

Database Systems CSCI760

Final Project Report

Library Management System

Group 13:

Pratikkumar Patel (1053030)

Kishenkumar Gabani (1116169)

Apoorva Kanchu (1049616)

Table of Contents

Introduction.....	2
ER MODEL	2
Relational Database Design	8
Implementation of Database and SQL Queries	8
Application Design	13
Screenshots	14
Team member's work.....	27

1. Introduction:

The main purpose of the project is managing the entire activity of a library. The key is to keep track of all the information related to every book in the library. Information such as: the cost of the book, the author, the title, the total number of books available and so on. This system is much easier to work with rather than writing down each detailed description. The system will be a useful tool that tracks all the books in the library and handle the necessary information.

We were able to build an android application that helps to reduce the manual work for managing the books, its authors and readers. It helps to track all the information about books, readers and if the books issued have been issued or not. It helps in searching facilities based on factors like: books, library branch, readers, authors and publishers. It helps in increasing the efficiency of the managing books, its readers, authors and so on. It also makes it possible to monitor the information and transactions of the borrower.

2. ER Diagram:

1. Book: This entity contains information pertaining to books. Attributes of this entity are:

ISBN (primary key)

Bookid

Title

Publisherid

Publication date

2. Publisher:

Publisher Id (primary key)

Publisher Address

Publisher Name

3. Author: Each book can have more than one author. Attributes of this entity are:

Author Id (primary key)

Author Name

4. Branch:

Branchid (primary key)

Name

Location

5. Reader:

Readerid (primary key)

Name

Address

Phone

6. Borrow:

Bookid (primary key)

Isbn (primary key)

Readerid

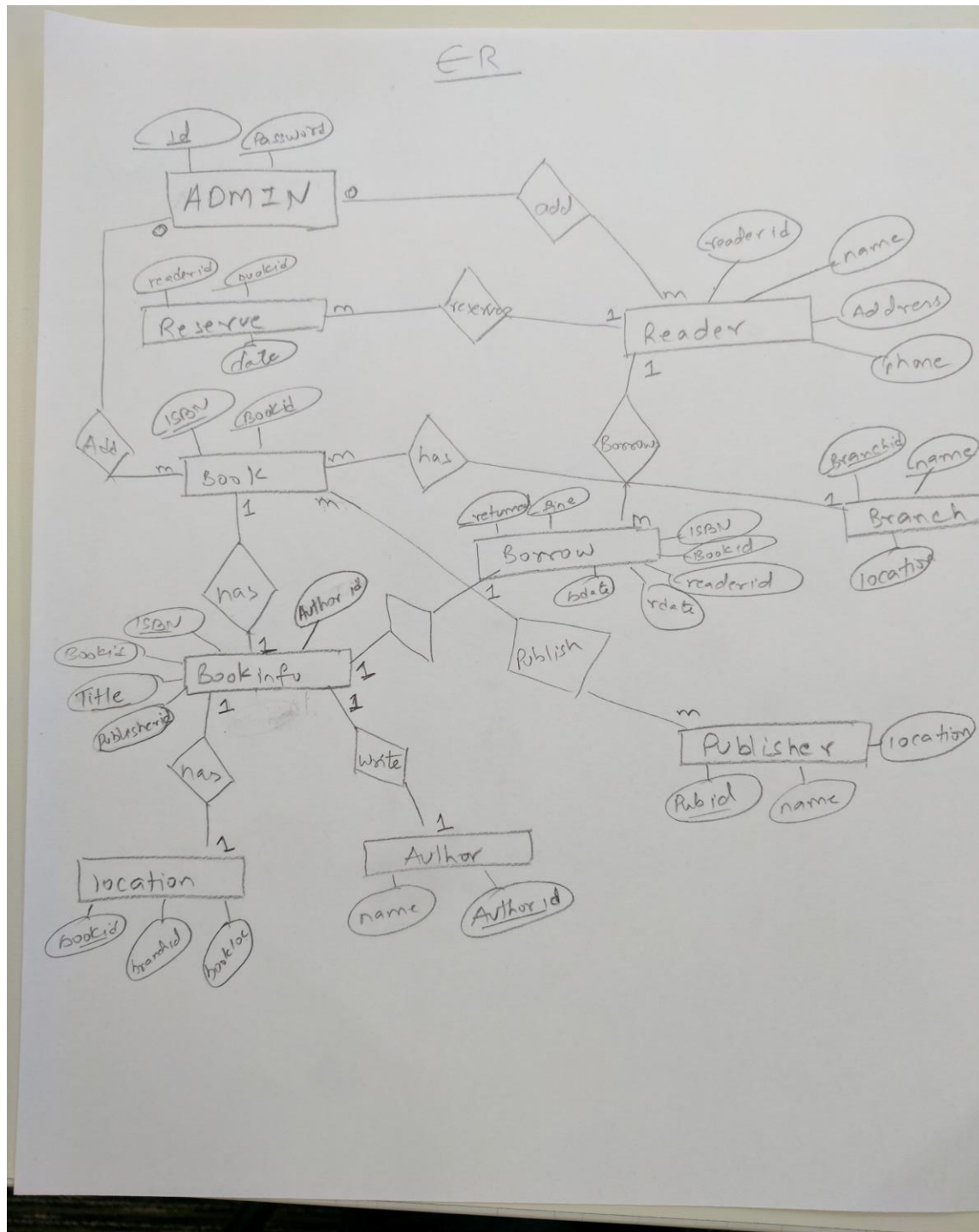
Bdate

Rdate

Branchid

Returned { if 0 = book not returned, if 1 = book returned }

ER- Diagram



3.Logical Design of the database:

Table t_branch

Columns		
* id	int	AUTOINCREMENT
* branch_id	int	
branch_name	varchar(100)	
location	varchar(100)	
address	varchar(500)	
pincode	int	
Indexes		
pk_t_branch	ON id	
branch_id	ON branch_id	

Table t_publisher

Columns		
* id	int	AUTOINCREMENT
* name	varchar(200)	
address	varchar(500)	
Indexes		
pk_t_publisher	ON id	
name	ON name	

Table t_book_author

Columns		
* book_id	int	

Columns		
*	author_id	int
Indexes		
	pk_t_book_author	ON book_id, author_id
	author_id	ON author_id
Foreign Keys		
	t_book_author_ibfk_2	(author_id) ref t_author (author_id)
	t_book_author_ibfk_1	(book_id) ref t_book (book_id)

Table [t_book_borrows](#)

Columns		
*	id	int AUTOINCREMENT
	reader_id	int
	borrow_date	date
*	expected_return_date	date
	return_date	date
	borrowed_book_id	int
	fine	float(4, 2) DEFAULT 0.00
Indexes		
	pk_t_book_borrows	ON id
	borrowed_book_id	ON borrowed_book_id
	reader_id	ON reader_id
Foreign Keys		
	t_book_borrows_ibfk_2	(borrowed_book_id) ref t_book_copies (id)
	t_book_borrows_ibfk_1	(reader_id) ref t_reader (reader_id)

Table [t_library](#)

Columns		
* id	int	AUTOINCREMENT
* lib_id	int	
lib_name	varchar(200)	
branch_id	int	
Indexes		
pk_t_library	ON	id
lib_id	ON	lib_id
branch_id	ON	branch_id
Foreign Keys		
t_library_ibfk_1	(branch_id)	ref t_branch (branch_id)

Table [t_reader](#)

Columns		
Readers		
* id	int	AUTOINCREMENT
reader_id	int	
name	varchar(100)	
address	varchar(300)	
phone_number	varchar(15)	
Indexes		
pk_t_reader	ON	id
reader_id	ON	reader_id

Table [t_author](#)

Columns		
Authors		
* id	int	AUTOINCREMENT
* author_id	int	
* name	varchar(150)	
details	varchar(500)	
Indexes		
pk_t_author	ON id	
author_id	ON author_id	
name	ON name	

Table [t_book_copies](#)

Columns		
Book Copies		
id book_id lib_id copy_no		
1 b1 l1 1		
2 b1 l2 1		
3 b2 l1 1		
4 b2 l1 2		
5 b3 l2 1		
* id	int	AUTOINCREMENT
book_id	int	
lib_id	int	
copy_no	int	
Indexes		
pk_t_book_copies	ON id	
book_id	ON book_id, lib_id, copy_no	
lib_id	ON lib_id	
Foreign Keys		
t_book_copies_ibfk_2	(book_id) ref t_book (book_id)	

Columns		
Book Copies		
id	book_id	lib_id copy_no
1	b1	l1 1
2	b1	l2 1
3	b2	l1 1
4	b2	l1 2
5	b3	l2 1
t_book_copies_ibfk_1 (lib_id) ref t_library (lib_id)		

Table t_book

Columns		
Books		
*	book_id	int
	Publisher	varchar(200)
	Title	varchar(200)
*	Isbn	varchar(13)
	publication_date	date
Indexes		
	pk_t_book	ON book_id
	Isbn	ON isbn
	Publisher	ON publisher
Foreign Keys		
	t_book_ibfk_1	(publisher) ref t_publisher (name)

3. Implementation of Database and SQL Query:

➤ Creation of the database:

1. Creating tables, admin and reader:

```
sqlite> CREATE TABLE `admin` (  
...> `id`VARCHAR NOT NULL UNIQUE,  
...> `password`VARCHAR NOT NULL,  
...> PRIMARY KEY(`id`)  
...> );  
sqlite> CREATE TABLE `reader` (  
...> `readerid`INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
...> `name`VARCHAR NOT NULL,  
...> `address`VARCHAR NOT NULL,  
...> `phone`VARCHAR NOT NULL  
...> );
```

2. Tables Book and Author:

```
sqlite> CREATE TABLE `book` (  
...> `isbn`VARCHAR NOT NULL,  
...> `bookid`INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE  
...> );  
sqlite> CREATE TABLE `author` (  
...> `authorid`INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
...> `name`VARCHAR NOT NULL  
...> );
```

3. Publisher and Branch:

```
sqlite> CREATE TABLE `publisher` (  
...> `publisherid`INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
...> `publishername`VARCHAR NOT NULL,  
...> `location`VARCHAR NOT NULL  
...> );  
sqlite> CREATE TABLE `branch` (  
...> `branchid`INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,  
...> `name`VARCHAR NOT NULL,  
...> `location`VARCHAR NOT NULL  
...> );
```

4. Bookinfo:

```
sqlite> CREATE TABLE `bookinfo` (  
...> `isbn`VARCHAR NOT NULL,  
...> `title`VARCHAR NOT NULL,  
...> `publisherid`INTEGER NOT NULL,  
...> `publicationdate`DATE NOT NULL DEFAULT (null),  
...> `authorid`INTEGER NOT NULL,  
...> PRIMARY KEY(`isbn`)  
...> );
```

5. Location, Borrow and Reserve:

```
sqlite> CREATE TABLE `location` (  
...> `bookid`INTEGER NOT NULL,  
...> `branchid`INTEGER NOT NULL,  
...> `position`VARCHAR NOT NULL,  
...> PRIMARY KEY(`branchid`,`position`)  
...> );  
sqlite> CREATE TABLE `borrow` (  
...> `bookid`INTEGER NOT NULL,  
...> `readerid`INTEGER NOT NULL,  
...> `branchid`INTEGER NOT NULL,  
...> `bdate`DATE NOT NULL,  
...> `rdate`DATE DEFAULT NULL,  
...> `fine`FLOAT NOT NULL DEFAULT 0,  
...> `returned`INTEGER DEFAULT 0,  
...> PRIMARY KEY(`bookid`,`readerid`,`bdate`)  
...> );  
sqlite> CREATE TABLE `reserve` (  
...> `readerid`INTEGER NOT NULL,  
...> `bookid`INTEGER NOT NULL,  
...> `date`DATE NOT NULL,  
...> PRIMARY KEY(`readerid`,`bookid`,`date`)  
...> );
```

```
sqlite> .tables  
admin      book       borrow     location  reader  
author     bookinfo  branch    publisher reserve
```

6.

```
C:\sqlite>sqlite3  
SQLite version 3.17.0 2017-02-13 16:02:40  
Enter ".help" for usage hints.  
Connected to a transient in-memory database.  
Use ".open FILENAME" to reopen on a persistent database.  
sqlite> .open my_db.db
```

7.

➤ SQL statements that create and populate the tables:

➤ SQL statements that query the database:

- Query to get all reader id:
SELECT readerid FROM reader;
- Query to retrieve book using ISBN:
SELECT title FROM Bookinfo WHERE ISBN={user input};
- Query to retrieve book by publisher:
SELECT title FROM Book, Publisher WHERE
publisher.publisherid=book.publisherid and publishername={user input};
- Query to add book location:
INSERT into Location (bookid, branch, position) value (user input);
- Query for Book Checkout:
SELECT returned FROM borrow WHERE bookid={user input}
If this returns 1, then it adds the book:
Insert into borrow (bookid, readerid, branch, bdate, returned) values (user input);
It sets returned as 0;
- Query to return book:
SELECT returned FROM borrow Where bookid={user input};
If returned=0 then return book- update borrow set rdate={date}
- Query to retrieve all books, titles and ID:
SELECT title, bookid FROM book, bookinfo WHERE book.isbn=bookinfo.isbn;
- Query to retrieve reserved book list:
SELECT title from reserve, book, bookinfo WHERE reserve.bookid=book.bookid
AND book.isbn=bookinfo.isbn and readerid={user input};
- Query to retrieve frequent borrowers:
SELECT name, count(*) AS count FROM borrow, reader WHERE
borrow.readerid=reader.readerid and branch={user input} group by name order by
count(*) DESC;
- Query to retrieve frequently borrowed books:

```
SELECT title, count(*) AS count FROM borrow, book, bookinfo WHERE  
branched= {user input} and book.bookid=borrow.bookid AND  
bookinfo.isbn=book.isbn GROUP BY title ORDER BY count DESC;
```

- Query to check reader login:

```
SELECT readerid FROM reader WHERE reader= {user input}
```

If there is no match it will return 1.

4. Application Design:

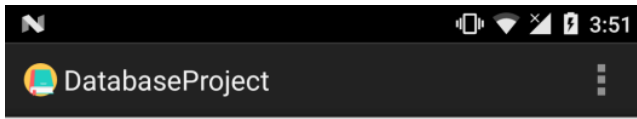
We have considered Android as our Operating system, because it is a powerful open source that also runs on Smart Phones and Tablets. The Operating System runs on Linux and uses its own Libraries and the SQLite Database is embedded within it. Our project provides a user friendly interface, which allows the user to view his/her due dates or hold on available books in the library.

The first step is when the user downloads the app from the Google play store and the application will appear on their Android device. The user will also be notified with the updates for the application. Initially, when the user logs in there will be a list of book available along with the search option. The users will also be able to check the availability of the books along with their due dates.

The main advantage of this application is to make it easier to access the library system, using their Smart Phone. But the downside to this is that only Android users can use this Application.

5. Window snapshots of the use of the program for each function:

1. Main Page:

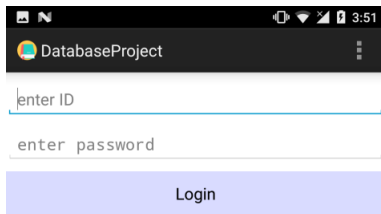


Admin

Reader



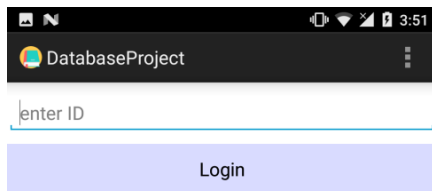
2. Admin Login:



A screenshot of an Android application titled "DatabaseProject". The screen displays two text input fields: the first is labeled "enter ID" and the second is labeled "enter password". Below these fields is a blue button labeled "Login". The status bar at the top shows the time as 3:51 and various system icons.



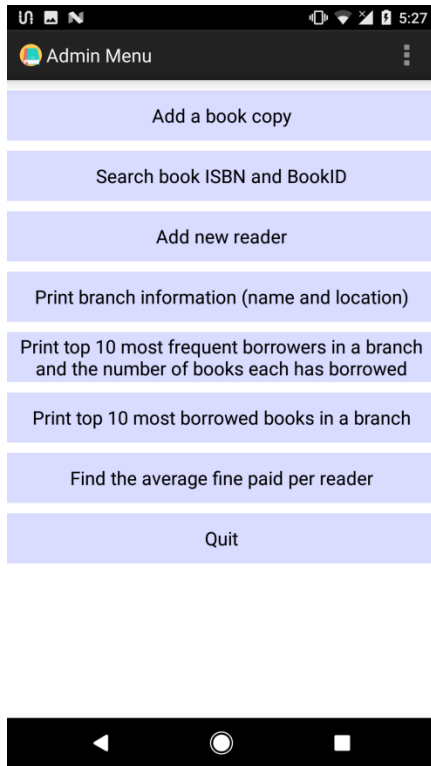
3. Reader Login:



A screenshot of an Android application titled "DatabaseProject". The screen displays a single text input field labeled "enter ID". Below this field is a blue button labeled "Login". The status bar at the top shows the time as 3:51 and various system icons.



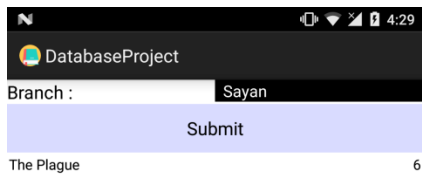
4. Admin menu:



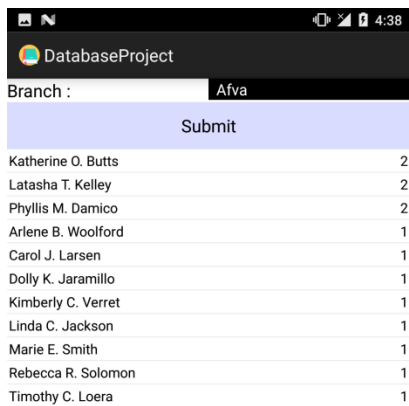
5. Finding the average fine per reader:



6. Print the top 10 frequently borrowed books from a branch:



7. Printing the top 10 Frequent Borrowers from a Branch and the number of times:

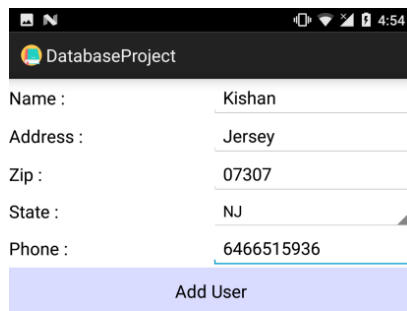


8. Branch name and Location:



Sitapur	Baben
Ram Nagar	Bardoli
Puna Patya	Ooty
Bhagal	Patan
Chakli Circle	Rajkot
Palsana	Udaipur
Isroli	Surat
Afva	Chikli
Sayan	Daman
Amroli	Mumbai
Dumas	Chennai
Piplod	Hyderabad
Moti Talav	Vadodara
Virar	Ahmedabad
Gandhi road	Borsad
VIP road	Navsari
Alkapuri	Porbandar
Raipur	Delhi
Kala Goda	Lucknow
Adajan	Kolkata
Lal Baug	kashmir

9. User details:



Name : Kishan

Address : Jersey

Zip : 07307

State : NJ

Phone : 6466515936

Add User

10. Adding a user:

The first screenshot shows the 'Admin Menu' screen with the following options: 'Add a book copy', 'Search book copy and check its status', 'Add new reader', 'Print branch information (name and location)', 'Print top 10 most frequent borrowers in a branch and the number of books each has borrowed', 'Print top 10 most borrowed books in a branch', 'Find the average fine paid per reader', and 'Quit'. Below this menu is a confirmation message: 'User 99033 Added'.

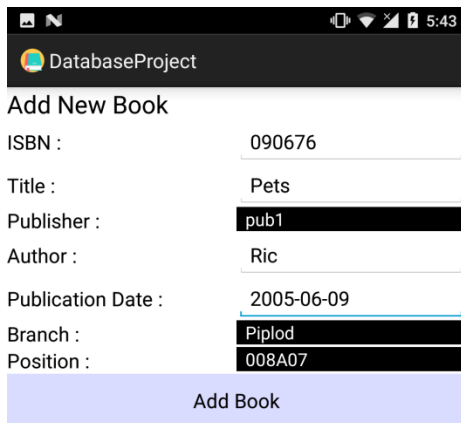
The second screenshot shows the 'DatabaseProject' screen for adding a new user. It contains the following fields and values: Name: Kishan, Address: Jersey, Zip: 07307, State: NJ, and Phone: 6466515936. An 'Add User' button is at the bottom.

11. Screen for searching a book:

The first screenshot shows the 'DatabaseProject' screen with a search input field containing '002004'. Below the input field are two buttons: 'Search By ISBN' and 'Search By BookID'. The 'Search By BookID' button is highlighted. Below the buttons, the text 'Moon Palace' is displayed.

The second screenshot shows the 'DatabaseProject' screen with a search input field containing '43'. Below the input field are two buttons: 'Search By ISBN' and 'Search By BookID'. The 'Search By BookID' button is highlighted. Below the buttons, the text 'Watch it Come Down' is displayed.

12. Adding a new book:



The screenshot shows a mobile application interface with a dark header bar containing the text "DatabaseProject" and a colorful icon. Below the header, the title "Add New Book" is displayed. The form contains several input fields with labels and values: ISBN : 090676, Title : Pets, Publisher : pub1, Author : Ric, Publication Date : 2005-06-09, Branch : Piplod, and Position : 008A07. At the bottom of the form is a blue button labeled "Add Book".

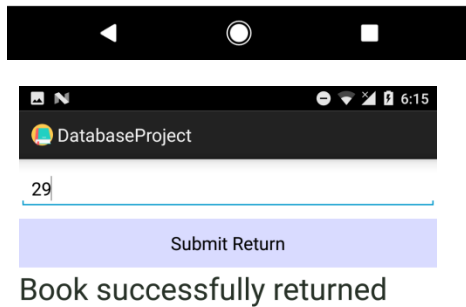
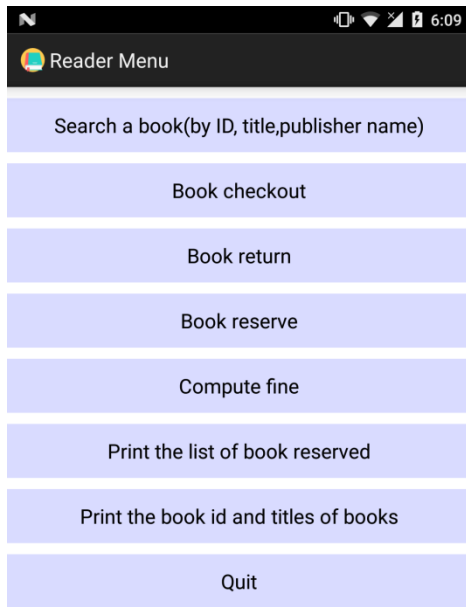
ISBN :	090676
Title :	Pets
Publisher :	pub1
Author :	Ric
Publication Date :	2005-06-09
Branch :	Piplod
Position :	008A07

Add Book

Book is added



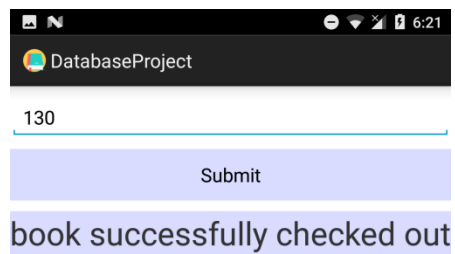
13. Reader menu options:



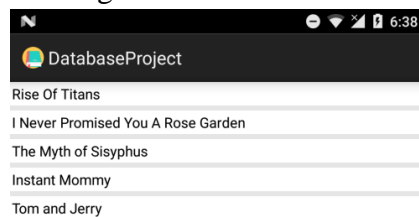
14. Book Returned:



15. Book Checkout screen:



16. Printing the reserved book list:

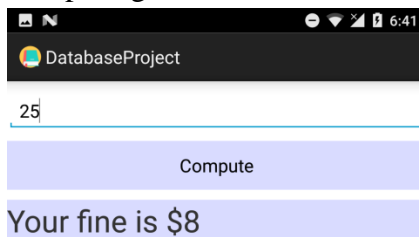


17. Printing book ID and titles:



Rise Of Titans	2
I Never Promised You A Rose Garden	3
The Myth of Sisyphus	4
The Plague	6
Joy and Anger	7
The Armageddon Inheritance	8
Oracle Night	20
Jacobs Room	21
A Touch of Spring	22
Jacobs Room	23
Moon Palace	24
The Magic Compass	26
Emergence	27
Rainbow Road	28
Everything Bad is Good for You	29
The Kings Persons	30
In Emeny Hands	31
Daddy's Angel	32
How Firm a Foundation	33
Love Diary	34
Fault In Our Stars	35
Challenge To Honor	36
Woods	37
Last Wish	38
Look Back In Anger	40
Night Of laughter	41

18. Computing the fine:



25

Compute

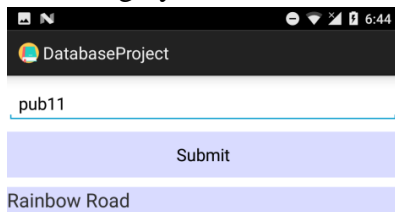
Your fine is \$8



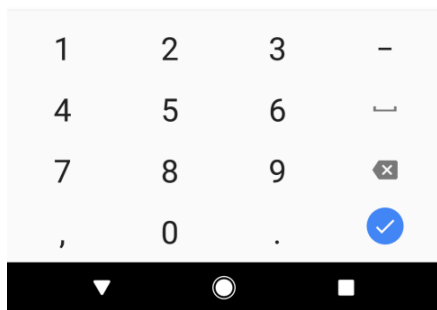
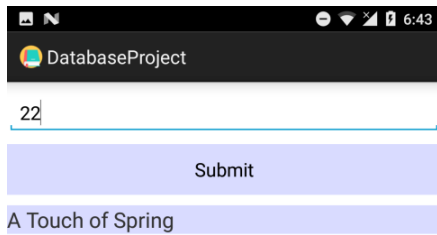
19. Searching for the book:



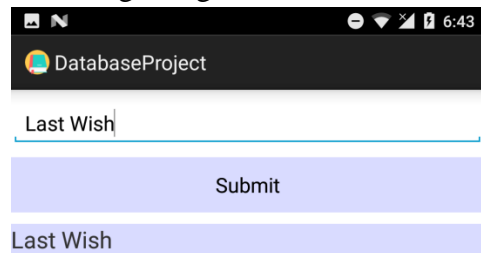
20. Searching by Publication:



21. Searching using the ID:



22. Searching using the Title:



Team member's work

In this project, each member of the team has given equal time to develop the system. We started from the scratch, designing the ER diagram to the front-end application. The most important task of the project was to design and develop the front end which communicates with the database. With the equal efforts from the members it made possible to develop.

Contribution of members:

1. Pratik Kumar Patel – Designing databases, developing front end and backend of system.
2. Kishan Gabani- Designing admin panel to using creating user. Fetching data from database and displaying different module. E.g. Add reader, Top 10 most borrowed books, fine paid by reader.
3. Apoorva Kanchu- Developing reader menu from user side. Designing different module from menu in book return, book reserve, total fine, self-checkout from the database