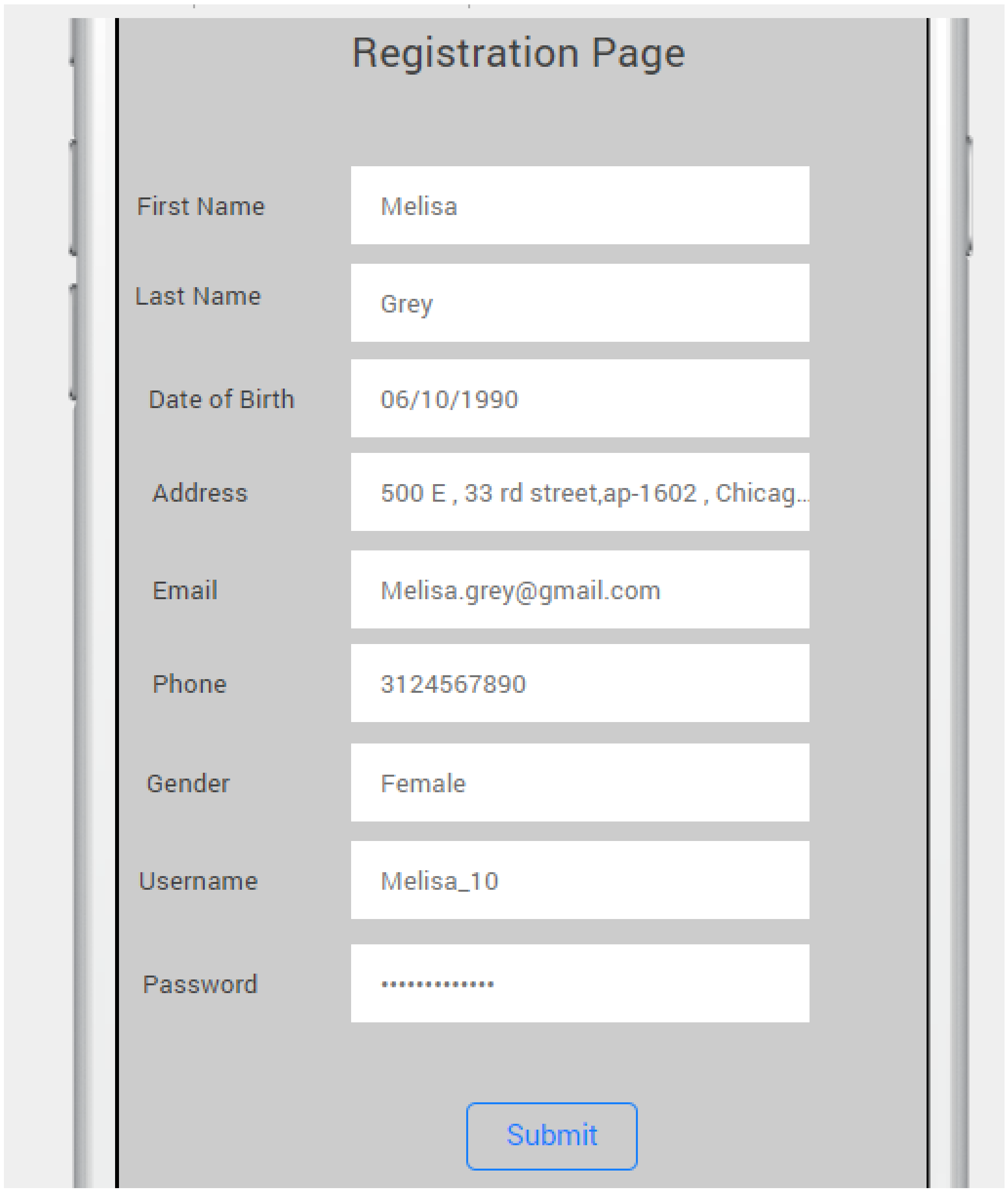
2) Forget password: If student forget password then click on forget password button and new page will be opened to enter username and emailed.



3)New password send to email id


New Password send to Emailid

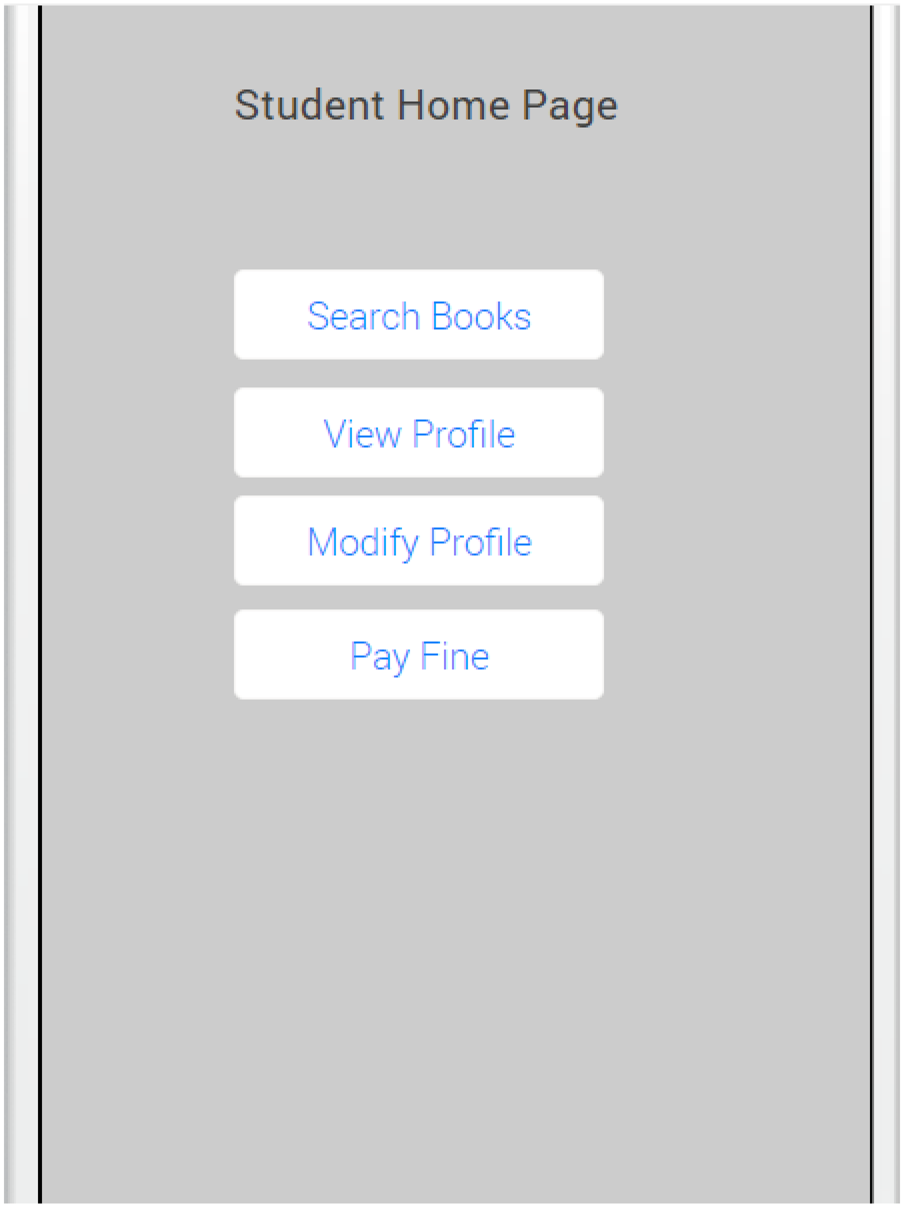
4) Register: Student can registered by entering his/her details in provided blocks.



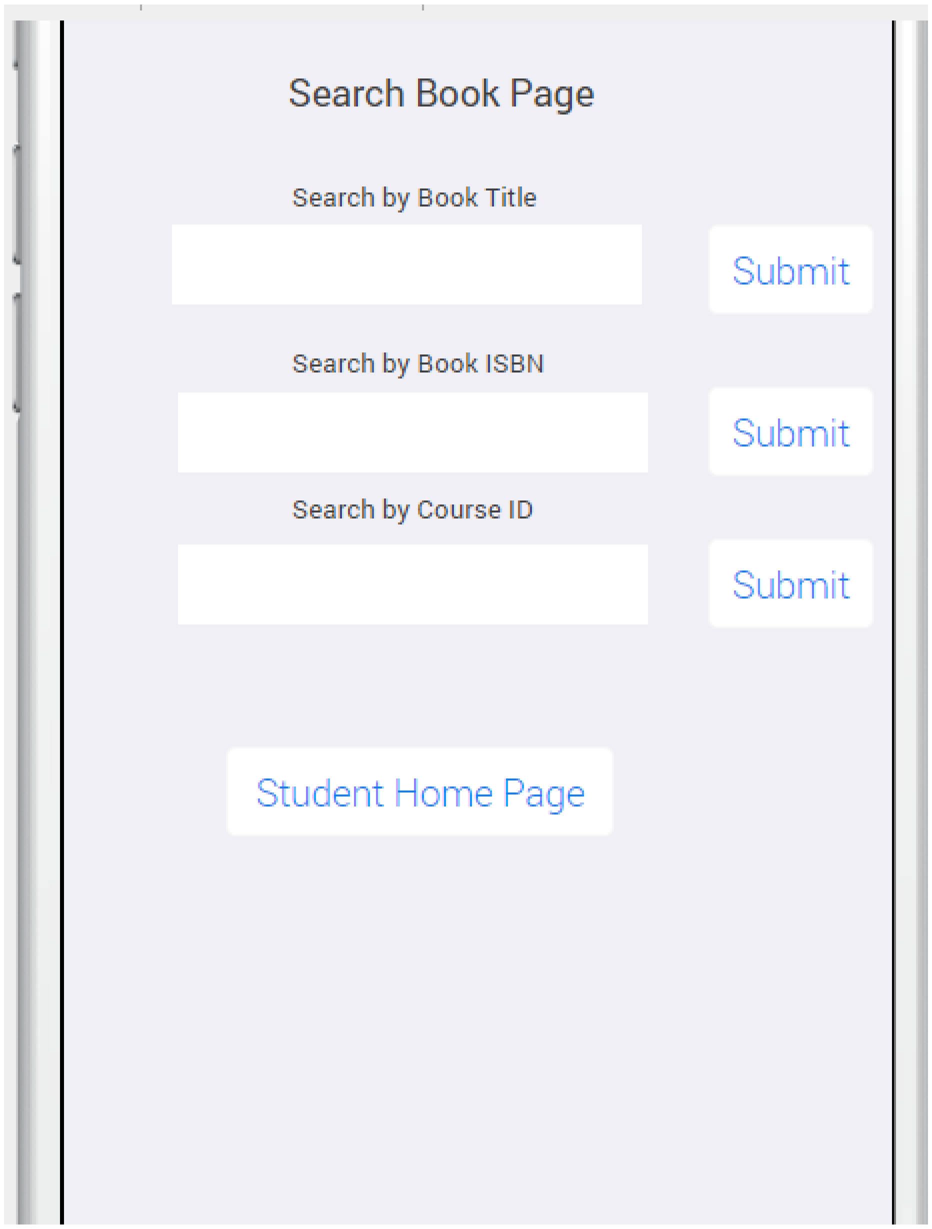
Registration Page

First Name	Melisa
Last Name	Grey
Date of Birth	06/10/1990
Address	500 E , 33 rd street,ap-1602 , Chicag...
Email	Melisa.grey@gmail.com
Phone	3124567890
Gender	Female
Username	Melisa_10
Password	•••••••••••••

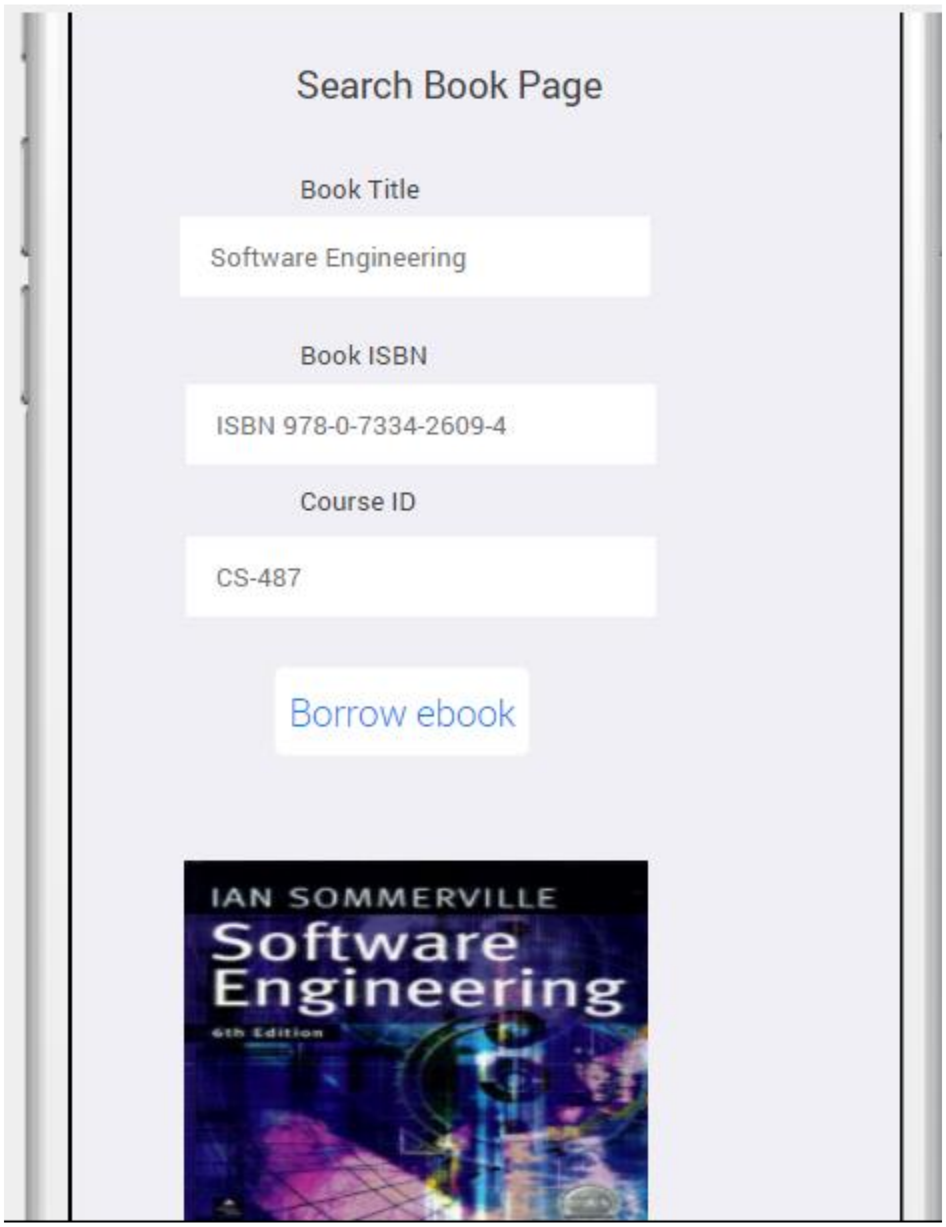
5) Student home page : will contain search books, view profile ,modify profile and pay fine buttons



6)Search Books: will contain search by name, course id and ISBN



7) Search book: student can borrow ebook by clicking the button .



Search Book Page

Book Title

Software Engineering

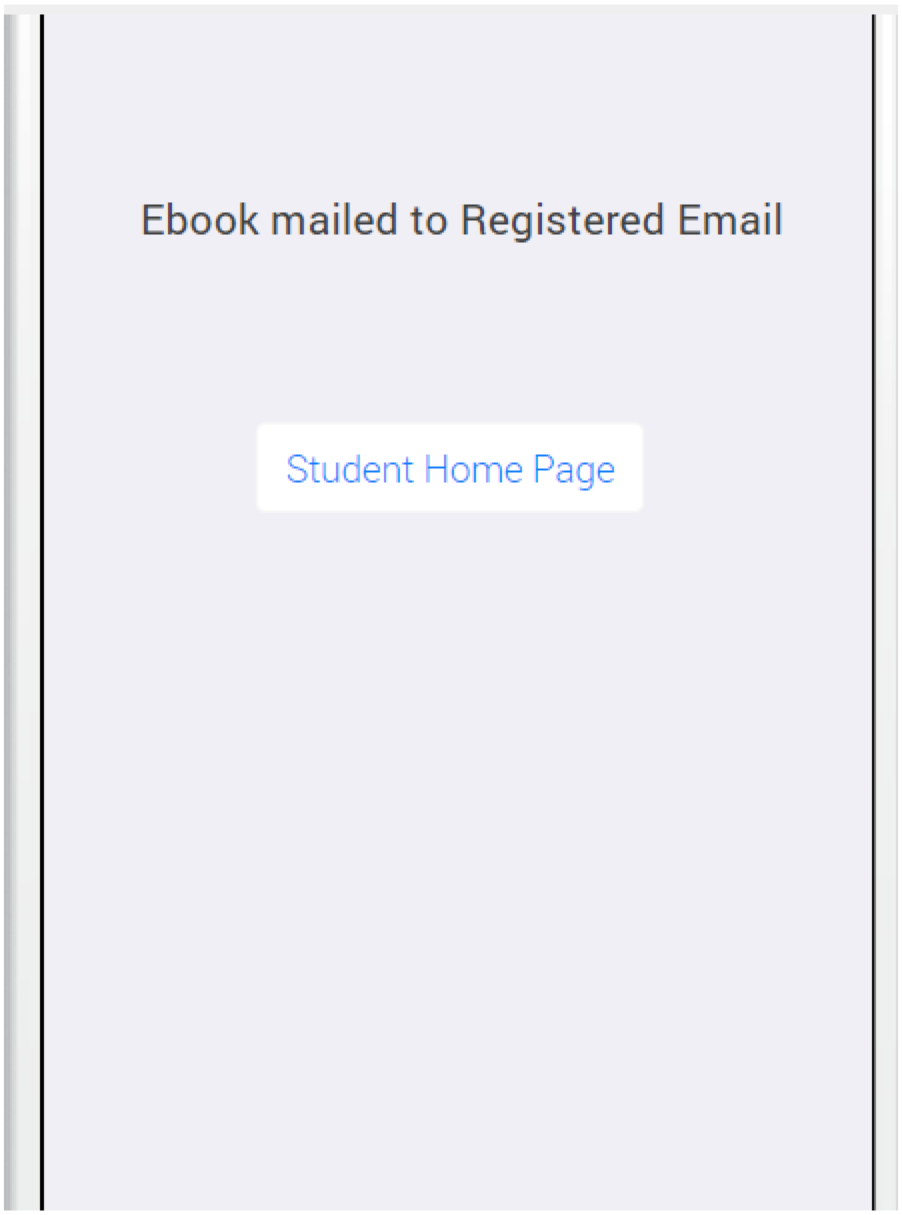
Book ISBN

ISBN 978-0-7334-2609-4

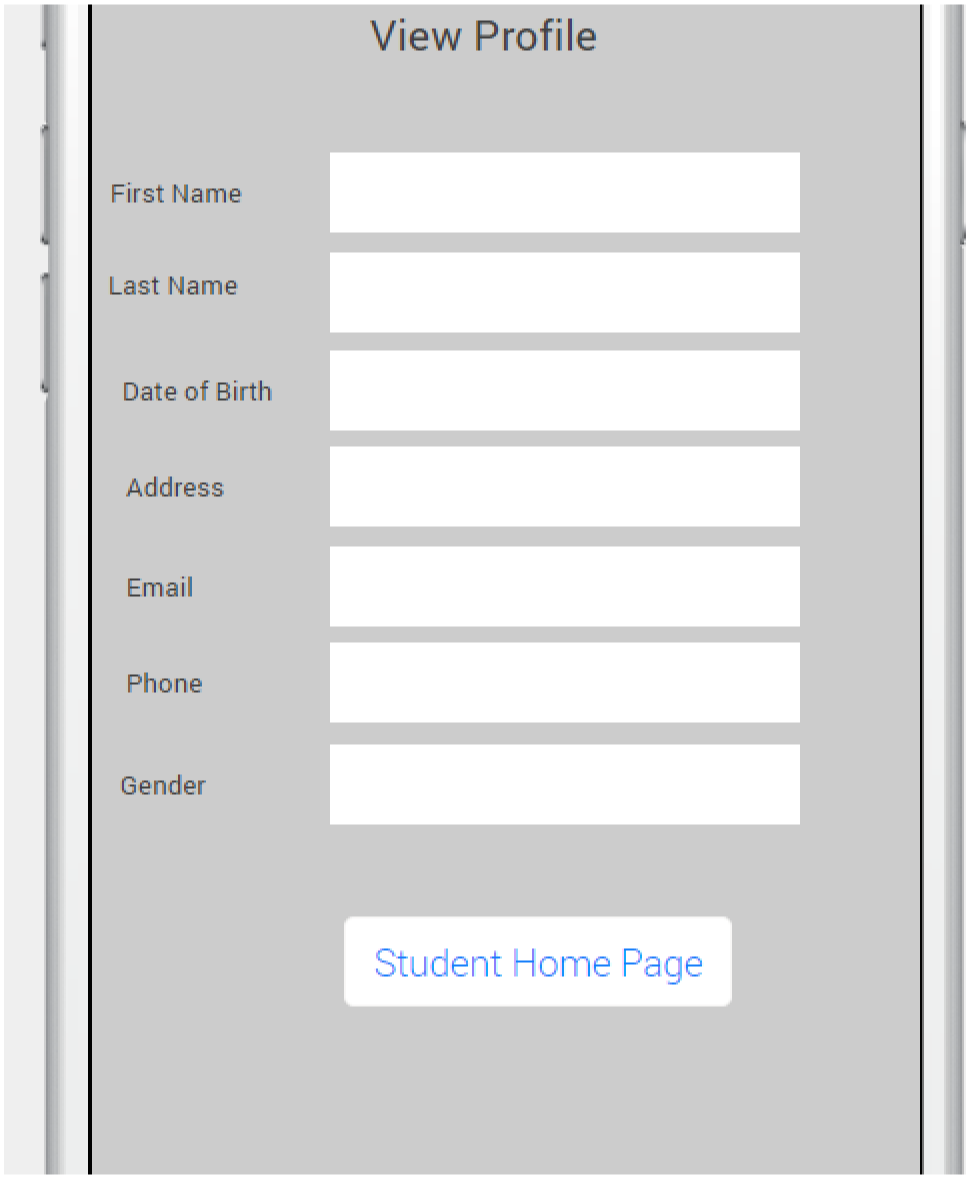
Course ID

CS-487

8) Book borrowed: ebook will be borrowed successfully

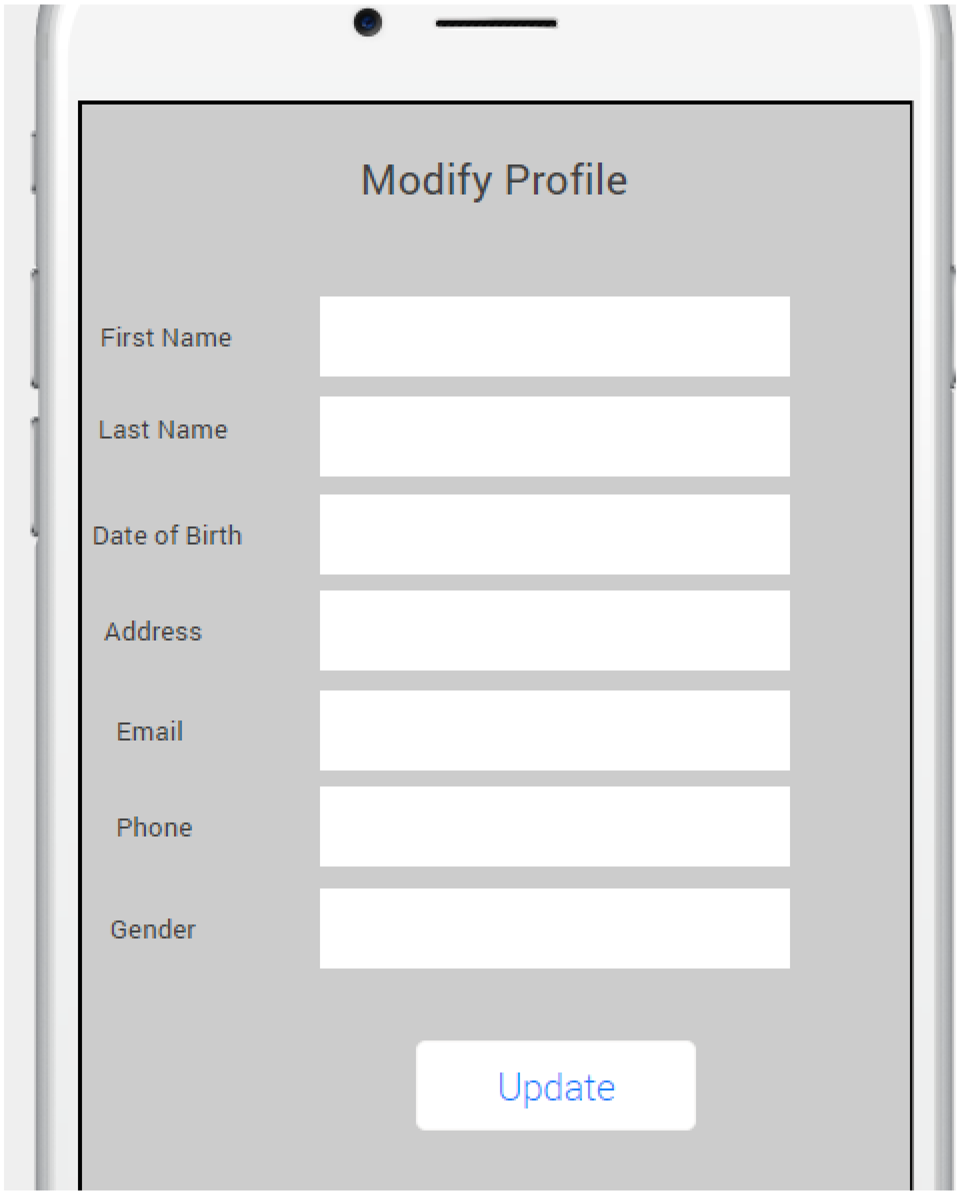

Ebook mailed to Registered Email

9)) View Profile: Student can view his/her profile by clicking on view profile button.

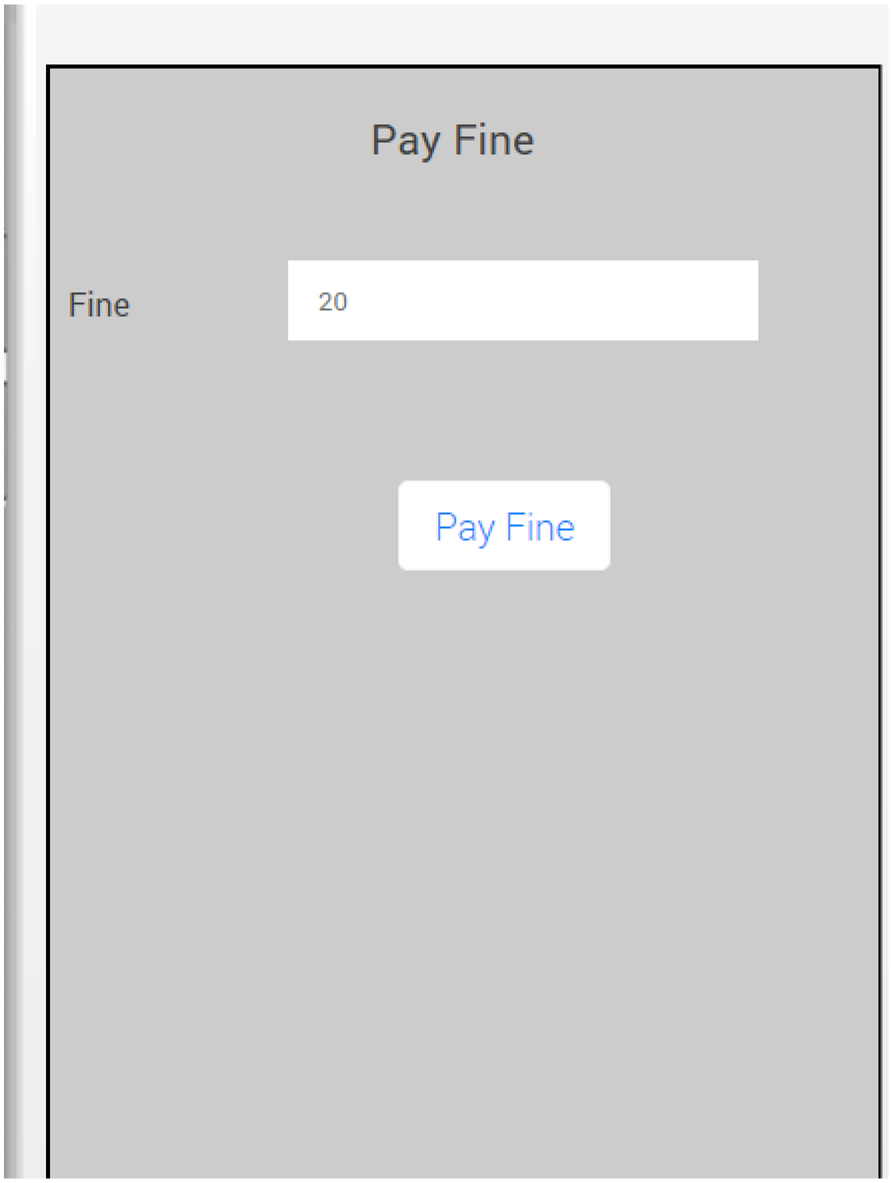

View Profile

First Name	
Last Name	
Date of Birth	
Address	
Email	
Phone	
Gender	

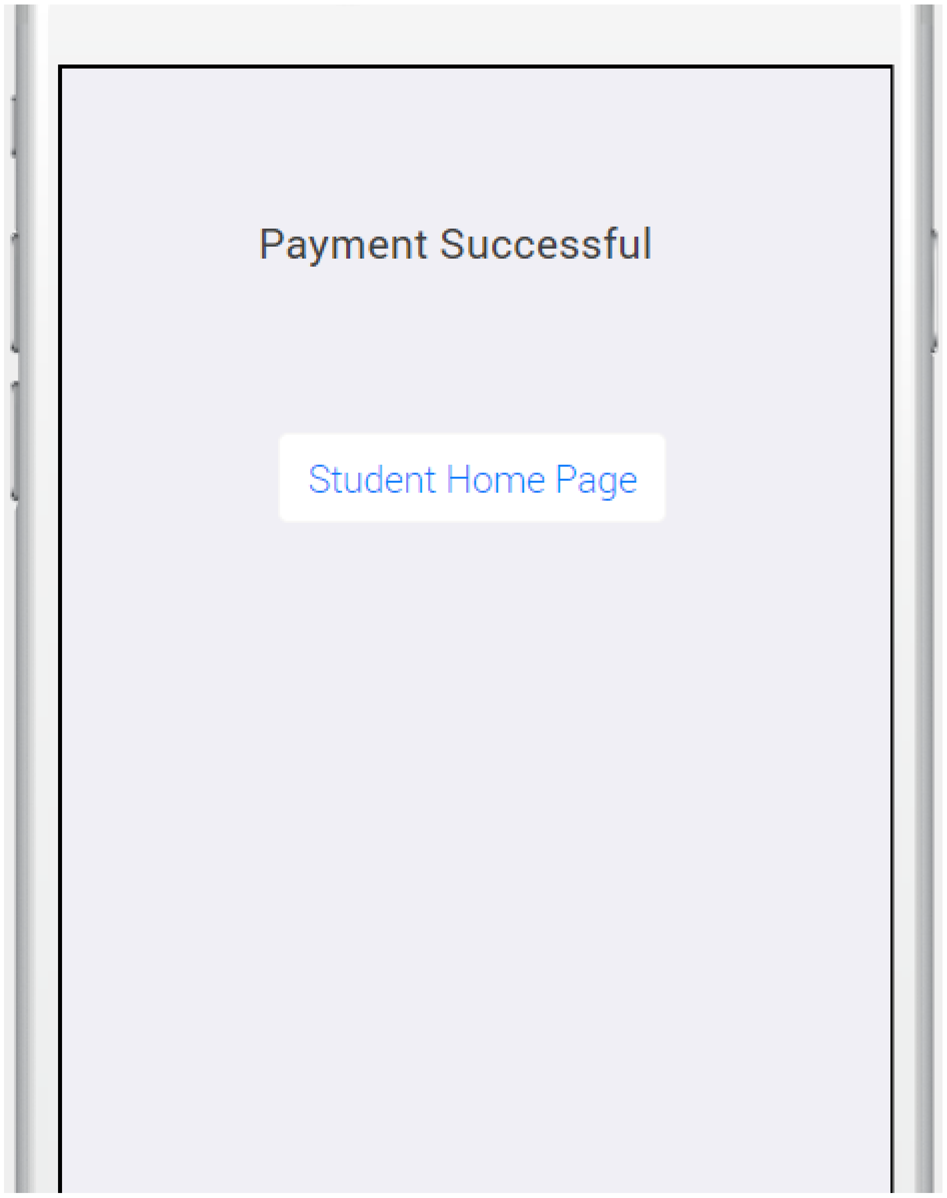
10)Modify profile: user can modify profile by clicking on modify button



11) Pay fine: fine will displayed on the app screen .

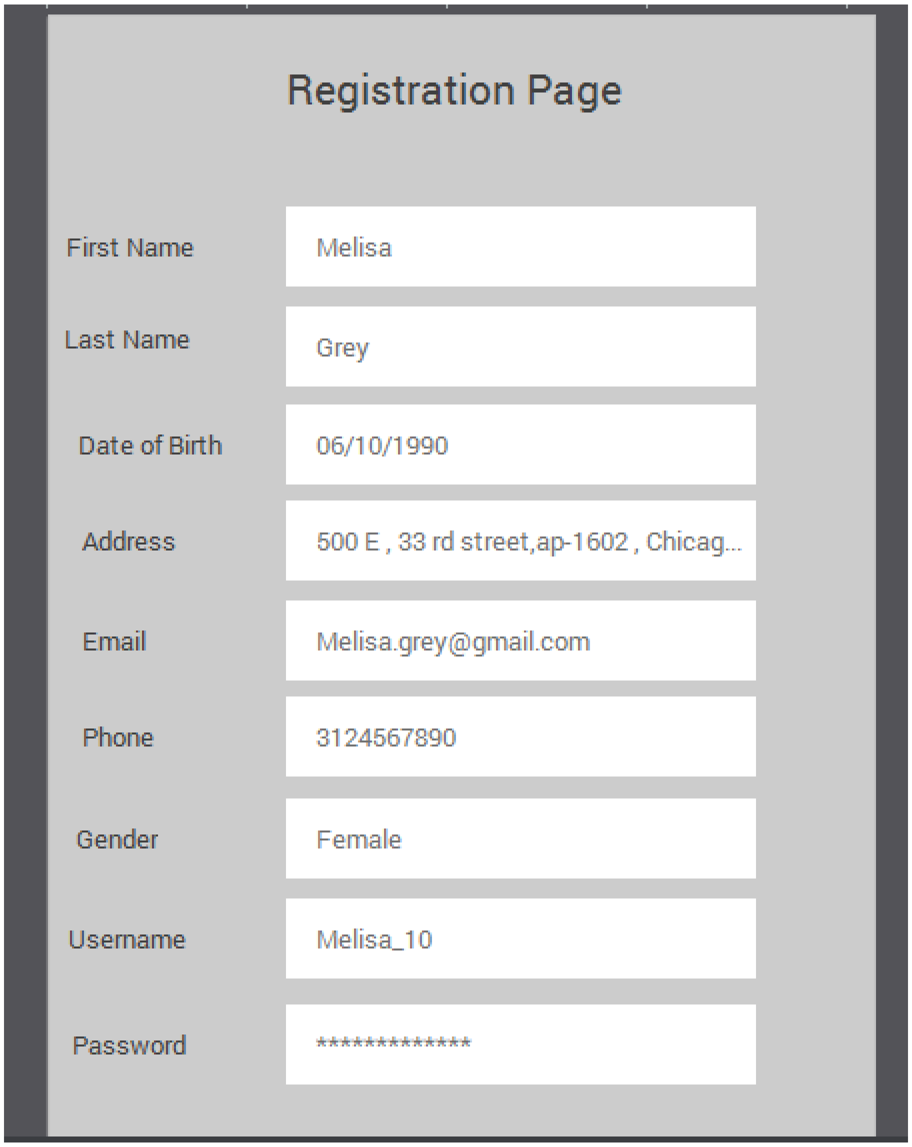


12) Pay successful: Payment will be successful and will be displayed on the screen.



4.6 Exception Handling Screenshots

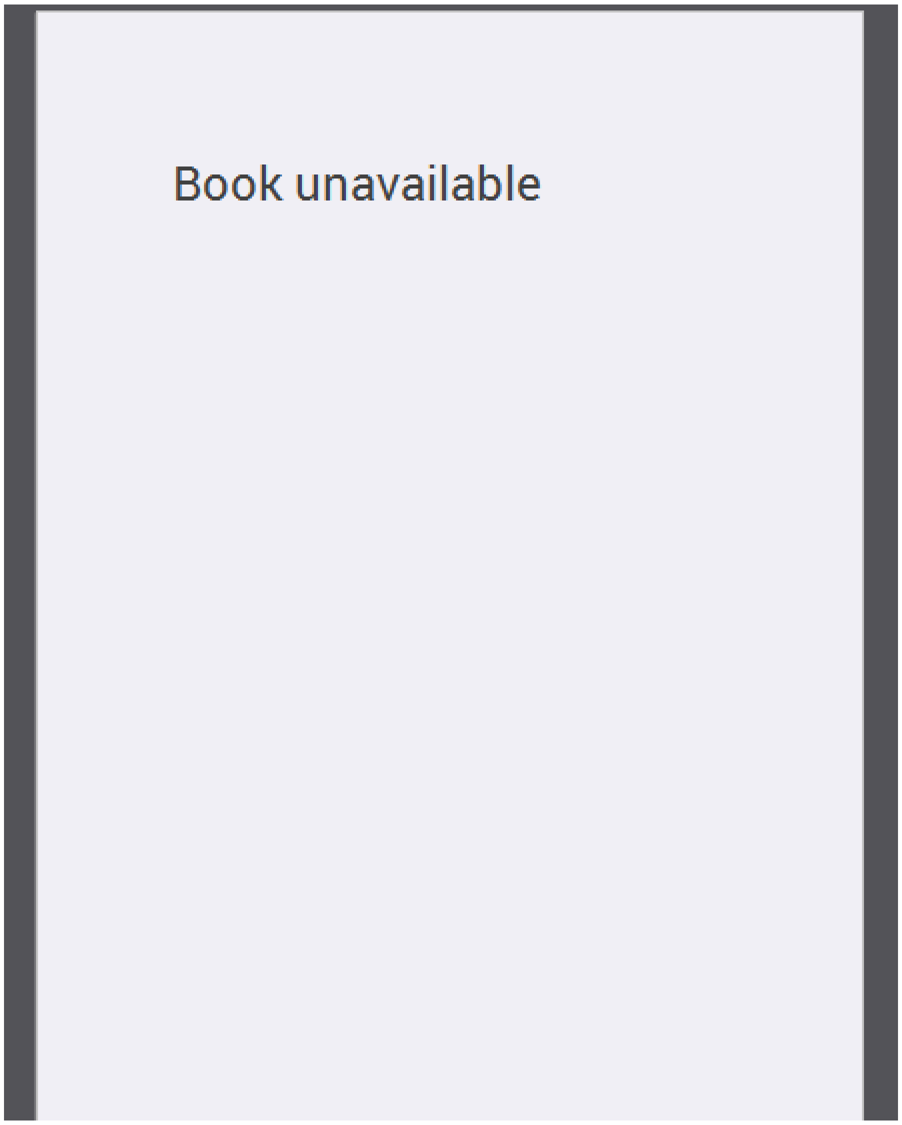
1)Username already exist : message will be displayed like username already exist , please enter another username .(Melisa_10 already exist)



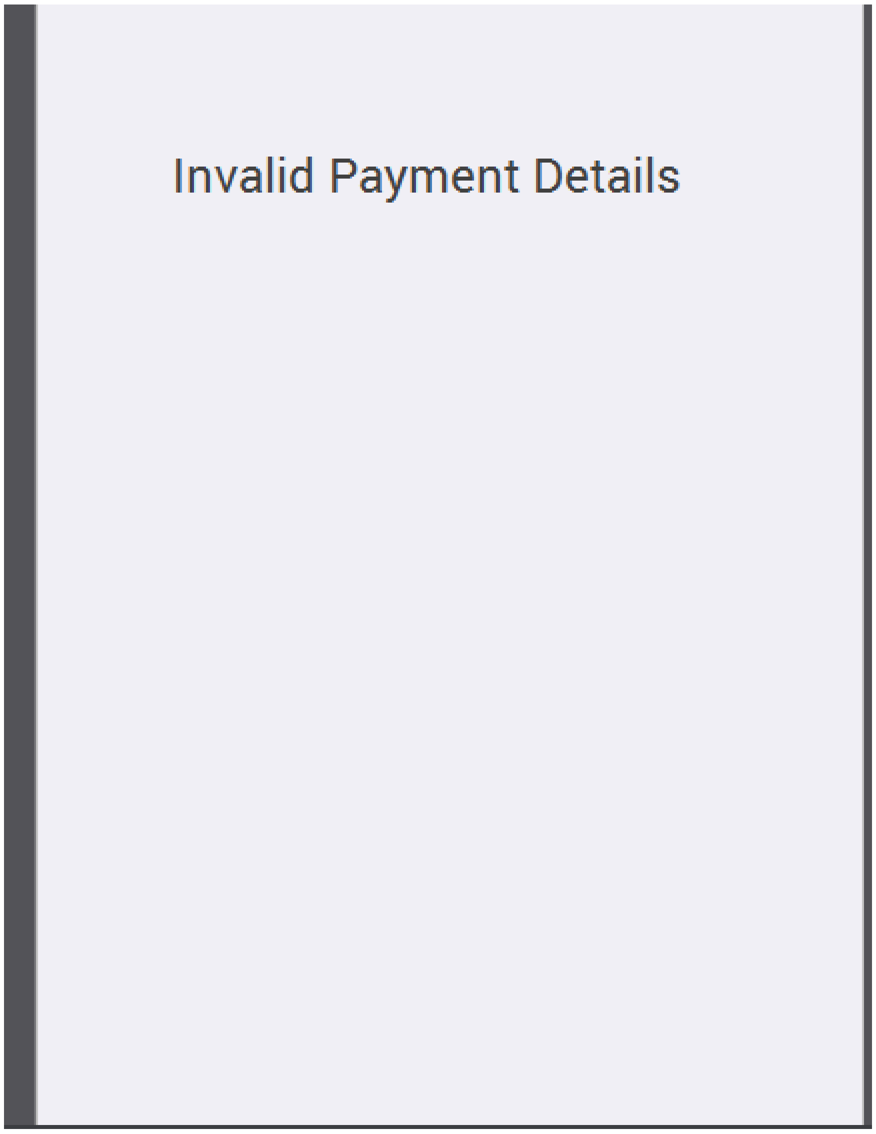
Registration Page

First Name	Melisa
Last Name	Grey
Date of Birth	06/10/1990
Address	500 E , 33 rd street,ap-1602 , Chicag...
Email	Melisa.grey@gmail.com
Phone	3124567890
Gender	Female
Username	Melisa_10
Password	*************

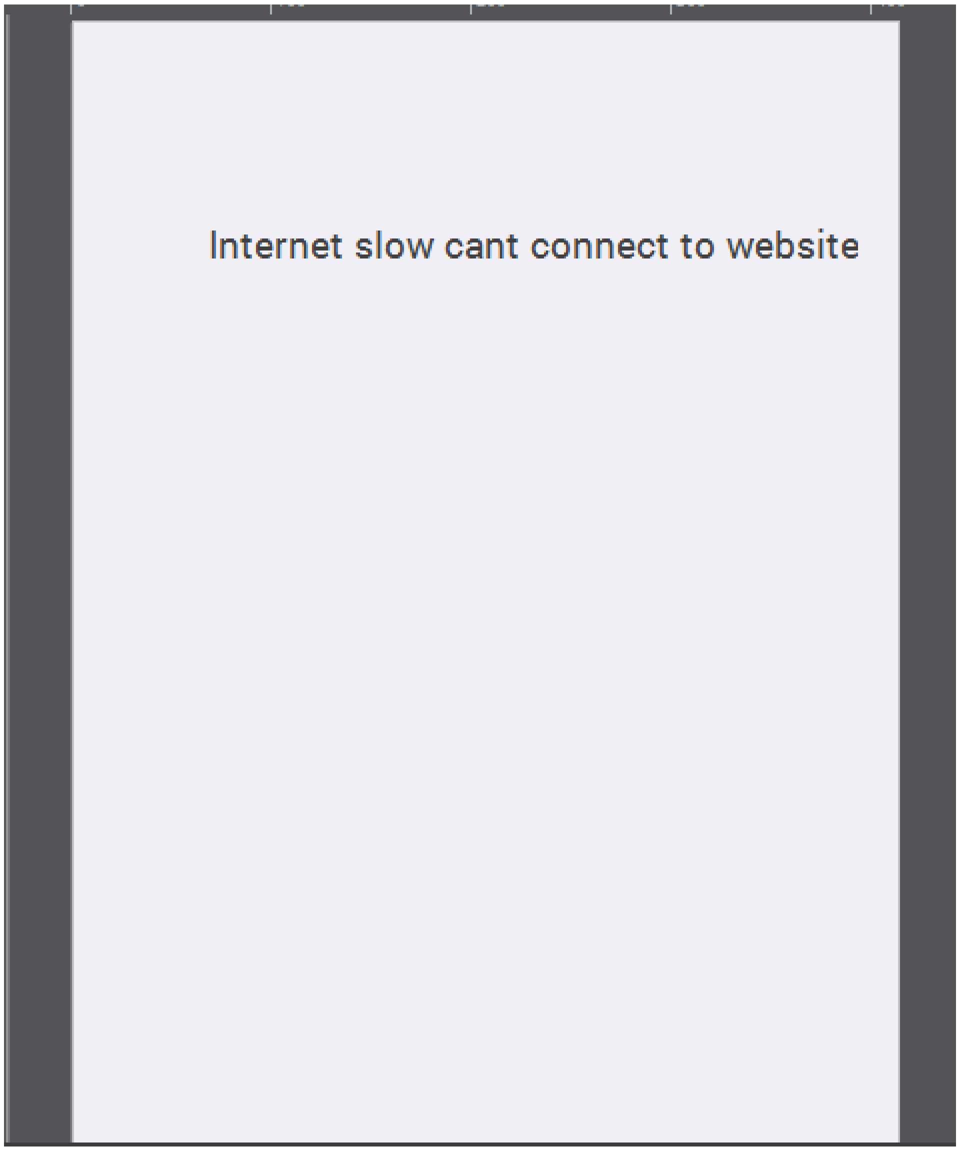
2)Book is already Borrowed : if book is already borrowed or unavailable then unavailability of book message will be displayed.


Book unavailable

3) Not correct payment method: card details may be wrong , cvv number or date of expiry.


Invalid Payment Details

4)Cannot login: Internet is too slow or not working .



Internet slow cant connect to website