IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 4, Ver. V (Jul – Aug. 2014), PP 23-30
www.iosrjournals.org

Blue-Droid: An Intelligent Library Management System on Android Platform

Subhadeep Bhattacharya[1]
[1]*Department of Information Technology, St. Thomas' College of Engineering and Technology, Kolkata, India*

Abstract : *Bluetooth Library Manager (BLM) is library management software which can be used in any library for maintaining different functions like book search, requisition submission, data entry etc. I have implemented this software on Android platform which has the advantage of portability and wide availability. I used Bluetooth as communication medium between the server and the client. BLM has additional capabilities (call, SMS, email) for communication between library card holder and the administration. I have implemented this specially for any educational institution which has a library having capacity not more than 20,000 users. An encryption algorithm is used to enhance the security of that software. This algorithm is used for the log-in purpose. Using the encryption algorithm we encrypt the password in the client side and the cipher-text and the username is send over Bluetooth to the server side. The decryption algorithm is used in the server side to decrypt the cipher-text. Then the decrypted password is matched with corresponding password of that user stored in the server side database. The user can successfully log-in into the system if and only if the password of the database matches with the decrypted password. The entire database of the library is implemented in a normalized way. Student can submit their requisition from there phone without using internet. They can contact their faculties and classmates via SMS, email or call and vice-versa. A student can check the availability using this software. She/he can also see the return-date of any book which is out of stock now and borrowed by another student. Faculties can also send any notification to any student via email or SMS.*

Keywords: *Android, Python, SL4A, Bluetooth, SQLite, Library Management System*

I. Introduction

Nowadays, smartphones are becoming more powerful with reinforced processors, larger storage capabilities, richer entertainment functions and more communication methods.

- **Android**

Android is a mobile operating system (OS) based on the Linux kernel that is currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers. Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices.

- **Bluetooth**

Bluetooth, which is mainly used for data exchange, add new features to smartphones. Bluetooth requires low power which results to long battery life. Bluetooth technology can be used within the range of 30-feet, which is reasonable within a small building. This range can be increased using Bluetooth repeater. Moreover, the data-transfer rate of the Bluetooth technology is quite acceptable, i.e. 3-4Mbps. So, finally, due to the aforesaid benefits, I decided to use the Bluetooth technology for the connectivity in my application.

- **SL4A**

I have used Scripting Layer for Android (SL4A) and python for developing my application. SL4A is a library that allows the creation and running of scripts written in various scripting languages directly on Android device.

- **Python**

Python is a high-level programming language which allows programmer to express concepts in fewer lines of code. Python is widely used as scripting language. The Java language is too heavy weight for some and is not entirely open source. Python has a wide usage and is open source. It also has seen the most interest as far as SL4A is concerned.

- **SQLite**

I have also used SQLite for my database related work. I use SQLite using sqlite3 module for python. I have used the different UI facades available in SL4A for user interactions. BLM is capable of performing different functions such as Searching any book, Requisition submission, Contact help, Book issue etc. An encryption algorithm is also included using ASCII conversion and some calculations. This encryption technique is used for enhancing the reliability of the software.

II. Motivation

Library management software is essential for increasing the flexibility of any library. It reduces the work force required for maintaining the daily work of any library system. In the present manual library system a student have to follow several steps for accessing any book. At first requisition for a book has to be written on paper and submitted to the librarian. Then she/he has to wait in a long queue to get that book if the book is available. So, it can be concluded that the manual system take lots of time of both students and the library authorities for accessing any particular book. Besides that searching details of any book also take lots of time as the student has to go to the library and search for that book. BLM helps to integrate several works of library and automate them. It saves time of not only the students but the library authority also. In today's world Android phones are widely available and widely used among the student community. So I have implemented the software in Android platform using python for Android. I have implemented it using Bluetooth communication because it will help students to operate that software anywhere in the institution without any internet connection. BLM can be used for maintaining the different library functions as well as for communication between the faculties and students. I also used an encryption algorithm to main the authentication of the user. To use the functionality of that software both student and authority have to log-in with his username and password. The authentication checking is done in the server. After successful log-in user can use the different functionalities of that software. The installation and maintenance cost of BLM is also very less so it can be used effectively.

III. Methodology

I. Database Structure

accession_detials

Name	Type	Details
accession_no	number(20)	primary key
book_id	number(7)	references book-details(book_id)
doa	date	
bill_no	number(20)	references bill_details(bill_no)
price	number(7)	
cd	number(1)	check(cd in (0,1))
dop	date	

bill_details

Name	Type	Details
bill_no	number(20)	primary key
bill_date	date	
source	varchar(20)	

author_details

Name	Type	Details
author_name	varchar(40)	primary key
author_mark	varchar(3)	

book_count

Name	Type	Details
book_id	number(7)	primary key, references book_details(book_id)
subject	varchar(20)	references class_details(subject)
publisher_name	varchar(20)	
count	number(10)	

book_details

Name	Type	Details
book_id	number(7)	primary key
book_name	varchar(40)	
author_name	varchar(40)	references author_details(author_name)
edition	varchar(4)	
series	varchar(20)	
rack	varchar(3)	

class_details

Name	Type	Details
subject class_no	varchar(20) varchar(10)	primary key

faculty_detials

Name	Type	Details
card_no f_name f_ph_no f_email	Varchar(7) Varchar(20) Varchar(15) Varchar(25)	primary key

issue_book

Name	Type	Details
book_id card_no return_date	number(7) varchar(7) date	references book_count(book_id)

requisition

Name	Type	Details
card_no	varchar(7)	primary key
book_id	number(7)	references book_count(book_id)
book_name	varchar(40)	
author_name	varchar(40)	
edition	varchar(4)	
series	varchar(20)	
class_no	varchar(10)	
author_mark	varchar(3)	

student_details

Name	Type	Details
card_no	varchar(7)	primary key
s_name	varchar(20)	
department	varchar(5)	
s_ph_no	varchar(15)	
s_email	varchar(25)	

user_details

Name	Type	Details
user_id	varchar(20)	primary
password	varchar(20)	key

2.1 Encryption Algorithm

Step1: Divide the entire plain text into 4 bit sub texts.
Step 2: For each sub text go to Step3.
Step 3: Determine the ASCII values of all character of keys and find out its summation (k).
Step 4: Determine the ASCII values of all characters of plain text and merge them sequentially.
If the length of ASCII value of any character is equal to 3, add '//' in the before and after that ASCII value.
Step5: Determine t1 by eliminating the '//'s.
Step6: Calculate L using L=length (t1)-2. Create a special number(s) using L.
Step7: Calculate x=t1*s.
Step8: Calculate y=k-s
Step9: Determine y1 by determining the ASCII values of all characters of y except '.' and merge them sequentially.
Step10: Create y1. y1=<L>.<number of characters having ASCII values of length=3><positions of characters having ASCII values of length=3>.<y1>
Go to step 2.
Step11: Determine the cipher text by converting each character into integer except '.' and replace them with the character having ASCII value=integer value.

2.2 Decryption algorithm

Step1: Convert the cipher text into y1 by determining the ASCII value of each character except '.'
Step2: Split the modified cipher text while any blank space is occurred and store those sub cipher text into a list li. Initialize d=[], decrypted_text=""
Step3: For each string in li go to step 4.
Step4: Determine L, number of characters having ASCII values of length=3, positions of characters having ASCII values of length=3 and y1.
Step5: Determine y1 into y by converting ASCII values into characters. Determine s.
Step6: Calculate t1= (k-y)/s
Step7: Determine t using the number of characters and positions of characters having ASCII

values of length=3.

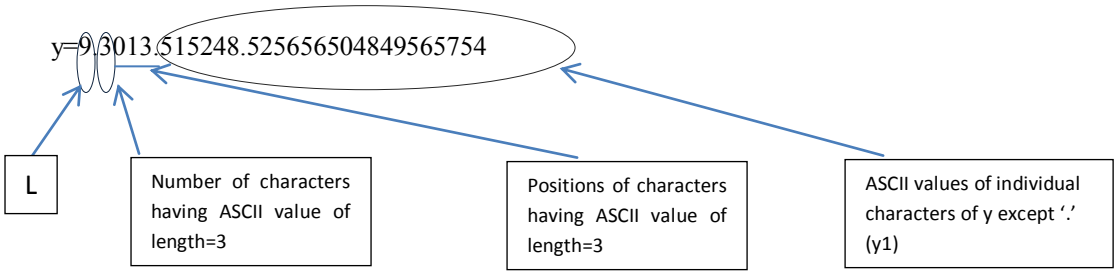
Step8: Convert each ASCII value into its corresponding character and append the entire string in a list d. Go to step 3

Step9: For each string in d merge it with the decrypted_text

2.3 Example

2.3.1 Encryption

Plain Text	Subh
key	DU/314
Key value after adding individual ASCII values of characters(k)	351
Modified value of text after merging ASCII values of all characters(t)	//115////117//98//104//
Modified t(t1)	11511798104
L=length(t1)-2	9
Special number(s)	0.000000001
x=t1*s	11.511798104
y=k-s	340.488201896
Modified y(y1)	515248.525656504849565754
Modified y1	9.3013.515248.525656504849565754
Cipher Text	.└ └.| |┐┘◘.|┐|-|-| ┘◘┘ |-| |┘ -..| ┘ | .|┘|||┘|||┘|-



2.3.2 Decryption

Cipher Text	.└ └.| |┐┘◘.|┐|-|-| ┘◘┘ |-| |┘ -..| ┘ | .|┘|||┘|||┘|-
key	DU/314
Key value after adding individual ASCII values of characters(k)	351
Modified Cipher Text(y1)	9.3013.515248.525656504849565754
L	9
s	0.000000001
Number of characters having ASCII value of length=3	3
position	0,1,3
y1	515248.525656504849565754
y	340.488201896
t1=(k-y)/s	11511798104
t	//115////117//98//104//
Decrypted Text	Subh

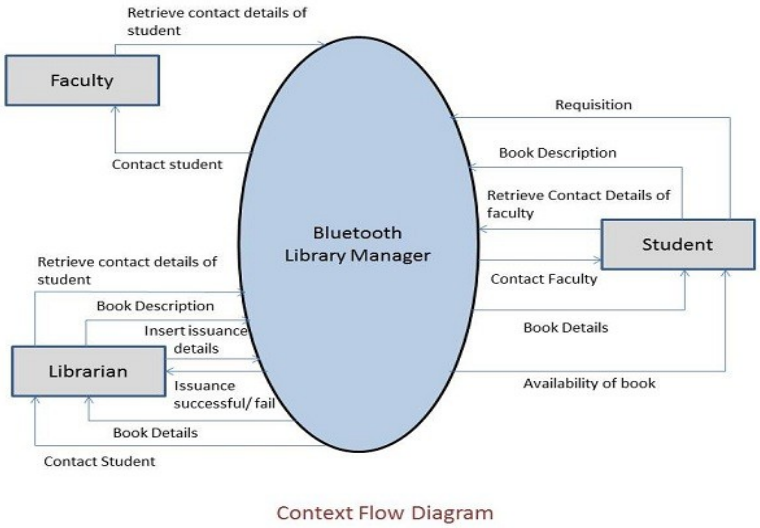


Fig 1: Context Flow Diagram

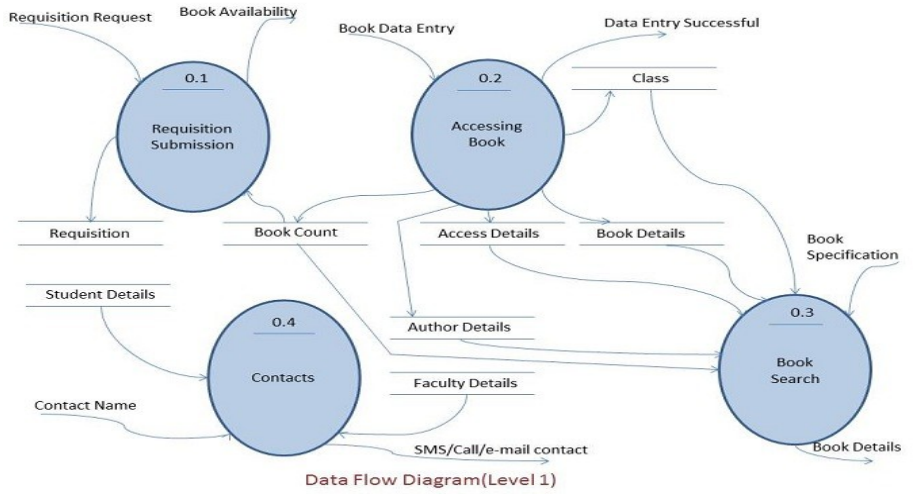


Fig 2: Data Flow Diagram(Level 1)

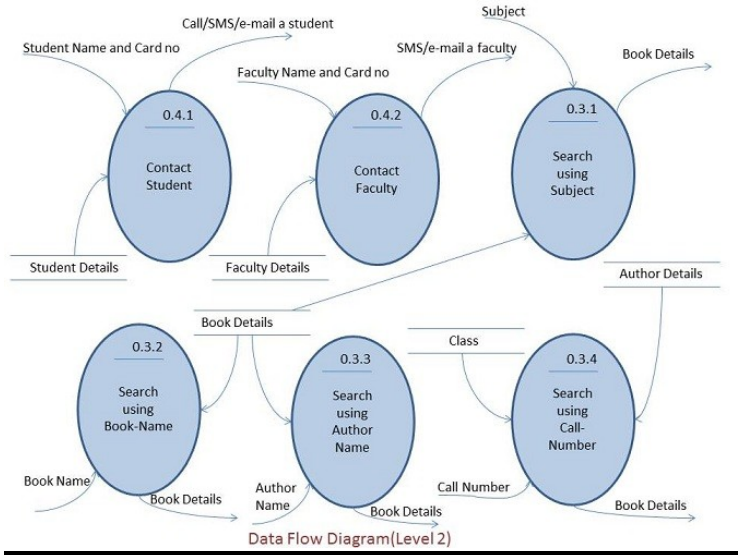


Fig 3: Data Flow Diagram(Level 2)

3 Algorithm

3.1 Client part:

Step1: Choose user type(admin or others). If user type=others go to step2. If user type=admin go to step 18.
Step2: Read User id and Password from the user. Determine cipher text using encryption algorithm.
Step3: Check current Bluetooth state of the device. If Bluetooth is off, activate it. Start Bluetooth discovery. Select the admin device from the list.
Step4: If connection established, prompt it.
Step5: Write "Bluetooth" to server.
Step6: Write user id and cipher text to the server.
Step7: Read ack from the server. If ack='1', prompt "Login Successful" and go to step 7
Step8: Choose option (Search, Requisition Submission, SMS, Call, e-mail) from the alert box.
If option=Search, go to step8
If option=Requisition Submission, go to step13
If option=SMS, go to step14
If option=Call, go to Step15
If option=e-mail, go to Step16
Step9: Choose Search option (Book_name, Author_name, Call_no., Subject) from the alert box.
Write "1" to server.
If search option=Book_name, go to step 9
If search option=Author_name, go to step10,
If search option=Call_no, go to step 11,
If search option= Subject, go to step12.
Step10: Read book_name, edition and series.
Fetch book_id, book_name, author_name, class_no, author_mark, edition and series from server side database using those details.
Step11: Read author_name.
Fetch book_id, book_name, author_name, class_no, author_mark, edition and series from server side database using author_name.
Step12: Read class_no.
Fetch book_id, book_name, author_name, class_no, author_mark, edition and series from server side database using author_name.
Step13: Read subject.
Fetch book_id, book_name, author_name, class_no, author_mark, edition and series from server side database using subject.
Step14: Write "2" to server. Read card_no, book_id, book_name, author_name, class_no, author_mark.
Insert card_no, book_id, book_name, author_name, class_no, author_mark into server side database.
Read return date from server database if any book is not available and show it to user.
Step15: Write "1" to server.
Read faculty or student name from the user. Fetch his/her phone number from the server side database.
Compose and send SMS to that number.
Step16: Write "1" to server.
Read faculty or student name from the user. Fetch his/her phone number from the server side database.
Make a call to that number.
Step17: Write "1" to server.
Read faculty or student name from the user. Fetch his/her email id from the server side database.
Compose and send email to that email id.
Step18: Check current Bluetooth state of the device. If Bluetooth is off, activate it. Start Bluetooth discovery.
Select the admin device from the list.
If connection established, prompt it.
Step19: Write "Admin" to server. Write user id and password to the server.
Step20: Read ack from the server. If ack='1', prompt "Login Successful" and go to step 21.
Step21: Read entire book description and modify the server side database.

3.2 Server part

Step1: Check current Bluetooth state of the device. If Bluetooth is off, activate it. Make Bluetooth device discoverable. Accept connection.
Step2: While True, repeat step 3 to 7.

Step3: Read user type from client. If user type='Admin', go to step3. If user type='Bluetooth', go to step 4.
Step4: Check user name and password. Write ack=1 to client if log in successfully. Modify the database.
Step5: Decrypt cipher text and check user id and password. Write ack=1 to client if log in successfully. Read option from client. If option='1', go to step5. If option='2', go to step6.
Step6: Fetch data from database and write them to client.
Step7: Modify database. Fetch data from database and write them to client.

IV. Result

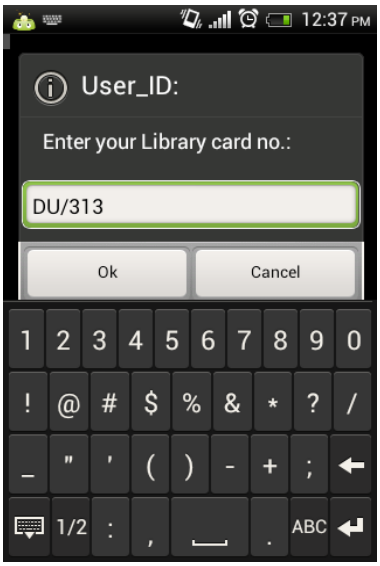


Fig 4(a):User id entry

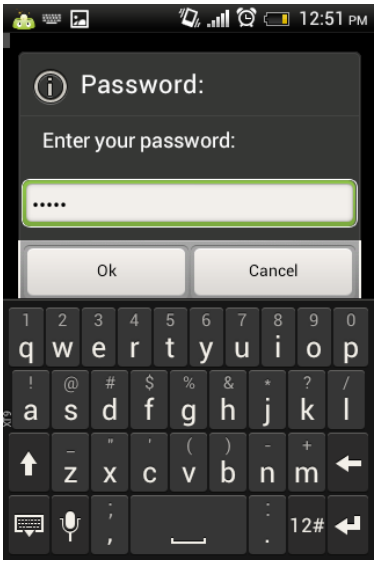


Fig 4(b):Password entry

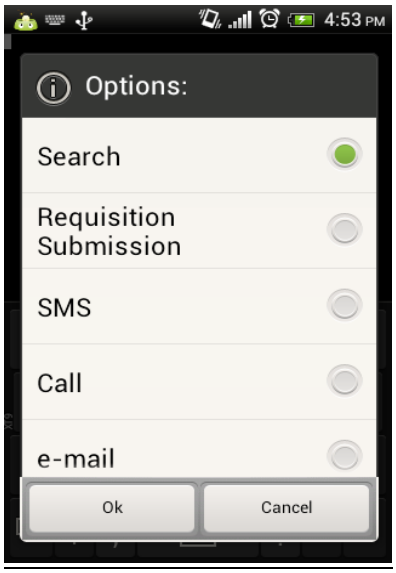


Fig 5: Different options for user.

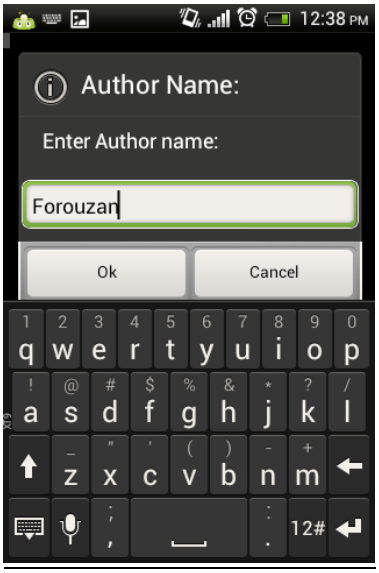


Fig 6: Search book details by author name

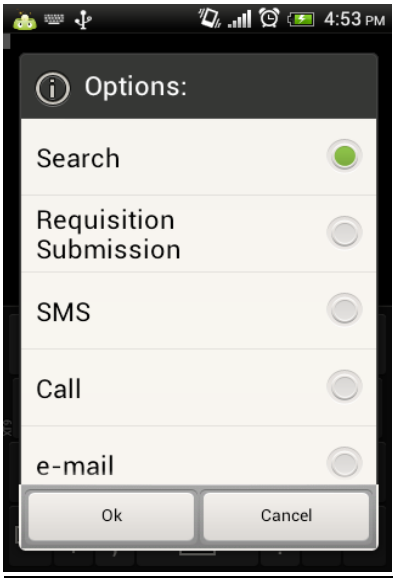


Fig 7: Search Result

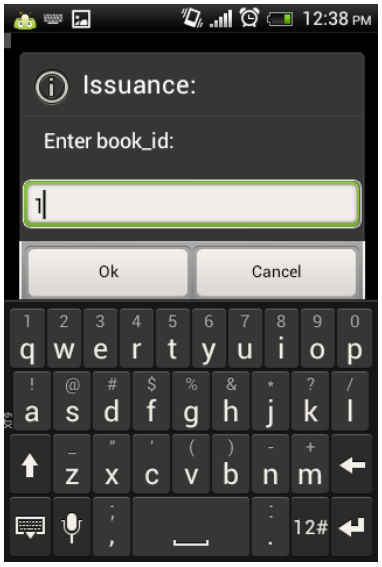


Fig 8: Book Issuance

V. Conclusion

Bluetooth Library manager is very useful software for library management of any institution. It can be installed cost efficiently as it does not require many resources. It help students to manage their library oriented work from any place inside the institution. It does not require internet connections which is also a major advantage of this software. It also help the faculties. They can access library and communicate with the students using this software. The library authority can manage the library in an autonomous way with less number of workforces. So, it can be concluded that this software is very helpful for any institutional library system.

I did not test the performance of this software. Performance of this software can be increased by adding additional capabilities and performance testing.