Table of Contents

Task 1 [1.1, M1]

Discuss the principles, characteristics and features of programming in Java.

Introduction

Java is a highly object oriented and platform independent programming language. It is simple yet very powerful language that is a lot similar to C++ in various aspects. Green team, a team consists of James Gosling, Mike Sheridan, and Patrick Naughton initiated java language project initiation at 1991.

According to mathbits (n.d.) first version of java was aimed to program home appliances. Java was named Oak at first but later it was renamed to java as there was already programming language called Oak. In 1995 Sun Microsystems' released first public implementation as java 1.0.

Principles

Java language is based on famous write-once, run-anywhere principle. The core principle is that program developed in java language should be able to run on any platform. Besides this, java follows four major object oriented principles; encapsulation, inheritance, abstraction and polymorphism. Each principles are discussed below in this document.

Encapsulation

In java, encapsulation controls access of a class from outside. This features provides security to the class. Encapsulation bundles methods and properties inside class to protect them from unauthorized modification. Access modifiers, public, private and protected defines which parts of class are accessible from outside. Private parts can only be accessed by properties and methods of own class. Whereas, protected data are accessible to own class members and child classes. Only public part of the class is accessible from outside using object and dot (.) operator.

Example of Encapsulation in Java

```
public class encapsulation {
    private int CARID=101;
    public String name = "Ram";
    protected String CARNAME= "TOYOTA";
    }
```

Figure 1 Example of Encapsulation

In example above (Figure 1), private variable CARID cannot be access from outside. Only public variable name is accessible from outside and variable CARNAME is accessible by classes that inherits class encapsulation.

Inheritance

Inheritance is principle that allows to create child class with ability to inherit public and protected variables and methods from parent class. According to INTROPROGRAMMING (n.d.) this feature of OOP enables the code reusability and improves code readability.

```
package Encapsulation;
public class Encapsulation {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Car objectcar = new Car();
        objectcar.color = "Blue";
        objectcar.speed = 200 ;
        objectcar.CC = 1000;
        objectcar.gears = 5;
        objectcar.attributescar();
    }
}
class Vehicle {
       String color;
       int speed;
       void attributes() {
          System.out.println("Color : " + color);
          System.out.println("Speed : " + speed);
       }
    }
class Car extends Vehicle {
    public   int CC;
    public   int gears;
      public void attributescar() {
          // The subclass refers to the members of the superclass
          System.out.println("Color of Car : " + color);
          System.out.println("Speed of Car : " + speed);
          System.out.println("CC of Car : " + CC);
          System.out.println("No of gears of Car : " + gears);
       }
    }
```

```
<terminated> Encapsulation [Java
Color of Car : Blue
Speed of Car : 200
CC of Car : 1000
No of gears of Car : 5
```

Figure 2 Example of Inheritance

In example above (Figure 2), class Car inherits properties and methods of class Vehicle. Public and protected part of vehicle can be used by class Car like their own properties and methods. Car class from this example is also called sub class which inherits features from another class. And Vehicle class which is being inherited is called super class.

Polymorphism

This principle of OOP is followed by java which provides capability to perform multiple action based on how it is called. In more general term, it allows methods of same names but with different

functionalities. As the name itself suggests, polymorphism means many forms. Polymorphism follows concept one interface many methods. According to BEGINNERSBOOK (n.d.) in java there are two different types of polymorphism.

a. Method overloading
b. Method overriding

```
package Encapsulation;

public class Polymorphism {


public static void main (String args [])
        {
            Overload Obj = new Overload();
            double result;
            Obj .demo(20);
            Obj .demo(30, 50);
            result = Obj .demo(6.5);
            System.out.println("O/P : " + result);
        }
    }
class Overload
{
    void demo (int num1)
    {
       System.out.println ("a: " + num1);
    }
    void demo (int num1, int num2)
    {
       System.out.println ("a and b: " + num1 + "," + num2);
    }
    double demo(double a) {
       System.out.println("double a: " + a);
       return a*a;
    }
}
```

```
<terminated> Polymorpl
a: 20
a and b: 30,50
double a: 6.5
O/P : 42.25
```

Figure 3 Method Overloading polymorphism

In example of polymorphism above (figure 3), demo method is overloaded in class overload. Method demo can perform three different task based on argument lists and parameter. This type of polymorphism is called method overloading. This type of polymorphism can used in constructor to achieve different constructor behavior.

In method overriding polymorphism, base class overrides the methods and properties of parent class that share same name. For example if parent has method called RUN () and sub class also has method with same name then RUN method from parent class is overridden by method from sub class.

```
package Encapsulation;

public class Polymorphism  {


public static void main (String args [])
        {
            Child Obj = new Child();
            double result;
            Obj .demo(20);

        }
    }
    class Child extends Overload
    {
        void demo (int num1)
        {
           System.out.println ("Child Value B: " + num1);
        }
    }
    class Overload
    {
    void demo (int num1)
    {
       System.out.println ("parent Value A: " + num1);
    }
  }
```

```
<terminated> Polymorphism
Child Value B: 20
```

Figure 4 Example of method overriding

Abstraction

Java follows abstraction feature of OOP that provides only required information to user and hides all internal structures. According to VISIONSDEVELOPER (n.d.) this feature of OOP hides implementation details from user and only provides functionality. In java this principle is achieved by use of abstract and interface class. Abstract class is not complete itself hence objects cannot be created.

Characteristics

Java Virtual Machine

JVM is virtual or abstract computing machine that provides execution environment for java byte code. Byte code is intermediate level code generated after compiling java code. The core functions of JVM is to load, verify, execute java codes and provide runtime environment.

JVM has two subsystems; Class loader and Execution engine, Zhebel (2014). Class loader is a mechanism that is responsible for loading extension libraries, core library, application class and user define libraries in hierarchical order; bootstrap loader, extension class loader, system class loader and user-defined class loader. Other subsystem of the JVM is execution engine which executes byte code to make them readable to machine.

Platform Independent

To run a program in platform they are not originally written for requires modification in program to run smoothly. Such programs are called program dependent. But thanks to JVM, java is platform independent. Program written on java for windows can run on Mac or Linux without any code alteration. The reason behind this is java byte code are readable to JVM, but not to actual platform, and JVM remains constant to all platforms WIDESKILL (n.d.). This allows java programs to run on any platform irrespective of the real platform.

Garbage Collection: Automatic memory management

One of the key characteristics of java is it has ability to manage memory automatically through garbage collection. Programmers does not requires to be worry about memory distribution as garbage collector automatically maintains memory for reuse, DYNATRACE (n.d.). In this implicit memory management, programmers does not require to write memory handling code as java memory management can automatically allocate memory for object instance, track them and reclaim allocated memory one object does not require that memory.

Multithreaded

Java is capable of running multiple threads simultaneously. Here, threads defines a smallest unit of process. Alemayehu (n.d.) states, multithreading is technique of allowing a process to perform different task at parallel time. One of the key benefits of multithreading is better use of CPU hence better performance. Multithreading also do possess some drawbacks. One of the drawback of multithreading is coding and debugging multithreaded applications are often complex.

Concurrency

Vogella (n.d.) describes concurrency as ability to run multiple parts of programs or multiple programs simultaneously (in parallel). Nowadays, computers with multiple processors are available in the market, in fact most of the computers are multiprocessors. Concurrency can utilize the processor to speed up the program. Java supports concurrency hence modern java programs are concurrent.

To understand concurrency let's take example, if a task A requires 20 second and Task B requires another 20 seconds, in general scenario it would take 40 seconds to complete both tasks but in if program is concurrent, task A would be processed in one processor and task B would be executed

in another processor in parallel . This would allow both tasks to complete in 20 seconds instead of 40 seconds taken in normal scenario.

Class Libraries

Java class libraries are pre written loadable group of classes that can be called upon requirement. Classes that offers similar functionality are bundled into package. Java offers hundreds of core java libraries and also supports third party libraries. These libraries includes fundamental classes required to build and run a java application. For example, java.io contains classes that are fundamental for input output operations. Programmer need to call java.io first to develop program that can take input and give output. Some other core class libraries are java.lang, java.net and java.swing etc. These class libraries are bundled with Java Development Kit.

Features

Simple

Java is very simple programming language that looks very much similar to c or C++ but it omits some confusing parts of C++ like multiple inheritance. Java does have break and continue statement that makes this language more simple than languages with go to statement. One of the key characteristic of java that makes it very simple language is garbage collection as programmer does not have to think about memory management. Additionally, there are large numbers of class libraries that makes life easier for developers.

Interpreted

Unlike C or C++, java is interpreted language where code is compiled into byte code. Even though interpreted languages are slower than compiled languages, java is faster than traditional interpreted language due to Java Just in time compiler.

Summary

Java is very popular programming language that is based on object oriented programming principle. This language allows to create platform independent application where developer need to code only once and application can run on any platform. This happens due to working mechanism of Java virtual machine. An abstract machine that can execute complied java byte codes. Java offers lots of other features and characteristics that helps to develop better application. Application developed with java are more performance efficient due to its garbage collection, multithreading and concurrency characteristics. This summarizes java as one of the most powerful and well-organized programming language.

References

- Alemayehu (n.d.) what is Multithreading in Java? [Online] Available: http://mahderjava.blogspot.com/2013/04/what-is-multithreading-in-java.html Accessed [11/13/2015]
- BEGINNERSBOOK (n.d.) Polymorphism in Java – Method Overloading and Overriding [Online] Available: http://beginnersbook.com/2013/03/polymorphism-in-java/ Accessed [11/14/2015]
- DYNATRACE (n.d.) Java Memory Management [Online] Available: http://www.dynatrace.com/en/javabook/how-garbage-collection-works.html
- INTROPROGRAMMING (n.d.) Object-Oriented Programming Principles (OOP) [Online] Available: http://www.introprogramming.info/english-intro-csharp-book/read-online/chapter-20-object-oriented-programming-principles/#_Toc362296570 Accessed [11/13/2015]
- Mathbits (n.d.) Brief History of Java [Online] Available: http://mathbits.com/MathBits/Java/Introduction/BriefHistory.htm Accessed [11/14/2015]
- VISIONSDEVELOPER (n.d.) java-abstraction [Online] Available: http://www.visionsdeveloper.com/tutorial/java/java-abstraction.jsp Accessed [11/14/2015]
- Vogella (n.d.) Java concurrency (multi-threading) - Tutorial [Online] Available: http://www.vogella.com/tutorials/JavaConcurrency/article.html#concurrency_processthreads Accessed [11/13/2015]
- WIDESKILL (n.d.) Introduction to Java Programming http://www.wideskills.com/java-tutorial/introduction-to-java-programming Accessed [11/14/2015]
- Zhebel (2014) Java Virtual Machine: the Essential Guide https://anturis.com/blog/java-virtual-machine-the-essential-guide/ Accessed [11/13/2015]

Task 2 [1.2]

Critically evaluate the environmental flexibility of programming in Java.

Introduction

Flexibility is capability of being altered or modified according to changes in requirements. To debug a program or add more features to them requires modification. Developer use existing system to develop better system. Flexibility describes how many different methods can be used to achieve requirement. This document evaluates how java offers mechanisms to support environmental flexibility.

Flexibility in Variable

Variables are used for storing value information by reserving memory location. When variable is created, it reserves some memory and allows to store value on it. Type of the value stored on memory is based on the type of variable itself. Java supports two major types of variable types.

Primitive Data type

Primitive data types are named by keywords and are predefined in language. Java supports eight different primitive types. Those eight data types are byte, short, int, long, float, double, Boolean, and char. Example,

```
public int int_test=123123123;
public double double_test=12312312123123123.12312312312;
public byte byte_test=123;
public char char_test='A';
public boolean check=false;
public long longtest=1232311231;
public float float_test=10.1f;
```

Object Data type

Object data type are also reference data type. Object of class is used as variable. Class objects, array variables are some of the Reference data types. Example,

```
public String string_test="RAM";
```

String data type in above example is instance of String class. Programmer can use user defined object data type as per requirements. Java support for both primitive and object data type offers environmental flexibility to code their program in various way.

Auto Boxing

Java environment has mechanism that can convert primitive data type into corresponding wrapping class type. For example int data type can be automatically converted to Integer. This mechanism offer flexibility of using primitive or object type interchangeably. Programmer does not require to be worry about data type as java takes care them itself. Example,

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
        Integer num=5;
        int num_2=num;
        System.out.print(num_2);
}
```

In above example of auto boxing, Integer data type num is automatically converted to int data type. In earlier versions of java, to get same result Integer first needed to convert to int type manually.

```
int num_2=num.intValue();
```

In latest version of Java this process happens behind the scene and programmer does not require to code them manually.

Memory Management

Java offers great flexibility when it comes to memory management as programmer does not require to worry too much as them. Programmers only need to design their code based on requirement and can forget about memory allocations as Java will do it for them automatically. It is basically process of reclaiming unused memory during runtime automatically. Program have unreferenced objects like null object and anonymous object, with Java memory allocated these objects reclaimed.

In other programming language such as C, automatic memory management feature is not available. When memory is allocated, it stays there until program is terminated or memory is freed manually. C has functions like free () that is used for freeing memory space.

Exceptional handling

Exceptional handling is process of handling exceptional events in programming. Here exceptional events describes cases occur during execution of program that can disturb the flow. In general cases, exception terminates the program preventing to continue. Exceptional case can occur due to several reasons such as failing to call required class, accessing non existing files, establishing wrong database connection etc.

Some of the famous exceptions in Java are:

- Divide-By-Zero exception
- Null-Pointer-Exception
- Arithmetic-Exception
- Array-Index-Out-Of-Bounds-Exception

Java supports great tools like try-catch, throw, finally and nested try catch to handle such exceptional events. In try catch, unexpected event is handled through catch. Java This offers great flexibility in as programmer can build different work flow based on how program behave. S/he can code alternative flow for the case of exceptional event.

```
public static void main(String[] args) {
    // TODO Auto-generated method stub

        int num1=90, num2=0;
        int result=num1/num2;
        System.out.println ("Output = " +result);
    }

}
```

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Encapsulation.Encapsulation.main(Encapsulation.java:8)
```

```
    public static void main(String[] args) {
        // TODO Auto-generated method stub
          try{
              int num1=90, num2=0;
              int result=num1/num2;
              System.out.println ("Output = " +result);
           }
           catch(ArithmeticException e){
              System.out.println ("Exception Occured: Cannot divide by ZERO");
           }
}
```

```
Exception Occured: Cannot divide by Z
```

In example above, programmer can code alternative flow with in catch block. If exceptional handling is not implemented, program would terminate.

Look and Feel

In software development, look describes the appearance and behavior of developed application. Java offers great flexibility in applying different look and feel. Programmer can customize appearance or behavior of application as requirement. This feature offers environmental flexibility to design software layout and interactivity.

Summary

This document critically evaluated flexibility in programming with Java. Java programming environment has some features and characteristics that allows programmer to design code in various ways to achieve the requirement. Feature such as automatic memory management allows

developer to code without worrying about memory management. Likewise, Java allows to use either of primitive or their corresponding wrapping class object as Java can automatically convert them as requirement. Programmers has flexibility to customize the looks and feels of application while programming with Java as well as can set alternative flow of program for the scenario of occurrence of exceptional event.

Task 3 [2.1, M2]

Design a java programming solution to problem given in the scenario.

Designed solution

To design a solution to given problem, several tools and techniques are used. These tools and techniques helps to describe flow of application, functionalities, and data relations etc. Tools and techniques like Pseudo-code, algorithm, flow-chart, ER-diagram, use case diagram, context diagram and zero-level data flow diagram are used for designing the solution.

Pseudo code

Pseudo code describes an algorithm in English-like friendly way of expression. Pseudo code are more precise than real English but are not as precise as a programming language. In more specific words, pseudo code is readable text-based description of how a computer application need to be developed or program need to written to achieve the required solution.

Pseudo code allows programmers/ developers to plan detailed template for coding the program. In the current scenario, pseudo code would allow to plan template for designing various components of library management system such as login system, add new student system, reserve book system and student wish list system.

a) Login System

1. START
2. PRINT Welcome to Library Management System
3. FOR First application load
   Create and load required Database and Tables
4. INPUT username and password
5. Connect to database
6. IF (Username and password verifies) THEN
   DISPLAY Main Panel
   PRINT Welcome Massage
   ELSE
   DISPLAY Failure Massage
   RETURN
7. END

b) Add New Student

1. START
2. PRINT Add Student
3. Initialize variables for Student ID, name, address, contact, faculty and Semester
4. INPUT Student ID, name, address, contact, faculty and Semester
5. For add button click event
6. IF Student ID does not exists already in database
   Insert information in database
   PRINT success massage
   ELSE IF student ID already Exists
   PRINT Warning and Error massage
   ELSE
   PRINT error massage
7. END

c) Reserve Book

1. START
2. PRINT Search Book
3. Initialize Book ID
4. IF INPUT Book ID
   Filter Book ID from database and DISPLAY in list view
   ELSE
   DISPLAY all available books
5. Long press in book Generate Menu with reserve option
6. Store Book name , book ID, Student ID in Bundle
7. Allow to click reserve
8. PROCESS Reserve Book Page
9. Initialize Book name , book ID, Student ID from Bundle
10. Connect to database
11. Insert information into database
    IF Success
    PRINT success massage
    ELSE

PRINT Error and Failure massage

12. END

d) Wish list Student

1. Start
2. PRINT Wish list
3. Process Wish list File
4. Initialize Book name, author, edition to null
5. INPUT Book name, author, edition
6. Connect to database
7. Insert information into database table

   IF Success

   PRINT Success massage

   ELSE

   PRINT Failure and Error massage
8. END

Algorithm

Algorithm in general word is detailed procedure for writing solution to the computer problem. Each steps in algorithms are unambiguous and clear. Input/output variables in algorithm are specified precisely. Algorithms are not actual code but it is more detailed template for solution that a pseudo code. No matter what programming language is chosen, following an algorithm always result same output that can solve the problem

Algorithm for components of various components of library management system are written below.

a) Return Book

1. Start
2. Declare Array list of String type ‘allbookid’
3. Declare Array list ‘bookinfo’ of type getter_setter
4. Declare Array Adapter ‘adapter’
5. Connect to database
6. Get list of reserved books from database
7. IF count of reserved book is not equal to zero

8. Declare temp of data type getter_setter
9. Initialize temp.bookname←bookname, temp.reserveddate←reserveddate, temp.studentname←studentname from the list
10. Add temp to bookinfo
11. REPEAT until all row from list is added to bookinfo
12. Initialize adapter←bookinfo
13. SET adapter as Source for List view
14. Display reserved Books in list view
15. TAKE Input through long press
16. Declare and initialize bookid
17. Delete reserved book information with whose id is bookid
18. Set status of book whose id is book id
19. Remove book whose id is bookid from bookinfo
20. Refresh List view
21. End

b) Wish list (Admin)

1. START
2. Declare Array list of String type 'bookName'
3. Declare Array list of String type 'bookAuthor'
4. Declare Array list of String type 'bookEdition'
5. Declare Array Adapter 'adapter'
6. Connect to database
7. Get list of wish list items from database
8. IF count of list row is not zero
9. Add book name from database to bookName
10. Add book author from database to bookAuthor
11. Add book edition from database to bookEdition
12. REPEAT Until all Row's information is retrieved
13. Declare and initialize String variables name =bookName, author= bookAuthor and edition=bookEdition
14. Display values of name, author and edition in table row tr
15. Repeat until every arraylist data is displayed in table

16. END

c) ***Fine system***

1. Start
2. Declare Array list of String type ‘bookName’
3. Declare Array list of String type ‘StudentName’
4. Declare Array list of String type ‘ReservedDate’
5. Declare Array list of String type ‘TotalFine’
6. Declare and initialize NUM_DAYS=5 (Number of days allowed by library to borrow book without paying fine)
7. Connect to database
8. Get list of all reserve table data
9. If Count of rows in list not zero
10. Add book name from database to bookName
11. Add student name from database to StudentName
12. Add reservation date from database to ReservedDate
13. REPEAT until all rows from list is retrieved
14. Get current date and store it to variable todaydate of Date data type
15. Calculate difference between current date and reserved date
16. Calculate to get difference in days and store it to Long variable diffindays
17. IF diffindays>NUM_DAYS
18. Calculate totalfine by (diffindays-NUM_DAYS)*10
19. Add totalfine to TotalFine array list
20. Declare and initialize String variables name =bookName, student= studentName, reserve= ReservedDate and displayFine= TotalFine
21. Display values of name, student, reserve and displayFine in table row tr
22. Repeat until every arraylist data is displayed in table
23. END

Flowchart

Flowchart is used for describing flow of a computer program. It drawing-based representation of algorithm. Unlike texts in pseudo-code and algorithms, flow of program is described using graphical objects.

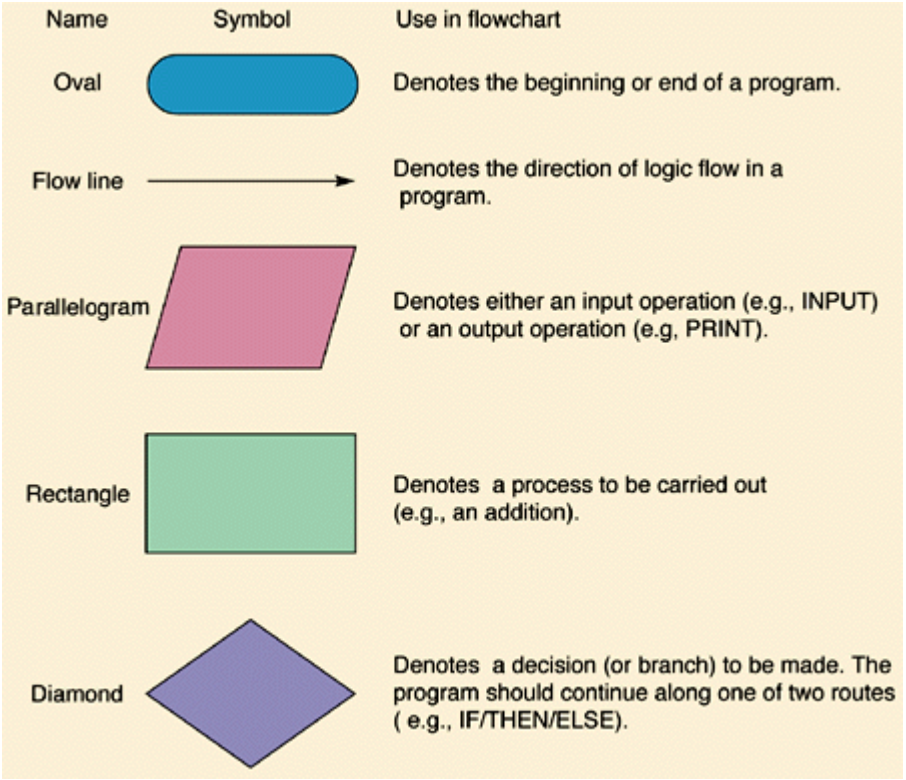


Figure 5 Various controls of Flowchart

Each flowchart has start and ending represented with oval. Input and output in programs are represented with parallelogram shape. Diamond shape is used for decision, which is used for showing one or two flow route of program. Two different route describes two different scenario i.e. one for if condition is true and another for condition is false. Process such as calculation, assignments etc. are described using rectangle shape. And finally, arrow sign is used to show flow of program. To design solution for the problem, flowchart for different components of library management system are drawn below.

1. Login System
2. Delete User
3. Fine system
4. Check wish list (Admin)

Context Diagram

Context diagram demonstrates high level structure of a project. It is graphical description of system that describes various elements of project and their connection with core process. Context diagram is used for describing high level overview of the system. Context diagram for Library management system is drawn below [Figure 2].

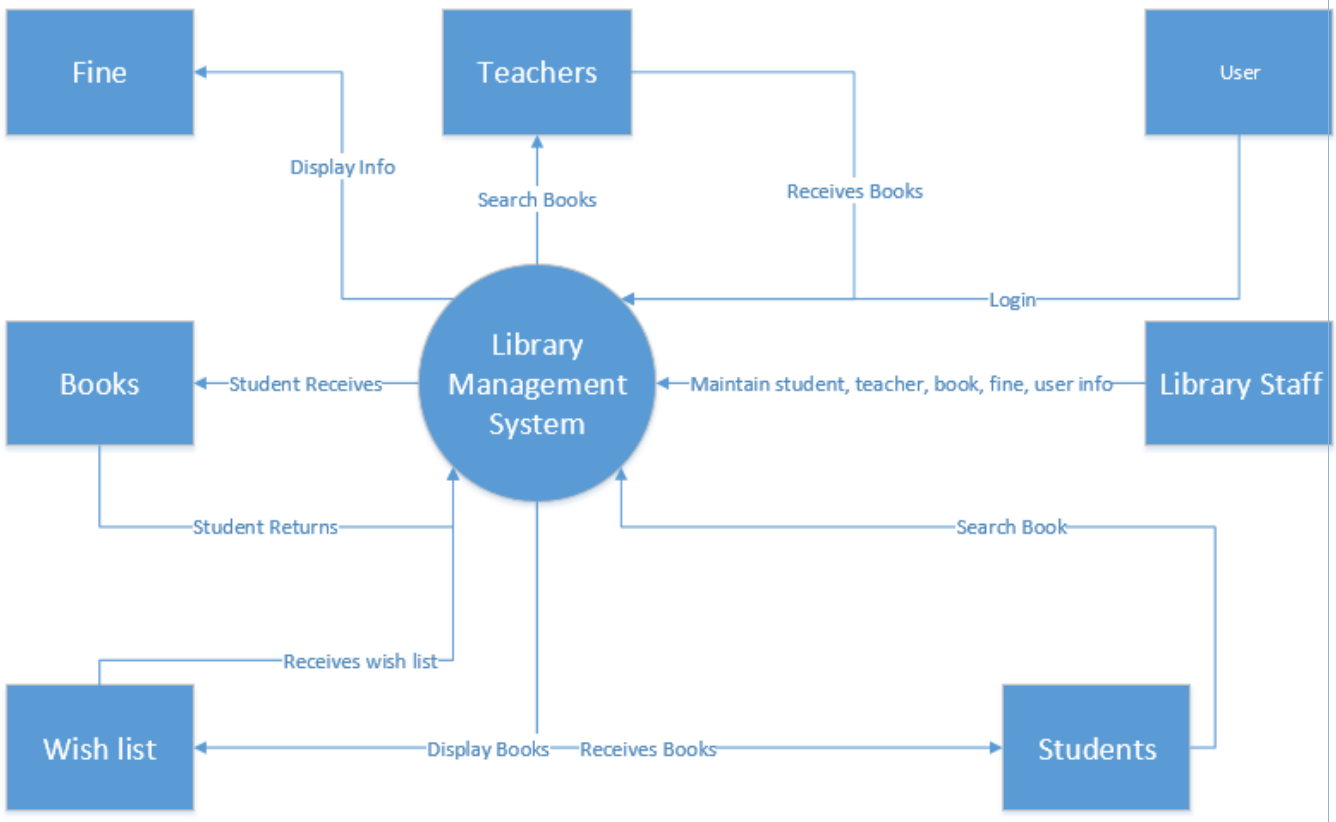


Figure 6 Context Diagram for Library Management system

In context diagram above [figure 2] of library management system has external entities such as students, library staffs, books etc. All External entities are linked with main process. This will help to analyze system boundaries. Programmer will understand what components are required to be included while developing the system.

0 Level DFD

A DFD (Data Flow Diagram) is graphical representation of data flow within a program or system. 0-level DFD is lower level form of context diagram. This diagram illustrates how data is data is processed in the system. . Flow of data and their direction between different processes, entities are shown. DFD for the library management system is drawn below.

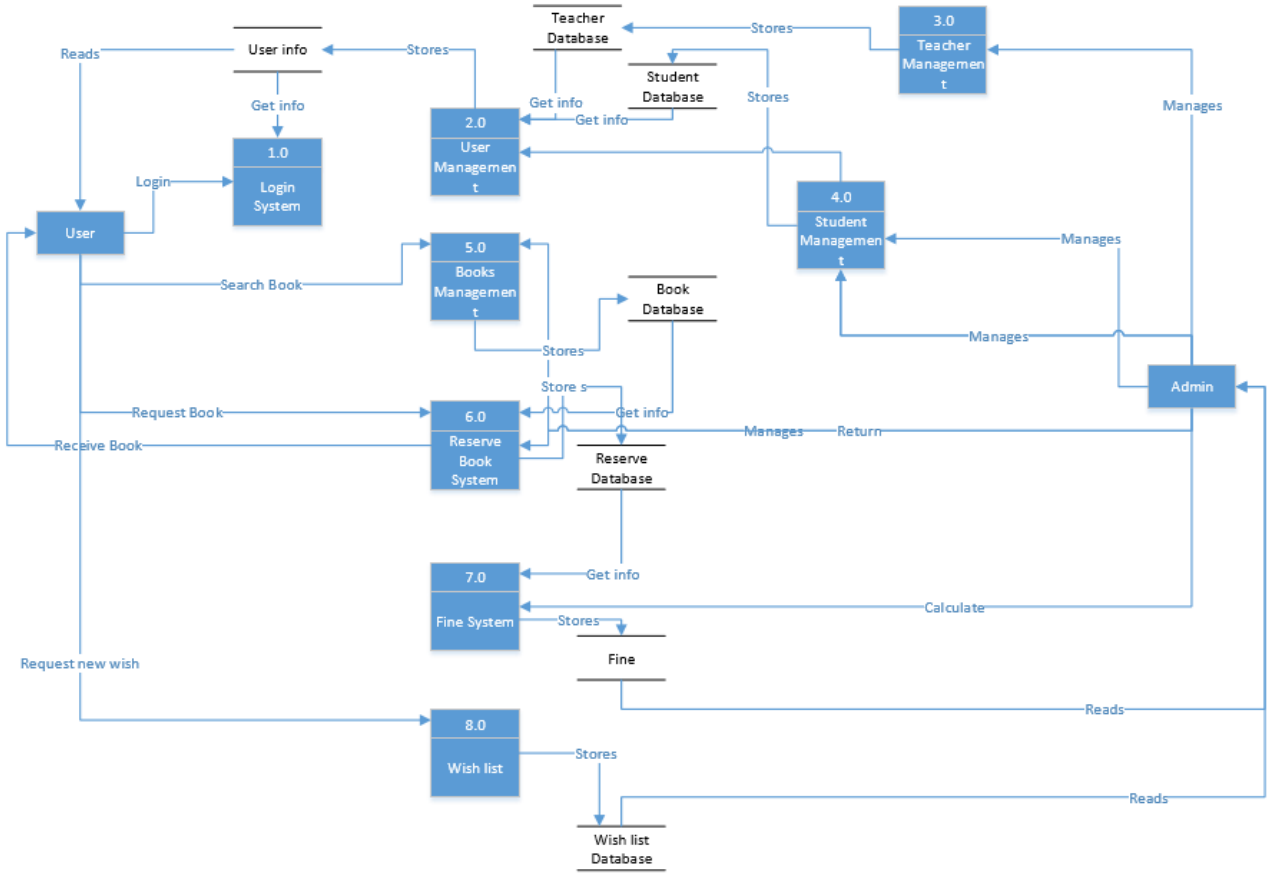


Figure 7 Data Flow Diagram for Library Management System

Designed zero level data flow diagram for library management system demonstrates various top level processes and their inter action with each other and external entities. According to prepared design, user will use user database to get user info while login into the system. Similarly, User information process will take student and teacher information to store data in database. User will request book, request new book wish and book system will provide book to user. File system will utilize information from reserve table to calculate and store fine in fine table. And similarly, designed DFD describes data flow in the system.

Use case diagram

Use case diagram is a tool that is utilized while analyzing a system for development. It simple yet has significant value while investing a system. Diagram describes different user/actor of the proposed system and their roles. This allows programmer to understand boundaries of the system. It helps to understand system requirements. Use case diagram for library management system is developed below.

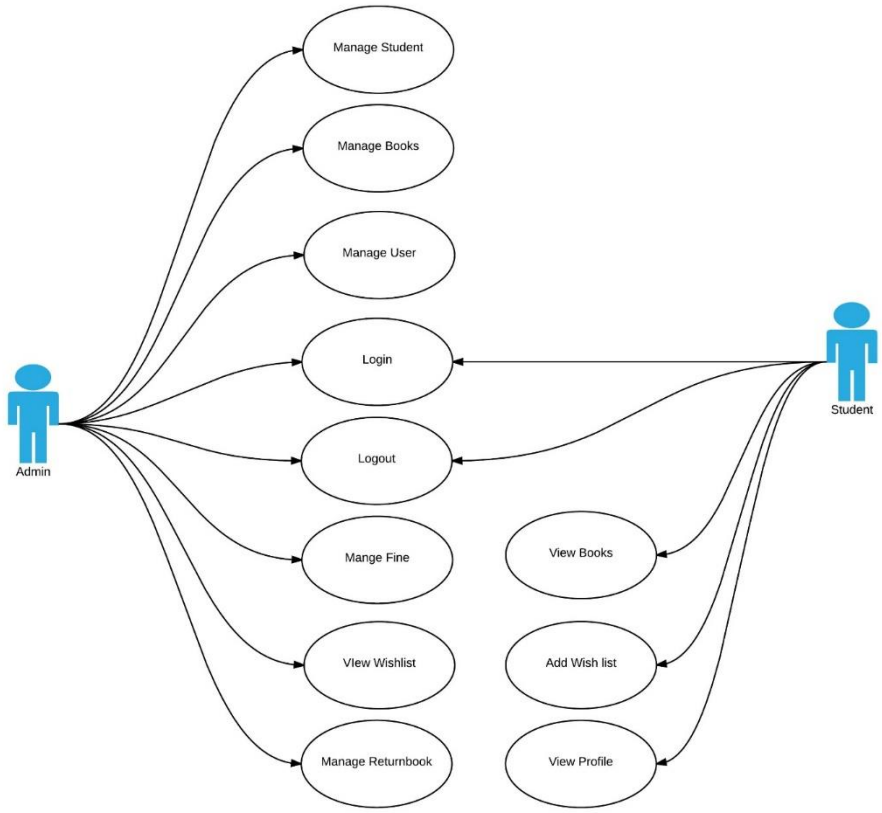


Figure 8Usecase diagram for the library management system

Above use case diagram allows programmer to understand various factors and their relation with different component of the system. Library system has two actors, admin and user. Admin interacts with manage books, students, user, fine, and return book. He can also view wish list data. Similarly, student has relation with view books, view profile and add new wish. Both student and admin uses login and logout use cases.

ER diagram

In application development, E-R diagram is used for graphically demonstrating various Entities, their attributes, relationships and their attributes.

Entity

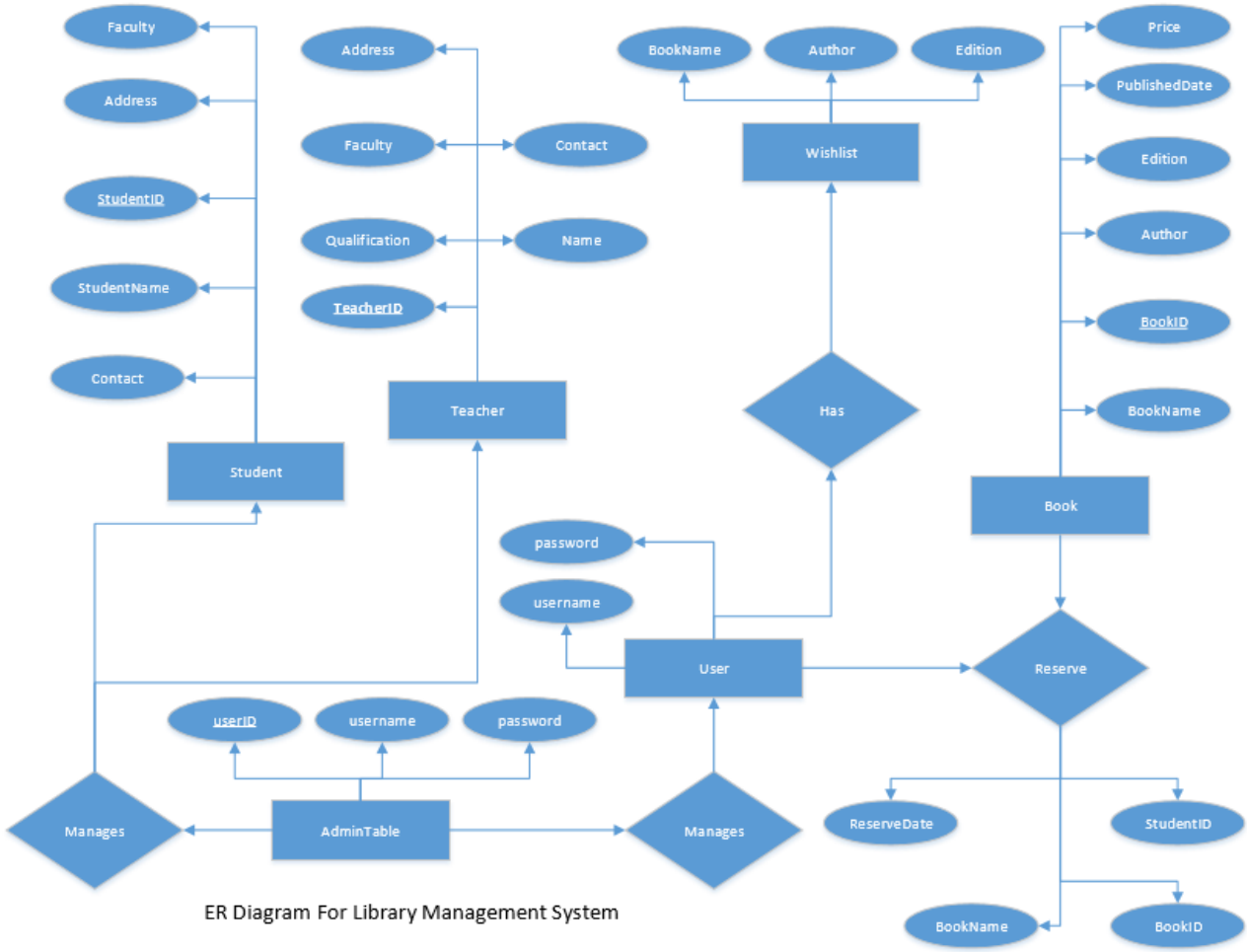
Entity represent objects relevant to the system. For example in current scenario, book or students can be entity. In ER diagram such entity is represented using rectangle shape.

Attributes

Attributes are characteristic, properties of an entity or relationship. For example, student can have attributes like name, age, faculty etc. Attributes in ER diagram is represented by oval shape.

ER diagram is used for designing database solution for the given problem. It is like graphical representation of tables in database, their attributes and relation between different tables. It is very useful to draw ER diagram as it makes creating real database lot easier.

ER for library management system consists of entities such as teacher, books, students, and admin and their relationship. ER diagram for current scenario is drawn below.



ER Diagram For Library Management System

Summary

This paper utilized various tools and technology while designing library management system for ISM College. Pseudo-code, algorithm, flowchart for the program is designed. To further analyze system DFD and use case diagram has been prepared. To design the system, range of methods and techniques has been utilized.

Task 4 [2.2]

Explain the components and data and file structures required to implement the given design.

Introduction

This paper identifies and explains various classes, files/tables and variables required while developing java solution for ISMT College.

Class

Android offers various pre-build class through APIs that programmer can utilized while developing a software. Inbuilt and custom classes used in developed system are listed in table below along with reason to use them.

Inbuilt class Name	**Requirement in given design**
android.content.Context;	permits access to application-specific assets and classes
android.database.sqlite.SQLiteDatabase;	Enables to manage SQLite database
android.widget.Toast;	Provides quick little message for the user
android.os.Bundle;	Allows to map from string to parceble types
android.widget.Button;	Allows to utilize push-button widget.
android.app.Activity	Used while creating UI for user

Custom class Name	**Requirement in design**
AdminPanelActivity	To create Admin Panel UI and write codes for Admin Panel
addBookActivity	To create Book windows UI and write codes for Add Book
UserpanelActivity	To create Client Panel UI and write codes for Client Panel
viewBookActivity	To create View Book UI and write codes for View Book
adminFineActivity	To create Fine window UI for admin and write codes for Fine
clickHandler	Inner class to handle Button click

Methods

Various oftenily used methods in the developed system is listed here along with the reason to use them

Name of Method	**Requirement in design**
onCreate	onCreate method is used for linking java class with UI and running statement that should run when that class is called.

onClick	onClick method is used for handling button click
Open	Open method is used for opening database connection
onCreateContextMenu	This method is used for handling context menu creation
onContextItemSelected	This method is used for handling context menu item selection

Files

File structure for the given system is planned below. System requires various java file, xml file, image files while developing the library system.

XML Files	Requirement in design
Main_menu	manage menu UI
Style	Plan and manage style sheet for application works similar to CSS
Activity_main	Create UI layout for Login window
Admin_panel	Create and manage UI layout for Admin dashboard
Admin_wishlist	Create and manage UI layout for Admin side of wish list

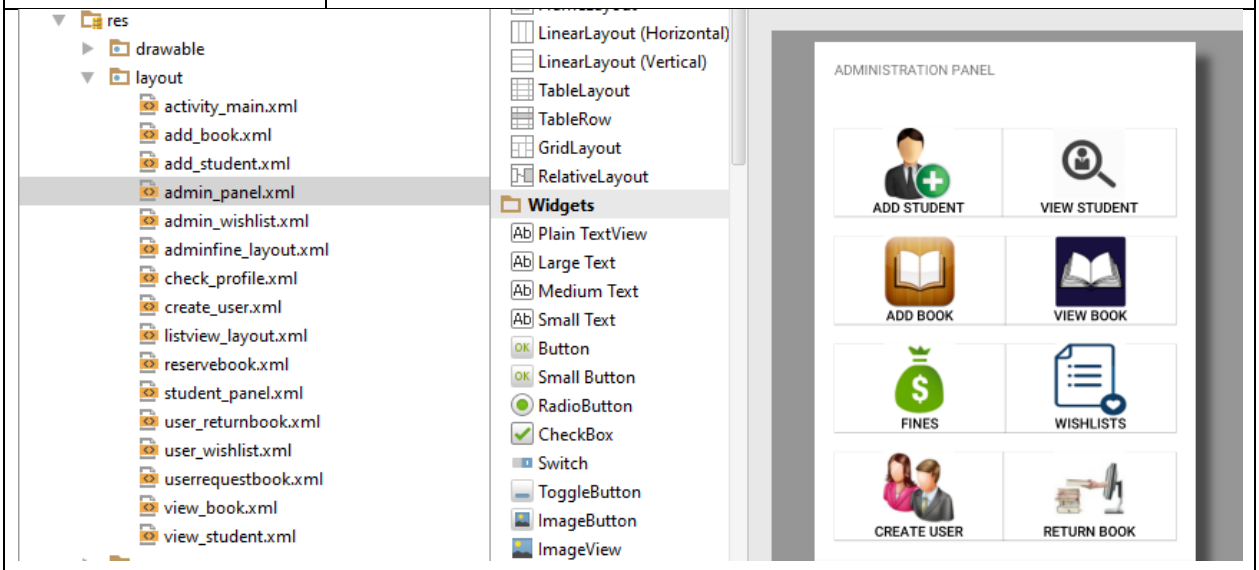


Figure 9 ALL Required XML file Structure for the system

Image Files	Requirement in design
Ic_launcher	Servers as ICON for the system
Rsz_fine	Fine logo in admin dashboard
Rsz_addStudent	Add Student logo in admin dashboard
Rsz_searchbook	Search book logo in admin dashboard
Rsz_wishlist	Wish list logo in admin dashboard
Rsz_AddBook	Add Book logo in admin dashboard

Rsz_Adduser	Create User logo in admin dashboard
Rsz_Returnbook	Return Book logo in admin dashboard

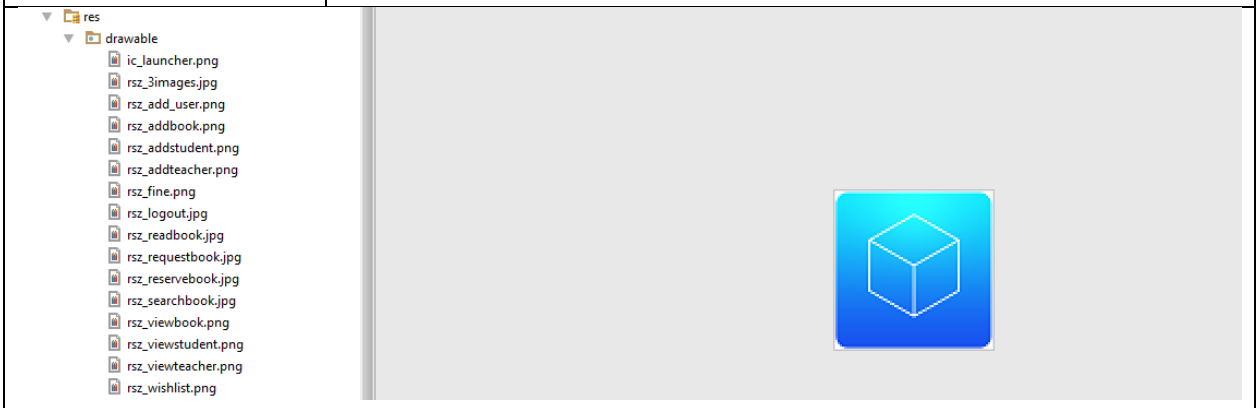


Figure 10 ALL Image Files required for system development

Java Files

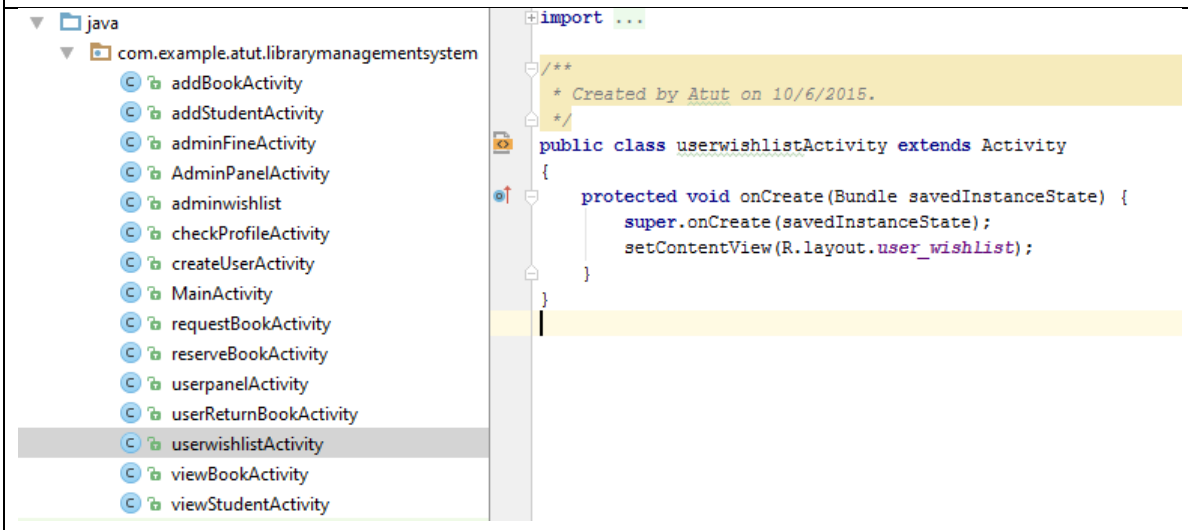


Figure 11All Required Java Class for the system

Variables: Various object/primitive variable required in system is listed here.

Class/Method	Object/Variable Name	Type	Requirement in design
addStudentActivity	dB	dBConnection	To connect to database
	sDatabase	SQLiteDatabase	Perform database related actiivity
	add	Button	to get activity from Add button

	bundle	Bundle	To pass data to another window
	studentid	EditText	To get value of student id from UI
AdminFineActivity	dB	dBConnection	To connect to database
	sDatabase	SQLiteDatabase	Perform database related actiivity
	Sql_Search_reserved	String	Query string for selecting reserve data from database
	bookName	ArrayList<String>	To store book names
	studentName	ArrayList<String>	To store student names
	reservedDate	ArrayList<String>	To store reserve dates
	TotalFine	ArrayList<String>	To store fines
	NUM_DAYS	int	For fix date to calculate fine
	tableLayout	TableLayout	To design table for UI
	tr	TableRow	To add row in table

Tables: Database structure required for system is listed here.

Name of Table	Attibutes	Requirement in design
tbl_Student	StudentID, Name, Address, Faculty, semester	Stores student information
tbl_Book	BookID, Name, price, edition, publisheddate	Stores books information
tbl_Reserved	BookiID,StudentID,Date	Stores reserve information of books
tbl_Wishlist	BookName, Edition, Author	Stores
tbl_User	Userid, username, password	Store user information
tbl_Teacher	TeacherID, Name, Address, Qualification, subject	Store teacher information
tbl_adminLogin	userID, username,password	Store admin user information

Summary

This document identified, planned and explained file structure and data as well as component structure for the system. Require classes, methods, variable (primitive and object), files were identified and listed along with their requirement.

Task 5

Implement a Java programming solution based on prepared design. **[3.1]**

Define relationship between objects to implement the design requirement. **[3.2]**

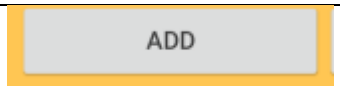
Implement object behaviors using control structures to meet the design algorithms. **[3.3, D2]**

Make effective use of and Integrated Development Environment (IDE) including code and screen templates. **[3.4]**

Identify and implement opportunities for error handling and reporting. **[3.4]**

Preparing UI Design using XML

To implement android Java programming, Graphical User Interface () need to be prepared. In android, separate layouts are prepared for each components of application. In this project, UI is designed using XML code. One of the key benefits of using XML to code UI is it allows to keep UI separate from application behavior. It means freedom in UI design. Each UI components is drawn using corresponding XML code. Table below demonstrates various components and layouts designed for library management system and XML required to render them.

Layout\Component Name	Screenshot
Button	

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add"
    android:layout_weight="5"
    android:id="@+id/btnAddBook"/>
<Button
```

Edit View	

```
<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/etEdition"
    android:singleLine="true"/>
```

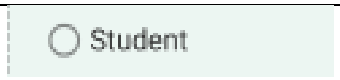
	

```
<TextView
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="Book Name" />
```

Calendar

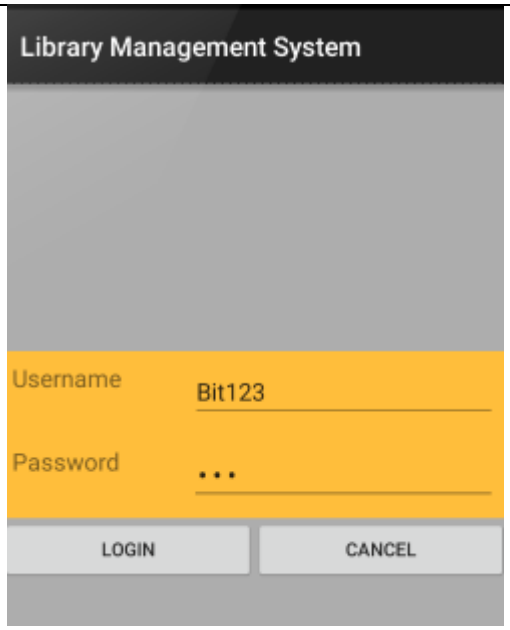

```
<DatePicker
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/dpPublished"
    android:calendarViewShown="false"
    android:layout_marginTop="20dp">
</DatePicker>
```

Radio Button

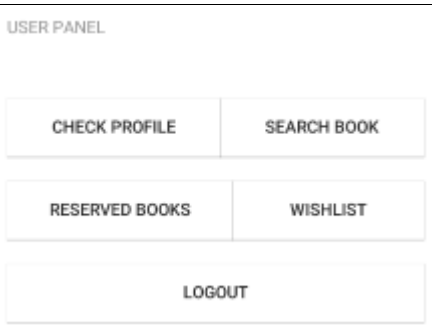
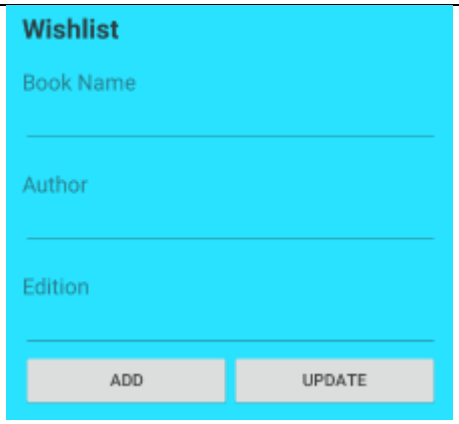
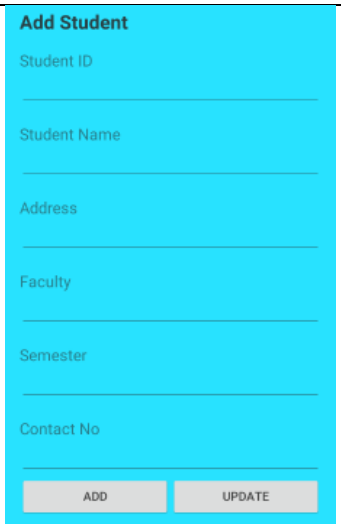
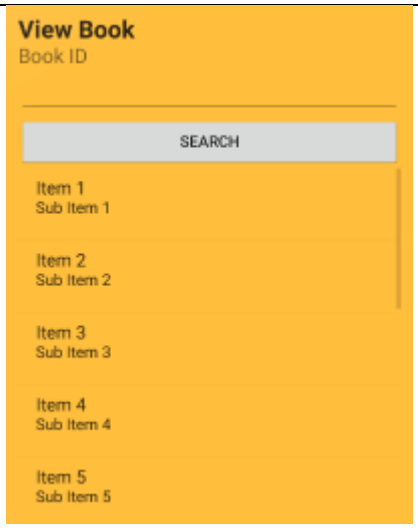
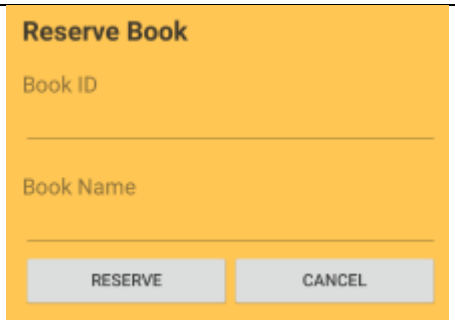
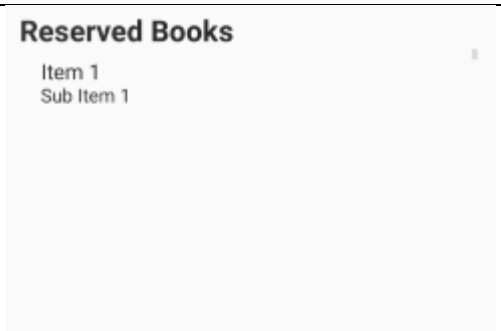



```
<RadioButton
    android:onClick="onRadioButtonClicked"
    android:layout_width="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_height="wrap_content"
    android:text="Student"
    android:id="@+id/rbRegisterUserStudent"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:layout_weight="5"
    />
```

Login Page



Admin Panel

User Panel	Wish list (Student)
	
Add Student	**View Book**
	
Reserve Book	**Return Book**
	

XML Code for Admin Panel Layout

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:id="@+id/svAddBook"
    android:background="@android:color/holo_orange_light">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="20dp"
        android:id="@+id/view">
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Reserve Book"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:textStyle="bold"
            android:layout_marginTop="20dp"
            android:id="@+id/tvReserveBook" />
        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Book ID"
            android:layout_marginTop="20dp"
            android:textAppearance="?android:attr/textAppearanceMedium"/>
        <EditText
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:id="@+id/etReserveBookID"
            android:singleLine="true"/>


        <TextView
            android:layout_marginTop="20dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="Book Name"/>
        <EditText
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:id="@+id/etReserveBookName"
            android:singleLine="true"/>

        <LinearLayout
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Reserve"
                android:layout_weight="5"
                android:id="@+id/btnReserveBook"/>
            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Cancel"
                android:layout_weight="5"
                android:id="@+id/btnReserveCancel"/>
        </LinearLayout>
    </LinearLayout>
</ScrollView>
```

XML code above if for designing UI layout for Admin Panel. It is basically combination of XML statements of various android UI components. Similarly, each layouts required for library management system are designed.

Relationship between Objects

Schema Diagram

To identify and demonstrate various database objects, attributes and their relationship with other objects, schema diagram is drawn below. This diagram is based on designed ER Diagram in analysis phase. A schema diagram is universally understood simple graphical representation of database tables and their attributes as well as their relation with other tables' i.e. primary key and foreign keys. Schema diagram helps developer to create analyze and design database for the application.

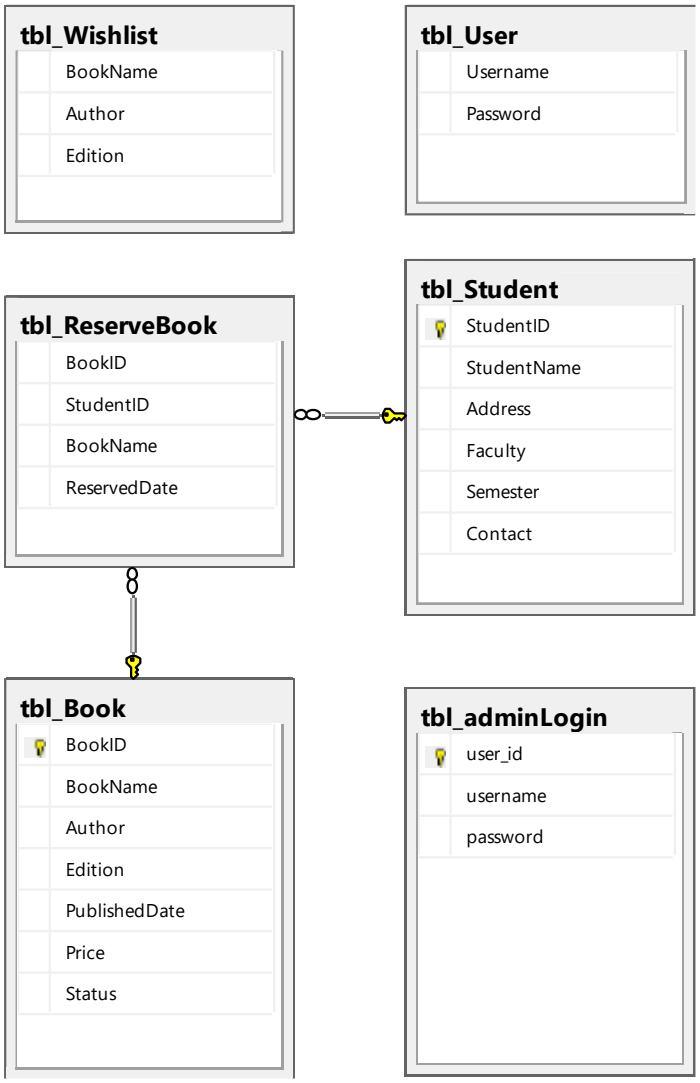


Figure 12Schema diagram for library management system

Implementation: Control Structures

In java programming, control structure refers to condition and looping structure. These structures are used in programming to implement programming logics. With help of these structures developer achieve business requirements. Various control structures used in designed library management system is listed below here in this document.

If Else Statement

If else statement allows to check condition and control flow of program. Keyword IF is followed by check condition. If the condition is true, it executes statements from if block. If condition is false, statements from else block is get executed.

```
public class Encapsulation {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
         int a = 3;
         if (a%3==0)
          System.out.println("a is divisible by 3");
         else
             System.out.println("a is not divisible by 3");
           }

    }
```

```
a is divisible by 3
```

Example 1 if else statement

```
if(cursor.moveToFirst()){
    do{
        temp.setBookname(cursor.getString(cursor.getColumnIndex("BookName")));
        temp.setStudentname(cursor.getString(cursor.getColumnIndex("StudentName")));
        temp.setReserveddate(cursor.getString(cursor.getColumnIndex("ReservedDate")));
        bookinfo.add(temp);
        allBookId.add(cursor.getString(cursor.getColumnIndex("BookID")));
    }
    while (cursor.moveToNext());
}
else{
    Toast.makeText(getApplicationContext(),"No Data Found",Toast.LENGTH_LONG).show();
}
```

Figure 13 Implementation of if else statement in program

In library management system if else statement is used on various occasions. For example, in figure 2 if else statement is checking if cursor has row(s). If cursor has rows, statements inside if block is executed otherwise toast massage is thrown to inform there is no data in the cursor.

Switch

Switch statement is alternative to if else statement. Unlike if else, switch can have multiple execution path. It allows to choose a block of statements based on the value of an expression. Various statements are written for cased based on value of expression. In example 2 below here show a's value is 2. This means, case 2 will be triggered resulting execution of statements inside case 2: block.

```
public class Encapsulation {

    public static void main(String[] args) {
        int a = 2;
        switch (a){
        case 1:
            System.out.print("Hello value of a is 1");
            break;
        case 2:
            System.out.print("Hello value of a is 2");
            break;
        default:
            System.out.print("Hello value of a is not 1 or 2");
            break;
        }
    }
    }
```

```
a is divisible by 3
```

Example 2 switch statement

```
switch (v.getId())
{
    case R.id.btnAddStudent:
    {

            String Sql_InsertData="Insert into tbl_Student(StudentID,StudentName,Address,Faculty,Semester,Contact)" +
                    " values ('"+studentid.getText()+"','"+studentname.getText()+"','"+studentaddress.getText()+"'," +
                    "'"+studentfaculty.getText()+"','"+studentsemester.getText()+"','"+studentcontact.getText()+"')";
            sDatabase.execSQL(Sql_InsertData);
            Toast.makeText(getApplicationContext(),"Information saved successfully!!!",Toast.LENGTH_LONG).show();

        break;
    }
  case R.id.btnUpdateStudent:{
        String Sql_UpdateData="Update tbl_Student SET StudentName='"+studentname.getText()+"',Address='"
                +studentaddress.getText()+"',Faculty='"+studentfaculty.getText()+"',Semester='"+studentsemester.getText()
                +"',Contact='"+studentcontact.getText()+"' WHERE StudentID='"+studentid.getText()+"' ";
        sDatabase.execSQL(Sql_UpdateData);
      Toast.makeText(getApplicationContext(),"Student update successfully", Toast.LENGTH_SHORT).show();
      break;
    }
}
```

Figure 14 switch statement in library management system

In android library management system switch case is used extensively. Mainly to identify which button is pressed by user and act based on button pressed. In example above (figure 3) show switch

case being used for providing alternative path based on id of view. If return value of expression is R.id.btnaddstudent then block of statements to add student in database is executed. If return value of expression is R.id.btnupdatestudent then block of statements to update student in database is executed.

Loop

Loop statement in programming is used for iterate certain code statements. To implement designed Java program, various looping statements are used.

Do While

Do while loop is used for executing block of statements (s) repeatedly until checking condition is true. But statement executes least once regardless of condition. This means statement(s) runs once even if condition is false. Example:

```
public class Encapsulation {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int i = 0;
         do {
           System.out.println(i);
           i++;
         } while (i < 10);
    }
}
```

```
0
1
2
3
4
5
6
7
8
9
```

Example 3 do while

Implementation of do while

```
do{
    bookName.add(cursor.getString(cursor.getColumnIndex("BookName")));
    bookAuthor.add(cursor.getString(cursor.getColumnIndex("Author")));
    bookEdition.add(cursor.getString(cursor.getColumnIndex("Edition")));
}
while (cursor.moveToNext());
```

Figure 15 do while in library management system

Data from database is extracted and stored in cursor. Cursor can have multiple rows of data based on SQL query executed. With help of do while data from each rows are retrieved and stored in variable. In figure 4, values from cursor is being stored to variable repeatedly while cursor moves to next row is true.

For

```
public class Encapsulation {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int a;
         for (a=0; a<=5; a++)
        {
         System.out.println("a = " +a);
        }
    }
}
```

```
a = 0
a = 1
a = 2
a = 3
a = 4
a = 5
```

Example 4 for loop

```
for(int i=0;i<bookName.size();i++)
{
    tr=new TableRow(this);
    tr.setLayoutParams(new
    ActionBar.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,ViewGroup.LayoutParams.WRAP_CONTENT));
    name=new TextView(this);
    author=new TextView(this);
    edition=new TextView(this);

    name.setText(bookName.get(i));
    author.setText(bookAuthor.get(i));
    edition.setText(bookEdition.get(i));
    tr.addView(name);
    tr.addView(author);
    tr.addView(edition);
    tableLayout.addView(tr,new TableLayout.LayoutParams(
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.WRAP_CONTENT));
}
```

Figure 16 for loop in library management system

Error handling

To handles errors and exception cases in developed system, it have error handling mechanism. Error handling allows identification and control for error in the system. Library management system uses try-catch method to handle both checked and unchecked exceptions.

Checked

These are the exceptions that are tested at compile phase. If certain statement within a method tosses a checked exception, then the method should either handle the case or it must indicate the exception case utilizing throw keyword.

Unchecked

Unlike checked, unchecked exception are not checked in compile phase. Exceptions are unchecked, so it is not constrained by the compiler to either handle or indicate the exemption. It is up to the developers to be cultivated, and determine or get the special cases.

Try Catch

Try catch method is utilized by developed system to handle exception events. A block of java statement is written inside try segment. If java statement has error or throws expectation, flow of program will go to catch segment preventing system from crashing/terminating. Usual catch segment is used for toasting error massage for users. Code below demonstrate implementation of Try-Catch method in library management system to tackle exceptional events.

```
try
{
    dB=new dBConnection(getApplicationContext());
    sDatabase=dB.Open();
    switch (v.getId())
    {
        case R.id.btnAddBook:
        {
            String actual_date=publisheddate.getDayOfMonth()+"-"+publisheddate.getMonth()+"-"+publisheddate.getYear();
            pickeddate.setText(actual_date);
            String Sql_InsertData="Insert into tbl_Book(BookID,BookName,Author,Edition,PublishedDate,PRICE,Status)" +
                    " values" + " ('"+bookid.getText()+"','"+bookname.getText()+"','"+bookauthor.getText()+"'," +
                    "'"+bookedition.getText()+"',"+"'"+pickeddate.getText()+"','"+price.getText()+"','AVAILABLE')";
            sDatabase.execSQL(Sql_InsertData);
            Toast.makeText(getApplicationContext(),"Information saved successfully",Toast.LENGTH_LONG).show();
            break;
        }

    }
}
catch (Exception ex)
{
    Toast.makeText(getApplicationContext(),ex.getMessage(),Toast.LENGTH_LONG).show();
}
```

Figure 17 try catch in library management system

User of IDE

Android Studio

Developed library management system utilizes features provided by Android studio tool. Android studio is android based Integrated Development Environment that provides extraordinary support for android development. It includes android SDK and helps to plan, troubleshoot, test and manage the development phase. Android studio is great tool for android development using JAVA

languages. While developing library management system, Android studio helped for managing codes, files, resources, finding errors and testing the apps.

Figure 18 Android Studio loading

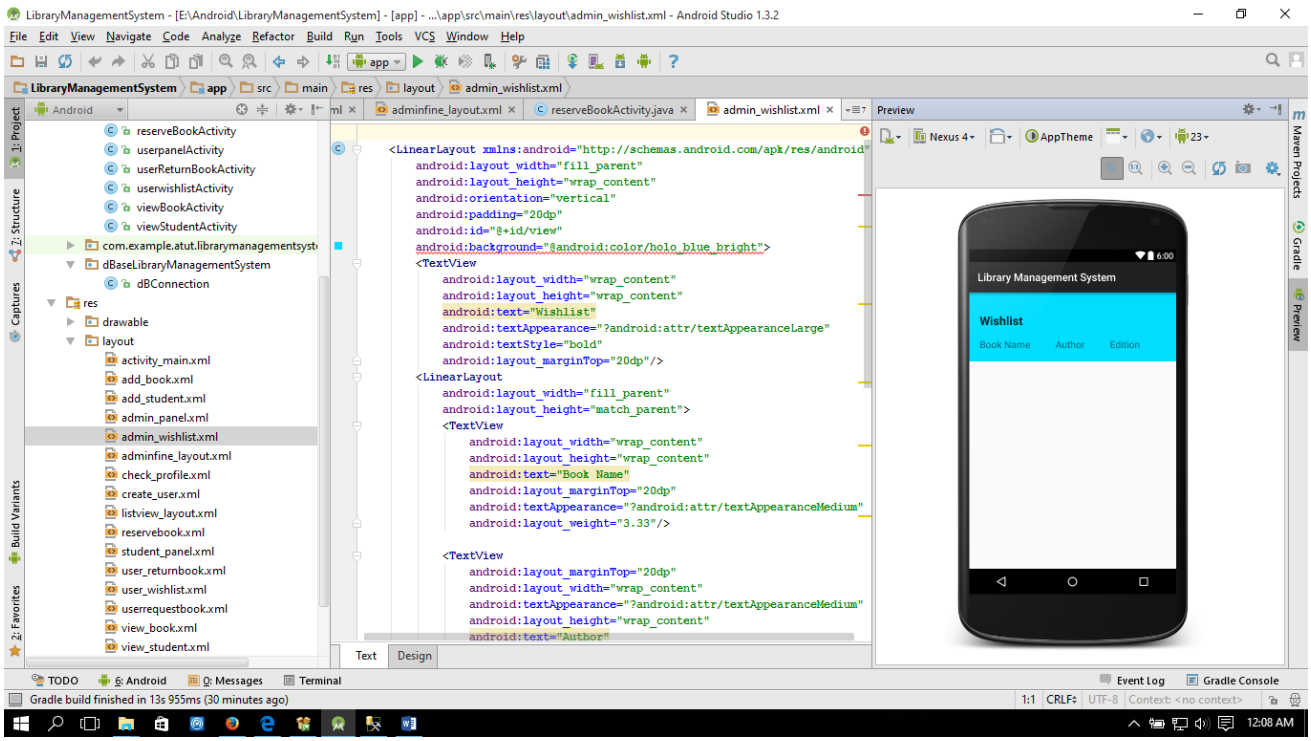


Figure 19 Android Development using Android Studio

Summary

Implementation of java programming to build library system for ISMT College has been documented in this report. While documenting, Implementation of UI, object relation, control structures and exception handling is emphasized. UI of the system is developed using XML. Various data objects used in database and their relationship is described using schema diagram.

To achieve system requirement, programmer has utilized various control structure such as for loop, switch case, if-else statement, and do while statement. Implementation of control structure is documented in this report. And Implementation of library system make effective use of android studio IDE.

Task 6 [4.1, 4.2, D1]

Critically review and test the Java programing solution and **analyze** actual test results against expected results to identify discrepancies.

Introduction

Java programming testing is a procedure used to recognize the rightness, culmination, and quality of developed android system. It incorporates an arrangement of exercises directed with the expectation of discovering blunders in programming so it could be redressed before the item is discharged to the end clients. This paper documents various test procedure carried out on developed system to identify and address issues in the system.

Unit Test

This type of testing involves testing of smallest testable module of the system. Though Unit test can be automated, manual unit testing is carried out for developed system. During unit test, functionality of various modules of system is tested.

Stress Test

Stress testing helps to study system's ability to work under extreme condition. Unfavorable condition is created to check application behavior. In current system, while performing stress test, validation testing is utilized as stress test.

Integrated Test

Integration test evaluates how different module of the system interacts with each other. Does UI module interact correctly with coding? To perform integration testing in current system, integration with database system is tested.

Compatibility Test

Compatibility test evaluates system's behavior in different environment. Such as different OS, different hardware system, different user etc. Developed library management system is tested by analyzing its behavior during running app in different Android version and different users.

Each of the test carried out is documented in pages below as test logs.

Critical Analysis of Test results

Various test methodologies are used while testing newly developed Android based library management system using Java programming. Each test result is documented as test log for further study. Utilized test methodologies are unit testing, stress testing, integration testing and compatibility test.

Test procedure was carried out successfully. While performing unit test of the system various unit's functioning was tested including function of student add button, search button, function of student edit menu, dropdown control etc. Test result was evaluated against expected output. All test results were similar to expected outputs. This allows programmer to understand that all modules of the system are functioning correctly. After unit testing, stress testing was carried out. While performing stress testing, invalid username and password were used for trying to access system and then study system's behavior. As expected, test result showed system blocked access to system and displayed warning massage. In another stress test, creating multiple users with same username was tried, but as expected system prevented creating multiple accounts. Stress test of the system showed stress management mechanism is taken care of in developed system.

During test process, integration test was carried out in system by testing systems integration with database. First new student was added using add student module. Test result showed data is stored in database. Similarly, other test were performed to test integration of android application with SQLite database. And finally, compatibility test was carried out. For this system was executed in android version 4.4.4. And 5.0. Test showed system was able to function normally in both environment. All test carried out helped developer to understand the system.

Critical Review

Introduction

ISMT College is established international institution that offers quality education. College does not have automated library management system hence, to improvise on quality of library system, college has decided to developed android based library management system. This paper critically review several components of newly implemented java solution. This solution is developed using designs that are prepared using various tool and technologies such as DFD, context diagram, flow chart etc.

Body

Implemented system emphasize on achieved requirement of the ISMT College. I has various necessary modules for library management such as user management, book management, fine management, student management etc. System utilizes SQLite technology to provide database support. All information from the system is stored in database. It has simple and lite user interface. Controls are smooth which makes interaction with developed system better. While critically reviewing system, following strength and limitations are noted.

Strengths

- System has all necessary modules to support library management system
- System is developed using Java programming which offers plenty of advantage of java
- Developed solution has authentication system that helps preventing unauthorized access
- Various test is carried out on system to identify bugs and errors

Limitations

- System does not have web support
- System is developed targeting Android user, hence other mobile carrier will not able to take advantage of developed system.

Summary

Developed android system is programmed using Java language. This utilizes all feature of java to create the system. System has all necessary modules for book, user, student, fine, wish list management. This solution possesses some of the highlight and some of limitations. One of the key significance of the system is it satisfies clients requirement, that is to automate library management system. Key limitation portrayed by system is it does not have web support which will cause number of issue. For example, it prevents multiple user login using multiple devices. At last, developed system has successful satisfied system and client requirement event though there are plenty of room for improvements.

Task 8 [4.4]

Create user documentation for the developed Java program solution.

Introduction

This documents assists as user manual for system operator of this library management system. This document will help users while using the system. This is further alienated into three different categories according to the nature of the guide.

General

General help will provide necessary guideline to perform basic activity such as installation, starting app and default login information.

Installation

To install the package into the device follow these procedure.

1. Download app from Google play store
   Alternative 1
   Download LIBRARY.APK file from college's official website
   Alternative 2
   Manually load LIBRARY.APK file into the mobile through USB cable.
2. For Alternative 1 and 2 follow these steps
   i. Locate LIBRARY.APK in mobile's my files [Figure1]
   ii. Click on LIBRARY.APK
   iii. Click on Setting[Figure2]
   iv. Tick on Unknown Sources [Figure3]
   v. Tick on Allow this installation only→ Ok [Figure 4]
   vi. Click on Install→ Done [Figure5]

Staring Up application

1. Locate Library Management System Logo [Figure6]
2. Click on the logo→ Click on →Done

Default Administrator login

To login the system first time, system has default admin authentication information.

1. Username: Admin
2. Password: 1234

Login

1. Start the application
2. Provide username and password [Figure7]
3. Click on login
4. Retry if login failed/wrong username password
5. Done

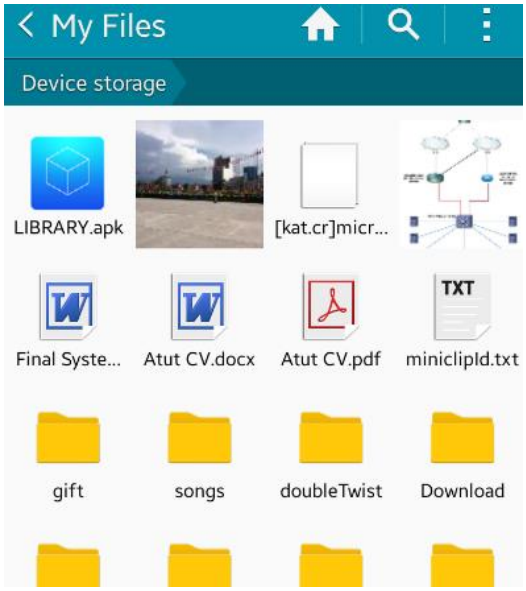


Figure 1

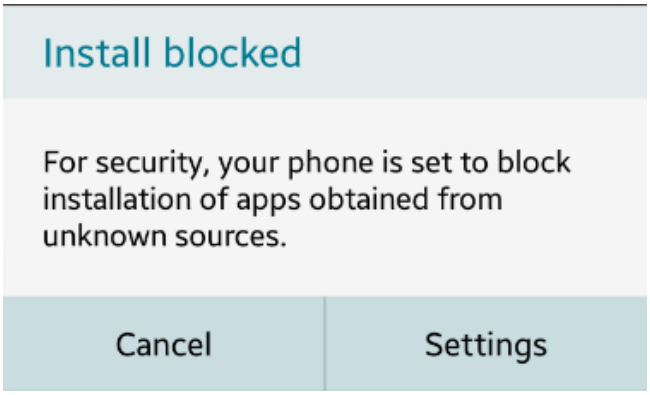


Figure 2

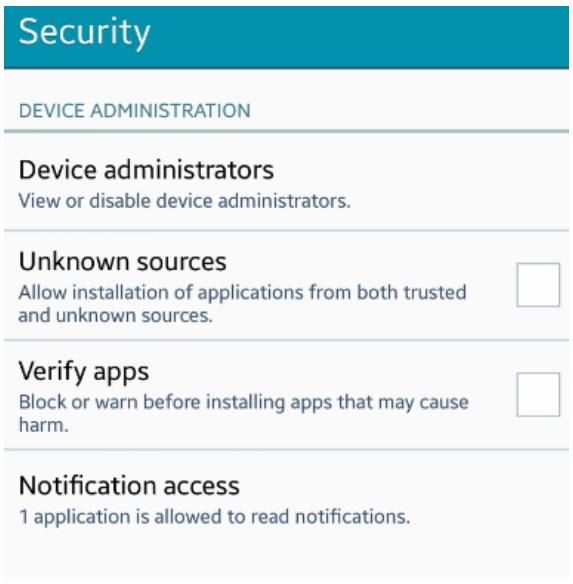


Figure 3

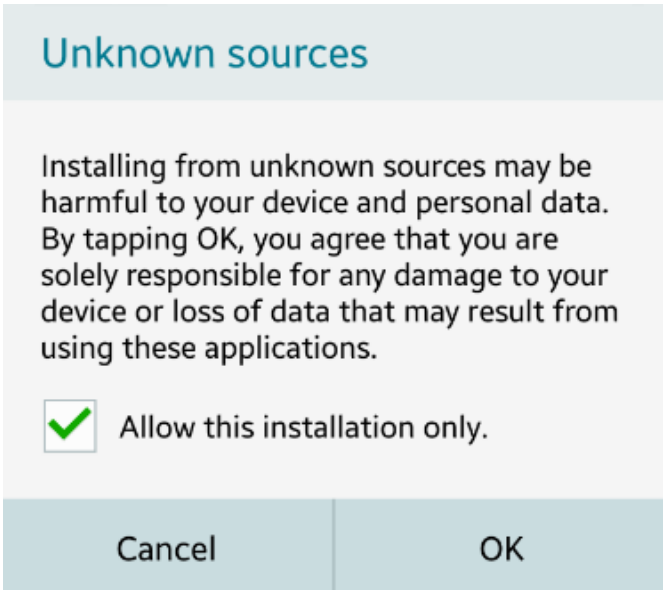


Figure 4

Figure 5

Figure 6

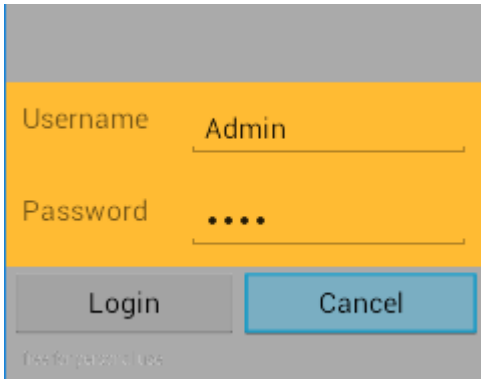


Figure 7

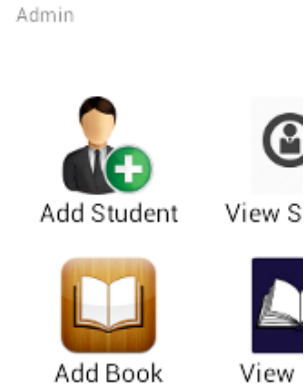


Figure 8

Administrator Operation

Student Management

Add Student

1. Login as Administrator
2. In Admin Panel click on Add student logo[Figure8]
3. Fill up the add student form
4. Click on Add student button[Figure9]
5. Done

Figure 9

Search Student

1. Login as Administrator
2. In Admin Panel click on View Student [Figure10]
3. To view all students click on search button without providing any ID in the text field [Figure11]
4. To filter student, provide student ID and click on search→ Done

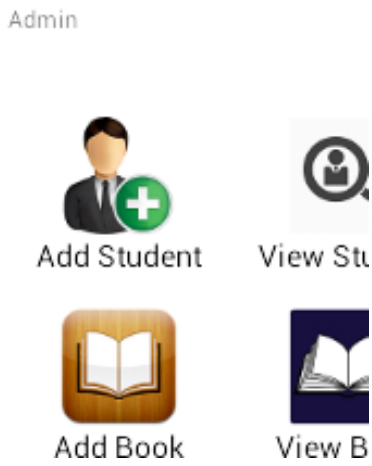


Figure 10

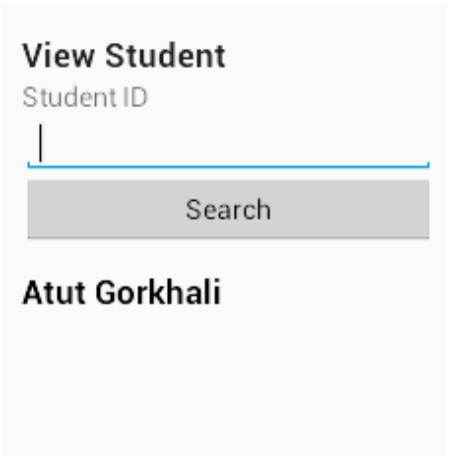


Figure 11

Edit Student

1. Login as Administrator
2. In Admin Panel click on View student
3. Search the student
4. Long press on student Name [Figure12]
5. Choose Edit from the menu
6. Modify student information [Figure13]
7. Click on Update Student→ Done

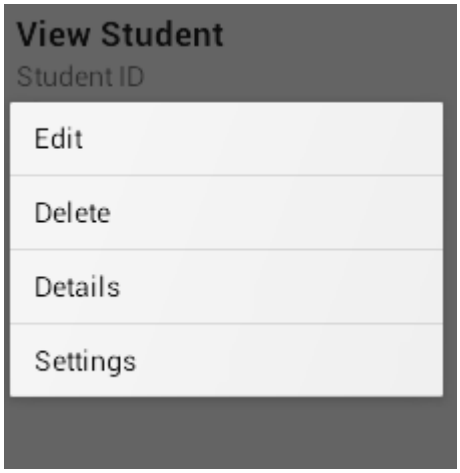


Figure 12

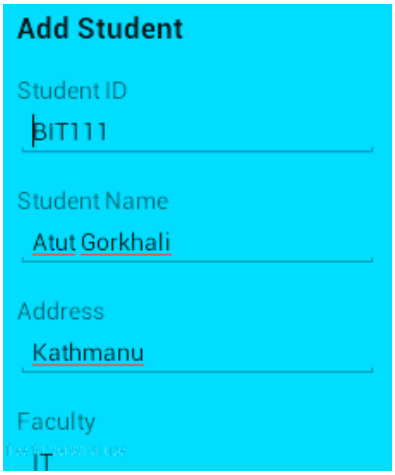


Figure 13

Delete Student

1. Login as Administrator
2. In Admin Panel click on View student

3. Search the student
4. Long press on student Name
8. Choose Delete from the menu[Figure14]
5. Confirm delete by pressing → Yes→ Done [Figure15]

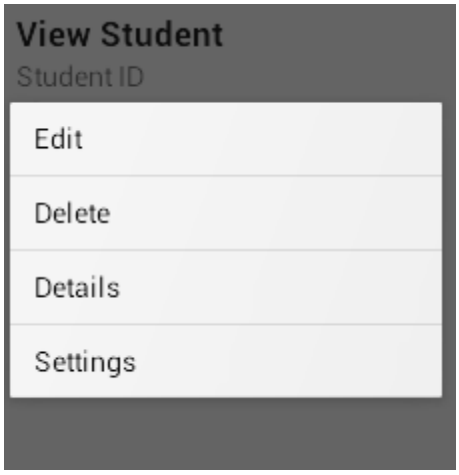


Figure 14

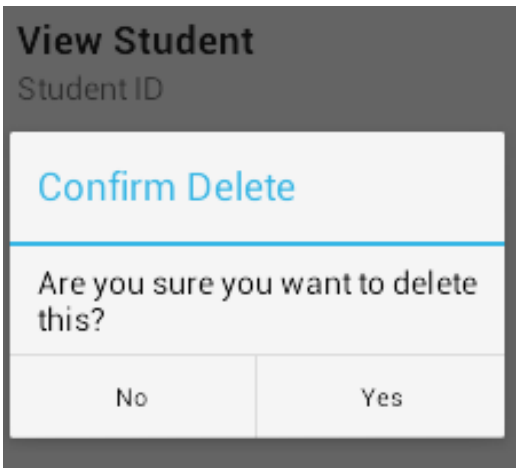


Figure 15

Create New User

1. Login as Administrator
2. In Admin Panel click on Create User
3. Select Member type→ Teacher or Student [Figure16]
4. Set password for the user [Figure17]
5. Confirm password
6. Click on Register→ Done

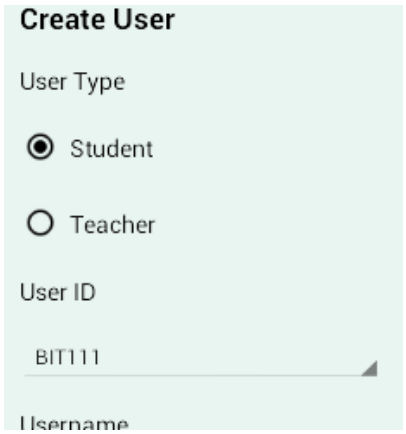


Figure 16

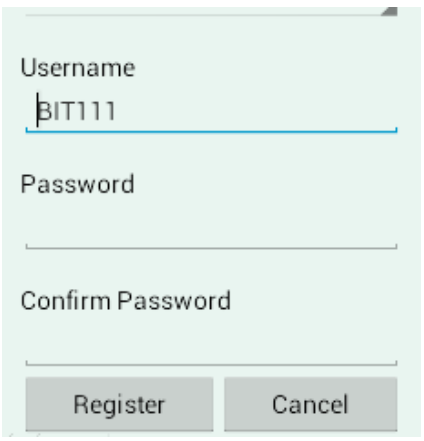


Figure 17

Return Book

When student wants to return borrowed book, follow these procedure.

1. Login as Administrator
2. In Admin Panel click on Return Book
3. Long Press on the book to be return [Figure 18]
4. Click on return→ Done

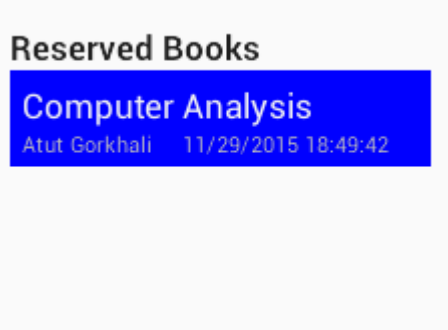


Figure 18

Book and Teacher Management

User need to follow similar procedure to manage book and teachers.

Member Operation

Check Profile

1. Login as Member user
2. In Admin Panel click on check profile
3. Done [Figure 19]

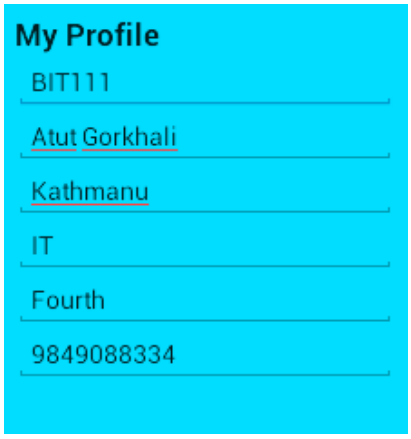


Figure 19

Reserve Books

1. Login as Member user
2. In Admin Panel click on Reserve Book
3. Search Required Book
4. Long Press on Book→ Click on Reserve [Figure 20]
5. Done

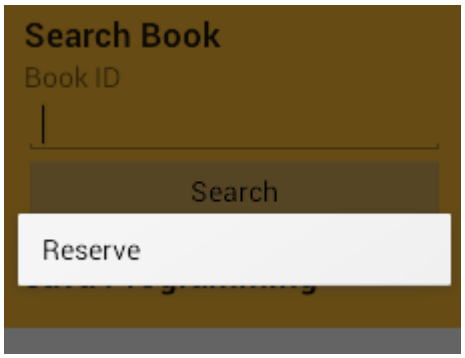


Figure 20

Add wish-list

1. Login as Member user
2. In Admin Panel click on wish list
3. Provide book's information [Figure 21]
4. Click on add→ Done

Figure 21

Logout

To logout while using the system as Admin or Member simply press on logout button from the main panel as shown in figure 22 and 23.

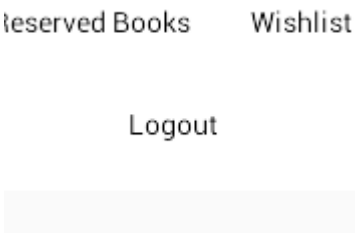


Figure 22

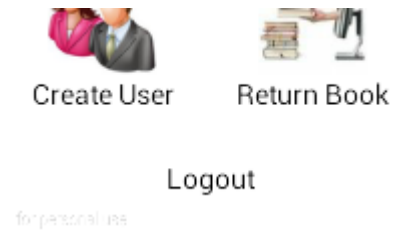


Figure 203

For more help/information

This document provide guideline for operating all features of developed library management system. If user still has any problem he/she may contact IT department or send query to help@ismt.com

Task 9 [4.5, M3]

Create technical documentation for the support and maintenance of Java Program solution.

Introduction

This document servers as technical instruction for support and maintenance of developed Android library management system using Java. This system provides automated library management service to ISMT College. This report includes all information about design, choice of tools, task breakdown approach, information collection mechanism, deployment mechanism, and maintenance techniques for the develop android system.

UI Design using XML

Graphical User Interface (GUI) need to be developed while developing android Java programming. Developed library system utilizes XML technology to design separate Layouts for different components of application. XML allows programmer to separate UI design from application behavior. This provides programmer much needed freedom while developing his/her system. Block of XML code is written for each UI control used in system.

Linking UI with Java Class

To program UI through programming, xml file first need to be linked with corresponding java class file. Following code is used for linking java class with xml file. This allows java class to load content of xml file in android screen when that class is called.

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

Figure 21Loading xml content in screen using java class.

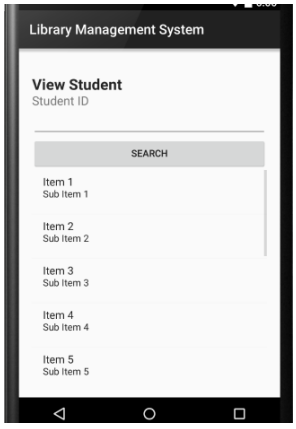


Figure 22 UI design of developed system

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="20dp"
    android:id="@+id/view">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="View Student"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textStyle="bold"
        android:layout_marginTop="20dp"/>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Student ID"/>
    <EditText
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/etSearchStudentID"
        android:singleLine="true"/>
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Search"
            android:id="@+id/btnSearchStudent"/>
    </LinearLayout>
    <ListView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listViewStudent"
        android:textAppearance="?android:attr/textAppearanceMedium"
        ></ListView>
    </LinearLayout>
```

Figure 23 XML code of developed system

AndroidManifest.XML

Each android system need to have an AndroidManifest.xml record (with exactly that name) in its root catalog. This file includes essential information about developed system. It includes required information that is must have to run the system. In addition to other things, this file does the accompanying:

1. It defines all java package in the developed system
2. It describes all system components such as activities, broadcast, services etc.
3. It defines all process that will host components
4. It defines all permission required to access system components
5. It describes all libraries that components requires
6. It also describes minimum level of android API that system requires.

Database Design

To store information about books, students, and fines developed system utilized SQLite as database.

SQLite

SQLite is an open source database, accessible on each android database. It underpins standard relations database elements, as SQL language structure, exchanges and SQL proclamations. SQLite is significantly, the lighter rendition of SQL database, where a large portion of the SQL is not supported on SQLite. SQLite requires small amount of memory which makes it suitable for most of the android phone. Following database is used in developed system.

Designed Database

Tables		
Name of Table	**Attibutes**	**Requirement in design**
tbl_Student	StudentID, Name, Address, Faculty, semester	Stores student information
tbl_Book	BookID, Name, price, edition, publisheddate	Stores books information
tbl_Reserved	BookiID,StudentID,Date	Stores reserve information of books
tbl_Wishlist	BookName, Edition, Author	Stores
tbl_User	Userid, username, password	Store user information
tbl_Teacher	TeacherID, Name, Address, Qualification, subject	Store teacher information
tbl_adminLogin	userID, username,password	Store admin user information

Choice of Tools

Android Studio

Android studio [figure 5] offers set of tool that can be utilized to develop android application. It's an Android centered IDE, composed extraordinarily for the Android advancement. Android studio contains all the Android SDK apparatuses to plan, test, troubleshoot and profile your application. Android studio is broad arrangement of development technology for implementing android system using Java language. Android studio allows programmer to take full advantage of java technology. Furthermore, android studio allows to maintain program file structure, find errors and debug the system.

GENYMOTION

Genymotion [figure 6] allows developer to run android application by providing virtual android environment. This project uses Genymotion to test developed android system.

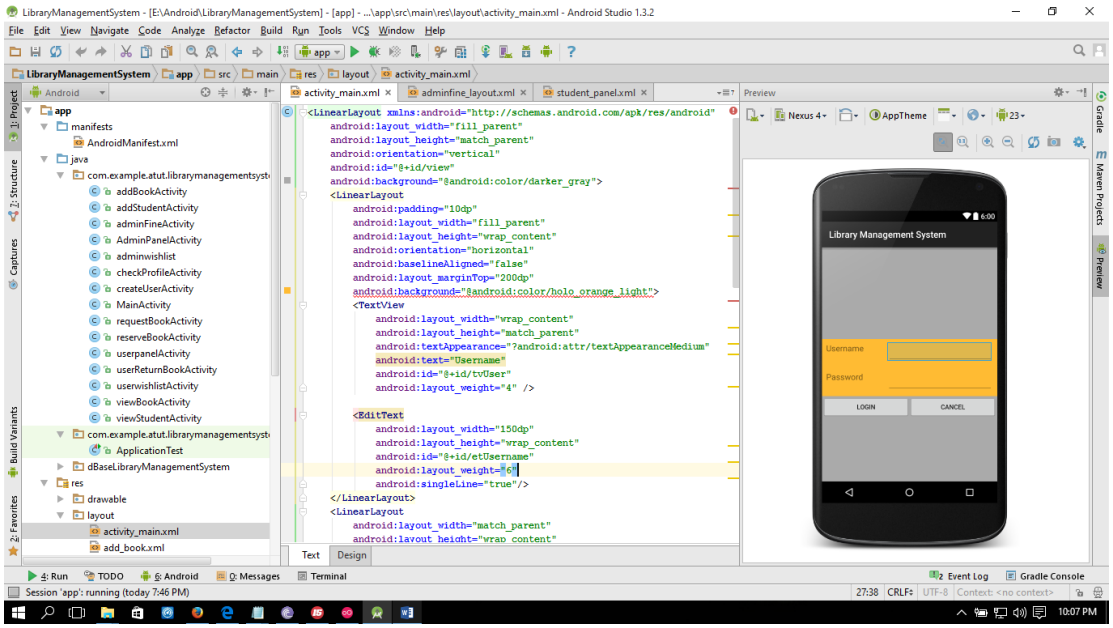


Figure 24Android Studio

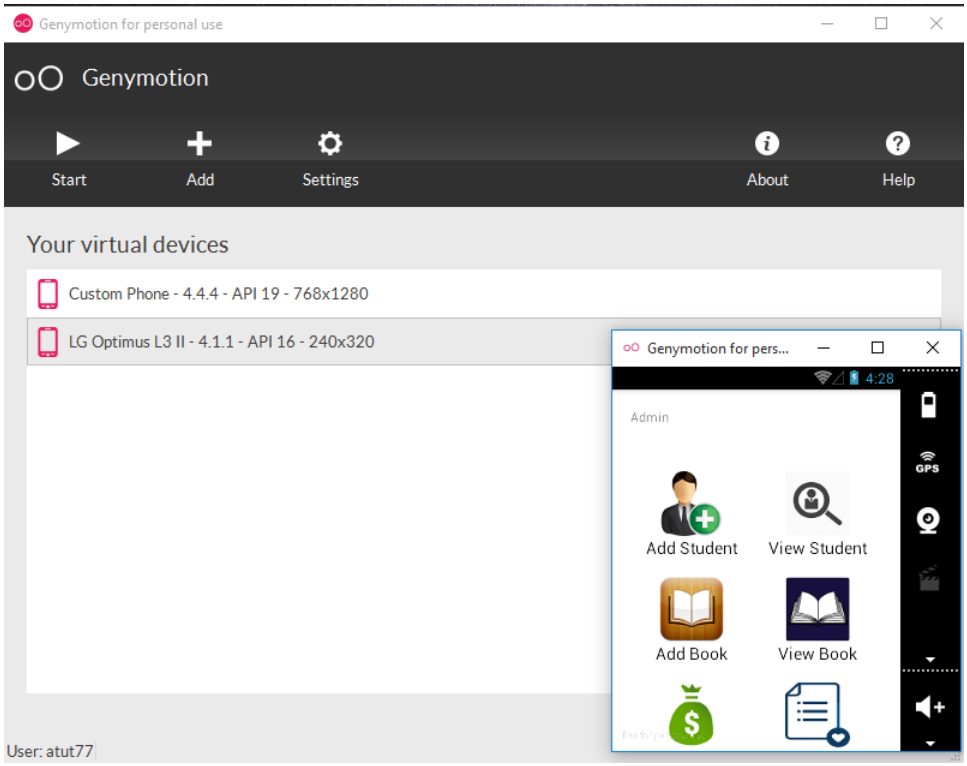


Figure 25Virtual Android Phone in Genymotion

Task Breakdown

Large project is broken down into smaller tasks in order to properly manage plan, design and implement the project. Project "library management system for ISMT" is broken into various phase to manage it properly. To demonstrate all phases and their timeline, Gantt chart for the project is prepare below.

Gantt chart

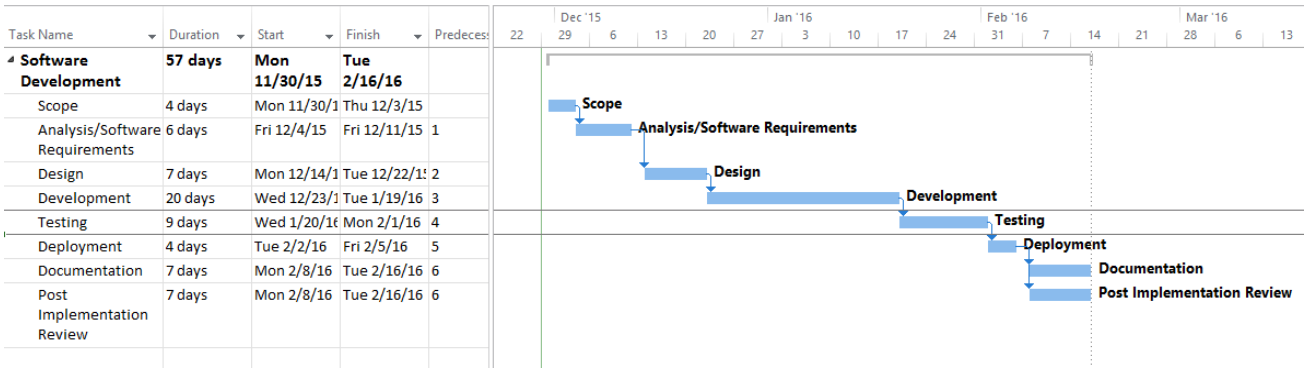


Figure 26Gantt char for the Library Management System

Information Collection

This project utilizes various data and information collection methodologies. To get proper feedback for the system, various fact finding methodologies were used. Feedback of the system included signification and limitations of the system. Following methodologies were used: 1. Interview 2. Surveys 3. Questionnaire.

The system itself has data collection mechanism to collect data from user and store in database. Android system collects information about books, students, user etc. Library management system uses form system to collect data.

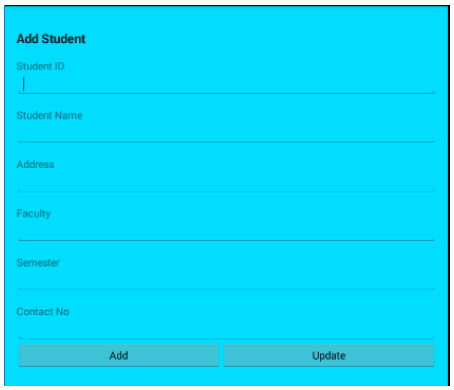


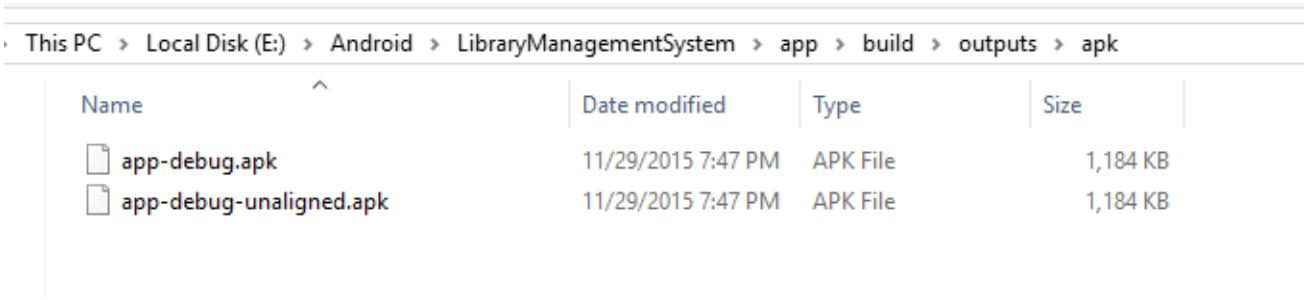
Figure 27 Data Collection method in developed system

Product Deployment

Product deployment is process of publishing/distributing developed app to its customer. Library management system is deployed using two different methods.

Through Manual Distribution/Website

In this method, .APK file is distributed to clients using website or through physical drive such as pen drive. .APK file is located under Root directory→Android→ LIBRARYMANAGEMENTSYSTEM→app→build→outputs→APK



Through Play Store

When app is deployed in store, anyone can download it via internet. To Publish the app in app store follow these procedure.

1. Locate .APK file [Follow Method 1]
2. Browse to developer console of Google play [Search Developer console-Google play] in Google [Figure 9]
3. Sign Up [Figure 10]
4. Click on Add new Application[Figure 11]
5. Provide title for the app→ click upload apk[Figure 12]
6. Select Upload your first APK to Production[Figure 13]
7. Browse and locate APK file from step 1. [Figure 14]
8. After APK upload click store listing from navigation→ Provide Description [Figure 15]
9. Click on + Add Screenshot to add 3 screenshots of the system[Figure 16]
10. Upload store listing icon. [Figure 17]
11. Finally in Store Listing, fill all necessary information then click Pricing & Distribution and provide necessary data[Figure 18]
12. To complete app publishing click on Publish this app" from the "Ready to Publish" from top right corner. →Done[Figure 19]

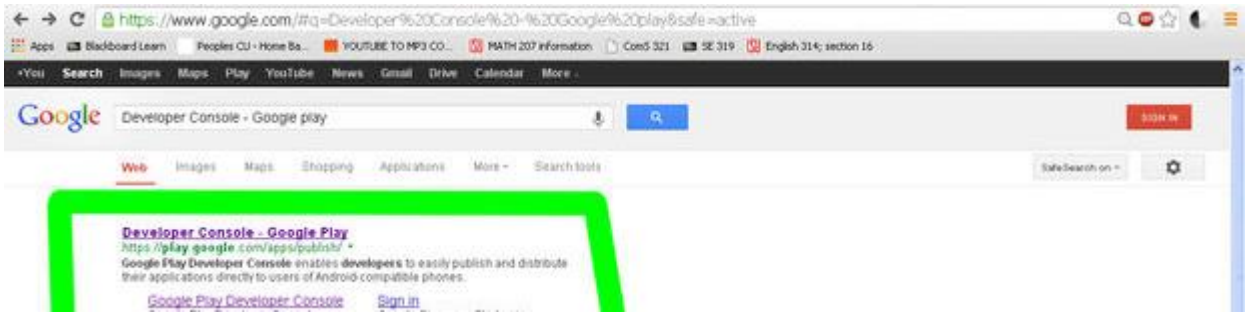


Figure 28 Searching Developer Console

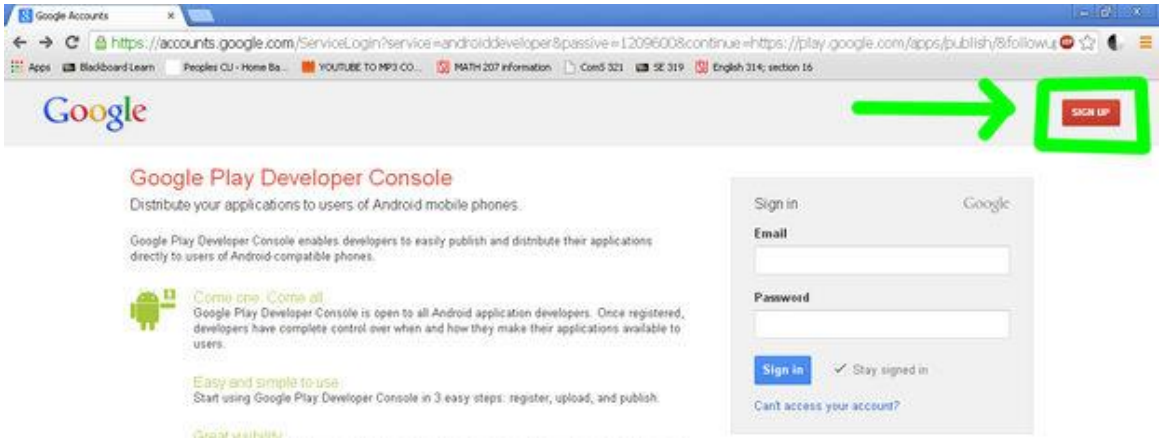


Figure 29 Sign Up

Figure 30 Add New Application

Figure 31 Provide App title

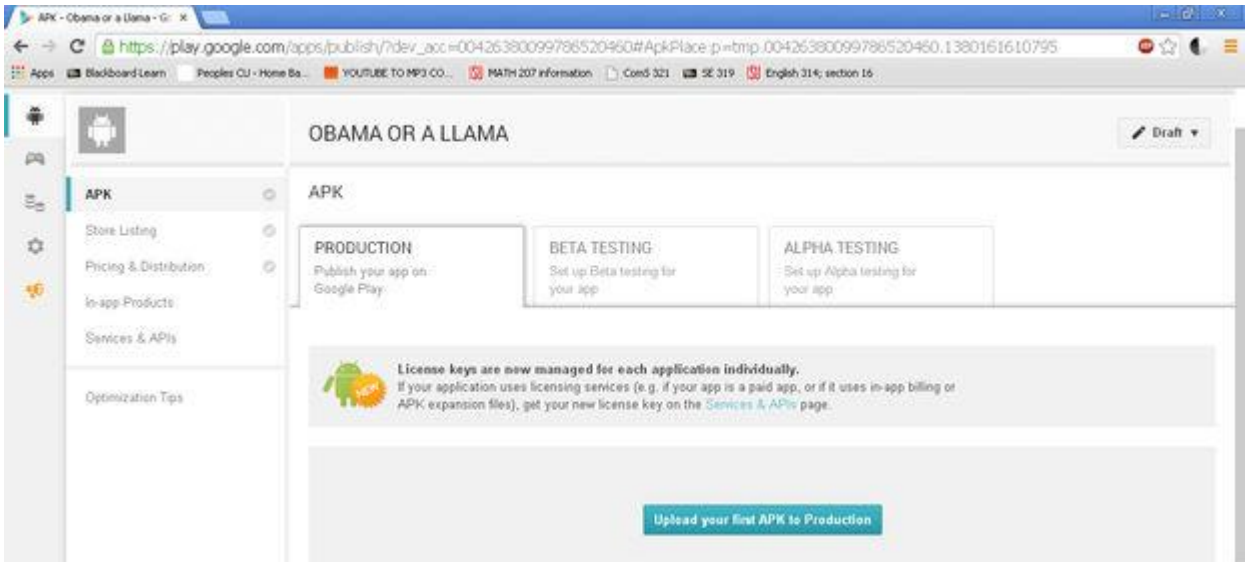


Figure 32Upload your first APK to Production

Figure 33 Uploading APK

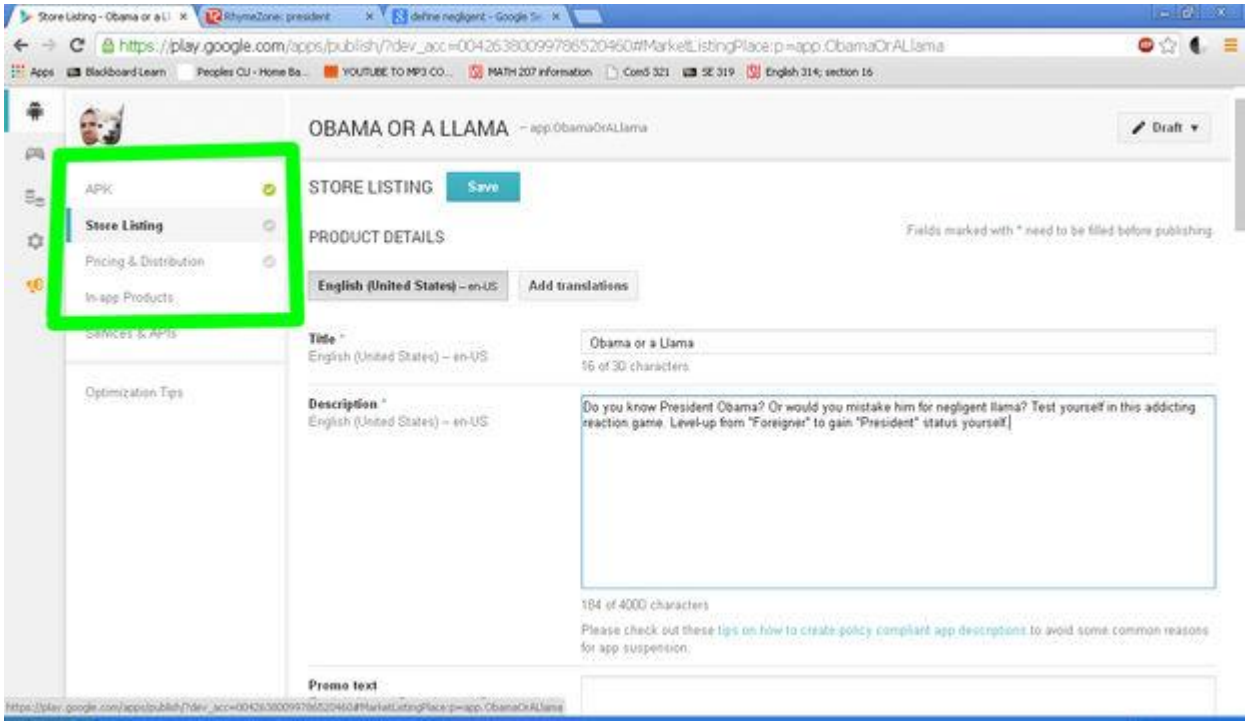


Figure 34 Fill Description

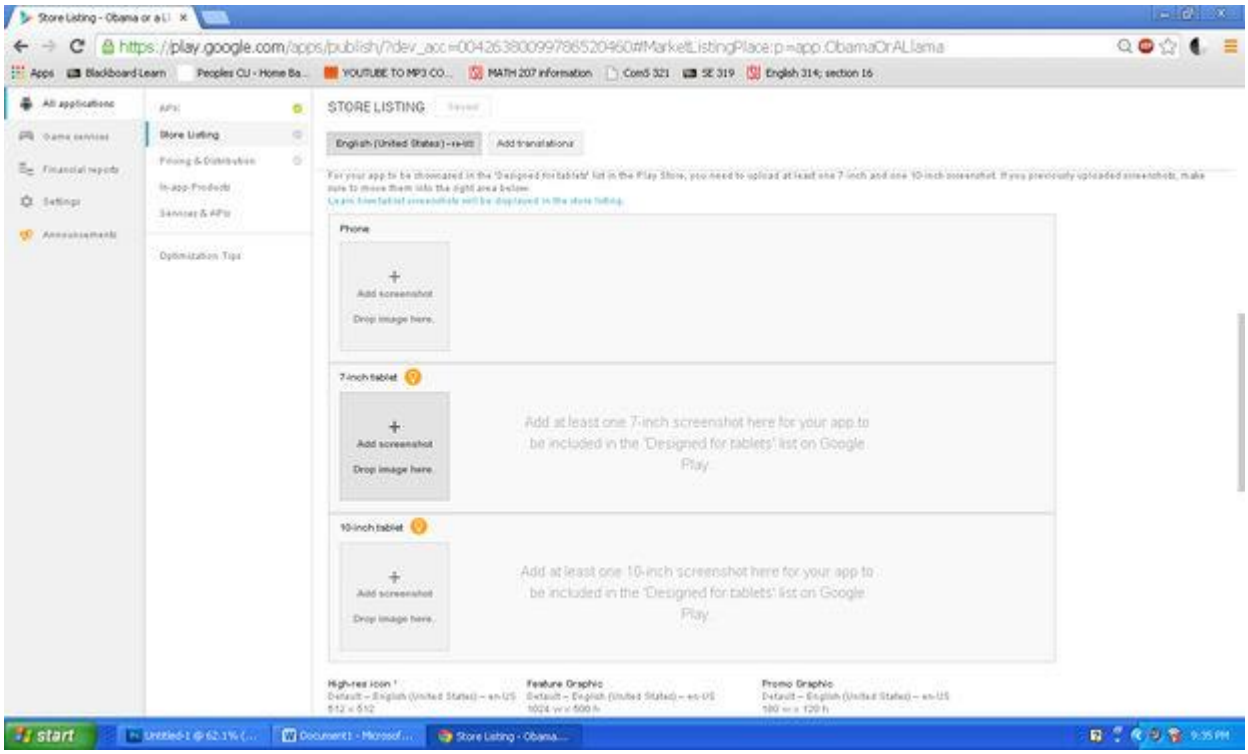


Figure 35 Upload Screenshot

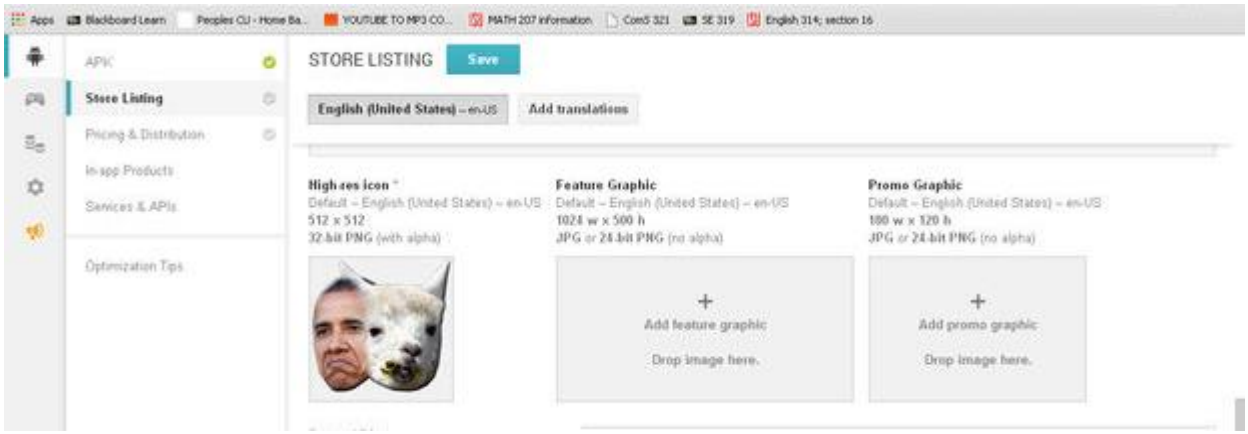


Figure 36 Set App icon

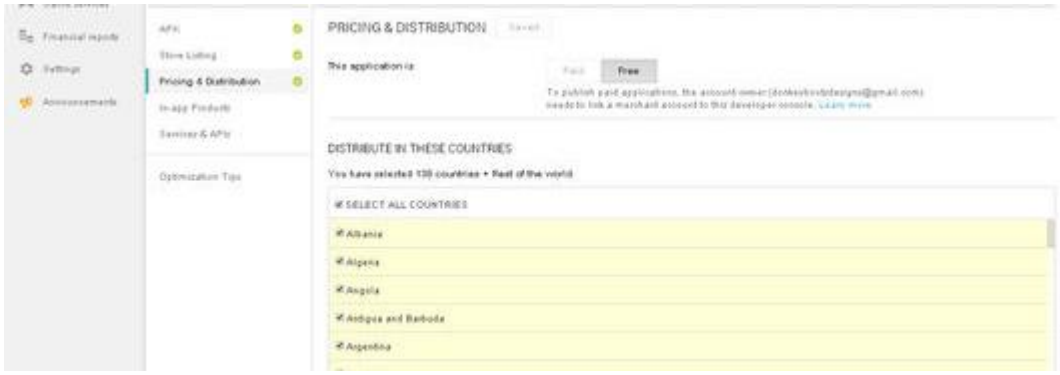


Figure 37 Fill in pricing and distribution

Figure 38Publish the app

Maintenance Techniques

For the maintenance of the system technician can follow guidance provided here below.

- To provide support to user/client follow user manual
- To add new page in the system, create XML layout and java class for the page. → Add information in AndriodManifest.XML
- To modify a page UI, modify relevant XML page

- To add new table in database, Locate java class DBCONNECTION and add new java statements to execute SQL query

For More help/information

This document provided comprehensive technical guidance. If reader still requires further support, kindly mail to support@ISMT.edu.np

Task 7 [4.3, D3]

Evaluate independent feedback on a developed Java programming solution and make recommendation for improvement.

Feedback

Feed is an essential procedure in software development. It allows to understand potential customer's view on the system. It is process of studying customer's reaction to the product. Proper study of feedback helps to improve product and services. To get proper feedback on developed system, it was given to Mr. Nabin K. Singh as part of feedback collection process. Feedback provided by him is documented here. This paper also evaluate feedback independently and provide necessary recommendation to address the feedback.

<table>
<tr><th colspan="2">Independent Feedback</th></tr>
<tr><td colspan="2">Product Title: Android Based Library Management System for ISMT College using JAVA</td></tr>
<tr><td>Feedback Given By:
Nabin k. Singh</td><td>Signature:
Date:</td></tr>
<tr><td colspan="2">Product Description (in brief): To develop android based library management system to increase facilities and condition of the library. Developed system would automated library system of ISMT college.</td></tr>
<tr><td>Important Features of the Product</td><td>While giving the feedback, following features were found important:
1. Authentication and authorization system
2. Separate Panel dash board/ panel for members and administrator
3. Admin has ability to manage books, students, reserved books
4. Members can reserve books
5. Reserved books are made unavailable during book search
6. Members can add book into wish list that are not available in library
7. Admins can view wish list
8. Fine system</td></tr>
</table>

Checklists	YES	NO	Comment
Have identified objects and data and file structures required to design the system?	Yes		Has identified classes, objects, files/tables, variables and methods
Have Front End and Back End of the system has been designed?	Yes		Has designed clear front end UI layout using XML code and back end database using SQLite
Have been critically tested and reviewed the system or not?	Yes		Has critically tested and reviewed. Final analysis based on various test results.
Does the system solve the problem of the client?	Yes		Partially yes
Does the system fulfill the requirements of the clients?	Yes		Partially yes
Have developed the user document help to assists the user of the mobile user?	Yes		User document is developed to assist users

Positive aspects of the Product	While giving feedback following positive points are noted: 1. The main significance of the product is it fulfill the minimum requirement set by organization. 2. It has user friendly GUI that is supported by tidily coded back end. 3. Each forms in application is well validated. 4. Bugs and errors are well tested and reviewed.
Product Limitation	While giving feedback for the developed system, following limitation are noted: 1. UI simple and quick but IT gets boring/dull after sometime 2. As application supports single mobile operation only, application cannot reach its full potential. 3. Application does not have cloud based data backup system 4. Application does not have forgot username/password system

Feedback Evaluation	
Feedback Evaluated By: ***Atut Gorkhali***	Signature: Date:

Evaluation of the System Feedback:

Library management system is implemented using properly analyzed designs such as flow chart, data flow diagram, use case diagram etc. Before implementation, system was thoroughly studied. Which allowed to understand client requirement and identify necessary designs to achieve those requirements. After designing solution using various tools, components and data and file structure was identified. All these procedures before implementation of system helped to satisfy client's requirement.

After development of the system, before deployment system was critically review and tested using various test methodologies such as unit test, stress test, compatibility test and integration test. All test result was documented as test logs. Proper analyze of testing and conducting necessary maintenance of the system resulted well validated and bug free system.

While evaluating the feedback, one of the key limitation of the system noted was absence of web support. Developed system uses local SQLite database. As there in so web support in current system, organization will be forced the use single mobile while managing library. Data in SQLite is not synced with hosted database. This means if for some reason data is lost from mobile database, there is no mechanism to retrieve lost data. Similarly, user cannot ask for new password if he/she forgets password as there is no forgot password feature. Generally, password retrieval feature uses email service and online database, hence implementation of such feature is very tough in current system.

Developed system is implemented without much emphasis on UI model. XML codes are used to build layout for the system. It is true that it may not have excellent UI layout as proper research before designing was not conducted and client was satisfied with prototype of the system UI. But if system was online application and had large number of potential clients, then UI should require much more focus and better design should be considered.

Recommendations for future consideration:

Evaluation of provided feedback allows to recommend following highlights:

Use Google's Material Design Guideline to design UI

Material design is Google's guideline for android users to create better User interface foundation. Material design includes guideline for handling motion, depths, lights etc. which allows to create beautiful UI layouts. With current system not following Material design, it is recommended to deploy UI lay for system using proper guideline.

Android App with PHP/MySQL support

As developed system is uses offline database and does not offer online database service, as feedback suggests, it has not achieved its full potential. With online database multiple user will able to use system accessing online database, unlike current situation where only one user can be use system at one time. To overcome all these issues, it is recommended to provide online database support for developed system. To achieve such support, one of the easiest method is to integrate PHP and MySQL with android application. This is extremely helpful in the event that you have a webserver, and you need to get to its information on your android application.

MYSQL is utilized as a database at the webserver and PHP is utilized to bring information from the database. Java application will speak with the PHP page with fundamental parameters and PHP will contact MYSQL database and will get the outcome and return the outcomes to our application. With this feature, multiple user will able to connect to system at same time and use same database. Additionally, this feature will open door for forgot username/password feature.