TITLE
LuPP SYSTEM

GROUP MEMBERS	REGISTRATION NUMBERS
Sheila Muyia	N11/3/1229/016
Nancy Muthoni Karanja	N11/3/1228/016
Maureen Wangari	N11/3/0514/016
Jedidah Waithera	N11/3/1165/015
Donatta Mutonyi	N11/3/1275/016

DECLARATION

This documentation is of a software project named LuPP system and it is our original work except where otherwise stated. It has not been presented for a degree in any other university or award.

..... 6/2/2019

Secretary Date

CERTIFICATION

The undersigned certify that he has read and recommended for acceptance of Laikipia Universiftware project named 'LuPP System.'

---------- ----------

Prof. Simon M. Karume Date

Department of Computer Science and IT

Laikipia University

COPYRIGHT

DEDICATION

The project is dedicated to the library of Laikipia University. This is for support and information required from the staff in the library. Thanks to each and every member of our group for working smart as well as hard to ensure that the project is complete. Our dedication also goes to friends who have supported and encouraged us during coding of the project to completion. We also thank Prof. Simon M. Karume for enabling us as the students understand how to build a working LuPP system.

ACKNOWLEDGEMENT

We would like to acknowledge our project instructor, Prof. Karume. He ensured that the team followed the guidelines for building working system. He also enabled us understand where the team went wrong and hence corrected our mistakes according to the instructions given to us. Our acknowledgement also goes to each member of the group for every effort that they have put in order to realize the goals set. They have sacrificed their own time to have LuPP App working. They have researched both online and offline so as to be able to get a lot of information concerning android studio ide and the tools required to come up with an android application. Library staff members have also given us relevant information concerning the current library services and management.

ABSTRACT

LuPP (Laikipia University Past Papers) system is an android application. It is supposed to make work and life easier. Accessing questions for any unit of the degree would be just a click away. This is in comparison with walking all the way from hostels to the library, queue to get past papers that one may get or wait for fellow student to be done with booklet. The system digitalizes services done in the library. A student can be able to register, log in, upload, download past papers of Laikipia University and share the app. The methods that were used to gather requirements for LuPP system include: Questionnaires, interviews and online materials. LuPP system has been developed using android studio IDE and tools such as Android SDK. This is to enable student of Laikipia University to access questions efficiently and with less time consumption. Knowledge is power hence the need to study smart using LUPP app.

Table of Contents

1 CHAPTER ONE: INTRODUCTION

1.1 Background information

The need to access a lot of information in a short time is what LuPP app is striving to uphold. With the current technology and the view of the world as a global village(McLuhan, 1962), information is shared a lot faster compared to the manual way of doing activities. By manual I mean, the need to be physically present in the library with a library card so as to access services that one requires. Not everyone has time on their side, students have a lot of work to do on a daily basis. In addition,

there are student who work (otherwise known as work study) and they are required to pass their examination. This is where LuPP app comes in, it caters for each and every student regardless of the work or commitment one maybe having. To ensure that the is security within the system, uploads will be conducted by the service provider at the library of Laikipia University, Main Campus. This is to ensure that what is posted is relevant and that it is up to date. On the other hand, students taking courses at Laikipia University will be able to download what they require during their revision as well as research.Downloads will be of examination papers that were done in Laikipia University hence the content will be what the students will be requiring as well as being of benefit to them. They will also be able to share to other platforms such as whatsup in order to distrbute the resourses to other students needing the information.Freedom is studying at anytime and anywhere.

Laikipia University Library Main Entrance

1.2 Problem Definition

The are a lot of challenges that students of Laikipia University, Main Campus face when they need to access past examination papers. These challenges include the following:

- ⇨ One can not be able to access library services without library card. This means that if a student loses his or her library card, he or she can not get past papers directly from the library.
- ⇨ It is clear that majority of students here at Laikipia University have a lot to research on, notes to study and discussion to hold. This is moreso before and during examination period. This means that less students have time to be physically in the library.
- ⇨ There are a lot of students admitted to Laikipia University but resources are limited. A student may want to revise for a certain unit but find out that the booklet with questions to be revised on has already been allocated to another student. This requires that the last student of the two to wait for one hour.

1.3 Description of The Current System

The current system (library) is manual. One is required to be physically present at the libary. The Laikipia University Universty has a lot of information to be explored from books, magazines, news papers and examination papers. In order to aceess the information one must present a library card to the staff at the Library. Only when your information is recorded manually are you allowed to take the the material to study. One can only use te material for a limited amount of time for an exam booklet is usually one hour, After the one hour has pased without returning the exam booklet you are fined.

1.3.1 How The Current Works

This is a step by step on how the current Laikipia University works.

1. A student physically arrive at library. One is required to follow some set rules for example not entering with bag, no wearing of hooded clothing among other rules.
2. The student chooses the reading material that he or she want in the library.
3. Present a library card to the service provider the Laikipia University Library.
4. Personal information about the student is manually recorded.
5. One is handed the required reading material that should be returned within the specified period of time otherwise fined.

1.3.2 Weakness of The Current System

The current weaknesses of the manual services at the library include the following:

- ⇨ The need to have a library card for one to get past examintion papers.
- ⇨ Time wastage during travel and waiting for services in the library.
- ⇨ Limited resources compared to the number of students using the library.

1.4 Proposed Solution

In order to curb the weaknesses and challenges, it is essential to digitalize the manual system currently being followed. This requires coming up with a digital system. LuPP App runs on majority of android devices. In addition, there is always WIFI at Laikipia University hence students will be able to access examinations at anywhere around school or at home because Internet Service Providers (ISP) offer affordable services. This will in turn eliminate the need to be physically present at the library, save time as well as resources being distributed and used by everyone.

1.4.1 Justification

LuPP App will be able to deliver all what is needed by students because:

1. Majority of student have android devices so as to enable them study and research efficiently.LUPP App which is an android based app will be of assistance to them.
2. There is availabity of WIFI so anyone in need of revising examination materials will acess by efficiently downloading the content required.They can as well share teir content with fellow students easily.

1.4.2 Project Objectives

GENERAL OBJECTIVES

=>Reducing time wastage. Getting examinations papers will be one click away compared to travelling all the way from hostel to the laibrary.

=>Digitalizing services in order to make it possible for everyone to access the same resources at the same time.

=>Reducing the cost of travel for those students who live far from the school.

SPECIFIC OBJECTIVES

=>Building system that is compatible with most mobile devices hence majority of student will access past papers

=> The librabry pastpaper service provider will be able to register, log in and upload pastpapers.

=>Students will be able to register, log in, download and share Laikipia University pastpapers.

=>Optimizing the use of Laikipia University resources, that is WIFI in order to do wide research.

1.5 Project Schedule

Task number	Task description	Starting date	Ending date	% of complition
1	Project proposal and documentation of chapter 1	1/2/2019	15/2/2019	Done
2	Documention of literature review and methodology	8/2/2019	22/2/2019	Done
3	Document analysis and design	18/2/2019	1/3/2019	Done
4	System implentation and testing, documention of appendices and system code	1/3/2019	2/4/2019	Done
5	Printing and binding	2/4/2019	9/4/2019	
6	Presentation	9/4/2019	2/4/2019	

1.6 Project Budget

EXPENSES	COST IN KSH
Online research	3,000
Printing services and Binding	4,000
Miscellaneous	1,500
Transport	3,000
Domain name and hosting	2,500
TOTAL	14,000

2 CHAPTER TWO: LITERATURE REVIEW

2.1 INTRODUCTION

Chapter 2 of this documentation present the review of systems that are similar to ours. Collection information from reliable sources is also relevant in this case. As a group, we have conducted research on how students get to access revision papers. Here are our findings:

2.1.1 CASE STUDY 1

In Laikipia University there is no specific application that has been created that one can be able to upload, download and share past papers. Instead interested students usually go to the library to revise. In this case they stay for long hours looking for answers and doing their research. Those who have smart phones take pictures of these papers in order for them to do further studies in their hostels. This is when their time is limited or when they want to extend their research in the comfort of their rooms.

2.1.2 CASE STUDY 2

Another way students only get to access Laikipia University past papers is by using social media such as WhatsApp. For those people who do not have time at all to be in the library download papers posted on whatsApp. This is usually around examination period when students post various past papers in their own whatsApp groups.

2.2 RESEARCH GAP

There exist gaps in this current system. The table below summarizes the areas that are addressed in the current systems and the gap that exist (areas that are not addressed)

EXISITING SYSTEM	AREAS ADDRESSED	AREAS THAT ARE NOT ADDRESSED
Laikipia University Library	It has addressed the issue of security. Booklets containing past papers are safe as the personal information of the user is noted. Time is also noted down to ensure that the material is returned where it belongs.	The issue of time management is not well addressed. If a student goes to library and find the recourses that he or she may be needing is being used then he or she has to wait for one hour. Another area where time is wasted is when one is required to travel to the library to get past papers. Cost minimization is not addressed. To be physically present in the library requires one to travel hence spending money. Some students live in far places requiring them to spend a lot of money locomoting to and from school.
WhatsApp	When using this platform, fast accessibility of revision papers is guaranteed. One can easily upload, download and share revision papers.	Issues regarding security are not addressed. A lot of irrelevant materials are posted on WhatsApp. A user might find out that he or she has downloaded what was not required.

	The application is compatible with most of devices that students have in Laikipia University. WhatsApp is an android based application hence compatible with most smart phones used during research and studies,	

2.3 BASIC ISSUES THAT LUPP APP WILL ADDRESS

TIME MANNAGEMENT

There is need of putting past papers done at Laikipia University, Main Campus on an app. It is convenient for every student in studying here to get past paper anytime and anywhere. One can be able to download the revision papers directly from their smart phones instead of walking all the way to the Library to get them. They will also be able to share the app among themselves.

COST EFFECTIVENESS

LuPP app will be cost effective. With accessibility of WiFi here at Laikipia University, cost is subsidized. The papers can be downloaded at anytime hence the hustle and transport cost will be taken care of as in this case as one studies, he or she can get the papers and not wait for last minute and incur transport cost.

INTEGRITY AND SECURITY OF LUPP APP

The security of LuPP app is guaranteed due to the fact that only the service provider at Library will be allowed to upload the examination papers ("WordPress Security - File Upload Vulnerabilities", 2018). This will ensure that only relevant materials are in the application. Students who might be having extra Laikipia University papers will be free to hand them over to staff members at the library in order for them to be uploaded.

3 CHAPTER THREE: METHODOLOGY

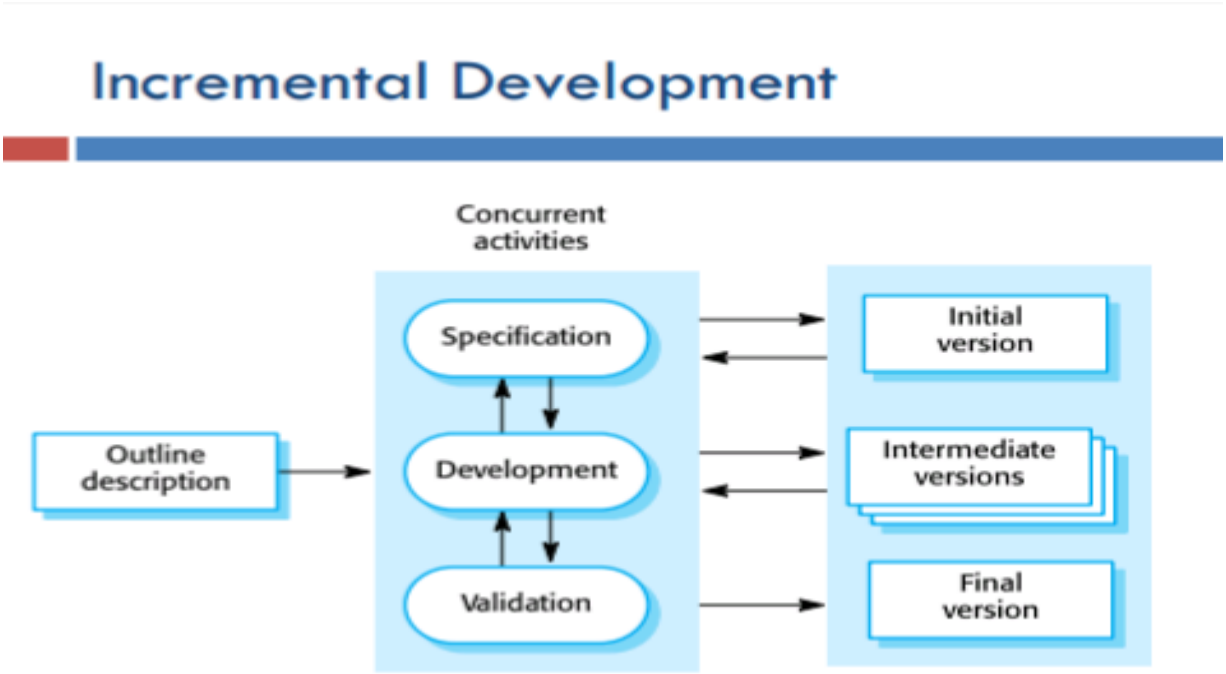
3.1 INTRODUCTION

This chapter of this documentation has addressed the software process model that has been used during development of LuPP App.

3.2 SOFTWARE PROCESS MODELS ADOPTED

LuPP application is based on incremental development which maybe plan-driven or agile process. In our case, the team chose agile process as it is easier to change the process to reflect changing customer requirements.

Incremental development model



3.2.1 STRENGTH OF MODEL ADOPTED

The cost of accommodating changing customer requirements is reduced. This is because the amount of analysis and documentation that has to be redone is less.

It is usually easier to get customers' feedback on the development work that has already been done.

When using incremental development, it is possible to rapidly deliver and deploy a useful system to the customers.

3.2.2 WEAKNESSES OF MODEL ADOPTED

With incremental development, it is difficult to clearly state where one is in software development process.

The structure of the system may degrade as new increments are added. This can be solved by investing in time and money to improve the system.

3.3 REQUIREMENTS GATHERING TOOLS

1. Observation of services being offered at Laikipia University library.
2. Interview with some of student in Laikipia University. We find out that some do not have
3. adequate time to be physically in the library in order to access services.

3.4 SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

A minimum of 1GB RAM memory.

A minimum of 500MB hard disk space.

Android version smart phone

The above requirements will be useful in coding and running of programs on computer.

SOFTWARE REQUIREMENTS

Android Operating System on smart phone

Android Studio IDE

Android Studio SDK

Virtual machine installed on Android Studio IDE

Gradle

The above requirements will be required for proper and complete functioning of LuPP App.

4 CHAPTER FOUR:SYSTEM ANALYSIS AND DESIGN

4.1 INTRODUCTION

This part of documentation presents the analysis of the system and design that was used in the LUPP APP project.

4.2 ARCHTECTURAL DESIGN

Architectural design is usually concerned with the understanding of how a system should be organized as well as designing the overall structure of that system.

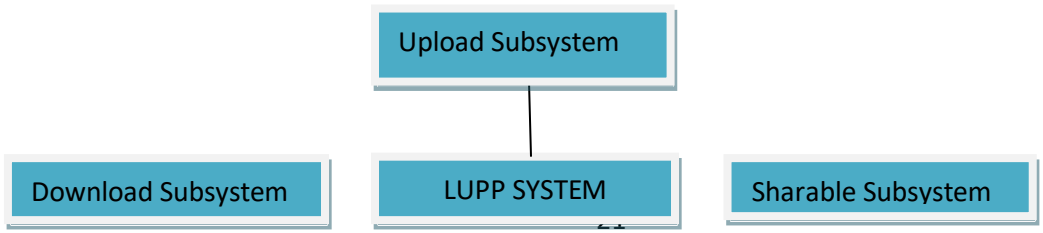
The architectural design that was used in the making of the application is client-server architecture.

This is because the librarian is required to upload examination papers to the server in order for it to be downloaded by students via a network. Network allows clients to access servers.

4.3 System Analysis

4.3.1 Context diagram

It usually shows inputs and outputs from the system. Below is a diagram showing the components of LuPP Application.

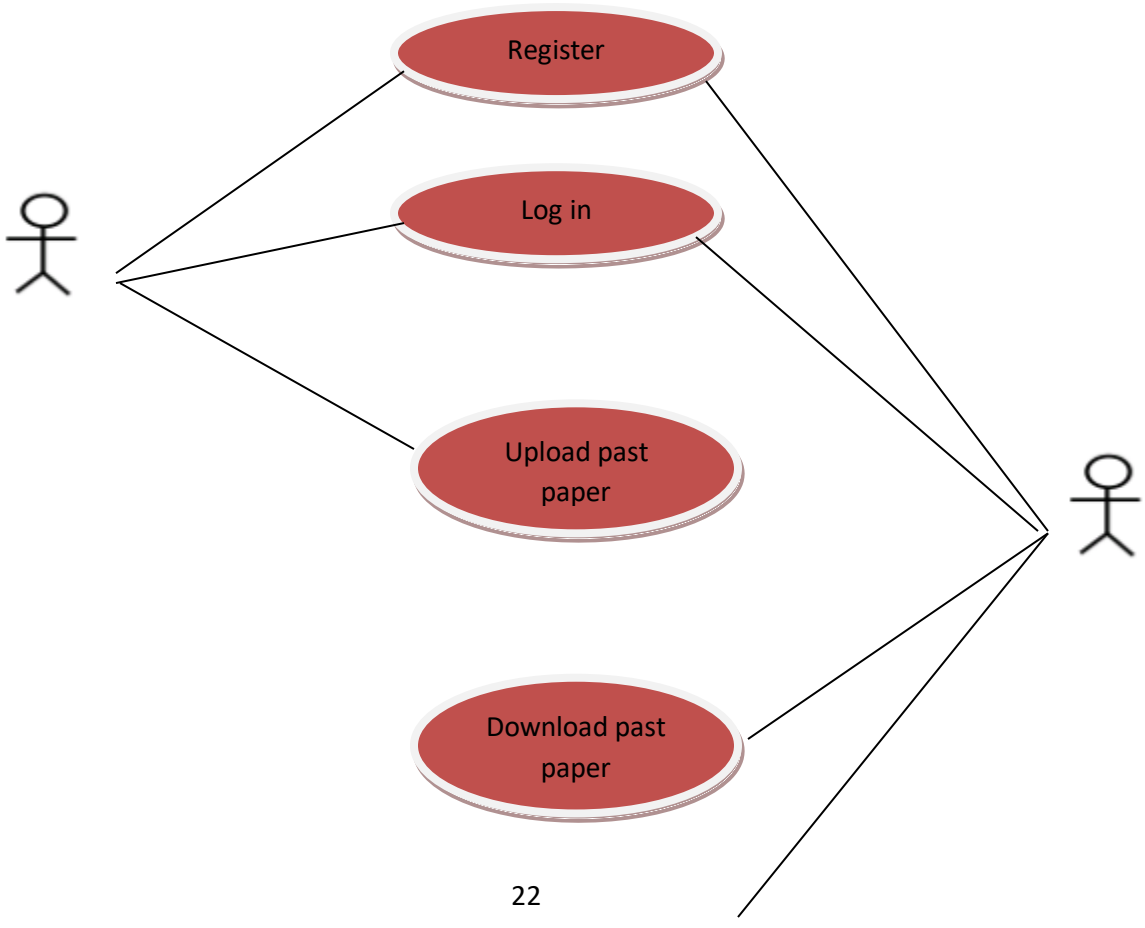


4.3.2 Domain analysis

Domain analysis identifies objects otherwise known as entities and their relationships. LuPP system has two entities; librarian and student. Laikipia University librarian is supposed to upload the softcopy of past examination papers and a student downloads or share the study aide material to other platforms.

4.3.3 Use case model

A use case is a simple scenario that describes what a user expects from the system. It identifies the actors involved in an interaction and names the type of interaction.

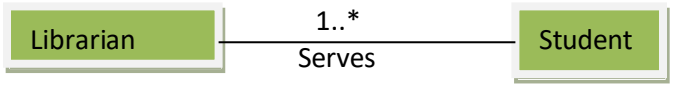


4.4

4.4.1 Class diagrams or entity relationship diagrams

It is a model that displays the organization of a system in terms of the components that make up that system and their relationships.

Below is structural model of LuPP system.



Librarian
Name Password
Upload()

Student
Name User name Reg No Password
Download() Share()

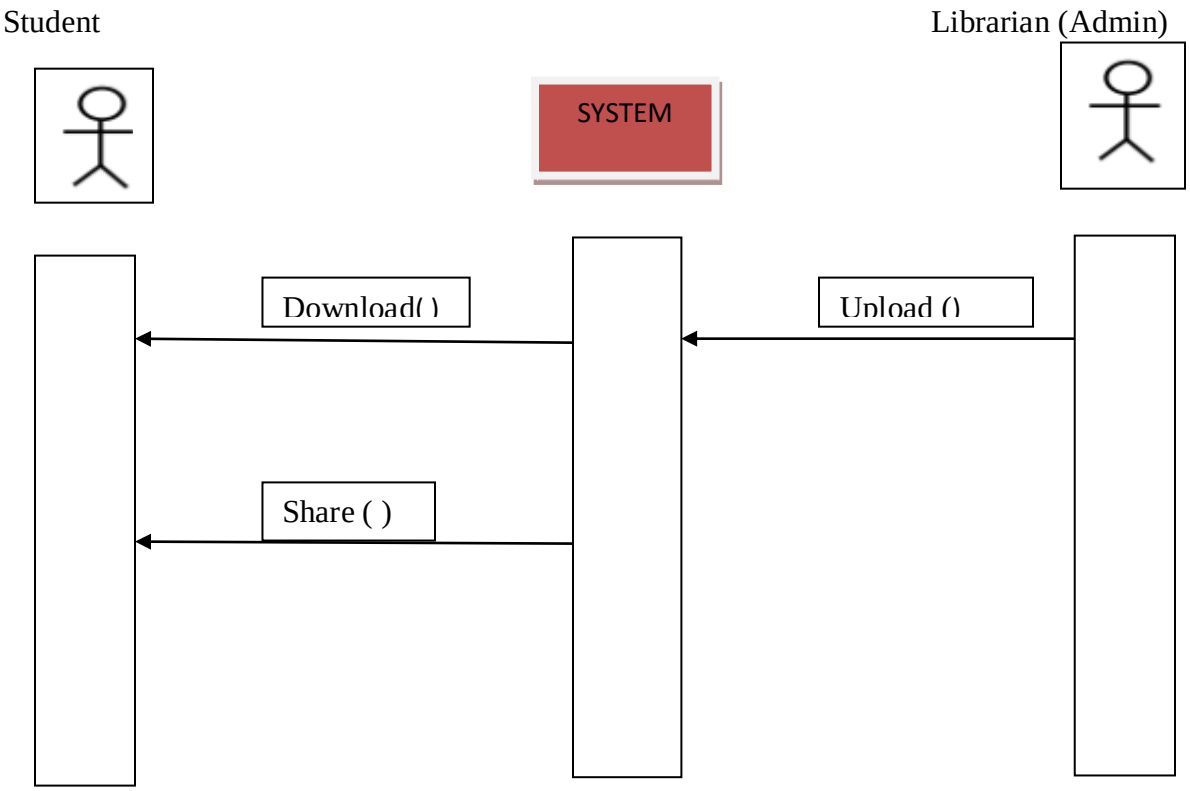
4.4.2 Sequence diagrams

Sequence diagrams are used to model the interactions between actors and objects within a system. Below is the sequence diagram of LUPP system.

Description

1. Student access LuPP application.
2. Student access LU past papers uploaded by the librarian (upload())
3. Student downloads LU past papers (download())

4. Student shares LuPP app (share())



4.5 Database design

librarian

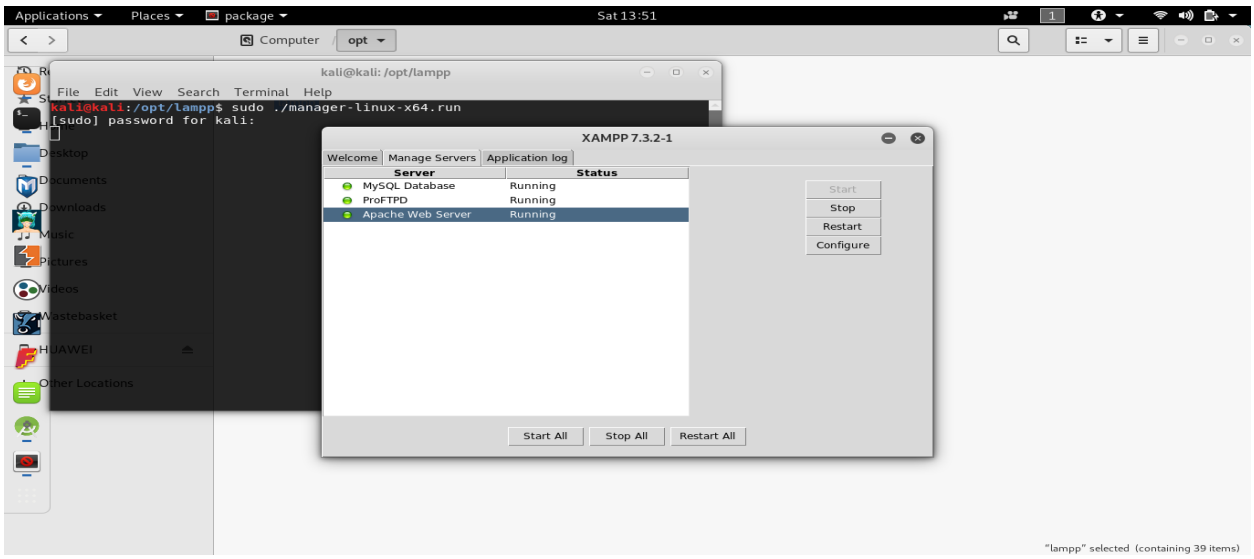
Username(Librarian)	Varchar	Unique
Email	Varchar	Unique

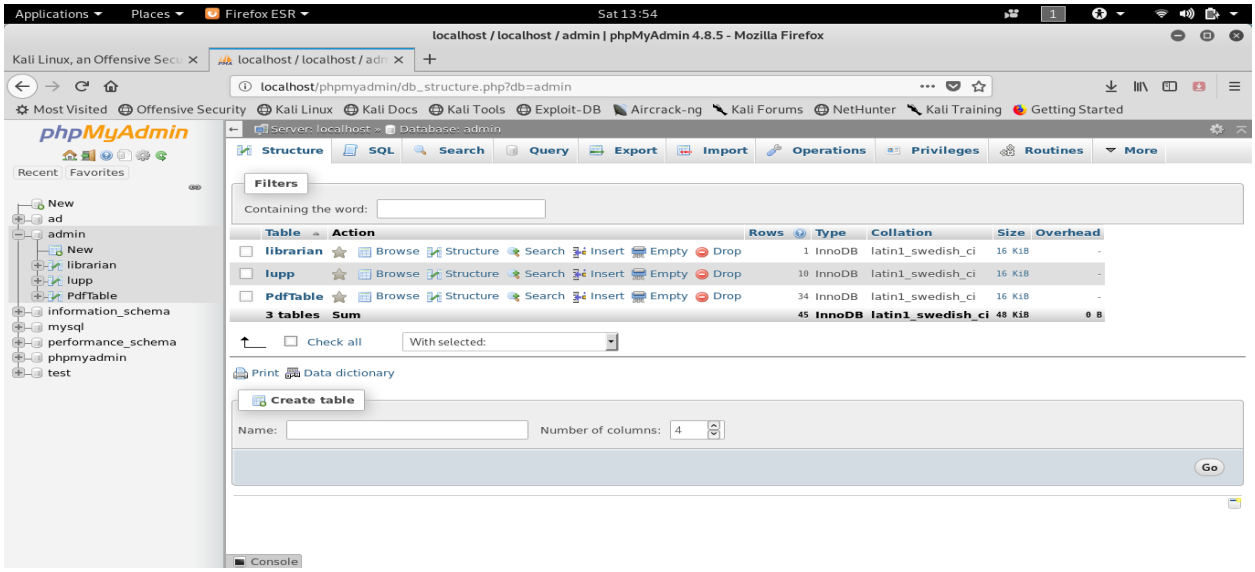
lupp

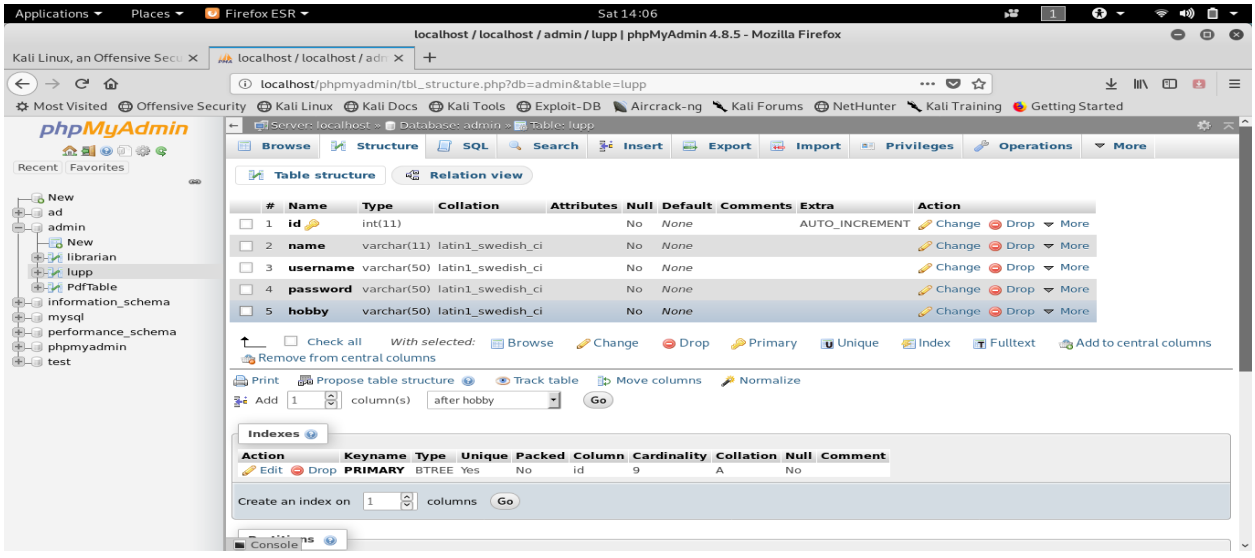
Id	Int	Primary key
name (student)	Varchar	Unique
Username	Varchar	Unique
RegNo	Varchar	Unique
Password	Varchar	Unique

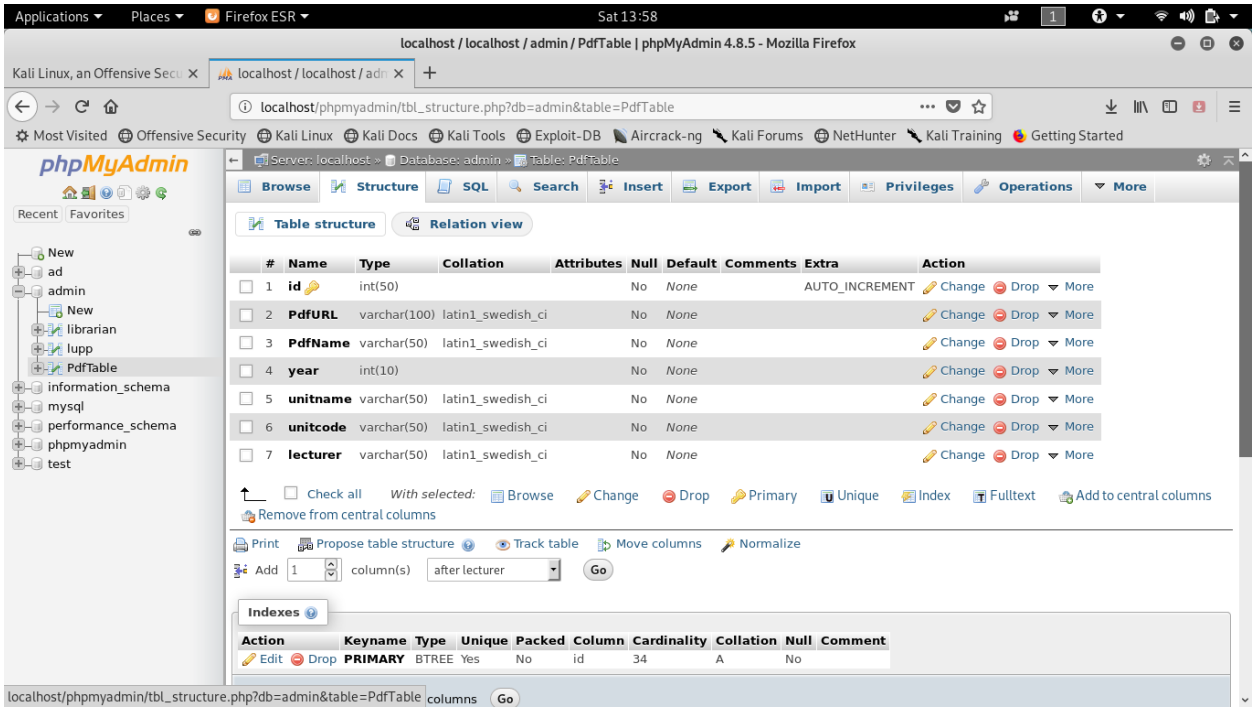
PdfTable

Id	Int	Primary key
PdfURL	Varchar	Unique
PdfName	Varchar	Unique
Year	Int	Unique
Unitname	Varchar	Unique
Unitcode	Varchar	Unique
Lecturer	Varchar	Unique




Applications
Places
Firefox ESR
Sat 13:54
localhost / localhost / admin | phpMyAdmin 4.8.5 - Mozilla Firefox
localhost/phpmyadmin/db_structure.php?db=admin
phpMyAdmin
Server: localhost » Database: admin
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Filters
Containing the word:
Table
Action
Rows
Type
Collation
Size
Overhead
librarian
lupp
PdfTable
3 tables Sum
Check all
With selected:
Print
Data dictionary
Create table
Name:
Number of columns:
Go
Console


Applications
Places
Firefox ESR
Sat 14:06
localhost / localhost / admin / lupp | phpMyAdmin 4.8.5 - Mozilla Firefox
localhost/phpmyadmin/tbl_structure.php?db=admin&table=lupp
phpMyAdmin
Server: localhost » Database: admin » Table: lupp
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
More
Table structure
Relation view
#
Name
Type
Collation
Attributes
Null
Default
Comments
Extra
Action
id
name
username
password
hobby
Change
Drop
More
Check all
With selected:
Browse
Change
Drop
Primary
Unique
Index
Fulltext
Add to central columns
Remove from central columns
Print
Propose table structure
Track table
Move columns
Normalize
Add
column(s)
after hobby
Go
Indexes
Action
Keyname
Type
Unique
Packed
Column
Cardinality
Collation
Null
Comment
Edit
Drop
PRIMARY
BTREE
Yes
No
id
9
A
No
Create an index on
columns
Console



THEIR RELATIONSHIP

	LIBRARIAN	STUDENT
LIBRARIAN	1:1	1:M
STUDENT	M:1	M:M

5 SYSTEM IMPLEMENTATION AND TESTING

5.1 INTRODUCTION

System implementation and testing is a phase which is necessary for the release of a system to be used.

5.2 SUMMARY OF THE MODULES

The modules of LuPP system include:

1. Librarian(Admin) module

   This module enable librarian to:

   =>Login with specific name and password.

   =>Upload past papers in form of pdf which are stored in database.

2. Student module
   This module allows students to:
   =>Register.
   =>Log in
   =>Download past papers.
3. Database
   This module:

   =>Stores student personal data entered during registration.

   =>Stores past papers in pdf format.

5.3 SUMMARY OF HOW THE SYSTEM WORKS

This is how LUPP system works;

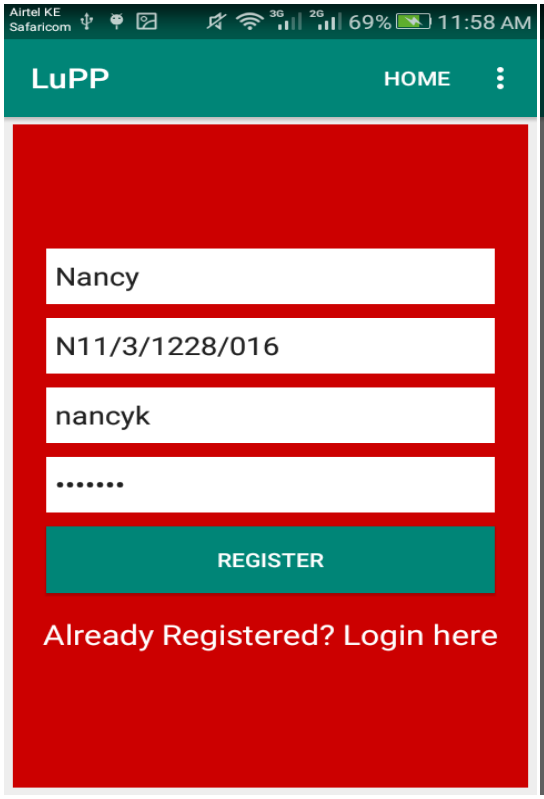
The admin (that is the librarian) registers and logs in the system. He or she then uploads examination papers in the form of pdfs.

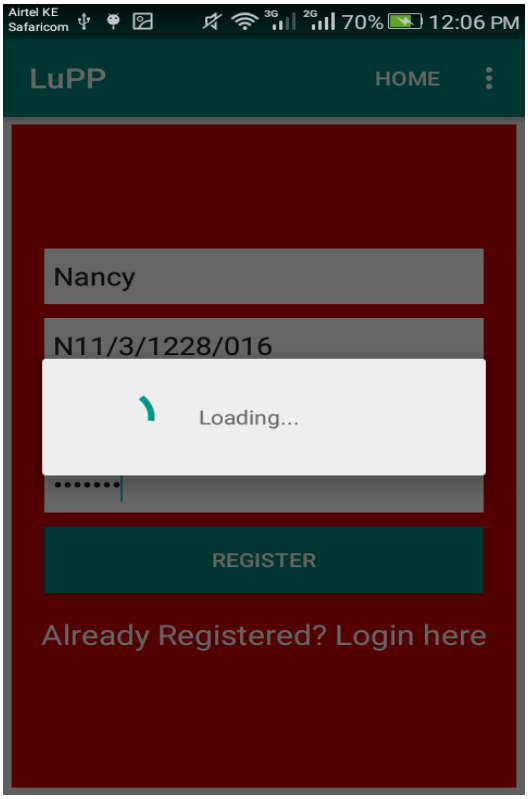
Students of Laikipia University, main campus also register and are allowed to log in by the system. They can download the examination papers as well as sharing the material they have downloaded.

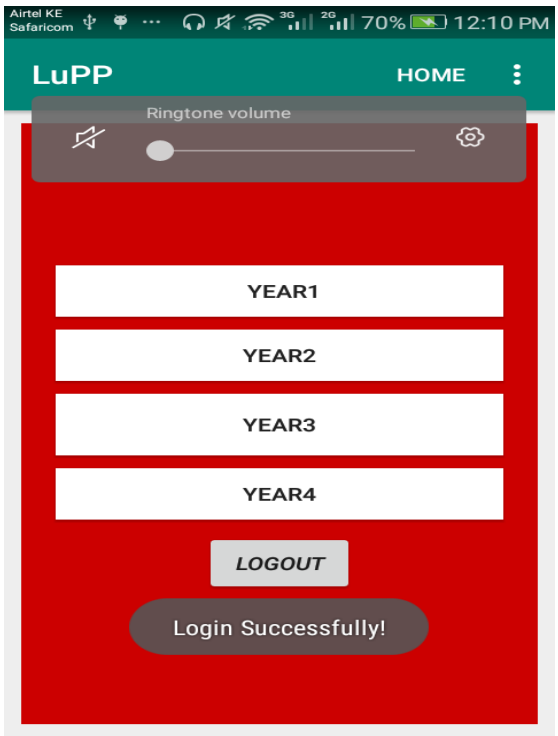
Email, password and the past papers are stored in the database. In this case information is not lost.

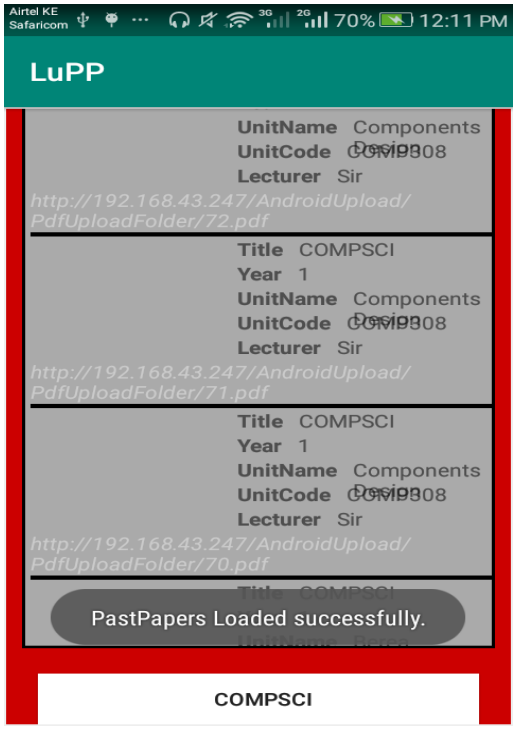
5.4 SCREENSHOTS

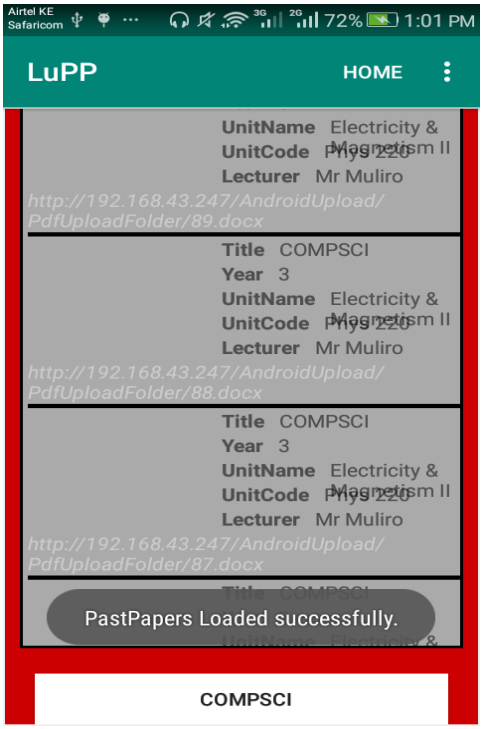
5.4.1 COMPUTER SCIENCE/BICT

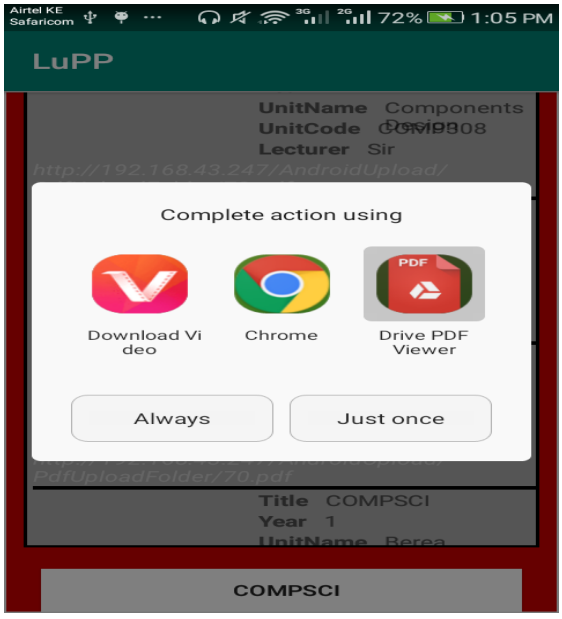




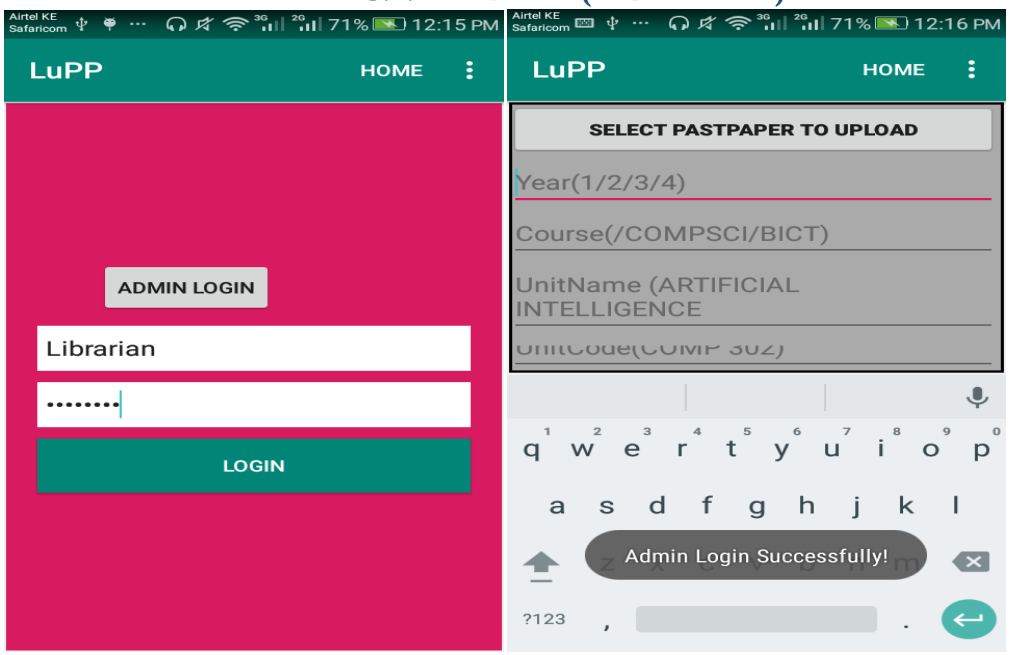


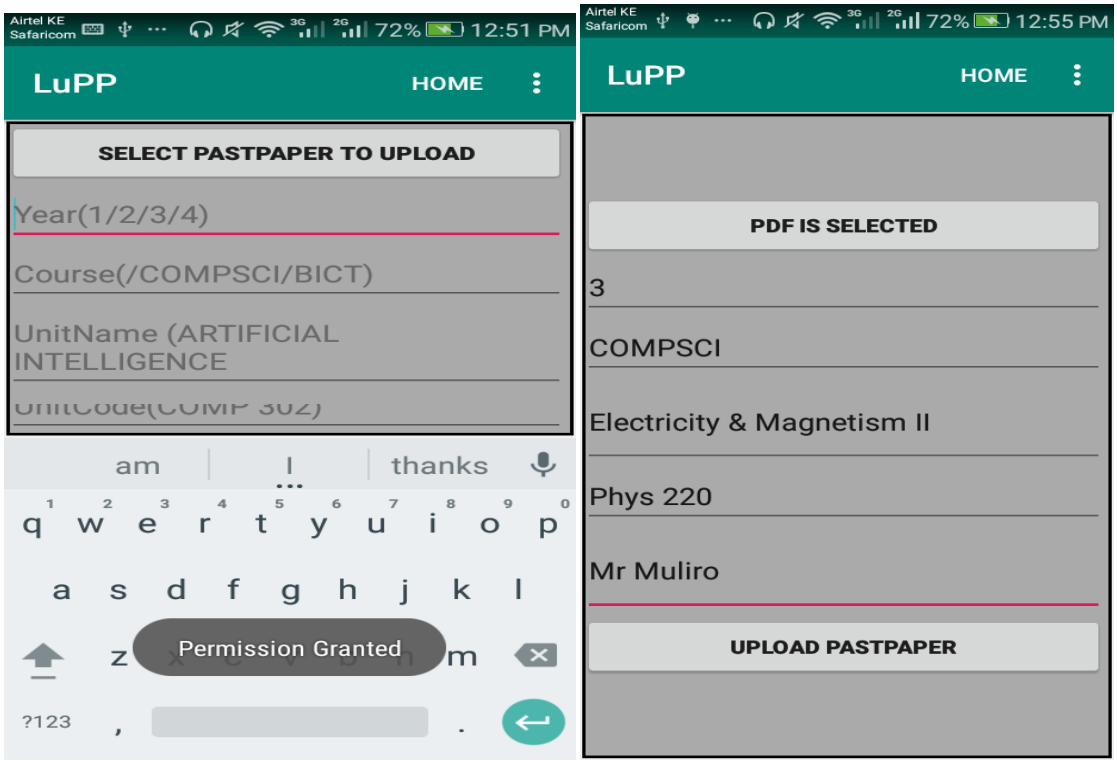

Airtel KE
Safaricom
70%
12:11 PM
LuPP
UnitName Components Design
UnitCode COMP308
Lecturer Sir
http://192.168.43.247/AndroidUpload/PdfUploadFolder/72.pdf
Title COMPSCI
Year 1
UnitName Components Design
UnitCode COMP308
Lecturer Sir
http://192.168.43.247/AndroidUpload/PdfUploadFolder/71.pdf
Title COMPSCI
Year 1
UnitName Components Design
UnitCode COMP308
Lecturer Sir
http://192.168.43.247/AndroidUpload/PdfUploadFolder/70.pdf
PastPapers Loaded successfully.
COMPSCI

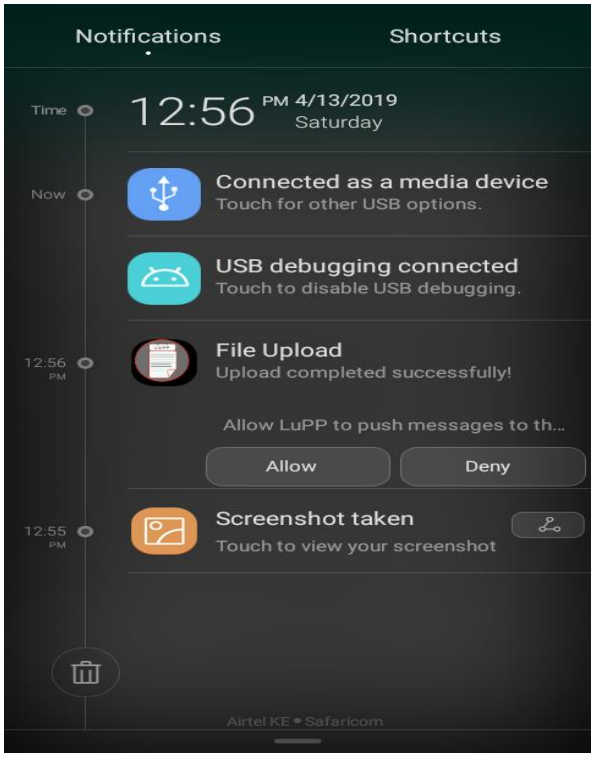

Airtel KE
Safaricom
72%
1:01 PM
LuPP
HOME
UnitName Electricity & Magnetism II
UnitCode Phys 220
Lecturer Mr Muliro
http://192.168.43.247/AndroidUpload/PdfUploadFolder/89.docx
Title COMPSCI
Year 3
UnitName Electricity & Magnetism II
UnitCode Phys 220
Lecturer Mr Muliro
http://192.168.43.247/AndroidUpload/PdfUploadFolder/88.docx
Title COMPSCI
Year 3
UnitName Electricity & Magnetism II
UnitCode Phys 220
Lecturer Mr Muliro
http://192.168.43.247/AndroidUpload/PdfUploadFolder/87.docx
PastPapers Loaded successfully.
COMPSCI

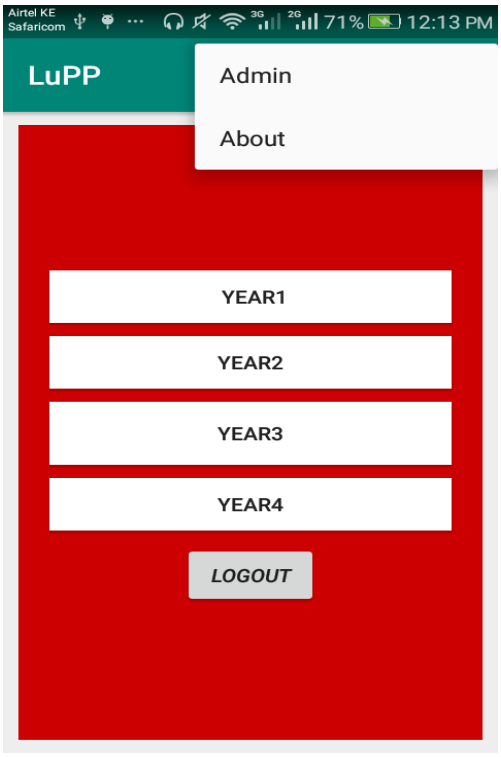

Airtel KE
Safaricom
72%
1:05 PM
LuPP
UnitName Components Design
UnitCode COMP308
Lecturer Sir
Complete action using
Download Video
Chrome
Drive PDF Viewer
Always
Just once
PdfUploadFolder/70.pdf
Title COMPSCI
Year 1
COMPSCI

5.4.2 ADMIN (LIBRARIAN)



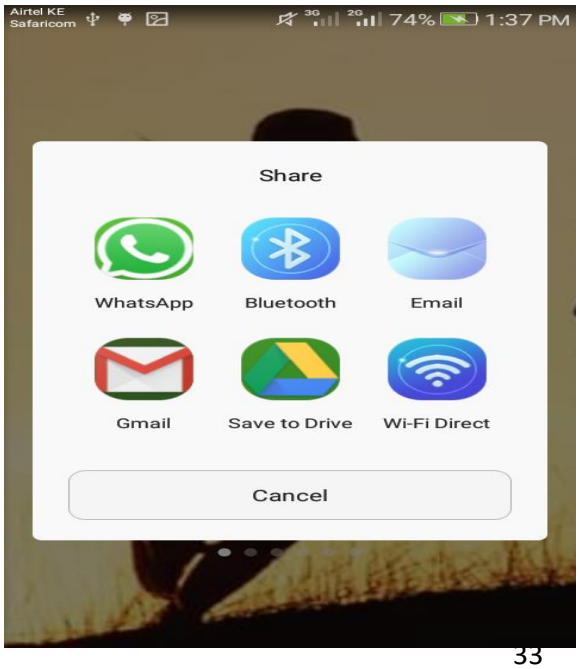


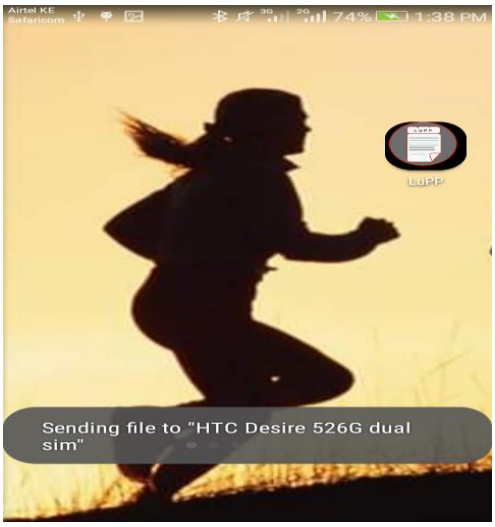




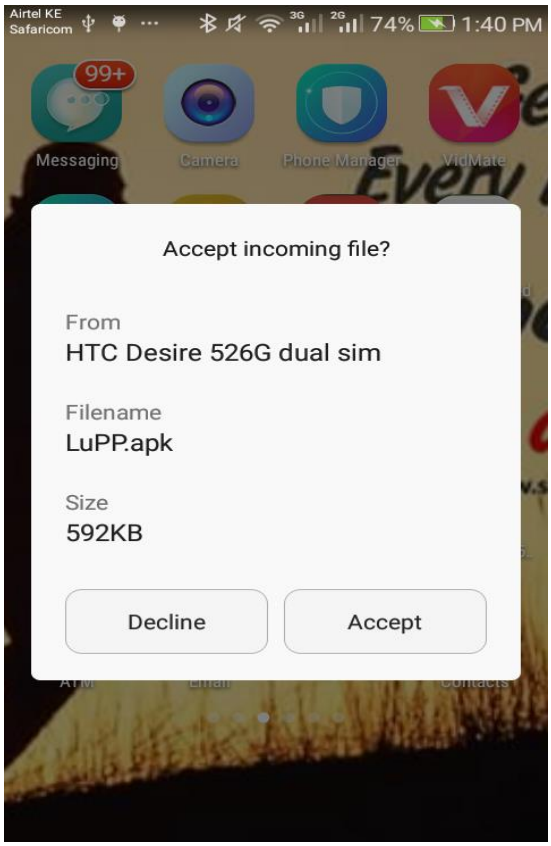
5.4.3 SHARING LuPP APPLICATION

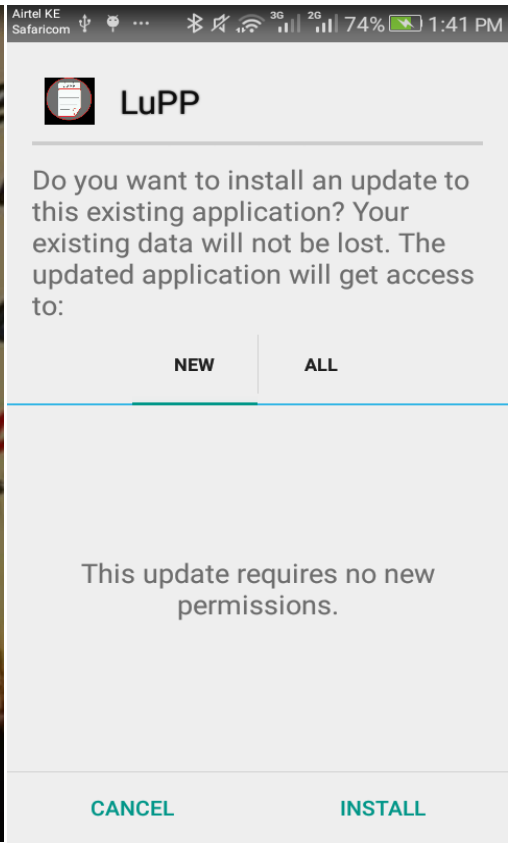
SENDING LuPP APK FILE

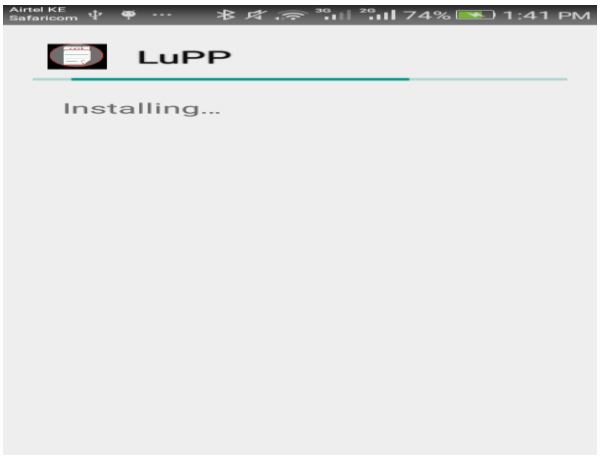




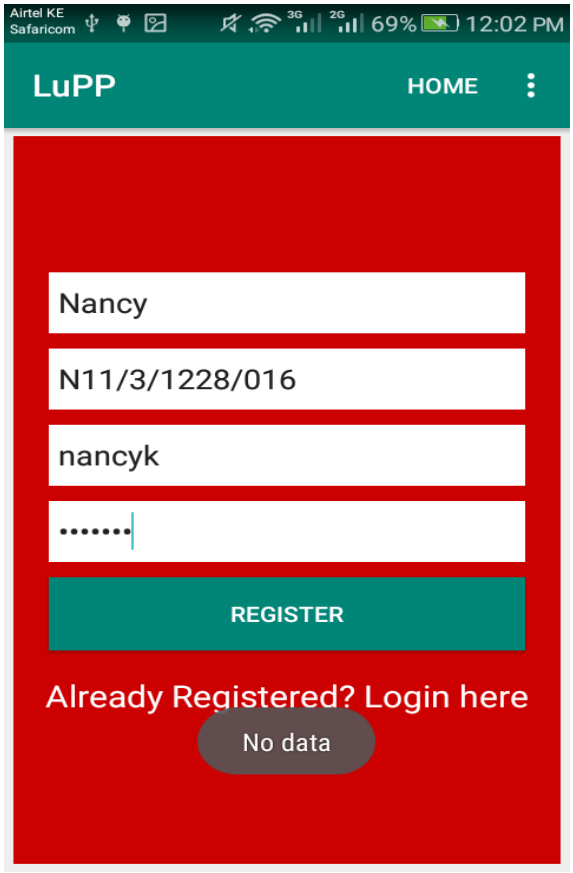
RECEIVING LuPP APK FILE

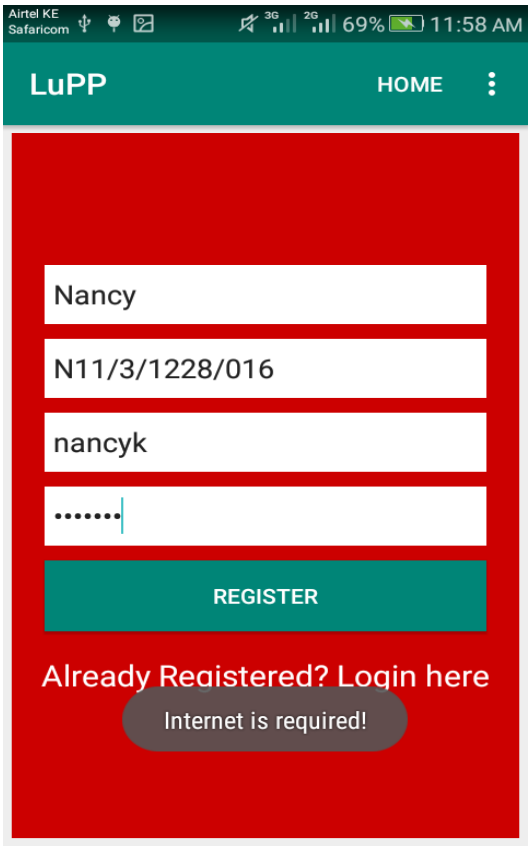


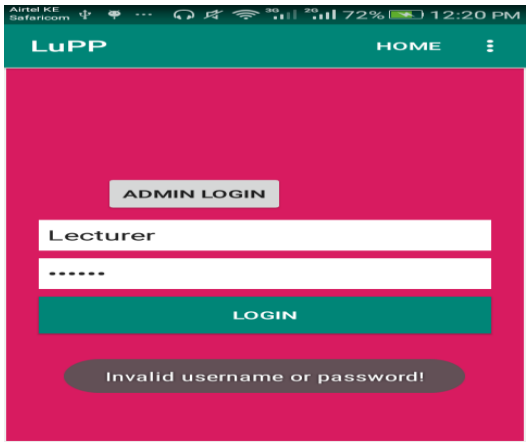




APP WHAT HAPPENS WHEN THERE IS AN ERROR WHILE ACCESSING LuPP







5.5 TEST REGIME

id	Description	Expected results	Actual results
1	Student registration	Registration successful. Information stored in lupp table in admin database.	Registration successful
2	Student Log in	Allows student into the system	Login successful
3	Librarian login	Allows librarian with a specific name and password	Permission granted
4	Uploading	Librarian upload past paper in pdf format	Upload completed successfully
5	Download	Student download past papers stored in database	Complete action with

5.6 CONCLUSION

LuPP application is there to ensure that students at Laikipia University study at any place; at school or at home. It eases the accessibility of past papers to all students by just clicking the download button. Library services are digitalized of efficiency study.

5.7 RECOMMENDATION

We urge students of Laikipia University taking Bachelor of Science and BICT to use this system. It will reduce the effort required to study. They will save their own time and concentrate on their revision as they will not be required to arrive at the library. It is convenient for these student to use it.

REFERENCES

McLuhan, M. (1962). Do we live in a global village?. Retrieved from https://www.library.illinois.edu/village/globalnews/mod1/pg1.htm

WordPress Security - File Upload Vulnerabilities. (2018). Retrieved from https://www.wordfence.com/learn/how-to-prevent-file-upload-vulnerabilities/

APPENDICES

APPENDIX 1: QUESTIONNAIRE

Dear respondent,

We would like you to answer the questions below. It will enable us build a system that is convinient for you and user friendly. Updates will also be released based on the the feedbacks.

Please or provide your thoughts where necessary.

COMPUTER SCIENCE / BICT STUDENT

PERSONAL DATA

1. Are you a student of Laikipia University?
   Yes ☐
   No ☐
2. What is your year of study?
   [1] [2] [3] [4]

3. Which course are you taking?

Computer Science ☐

BICT ☐

TECHNOLOGICAL DETAILS

4. Do you have an android phone?

   Yes ☐

   No ☐

5. Do you have an access to the internet?

Yes ☐

No ☐

If yes, how often?

Every day ☐

Three times a week ☐

Twice a week ☐

Once a week ☐

SYSTEM DETAILS

6. Rate how LuPP application is easy for you to use? (tick where applicable)

Very easy	Easy	Average	Hard	Very hard

7. Do you access all the materials from this application?

Yes ☐

No ☐

8. How relevant are the materials you access?

Not relevant at all	Relevant	Very relevant

9. Are there challenges you have experienced with the application?

Yes ☐

☐

No

10. Would you suggest any changes and updates required?

.....
.....
.....
.....
.....
.....

APPENDIX 2:SYSTEM CODES

adminlogin.php

```
<?php


  if($_SERVER['REQUEST_METHOD']=='POST'){

 // echo $_SERVER["DOCUMENT_ROOT"];  // /home1/demonuts/public_html

//including the database connection file

     include_once("config.php");


      $username = $_POST['username'];

          $password = $_POST['password'];


           if( $username == '' || $password == '' ){

                echo json_encode(array( "status" => "false","message" => "Parameter missing!") );

           }else{

                    $query= "SELECT * FROM librarian WHERE name='$username' AND
password='$password'";

                $result= mysqli_query($con, $query);


                if(mysqli_num_rows($result) > 0){

                 $query= "SELECT * FROM librarian WHERE name='$username' AND
password='$password'";

                        $result= mysqli_query($con, $query);

                     $emparray = array();

                    if(mysqli_num_rows($result) > 0){
```

```
                    while ($row = mysqli_fetch_assoc($result)) {
                     $emparray[] = $row;
                    }
                    }
                echo json_encode(array( "status" => "true","message" => "Admin Login successfully!", "data" => $emparray) );
            }else{
                echo json_encode(array( "status" => "false","message" => "Invalid username or password!") );
            }
             mysqli_close($con);
        }
       } else{
                        echo json_encode(array( "status" => "false","message" => "Error occured, please try again!") );
       }
?>
```

config.php

```
<?php
$host="localhost";
$user="root";
$password="";
$db = "admin";
```

```
$con = mysqli_connect($host,$user,$password,$db);

// Check connection
if (mysqli_connect_errno())
 {
 echo "Failed to connect to MySQL: " . mysqli_connect_error();
 }else{  //echo "Connect";



  }

?>
```

simplelogin.php

```
<?php

  if($_SERVER['REQUEST_METHOD']=='POST'){
 // echo $_SERVER["DOCUMENT_ROOT"];  // /home1/demonuts/public_html
//including the database connection file
      include_once("config.php");

       $username = $_POST['username'];
```

```
$password = $_POST['password'];


if( $username == '' || $password == '' ){

    echo json_encode(array( "status" => "false","message" => "Parameter missing!") );

}else{

        $query= "SELECT * FROM lupp WHERE username='$username' AND
password='$password'";

    $result= mysqli_query($con, $query);


    if(mysqli_num_rows($result) > 0){

     $query= "SELECT * FROM lupp WHERE username='$username' AND
password='$password'";

            $result= mysqli_query($con, $query);

             $emparray = array();

            if(mysqli_num_rows($result) > 0){

            while ($row = mysqli_fetch_assoc($result)) {

             $emparray[] = $row;

            }

            }

      echo json_encode(array( "status" => "true","message" => "Login successfully!", "data" =>
$emparray) );

    }else{

        echo json_encode(array( "status" => "false","message" => "Invalid username or
password!") );

    }

     mysqli_close($con);

}
```

```
        } else{
                        echo json_encode(array( "status" => "false","message" => "Error occured,
please try again!") );
        }
?>
```

simpleregister.php

```
<?php

  if($_SERVER['REQUEST_METHOD']=='POST'){
 // echo $_SERVER["DOCUMENT_ROOT"];  // /home1/demonuts/public_html
//including the database connection file
    include_once("config.php");

     $name = $_POST['name'];
        $username = $_POST['username'];
        $password = $_POST['password'];
        $hobby= $_POST['hobby'];

         if($name == '' || $username == '' || $password == '' || $hobby == ''){
            echo json_encode(array( "status" => "false","message" => "Parameter missing!") );
         }else{
```

```
$query= "SELECT * FROM lupp WHERE username='$username'";

$result= mysqli_query($con, $query);


if(mysqli_num_rows($result) > 0){

  echo json_encode(array( "status" => "false","message" => "Username already exist!") );

}else{

        $query = "INSERT INTO lupp (name,hobby,username,password) VALUES
('$name','$hobby','$username','$password')";

        if(mysqli_query($con,$query)){


          $query= "SELECT * FROM lupp WHERE username='$username'";

    $result= mysqli_query($con, $query);

     $emparray = array();

    if(mysqli_num_rows($result) > 0){

    while ($row = mysqli_fetch_assoc($result)) {

     $emparray[] = $row;

    }

    }

          echo json_encode(array( "status" => "true","message" => "Successfully
registered!" , "data" => $emparray) );

         }else{

              echo json_encode(array( "status" => "false","message" => "Error
occured, please try again!") );

        }

  }

      mysqli_close($con);

}
```

```
    } else{
                                echo json_encode(array( "status" => "false","message" => "Error occured,
please try again!") );
            }

 ?>
```

dbDetails.php

```
<?php
define('DB_HOST','localhost');
define('DB_USERNAME','root');
define('DB_PASSWORD','');
define('DB_NAME','admin');
?>
```

file_upload.php

```
<?php

ServerConfig();

$PdfUploadFolder = 'PdfUploadFolder/';
```

```
$ServerURL = 'http://192.168.43.247/AndroidUpload/'.$PdfUploadFolder;

if($_SERVER['REQUEST_METHOD']=='POST'){

 if(isset($_POST['name']) and isset($_FILES['pdf']['name'])){

 $con = mysqli_connect(HostName,HostUser,HostPass,DatabaseName);

 $PdfName = $_POST['name'];
 $PdfYear = $_POST['year'];
 $PdfUnitname = $_POST['unitname'];
 $PdfUnitcode = $_POST['unitcode'];
 $Pdflecturer = $_POST['lecturer'];


 $PdfInfo = pathinfo($_FILES['pdf']['name']);

 $PdfFileExtension = $PdfInfo['extension'];

 $PdfFileURL = $ServerURL . GenerateFileNameUsingID() . '.' . $PdfFileExtension;

 $PdfFileFinalPath = $PdfUploadFolder . GenerateFileNameUsingID() . '.'. $PdfFileExtension;

 try{
```

```
 move_uploaded_file($_FILES['pdf']['tmp_name'],$PdfFileFinalPath);

 $InsertTableSQLQuery = "INSERT INTO PdfTable (PdfURL, PdfName, year, unitname, unitcode, lecturer)
VALUES ('$PdfFileURL', '$PdfName','$PdfYear','$PdfUnitname','$PdfUnitcode','$Pdflecturer') ;";

 mysqli_query($con,$InsertTableSQLQuery);

 }catch(Exception $e){}
 mysqli_close($con);

 }
}

function ServerConfig(){

define('HostName','localhost');
define('HostUser','root');
define('HostPass','');
define('DatabaseName','admin');

}

function GenerateFileNameUsingID(){

 $con2 = mysqli_connect(HostName,HostUser,HostPass,DatabaseName);
```

```
$GenerateFileSQL = "SELECT max(id) as id FROM PdfTable";

$Holder = mysqli_fetch_array(mysqli_query($con2,$GenerateFileSQL));

mysqli_close($con2);

if($Holder['id']==null)
{
return 1;
}
else
{
return ++$Holder['id'];
}
}

?>
```

getPdfs.pdf

```
<?php

require_once 'dbDetails.php';

//connecting to the db
```

```
$con = mysqli_connect(DB_HOST,DB_USERNAME,DB_PASSWORD,DB_NAME) or die("Unable to connect");


//sql query


$onebict= "SELECT * FROM `PdfTable` WHERE PdfName ='BICT' AND year = '1' ORDER BY id DESC";

$onecompsci= "SELECT * FROM `PdfTable` WHERE PdfName ='COMPSCI' AND year = '1' ORDER BY id DESC";


$twobict= "SELECT * FROM `PdfTable` WHERE PdfName ='BICT' AND year = '2' ORDER BY id DESC";

$twocompsci= "SELECT * FROM `PdfTable` WHERE PdfName ='COMPSCI' AND year = '2' ORDER BY id DESC";


$threebict= "SELECT * FROM `PdfTable` WHERE PdfName ='BICT' AND year = '3' ORDER BY id DESC";

$threecompsci= "SELECT * FROM `PdfTable` WHERE PdfName ='COMPSCI' AND year = '3' ORDER BY id DESC";


$fourbict= "SELECT * FROM `PdfTable` WHERE PdfName ='BICT' AND year = '4' ORDER BY id DESC";

$fourcompsci= "SELECT * FROM `PdfTable` WHERE PdfName ='COMPSCI' AND year = '4' ORDER BY id DESC";




//getting result on execution the sql query


 $bictyr1= mysqli_query($con,$onebict);

 $compsciyr1 = mysqli_query($con,$onecompsci);
```

```
$bictyr2= mysqli_query($con,$twobict);
 $compsciyr2 = mysqli_query($con,$twocompsci);

$bictyr3= mysqli_query($con,$threebict);
 $compsciyr3 = mysqli_query($con,$threecompsci);

$bictyr4= mysqli_query($con,$fourbict);
 $compsciyr4 = mysqli_query($con,$fourcompsci);

//response array
$response = array();

$response['error'] = false;

$response['message'] = "PastPapers Loaded successfully.";


$response['onebict'] = array();
$response['onecompsci'] = array();

$response['twobict'] = array();
$response['twocompsci'] = array();

$response['threebict'] = array();
```

```
$response['threecompsci'] = array();

$response['fourbict'] = array();
$response['fourcompsci'] = array();

//traversing through all the rows
while($row =mysqli_fetch_array($bictyr1)){
    $temp = array();
    $temp['id'] = $row['id'];
    $temp['PdfURL'] = $row['PdfURL'];
    $temp['PdfName'] = $row['PdfName'];
    $temp['year'] = $row['year'];
    $temp['unitname'] = $row['unitname'];
    $temp['unitcode'] = $row['unitcode'];
    $temp['lecturer'] = $row['lecturer'];


            array_push($response['onebict'],$temp);

}
while($row =mysqli_fetch_array($compsciyr1)){
    $temp = array();
    $temp['id'] = $row['id'];
    $temp['PdfURL'] = $row['PdfURL'];
    $temp['PdfName'] = $row['PdfName'];
```

```
    $temp['year'] = $row['year'];
    $temp['unitname'] = $row['unitname'];
    $temp['unitcode'] = $row['unitcode'];
    $temp['lecturer'] = $row['lecturer'];


            array_push($response['onecompsci'],$temp);

}


//2
while($row =mysqli_fetch_array($bictyr2)){
    $temp = array();
    $temp['id'] = $row['id'];
    $temp['PdfURL'] = $row['PdfURL'];
    $temp['PdfName'] = $row['PdfName'];
    $temp['year'] = $row['year'];
    $temp['unitname'] = $row['unitname'];
    $temp['unitcode'] = $row['unitcode'];
    $temp['lecturer'] = $row['lecturer'];


            array_push($response['twobict'],$temp);
```

```
}
while($row =mysqli_fetch_array($compsciyr2)){
   $temp = array();
   $temp['id'] = $row['id'];
   $temp['PdfURL'] = $row['PdfURL'];
   $temp['PdfName'] = $row['PdfName'];
   $temp['year'] = $row['year'];
   $temp['unitname'] = $row['unitname'];
   $temp['unitcode'] = $row['unitcode'];
   $temp['lecturer'] = $row['lecturer'];


            array_push($response['twocompsci'],$temp);

}

//3
while($row =mysqli_fetch_array($bictyr3)){
   $temp = array();
   $temp['id'] = $row['id'];
   $temp['PdfURL'] = $row['PdfURL'];
   $temp['PdfName'] = $row['PdfName'];
   $temp['year'] = $row['year'];
   $temp['unitname'] = $row['unitname'];
   $temp['unitcode'] = $row['unitcode'];
```

```
    $temp['lecturer'] = $row['lecturer'];



            array_push($response['threebict'],$temp);

}
while($row =mysqli_fetch_array($compsciyr3)){
    $temp = array();
    $temp['id'] = $row['id'];
    $temp['PdfURL'] = $row['PdfURL'];
    $temp['PdfName'] = $row['PdfName'];
    $temp['year'] = $row['year'];
    $temp['unitname'] = $row['unitname'];
    $temp['unitcode'] = $row['unitcode'];
    $temp['lecturer'] = $row['lecturer'];



            array_push($response['threecompsci'],$temp);

}



//4
while($row =mysqli_fetch_array($bictyr4)){
    $temp = array();
```

```
    $temp['id'] = $row['id'];
    $temp['PdfURL'] = $row['PdfURL'];
    $temp['PdfName'] = $row['PdfName'];
    $temp['year'] = $row['year'];
    $temp['unitname'] = $row['unitname'];
    $temp['unitcode'] = $row['unitcode'];
    $temp['lecturer'] = $row['lecturer'];


            array_push($response['fourbict'],$temp);

}
while($row =mysqli_fetch_array($compsciyr4)){
    $temp = array();
    $temp['id'] = $row['id'];
    $temp['PdfURL'] = $row['PdfURL'];
    $temp['PdfName'] = $row['PdfName'];
    $temp['year'] = $row['year'];
    $temp['unitname'] = $row['unitname'];
    $temp['unitcode'] = $row['unitcode'];
    $temp['lecturer'] = $row['lecturer'];


            array_push($response['fourcompsci'],$temp);
```

```
}
///
echo json_encode($response);
?>
```

activity_year

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Year"
    android:background="@android:color/holo_red_dark"
    android:layout_margin="10dp"
    android:gravity="center"
    android:orientation="vertical"
    >

    <Button
        android:layout_width="match_parent"
        android:layout_height="40dp"
```

```
    android:id="@+id/yr1"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:text="year1"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/yr2"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:background="#fff"
    android:text="year2"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/yr3"
    android:text="year3"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:background="#fff"/>
```

```
    <Button
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:id="@+id/yr4"
        android:gravity="center"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:background="#fff"
        android:text="year4"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:layout_marginTop="10dp"
        android:text="Logout"
        android:textStyle="italic"

     />

</LinearLayout
```

activity_courses3

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Courses3"
    android:background="@android:color/holo_red_dark"
    android:gravity="center"
    android:orientation="vertical"
    android:layout_margin="1dp"

    >

    <ListView
        android:layout_width="match_parent"
        android:id="@+id/listView"
        android:layout_height="wrap_content"
        android:background="@drawable/border"
        android:verticalScrollbarPosition="right"
        android:dividerHeight="3dp"
        android:divider="@android:color/background_dark"
        android:layout_marginLeft="10dp"
```

```
    android:layout_marginRight="10dp"
    android:layout_margin="10dp"
    android:paddingLeft="5dp"
    android:paddingTop="5dp"
    android:layout_marginTop="10dp"


    >
</ListView>

<Button
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/compsci"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:text="CompSci"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/bict"
    android:layout_marginTop="10dp"
```

```
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:background="#fff"
        android:text="BICT"/>


</LinearLayout>
```

activity_courses4

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Courses4"
    android:background="@android:color/holo_red_dark"
    android:gravity="center"
    android:orientation="vertical"
    android:layout_margin="1dp"

    >
```

```
<ListView
   android:layout_width="match_parent"
   android:id="@+id/listView"
   android:layout_height="wrap_content"
   android:background="@drawable/border"
   android:verticalScrollbarPosition="right"
   android:dividerHeight="3dp"
   android:divider="@android:color/background_dark"
   android:layout_marginLeft="10dp"
   android:layout_marginRight="10dp"
   android:layout_margin="10dp"
   android:paddingLeft="5dp"
   android:paddingTop="5dp"
   android:layout_marginTop="10dp"



   >
</ListView>

<Button
   android:layout_width="match_parent"
   android:layout_height="40dp"
   android:id="@+id/compsci"
   android:layout_marginTop="10dp"
```

```
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:paddingLeft="5dp"
        android:background="#fff"
        android:text="CompSci"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:id="@+id/bict"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:background="#fff"
        android:text="BICT"/>



</LinearLayout>
```

activity_admin_login

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".AdminLogin"
android:background="@color/colorAccent"
android:gravity="center"
android:orientation="vertical"
android:layout_margin="1dp">

<RelativeLayout
   android:layout_width="wrap_content"
   android:layout_height="wrap_content">

   <Button
      android:id="@+id/admin"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_alignParentRight="true"
      android:layout_marginTop="0dp"
      android:layout_marginRight="146dp"
      android:text="Admin Login" />
</RelativeLayout>
```

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/adname"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter Username" />
<EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/adpassword"
    android:inputType="textPassword"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter Password" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/loginbtn"
```

```
        android:text="Login"
        android:background="@color/colorPrimary"
        android:textColor="#fff"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"/>

</LinearLayout>
```

Activity_about

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".About">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="match_parent" />

</RelativeLayout>
```

activity_main

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:gravity="center"
    android:background="@android:color/holo_red_dark"
    android:orientation="vertical"
    android:layout_marginBottom="5dp"
    android:layout_margin="5dp">

    <EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/etname"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter Name" />

    <EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/ethobby"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter RegNo" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/etusername"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter Username" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:inputType="textPassword"
```

```
    android:id="@+id/etpassword"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter Password" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btn"
    android:text="Register"
    android:background="@color/colorPrimary"
    android:textColor="#fff"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/tvlogin"
    android:gravity="center"
    android:layout_marginTop="10dp"
    android:textColor="#fff"
```

```
        android:textSize="20sp"
        android:text="Already Registered? Login here"/>

</LinearLayout>
```

activity_upload_pdf

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".UploadPdf"
    android:layout_margin="1dp"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/border"
    android:scrollbars="vertical"
    android:scrollbarStyle="insideInset">

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
```

```
    android:text="Select PastPaper to Upload"
    android:id="@+id/button"
    android:textStyle="bold"
    />

<EditText
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:hint="Year(1/2/3/4)"
   android:id="@+id/year"/>

 <EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:hint="Course(/COMPSCI/BICT)"
    />
 <EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/unitname"
    android:hint="UnitName (ARTIFICIAL INTELLIGENCE"
    />
 <EditText
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:id="@+id/unitcode"
        android:hint="UnitCode(COMP 302)"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/lecturer"
        android:hint="LecturerName (Sir/DR/Prof/MIss/Madam Ireri"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
        android:text="Upload PastPaper"
        android:textStyle="bold"
        />



</LinearLayout>
```

activity_courses

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"

android:layout_height="match_parent"

tools:context=".Courses"

android:background="@android:color/holo_red_dark"

android:gravity="center"

android:orientation="vertical"

android:layout_margin="1dp"


>


<ListView

   android:layout_width="match_parent"

   android:id="@+id/listView"

   android:layout_height="wrap_content"

   android:background="@drawable/border"

   android:verticalScrollbarPosition="right"

   android:dividerHeight="3dp"

   android:divider="@android:color/background_dark"

   android:layout_marginLeft="10dp"

   android:layout_marginRight="10dp"

   android:layout_margin="10dp"

   android:paddingLeft="5dp"

   android:paddingTop="5dp"

   android:layout_marginTop="10dp"
```

```
    >
  </ListView>

  <Button
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/compsci"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:text="CompSci"/>
  <Button
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/bict"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:background="#fff"
    android:text="BICT"/>

</LinearLayout>
```

activity_courses2

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Courses2"
    android:background="@android:color/holo_red_dark"
    android:gravity="center"
    android:orientation="vertical"
    android:layout_margin="1dp"

    >

    <ListView
        android:layout_width="match_parent"
        android:id="@+id/listView"
        android:layout_height="wrap_content"
        android:background="@drawable/border"
        android:verticalScrollbarPosition="right"
        android:dividerHeight="3dp"
        android:divider="@android:color/background_dark"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
```

```
    android:layout_margin="10dp"
    android:paddingLeft="5dp"
    android:paddingTop="5dp"
    android:layout_marginTop="10dp"



    >
</ListView>

<Button
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/compsci"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:text="CompSci"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/bict"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
```

```
        android:layout_marginRight="20dp"
        android:background="#fff"
        android:text="BICT"/>


</LinearLayout>
```

activity_login

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LoginActivity"
    android:background="@android:color/holo_red_dark"
    android:gravity="center"
    android:orientation="vertical"
    android:layout_margin="10dp">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

```
    <Button
        android:id="@+id/admin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_marginTop="0dp"
        android:layout_marginRight="141dp"
        android:text="Student Login" />
</RelativeLayout>

<EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/etusername"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter Username" />
<EditText
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/etpassword"
```

```
    android:inputType="textPassword"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp"
    android:layout_marginLeft="20dp"
    android:paddingLeft="5dp"
    android:background="#fff"
    android:hint="Enter Password" />
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btn"
    android:text="Login"
    android:background="@color/colorPrimary"
    android:textColor="#fff"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:id="@+id/tvreg"
    android:gravity="center"
    android:layout_marginTop="10dp"
    android:textColor="#fff"
    android:textSize="20sp"
```

```
    android:text="Guest? Register here"/>




</LinearLayout>
```

listlayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
   android:layout_height="match_parent">


   <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
      android:orientation="vertical"
      android:layout_width="match_parent"
      android:layout_height="match_parent">
      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Title"
         android:layout_marginRight="10dp"
         android:layout_marginLeft="130dp"
         android:id="@+id/pdfname"
```

```
    android:gravity="center"
    android:textStyle="bold" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@id/pdfname"
    android:id="@+id/textViewName"/>

<TextView
    android:textStyle="bold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Year"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="130dp"
    android:layout_below="@id/pdfname"
    android:id="@+id/yr" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/pdfname"
```

```
    android:layout_toRightOf="@+id/yr"
    android:id="@+id/year" />

<TextView
    android:textStyle="bold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="UnitName"
    android:layout_marginRight="10dp"
    android:layout_marginLeft="130dp"
    android:layout_below="@id/year"
    android:id="@+id/uname"
    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/year"
    android:layout_toRightOf="@+id/uname"
    android:id="@+id/unitname"/>
```

```
<TextView
   android:textStyle="bold"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="UnitCode"
   android:layout_marginRight="10dp"
   android:layout_marginLeft="130dp"
   android:layout_below="@id/uname"
   android:id="@+id/ucode"
   />
<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_below="@id/uname"
   android:layout_toRightOf="@+id/ucode"
   android:id="@+id/unitcode"/>




<TextView
   android:textStyle="bold"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/ucode"
    android:text="Lecturer"
    android:layout_marginLeft="130dp"
    android:layout_marginRight="10dp"
    android:id="@+id/lec"
    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/ucode"
    android:id="@+id/lecturer"
    android:layout_alignParentRight="true"
    android:layout_toRightOf="@+id/lec"
    />




<TextView
    android:textStyle="italic"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/lecturer"
```

```
        android:textColor="#ccc"
        android:id="@+id/textViewUrl"/>

    </RelativeLayout>

</LinearLayout>
```

Courses

```
package com.example.lupp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.ProgressDialog;
import android.support.annotation.StringDef;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
```

```
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;
```

```
import java.util.List;

import java.util.UUID;


public class Courses extends AppCompatActivity implements View.OnClickListener {


    private EditText editText;

    public static final String PDF_FETCH_URL =
"http://192.168.43.247:80/AndroidUpload/getPdfs.php";

    //Image request code

    private int PICK_PDF_REQUEST = 1;

    //storage permission code

    private static final int STORAGE_PERMISSION_CODE = 123;

    //Uri to store the image uri

    private Uri filePath;

    //ListView to show the fetched Pdfs from the server

    ListView listView;

    //button to fetch the intiate the fetching of pdfs.

    Button buttonFetch;

    Button bict1;

    //Progress bar to check the progress of obtaining pdfs

    ProgressDialog progressDialog;

    //an array to hold the different pdf objects

    ArrayList<Pdf> pdfList= new ArrayList<Pdf>();

    //pdf adapter

    PdfAdapter pdfAdapter;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_courses);

    //initializing ListView
    listView = (ListView) findViewById(R.id.listView);

    //initializing buttonFetch
    buttonFetch = (Button) findViewById(R.id.compsci);
    bict1 = (Button) findViewById(R.id.bict);
    //initializing progressDialog

    progressDialog = new ProgressDialog(this);

    //Setting clicklistener

    buttonFetch.setOnClickListener(this);
    bict1.setOnClickListener(this);///this crefers to implement onclick in current class



    //setting listView on item click listener
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
```

```
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {



            Pdf pdf = (Pdf) parent.getItemAtPosition(position);
            Intent intent = new Intent();
            intent.setAction(Intent.ACTION_VIEW);
            intent.addCategory(Intent.CATEGORY_BROWSABLE);
            intent.setData(Uri.parse(pdf.getUrl()));
            startActivity(intent);


        }
    });



}



/*
 * This is the method responsible for pdf upload
 * We need the full pdf path and the name for the pdf in this method
 * */

//method to show file chooser
```

```
    //handling the ima chooser activity result
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PICK_PDF_REQUEST && resultCode == RESULT_OK && data != null && data.getData() != null) {
            filePath = data.getData();

        }
    }


    //Requesting permission


    //This method will be called when the user will tap on allow or deny
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

        //Checking the request code of our request
        if (requestCode == STORAGE_PERMISSION_CODE) {

            //If permission is granted
```

```
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

            //Displaying a toast

            Toast.makeText(this, "Permission granted now you can read the storage",
Toast.LENGTH_LONG).show();

        } else {

            //Displaying another toast if permission is not granted

            Toast.makeText(this, "Oops you just denied the permission",
Toast.LENGTH_LONG).show();

        }

    }

}



@Override

public void onClick(View v) {

    if(v==buttonFetch){

        getPdfs();

    }

    if (v == bict1){

        getbict1();

    }


}

//onebict fetchpdfs

private void getbict1() {
```

```
progressDialog.setMessage("Loading PastPapers... Please Wait");
progressDialog.show();
StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {


                progressDialog.dismiss();
                try {
                    JSONObject obj = new JSONObject(response);
                    Toast.makeText(Courses.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();

                    JSONArray jsonArray = obj.getJSONArray("onebict");

                    for(int i=0;i<jsonArray.length();i++){

                        //Declaring a json object corresponding to every pdf object in our json Array
                        JSONObject jsonObject = jsonArray.getJSONObject(i);
                        //Declaring a Pdf object to add it to the ArrayList  pdfList
                        Pdf pdf  = new Pdf();
                        String pdfName = jsonObject.getString("PdfName");
                        String pdfUrl = jsonObject.getString("PdfURL");
```

```
            String year = jsonObject.getString("year");
            String unitname = jsonObject.getString("unitname");
            String unitcode = jsonObject.getString("unitcode");
            String lecturer =  jsonObject.getString("lecturer");
            pdf.setName(pdfName);
            pdf.setUrl(pdfUrl);
            pdf.setYear(year);
            pdf.setUnitname(unitname);
            pdf.setUnitcode(unitcode);
            pdf.setLecturer(lecturer);
            pdfList.add(pdf);



        }

        pdfAdapter=new PdfAdapter(Courses.this,R.layout.listlayout, pdfList);

        listView.setAdapter(pdfAdapter);

        pdfAdapter.notifyDataSetChanged();

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

```
            },

            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {

                }
            }
        );

        RequestQueue request = Volley.newRequestQueue(this);
        request.add(stringRequest);

    }

//onecompsci fetchpdf
    private void getPdfs() {

        progressDialog.setMessage("Loading PastPapers... Please Wait");
        progressDialog.show();
        StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

            new Response.Listener<String>() {
                @Override
```

```
public void onResponse(String response) {



    progressDialog.dismiss();
    try {
        JSONObject obj = new JSONObject(response);
        Toast.makeText(Courses.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();


        JSONArray jsonArray = obj.getJSONArray("onecompsci");


        for(int i=0;i<jsonArray.length();i++){


            //Declaring a json object corresponding to every pdf object in our json Array
            JSONObject jsonObject = jsonArray.getJSONObject(i);
            //Declaring a Pdf object to add it to the ArrayList  pdfList
            Pdf pdf  = new Pdf();
            String pdfName = jsonObject.getString("PdfName");
            String pdfUrl = jsonObject.getString("PdfURL");
            String year = jsonObject.getString("year");
            String unitname = jsonObject.getString("unitname");
            String unitcode = jsonObject.getString("unitcode");
            String lecturer =  jsonObject.getString("lecturer");
            pdf.setName(pdfName);
            pdf.setUrl(pdfUrl);
```

```
            pdf.setYear(year);
            pdf.setUnitname(unitname);
            pdf.setUnitcode(unitcode);
            pdf.setLecturer(lecturer);
            pdfList.add(pdf);


        }

        pdfAdapter=new PdfAdapter(Courses.this,R.layout.listlayout, pdfList);

        listView.setAdapter(pdfAdapter);

        pdfAdapter.notifyDataSetChanged();

      } catch (JSONException e) {
        e.printStackTrace();
      }


  }
},

new Response.ErrorListener() {
  @Override
```

```
            public void onErrorResponse(VolleyError error) {


            }
        }
    );

    RequestQueue request = Volley.newRequestQueue(this);
    request.add(stringRequest);


}


}
```

Courses2

```
package com.example.lupp;


import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.ProgressDialog;
import android.support.annotation.StringDef;
import android.Manifest;
import android.content.Intent;
```

```
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
```

```
import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;

public class Courses2 extends AppCompatActivity implements View.OnClickListener {

    private EditText editText;
    public static final String PDF_FETCH_URL =
"http://192.168.43.247:80/AndroidUpload/getPdfs.php";
    //Image request code
    private int PICK_PDF_REQUEST = 1;
    //storage permission code
    private static final int STORAGE_PERMISSION_CODE = 123;
    //Uri to store the image uri
    private Uri filePath;
    //ListView to show the fetched Pdfs from the server
    ListView listView;
    //button to fetch the intiate the fetching of pdfs.
    Button buttonFetch;
```

```
Button bict2;
//Progress bar to check the progress of obtaining pdfs
ProgressDialog progressDialog;
//an array to hold the different pdf objects
ArrayList<Pdf> pdfList= new ArrayList<Pdf>();
//pdf adapter
PdfAdapter pdfAdapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_courses2);

    //initializing ListView
    listView = (ListView) findViewById(R.id.listView);

    //initializing buttonFetch
    buttonFetch = (Button) findViewById(R.id.compsci);
    bict2 = (Button) findViewById(R.id.bict);
    //initializing progressDialog

    progressDialog = new ProgressDialog(this);

    //Setting clicklistener
```

```
    buttonFetch.setOnClickListener(this);
    bict2.setOnClickListener(this);///this crefers to implement onclick in current class



    //setting listView on item click listener
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {


            Pdf pdf = (Pdf) parent.getItemAtPosition(position);
            Intent intent = new Intent();
            intent.setAction(Intent.ACTION_VIEW);
            intent.addCategory(Intent.CATEGORY_BROWSABLE);
            intent.setData(Uri.parse(pdf.getUrl()));
            startActivity(intent);


        }
    });


}
```

```
    /*
     * This is the method responsible for pdf upload
     * We need the full pdf path and the name for the pdf in this method
     * */



    //method to show file chooser


    //handling the ima chooser activity result
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);


        if (requestCode == PICK_PDF_REQUEST && resultCode == RESULT_OK && data !=
null && data.getData() != null) {
            filePath = data.getData();


        }
    }



    //Requesting permission
```

```
    //This method will be called when the user will tap on allow or deny

    @Override

    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {


        //Checking the request code of our request

        if (requestCode == STORAGE_PERMISSION_CODE) {


            //If permission is granted

            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

                //Displaying a toast

                Toast.makeText(this, "Permission granted now you can read the storage",
Toast.LENGTH_LONG).show();

            } else {

                //Displaying another toast if permission is not granted

                Toast.makeText(this, "Oops you just denied the permission",
Toast.LENGTH_LONG).show();

            }

        }

    }




    @Override

    public void onClick(View v) {

        if(v==buttonFetch){
```

```
        getPdfs();
    }
    if (v == bict2){
        getbict2();
    }

}
//onebict fetchpdfs
private void getbict2() {


    progressDialog.setMessage("Loading PastPapers... Please Wait");
    progressDialog.show();
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {


                    progressDialog.dismiss();
                    try {
                        JSONObject obj = new JSONObject(response);
                        Toast.makeText(Courses2.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();
```

```
JSONArray jsonArray = obj.getJSONArray("twobict");

for(int i=0;i<jsonArray.length();i++){

    //Declaring a json object corresponding to every pdf object in our json Array
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    //Declaring a Pdf object to add it to the ArrayList  pdfList
    Pdf pdf  = new Pdf();
    String pdfName = jsonObject.getString("PdfName");
    String pdfUrl = jsonObject.getString("PdfURL");
    String year = jsonObject.getString("year");
    String unitname = jsonObject.getString("unitname");
    String unitcode = jsonObject.getString("unitcode");
    String lecturer =  jsonObject.getString("lecturer");
    pdf.setName(pdfName);
    pdf.setUrl(pdfUrl);
    pdf.setYear(year);
    pdf.setUnitname(unitname);
    pdf.setUnitcode(unitcode);
    pdf.setLecturer(lecturer);
    pdfList.add(pdf);


}
```

```
                pdfAdapter=new PdfAdapter(Courses2.this,R.layout.listlayout, pdfList);

                listView.setAdapter(pdfAdapter);

                pdfAdapter.notifyDataSetChanged();

            } catch (JSONException e) {
                e.printStackTrace();
            }


        }
    },

    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
    }
);

RequestQueue request = Volley.newRequestQueue(this);
request.add(stringRequest);
```

```
    }

    //onecompsci fetchpdf
    private void getPdfs() {

        progressDialog.setMessage("Loading PastPapers... Please Wait");
        progressDialog.show();
        StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {



                        progressDialog.dismiss();
                        try {
                            JSONObject obj = new JSONObject(response);
                            Toast.makeText(Courses2.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();

                            JSONArray jsonArray = obj.getJSONArray("twocompsci");

                            for(int i=0;i<jsonArray.length();i++){
```

```
    //Declaring a json object corresponding to every pdf object in our json Array
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    //Declaring a Pdf object to add it to the ArrayList  pdfList
    Pdf pdf  = new Pdf();
    String pdfName = jsonObject.getString("PdfName");
    String pdfUrl = jsonObject.getString("PdfURL");
    String year = jsonObject.getString("year");
    String unitname = jsonObject.getString("unitname");
    String unitcode = jsonObject.getString("unitcode");
    String lecturer =  jsonObject.getString("lecturer");
    pdf.setName(pdfName);
    pdf.setUrl(pdfUrl);
    pdf.setYear(year);
    pdf.setUnitname(unitname);
    pdf.setUnitcode(unitcode);
    pdf.setLecturer(lecturer);
    pdfList.add(pdf);


}

pdfAdapter=new PdfAdapter(Courses2.this,R.layout.listlayout, pdfList);

listView.setAdapter(pdfAdapter);
```

```
                pdfAdapter.notifyDataSetChanged();

            } catch (JSONException e) {
                e.printStackTrace();
            }


        }
    },

    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
    }
);

RequestQueue request = Volley.newRequestQueue(this);
request.add(stringRequest);

}


}
```

Utils

```
package com.example.lupp;


import android.app.ProgressDialog;

import android.content.Context;

import android.net.ConnectivityManager;

import android.net.NetworkInfo;


public class Utils {


    private static ProgressDialog mProgressDialog;

    public static void showSimpleProgressDialog(Context context, String title, String msg,
boolean isCancelable) {

        try {

            if (mProgressDialog == null) {

                mProgressDialog = ProgressDialog.show(context, title, msg);

                mProgressDialog.setCancelable(isCancelable);

            }

            if (!mProgressDialog.isShowing()) {

                mProgressDialog.show();

            }


        } catch (IllegalArgumentException ie) {

            ie.printStackTrace();
```

```
    } catch (RuntimeException re) {
        re.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
public static void showSimpleProgressDialog(Context context) {
    showSimpleProgressDialog(context, null, "Loading...", false);
}
public static void removeSimpleProgressDialog() {
    try {
        if (mProgressDialog != null) {
            if (mProgressDialog.isShowing()) {
                mProgressDialog.dismiss();
                mProgressDialog = null;
            }
        }
    } catch (IllegalArgumentException ie) {
        ie.printStackTrace();

    } catch (RuntimeException re) {
        re.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
```

```
    }
    public static boolean isNetworkAvailable(Context context) {
        ConnectivityManager connectivity = (ConnectivityManager) context
                .getSystemService(Context.CONNECTIVITY_SERVICE);
        if (connectivity == null) {
            return false;
        } else {
            NetworkInfo[] info = connectivity.getAllNetworkInfo();
            if (info != null) {
                for (int i = 0; i < info.length; i++) {
                    if (info[i].getState() == NetworkInfo.State.CONNECTED) {
                        return true;
                    }
                }
            }
        }
        return false;
    }

}
```

FilePath.java

```
package com.example.lupp;

import android.content.ContentUris;

import android.content.Context;

import android.database.Cursor;

import android.net.Uri;

import android.os.Build;

import android.os.Environment;

import android.provider.DocumentsContract;

import android.provider.MediaStore;


public class FilePath

{

   /**

    * Method for return file path of Gallery image

    *

    * @param context

    * @param uri

    * @return path of the selected image file from gallery

    */


   public static String getPath(final Context context, final Uri uri)

   {

      //check here to KITKAT or new version

      final boolean isKitKat = Build.VERSION.SDK_INT >=
Build.VERSION_CODES.KITKAT;
```

```
// DocumentProvider
if (isKitKat && DocumentsContract.isDocumentUri(context, uri)) {

    // ExternalStorageProvider
    if (isExternalStorageDocument(uri)) {
        final String docId = DocumentsContract.getDocumentId(uri);
        final String[] split = docId.split(":");
        final String type = split[0];

        if ("primary".equalsIgnoreCase(type)) {
            return Environment.getExternalStorageDirectory() + "/" + split[1];
        }
    }

    //DownloadsProvider
    else if (isDownloadsDocument(uri)) {

        final String id = DocumentsContract.getDocumentId(uri);
        final Uri contentUri = ContentUris.withAppendedId(
                Uri.parse("content://downloads/public_downloads"), Long.valueOf(id));

        return getDataColumn(context, contentUri, null, null);
    }
```

```
    // MediaProvider
    else if (isMediaDocument(uri)) {
        final String docId = DocumentsContract.getDocumentId(uri);
        final String[] split = docId.split(":");
        final String type = split[0];

        Uri contentUri = null;
        if ("image".equals(type)) {
            contentUri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
        } else if ("video".equals(type)) {
            contentUri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
        } else if ("audio".equals(type)) {
            contentUri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
        }

        final String selection = "_id=?";
        final String[] selectionArgs = new String[] {
                split[1]
        };

        return getDataColumn(context, contentUri, selection, selectionArgs);
    }
}
// MediaStore (and general)
else if ("content".equalsIgnoreCase(uri.getScheme())) {
```

```
        // Return the remote address
        if (isGooglePhotosUri(uri))
            return uri.getLastPathSegment();

        return getDataColumn(context, uri, null, null);
    }
    // File
    else if ("file".equalsIgnoreCase(uri.getScheme())) {
        return uri.getPath();
    }

    return null;
}

/**
 * Get the value of the data column for this Uri. This is useful for
 * MediaStore Uris, and other file-based ContentProviders.
 *
 * @param context The context.
 * @param uri The Uri to query.
 * @param selection (Optional) Filter used in the query.
 * @param selectionArgs (Optional) Selection arguments used in the query.
 * @return The value of the _data column, which is typically a file path.
 */
```

```
public static String getDataColumn(Context context, Uri uri, String selection,
                                   String[] selectionArgs) {

    Cursor cursor = null;
    final String column = "_data";
    final String[] projection = {
            column
    };

    try {
        cursor = context.getContentResolver().query(uri, projection, selection, selectionArgs,
                null);
        if (cursor != null && cursor.moveToFirst()) {
            final int index = cursor.getColumnIndexOrThrow(column);
            return cursor.getString(index);
        }
    } finally {
        if (cursor != null)
            cursor.close();
    }
    return null;
}

/**
 * @param uri The Uri to check.
```

```
 * @return Whether the Uri authority is ExternalStorageProvider.
 */
public static boolean isExternalStorageDocument(Uri uri) {
    return "com.android.externalstorage.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is DownloadsProvider.
 */
public static boolean isDownloadsDocument(Uri uri) {
    return "com.android.providers.downloads.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is MediaProvider.
 */
public static boolean isMediaDocument(Uri uri) {
    return "com.android.providers.media.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is Google Photos.
```

```
     */
    public static boolean isGooglePhotosUri(Uri uri) {
        return "com.google.android.apps.photos.content".equals(uri.getAuthority());
    }
}
```

MainActivity.java

```
package com.example.lupp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
```

```
import android.widget.Toast;
import org.json.JSONException;
import java.io.IOException;
import java.util.HashMap;

public class MainActivity extends AppCompatActivity {

    private EditText etname, ethobby, etusername, etpassword;
    private Button btnregister;
    private TextView tvlogin;
    private ParseContent parseContent;
    private PreferenceHelper preferenceHelper;
    private final int RegTask = 1;

    //menu
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()){
```

```
        case R.id.home:
            Intent intent2 = new Intent(MainActivity.this, MainActivity.class);
            startActivity(intent2);
            return true;

        case R.id.admin:
            Intent intent1 = new Intent(MainActivity.this, AdminLogin.class);
            startActivity(intent1);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}



//layout
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);



    preferenceHelper = new PreferenceHelper(this);
    parseContent = new ParseContent(this);
```

```
    if(preferenceHelper.getIsLogin()){
        Intent intent = new Intent(MainActivity.this,Year.class);
        ///intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        this.finish();
    }

    etname = (EditText) findViewById(R.id.etname);
    ethobby = (EditText) findViewById(R.id.ethobby);
    etusername = (EditText) findViewById(R.id.etusername);
    etpassword = (EditText) findViewById(R.id.etpassword);

    btnregister = (Button) findViewById(R.id.btn);
    tvlogin = (TextView) findViewById(R.id.tvlogin);

    tvlogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this,LoginActivity.class);
            startActivity(intent);
        }
    });
```

```
        btnregister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    register();
                } catch (IOException e) {
                    e.printStackTrace();
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });

    }

    private void register() throws IOException, JSONException {

        if (!Utils.isNetworkAvailable(MainActivity.this)) {
            Toast.makeText(MainActivity.this, "Internet is required!",
Toast.LENGTH_SHORT).show();
            return;
        }
        Utils.showSimpleProgressDialog(MainActivity.this);
        final HashMap<String, String> map = new HashMap<>();
        map.put(Constants.Params.NAME, etname.getText().toString());
```

```
        map.put(Constants.Params.HOBBY, ethobby.getText().toString());
        map.put(Constants.Params.USERNAME, etusername.getText().toString());
        map.put(Constants.Params.PASSWORD, etpassword.getText().toString());
        new AsyncTask<Void, Void, String>(){
            protected String doInBackground(Void[] params) {
                String response="";
                try {
                    HttpRequest req = new HttpRequest(Constants.ServiceType.REGISTER);
                    response =
req.prepare(HttpRequest.Method.POST).withData(map).sendAndReadString();
                } catch (Exception e) {
                    response=e.getMessage();
                }
                return response;
            }
            protected void onPostExecute(String result) {
                //do something with response
                Log.d("newwwss", result);
                onTaskCompleted(result, RegTask);
            }
        }.execute();
    }

    private void onTaskCompleted(String response,int task) {
        Log.d("responsejson", response.toString());
```

```
        Utils.removeSimpleProgressDialog();  //will remove progress dialog
        switch (task) {
            case RegTask:

                if (parseContent.isSuccess(response)) {

                    parseContent.saveInfo(response);
                    Toast.makeText(MainActivity.this, "Registered Successfully!",
Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(MainActivity.this, Year.class);
                    ///intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
                    startActivity(intent);
                    this.finish();

                } else {
                    Toast.makeText(MainActivity.this, parseContent.getErrorMessage(response),
Toast.LENGTH_SHORT).show();
                }

        }
    }
}
```

Year.java

```
package com.example.lupp;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

public class Year extends AppCompatActivity {
    private PreferenceHelper preferenceHelper;

    //menu
    //menu
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
```

```
    switch (item.getItemId()){
        case R.id.home:
            Intent intent2 = new Intent(Year.this, MainActivity.class);
            startActivity(intent2);
            return true;

        case R.id.admin:
            Intent intent1 = new Intent(Year.this, AdminLogin.class);
            startActivity(intent1);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}



@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_year);

    Button year1 = (Button) findViewById(R.id.yr1);
    Button year2 = (Button) findViewById(R.id.yr2);
    Button year3 = (Button) findViewById(R.id.yr3);
```

```
        Button year4 = (Button) findViewById(R.id.yr4);
       Button  btnlogout = (Button) findViewById(R.id.btn);
        preferenceHelper = new PreferenceHelper(this);


        btnlogout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                preferenceHelper.putIsLogin(false);
                Intent intent = new Intent(Year.this,MainActivity.class);
                /// intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
                Year.this.finish();
            }
        });
        year1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Year.this, Courses.class);
                startActivity(intent);
            }
        });


        year2.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            Intent intent = new Intent(Year.this, Courses2.class);
            startActivity(intent);
        }
    });

    year3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(Year.this, Courses3.class);
            startActivity(intent);
        }
    });

    year4.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(Year.this, Courses4.class);
            startActivity(intent);
        }
    });
  }
}
```

LoginActivity.java

```
package com.example.lupp;


import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import org.json.JSONException;
import java.io.IOException;
import java.util.HashMap;
```

```
public class LoginActivity extends AppCompatActivity {

    private EditText etusername, etpassword;
    private Button btnlogin;
    private TextView tvreg;
    private ParseContent parseContent;
    private final int LoginTask = 1;
    private PreferenceHelper preferenceHelper;
    Button adlogin;



    //menu
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()){
            case R.id.home:
                Intent intent2 = new Intent(LoginActivity.this, MainActivity.class);
                startActivity(intent2);
```

```
            return true;


        case R.id.admin:
            Intent intent1 = new Intent(LoginActivity.this, AdminLogin.class);
            startActivity(intent1);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}




//layout
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);


    parseContent = new ParseContent(this);
    preferenceHelper = new PreferenceHelper(this);


    etusername = (EditText) findViewById(R.id.etusername);
    etpassword = (EditText) findViewById(R.id.etpassword);
```

```
btnlogin = (Button) findViewById(R.id.btn);
///adlogin = (Button) findViewById(R.id.admin);
tvreg = (TextView) findViewById(R.id.tvreg);

tvreg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(LoginActivity.this, MainActivity.class);
        startActivity(intent);
    }
});
/*adlogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(LoginActivity.this,AdminLogin.class);
        startActivity(intent);
    }
});*/

btnlogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            login();
```

```
            } catch (IOException e) {
                e.printStackTrace();
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });

}

private void login() throws IOException, JSONException {

    if (!Utils.isNetworkAvailable(LoginActivity.this)) {
        Toast.makeText(LoginActivity.this, "Internet is required!",
Toast.LENGTH_SHORT).show();
        return;
    }
    Utils.showSimpleProgressDialog(LoginActivity.this);
    final HashMap<String, String> map = new HashMap<>();
    map.put(Constants.Params.USERNAME, etusername.getText().toString());
    map.put(Constants.Params.PASSWORD, etpassword.getText().toString());
    new AsyncTask<Void, Void, String>(){
        protected String doInBackground(Void[] params) {
            String response="";
            try {
```

```
                HttpRequest req = new HttpRequest(Constants.ServiceType.LOGIN);
                response =
req.prepare(HttpRequest.Method.POST).withData(map).sendAndReadString();
            } catch (Exception e) {
                response=e.getMessage();
            }
            return response;
        }
        protected void onPostExecute(String result) {
            //do something with response
            Log.d("newwwss", result);
            onTaskCompleted(result,LoginTask);
        }
    }.execute();
  }


  private void onTaskCompleted(String response,int task) {
    Log.d("responsejson", response.toString());
   Utils.removeSimpleProgressDialog();  //will remove progress dialog
    switch (task) {
        case LoginTask:
            if (parseContent.isSuccess(response)) {
                parseContent.saveInfo(response);
                Toast.makeText(LoginActivity.this, "Login Successfully!",
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(LoginActivity.this,Year.class);
```

```
                ////intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
                this.finish();
            }else {
                Toast.makeText(LoginActivity.this, parseContent.getErrorMessage(response),
Toast.LENGTH_SHORT).show();
            }
        }
    }
    }
```

AdminLogin.java

```
package com.example.lupp;


import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONException;

import java.io.IOException;
import java.util.HashMap;

public class AdminLogin extends AppCompatActivity {

    private EditText etusername, etpassword;
    private Button btnlogin;
    private TextView tvreg;
    private ParseContent parseContent;
    private final int LoginTask = 1;
    private PreferenceHelper preferenceHelper;
    Button adlogin;

    //MENU
    //menu
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.menu, menu);
```

```
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()){
        case R.id.home:
            Intent intent2 = new Intent(AdminLogin.this, MainActivity.class);
            startActivity(intent2);
            return true;

        case R.id.admin:
            Intent intent1 = new Intent(AdminLogin.this, AdminLogin.class);
            startActivity(intent1);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_admin_login);
```

```
parseContent = new ParseContent(this);
preferenceHelper = new PreferenceHelper(this);

etusername = (EditText) findViewById(R.id.adname);
etpassword = (EditText) findViewById(R.id.adpassword);
btnlogin = (Button) findViewById(R.id.loginbtn);

/// tvreg = (TextView) findViewById(R.id.tvreg);




btnlogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            login();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (JSONException e) {
            e.printStackTrace();
        }
```

```
        }
    });


}
private void login() throws IOException, JSONException {


    if (!Utils.isNetworkAvailable(AdminLogin.this)) {
        Toast.makeText(AdminLogin.this, "Internet is required!",
Toast.LENGTH_SHORT).show();
        return;
    }
    Utils.showSimpleProgressDialog(AdminLogin.this);
    final HashMap<String, String> map = new HashMap<>();
    map.put(Constants.Params.USERNAME, etusername.getText().toString());
    map.put(Constants.Params.PASSWORD, etpassword.getText().toString());
    new AsyncTask<Void, Void, String>(){
        protected String doInBackground(Void[] params) {
            String response="";
            try {
                HttpRequest req = new HttpRequest(Constants.ServiceType.ADMINLOGIN);
                response =
req.prepare(HttpRequest.Method.POST).withData(map).sendAndReadString();
            } catch (Exception e) {
                response=e.getMessage();
            }
            return response;
```

```
        }
        protected void onPostExecute(String result) {
            //do something with response
            Log.d("newwwss", result);
            onTaskCompleted(result,LoginTask);
        }
    }.execute();
  }


  private void onTaskCompleted(String response,int task) {
    Log.d("responsejson", response.toString());
    Utils.removeSimpleProgressDialog();  //will remove progress dialog
    switch (task) {
      case LoginTask:
        if (parseContent.isSuccess(response)) {
          parseContent.saveInfo(response);
          Toast.makeText(AdminLogin.this, "Admin Login Successfully!",
Toast.LENGTH_SHORT).show();
          Intent intent = new Intent(AdminLogin.this, UploadPdf.class);
          ////intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
          startActivity(intent);
          this.finish();
        } else {
          Toast.makeText(AdminLogin.this, parseContent.getErrorMessage(response),
Toast.LENGTH_SHORT).show();
```

```
            }
        }

    }
}
```

HttpRequest.java

```
package com.example.lupp;

import android.util.Log;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
```

```
import java.net.URL;
import java.util.HashMap;
import java.util.Map;

public class HttpRequest {

    public static enum Method{
        POST,PUT,DELETE,GET;
    }
    private URL url;
    private HttpURLConnection con;
    private OutputStream os;
    //After instantiation, when opening connection - IOException can occur
    public HttpRequest(URL url)throws IOException {
        this.url=url;
        con = (HttpURLConnection)this.url.openConnection();
    }
    //Can be instantiated with String representation of url, force caller to check for IOException
which can be thrown
    public HttpRequest(String url)throws IOException{
        this(new URL(url));
        Log.d("parameters", url);
    }

    /**
```

```
     * Sending connection and opening an output stream to server by pre-defined instance variable
url
     *
     * @param //isPost boolean - indicates whether this request should be sent in POST method
     * @throws IOException - should be checked by caller
     * */
    private void prepareAll(Method method)throws IOException{
        con.setDoInput(true);
        con.setRequestMethod(method.name());
        if(method== Method.POST||method== Method.PUT){
            con.setDoOutput(true);
            os = con.getOutputStream();
        }
    }
    //prepare request in GET method
    //@return HttpRequest this instance -> for chaining method @see line 22
    public HttpRequest prepare() throws IOException{
        prepareAll(Method.GET);
        return this;
    }
    /**
     * Prepares HttpRequest method with for given method, possible values:
HttpRequest.Method.POST,
     * HttpRequest.Method.PUT, HttpRequest.Method.GET & HttpRequest.Method.DELETE
     *
     * @param method HttpRequest.Method - nested enum HttpRequest.Method constant
```

```
 * @return HttpRequest this instance -> for chaining method @see line 22
 * @throws IOException - should be checked by caller
 * */
public HttpRequest prepare(HttpRequest.Method method)throws IOException{
    prepareAll(method);
    return this;
}
/**
 * Adding request headers (standard format "Key":"Value")
 *
 * @param headers String variadic params in standard format "Key":"Value"
 * @return HttpRequest this instance -> for chaining method @see line 22
 * */
public HttpRequest withHeaders(String... headers){
    for(int i=0,last=headers.length;i<last;i++) {
        String[]h=headers[i].split("[:]");
        con.setRequestProperty(h[0],h[1]);
    }
    return this;
}

/**
 * Writes query to open stream to server
 *
 * @param query String params in format of key1=v1&key2=v2 to open stream to server
```

```
     * @return HttpRequest this instance -> for chaining method @see line 22
     * @throws IOException - should be checked by caller
     * */
    public HttpRequest withData(String query) throws IOException{
        BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, "UTF-8"));
        writer.write(query);
        writer.close();
        return this;
    }
    /**
     * Builds query on format of key1=v1&key2=v2 from given hashMap structure
     * for map: {name=Bubu, age=29} -> builds "name=Bubu&age=29"
     * for map: {Iam=Groot} -> builds "Iam=Groot"
     *
     * @param params HashMap consists of key-> value pairs to build query from
     * @return HttpRequest this instance -> for chaining method @see line 22
     * @throws IOException - should be checked by caller
     * */
    public HttpRequest withData(HashMap<String,String> params) throws IOException{
        StringBuilder result=new StringBuilder();
        for(Map.Entry<String,String>entry : params.entrySet()){

result.append((result.length()>0?"&":"")+entry.getKey()+"="+entry.getValue());//appends:
key=value (for first param) OR &key=value(second and more)
            Log.d("parameters",entry.getKey()+"  ===>  "+ entry.getValue());
        }
```

```
        withData(result.toString());
        return this;
    }
    //When caller only need to send, and don't need String response from server
    public int send() throws IOException{
        return con.getResponseCode(); //return HTTP status code to indicate whether it successfully sent
    }
    /**
     * Sending request to the server and pass to caller String as it received in response from server
     *
     * @return String printed from server's response
     * @throws IOException - should be checked by caller
     * */
    public String sendAndReadString() throws IOException{
        BufferedReader br=new BufferedReader(new InputStreamReader(con.getInputStream()));
        StringBuilder response=new StringBuilder();
        for(String line;(line=br.readLine())!=null;)response.append(line+"\n");
        Log.d("ressss",response.toString());
        return response.toString();
    }
    /**
     * Sending request to the server and pass to caller its raw contents in bytes as it received from server.
     *
     * @return byte[] from server's response
```

```
     * @throws IOException - should be checked by caller
     * */
    public byte[] sendAndReadBytes() throws IOException{
        byte[] buffer = new byte[8192];
        InputStream is = con.getInputStream();
        ByteArrayOutputStream output = new ByteArrayOutputStream();
        for (int bytesRead;(bytesRead=is.read(buffer))>=0;)output.write(buffer, 0, bytesRead);
        return output.toByteArray();
    }
    //JSONObject representation of String response from server
    public JSONObject sendAndReadJSON() throws JSONException, IOException{
        return new JSONObject(sendAndReadString());
    }
}
```

UploadPdf.java

```
package com.example.lupp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.Manifest;
import android.content.Intent;
```

```
import android.content.pm.PackageManager;
import android.net.Uri;
import android.support.v4.app.ActivityCompat;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;
import java.util.UUID;

public class UploadPdf extends AppCompatActivity {

    Button SelectButton, UploadButton;



    EditText Year, Unitcode,Unitname, Coursename, Lecturer;


    Uri uri;


    public static final String PDF_UPLOAD_HTTP_URL =
"http://192.168.43.247:80/AndroidUpload/file_upload.php";
```

```
public int PDF_REQ_CODE = 1;


String PdfNameHolder, PdfPathHolder, PdfID;
String YearNameHolder;/// YearPathHolder, YearID;
String UnitcodeHolder; ///UnitCoursePathHolder, UnitCourseID;
String UnitNameHolder; ///UnitNamePathHolder, UNitNameID;
String LecturerNameHolder;///LecturerPathHolder, LecturerID;


//MENU
//menu
@Override
public boolean onCreateOptionsMenu(Menu menu) {
   MenuInflater inflater = getMenuInflater();
   inflater.inflate(R.menu.menu, menu);
   return true;
}


@Override
public boolean onOptionsItemSelected(MenuItem item) {
   switch (item.getItemId()){
      case R.id.home:
         Intent intent2 = new Intent(UploadPdf.this, MainActivity.class);
         startActivity(intent2);
         return true;
```

```
        case R.id.admin:
            Intent intent1 = new Intent(UploadPdf.this, AdminLogin.class);
            startActivity(intent1);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}


@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_upload_pdf);

    AllowRunTimePermission();

    SelectButton = (Button) findViewById(R.id.button);
    UploadButton = (Button) findViewById(R.id.button2);

    Year = (EditText)findViewById(R.id.year);
    Unitcode = (EditText)findViewById(R.id.unitcode);
    Unitname  = (EditText)findViewById(R.id.unitname);
    Coursename = (EditText)findViewById(R.id.editText);
```

```
Lecturer = (EditText)findViewById(R.id.lecturer);



SelectButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // PDF selection code start from here .

        Intent intent = new Intent();

        intent.setType("application/pdf");

        intent.setAction(Intent.ACTION_GET_CONTENT);

        startActivityForResult(Intent.createChooser(intent, "Select Pdf"), PDF_REQ_CODE);

    }
});

UploadButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        PdfUploadFunction();
```

```
            }
        });

    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == PDF_REQ_CODE && resultCode == RESULT_OK && data != null
&& data.getData() != null) {

            uri = data.getData();

            SelectButton.setText("Document is Selected");
        }
    }

    public void PdfUploadFunction() {

        PdfNameHolder = Coursename.getText().toString().trim();
        YearNameHolder = Year.getText().toString().trim();
        UnitNameHolder = Unitname.getText().toString().trim();
        UnitcodeHolder = Unitcode.getText().toString().trim();
        LecturerNameHolder = Lecturer.getText().toString().trim();
```

```
PdfPathHolder = FilePath.getPath(this, uri);

if (PdfPathHolder == null) {

    Toast.makeText(this, "Please move your PDF file to internal storage & try again.", Toast.LENGTH_LONG).show();

} else {

   try {

      PdfID = UUID.randomUUID().toString();

      new MultipartUploadRequest(this, PdfID, PDF_UPLOAD_HTTP_URL)
            .addFileToUpload(PdfPathHolder, "pdf")
            .addParameter("name", PdfNameHolder)
            .addParameter("year",YearNameHolder)
            .addParameter("unitname", UnitNameHolder)
            .addParameter("unitcode", UnitcodeHolder)
            .addParameter("lecturer",LecturerNameHolder)
            .setNotificationConfig(new UploadNotificationConfig())
            .setMaxRetries(5)
```

```
                .startUpload();


        } catch (Exception exception) {


            Toast.makeText(this, exception.getMessage(), Toast.LENGTH_SHORT).show();

        }

    }

}



public void AllowRunTimePermission(){


    if (ActivityCompat.shouldShowRequestPermissionRationale(UploadPdf.this,
Manifest.permission.READ_EXTERNAL_STORAGE))

    {


        Toast.makeText(UploadPdf.this,"READ_EXTERNAL_STORAGE permission Access
Dialog", Toast.LENGTH_LONG).show();


    } else {


        ActivityCompat.requestPermissions(UploadPdf.this,new String[]{
Manifest.permission.READ_EXTERNAL_STORAGE}, 1);


    }

}
```

```
    @Override
    public void onRequestPermissionsResult(int RC, String per[], int[] Result) {

        switch (RC) {

            case 1:

                if (Result.length > 0 && Result[0] == PackageManager.PERMISSION_GRANTED) {

                    Toast.makeText(UploadPdf.this,"Permission Granted",
Toast.LENGTH_LONG).show();

                } else {

                    Toast.makeText(UploadPdf.this,"Permission Canceled",
Toast.LENGTH_LONG).show();

                }
                break;
        }
    }
}
```

Courses3.java

```
package com.example.lupp;


import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Bundle;import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.ProgressDialog;
import android.support.annotation.StringDef;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuInflater;
```

```
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;
```

```
public class Courses3 extends AppCompatActivity  implements View.OnClickListener{


    private EditText editText;

    public static final String PDF_FETCH_URL =
"http://192.168.43.247:80/AndroidUpload/getPdfs.php";

    //Image request code

    private int PICK_PDF_REQUEST = 1;

    //storage permission code

    private static final int STORAGE_PERMISSION_CODE = 123;

    //Uri to store the image uri

    private Uri filePath;

    //ListView to show the fetched Pdfs from the server

    ListView listView;

    //button to fetch the intiate the fetching of pdfs.

    Button buttonFetch;

    Button bict2;

    //Progress bar to check the progress of obtaining pdfs

    ProgressDialog progressDialog;

    //an array to hold the different pdf objects

    ArrayList<Pdf> pdfList= new ArrayList<Pdf>();

    //pdf adapter

    PdfAdapter pdfAdapter;
```

```
//menu
//menu
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()){
        case R.id.home:
            Intent intent2 = new Intent(Courses3.this, MainActivity.class);
            startActivity(intent2);
            return true;

        case R.id.admin:
            Intent intent1 = new Intent(Courses3.this, AdminLogin.class);
            startActivity(intent1);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
```

```
}


@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_courses3);

    //initializing ListView
    listView = (ListView) findViewById(R.id.listView);

    //initializing buttonFetch
    buttonFetch = (Button) findViewById(R.id.compsci);
    bict2 = (Button) findViewById(R.id.bict);
    //initializing progressDialog

    progressDialog = new ProgressDialog(this);

    //Setting clicklistener

    buttonFetch.setOnClickListener(this);
    bict2.setOnClickListener(this);///this crefers to implement onclick in current class
```

```
    //setting listView on item click listener
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {



            Pdf pdf = (Pdf) parent.getItemAtPosition(position);
            Intent intent = new Intent();
            intent.setAction(Intent.ACTION_VIEW);
            intent.addCategory(Intent.CATEGORY_BROWSABLE);
            intent.setData(Uri.parse(pdf.getUrl()));
            startActivity(intent);


        }
    });



}



/*
 * This is the method responsible for pdf upload
 * We need the full pdf path and the name for the pdf in this method
 * */
```

```
    //method to show file chooser



    //handling the ima chooser activity result
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);


        if (requestCode == PICK_PDF_REQUEST && resultCode == RESULT_OK && data != null && data.getData() != null) {
            filePath = data.getData();


        }
    }



    //Requesting permission



    //This method will be called when the user will tap on allow or deny
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
```

```
        //Checking the request code of our request
        if (requestCode == STORAGE_PERMISSION_CODE) {


            //If permission is granted
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                //Displaying a toast
                Toast.makeText(this, "Permission granted now you can read the storage",
Toast.LENGTH_LONG).show();
            } else {
                //Displaying another toast if permission is not granted
                Toast.makeText(this, "Oops you just denied the permission",
Toast.LENGTH_LONG).show();
            }
        }
    }



    @Override
    public void onClick(View v) {
        if(v==buttonFetch){
            getPdfs();
        }
        if (v == bict2){
            getbict2();
        }
```

```
    }
    //onebict fetchpdfs
    private void getbict2() {


        progressDialog.setMessage("Loading PastPapers... Please Wait");
        progressDialog.show();
        StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {



                        progressDialog.dismiss();
                        try {
                            JSONObject obj = new JSONObject(response);
                            Toast.makeText(Courses3.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();


                            JSONArray jsonArray = obj.getJSONArray("threebict");


                            for(int i=0;i<jsonArray.length();i++){
```

```
    //Declaring a json object corresponding to every pdf object in our json Array
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    //Declaring a Pdf object to add it to the ArrayList  pdfList
    Pdf pdf  = new Pdf();
    String pdfName = jsonObject.getString("PdfName");
    String pdfUrl = jsonObject.getString("PdfURL");
    String year = jsonObject.getString("year");
    String unitname = jsonObject.getString("unitname");
    String unitcode = jsonObject.getString("unitcode");
    String lecturer =  jsonObject.getString("lecturer");
    pdf.setName(pdfName);
    pdf.setUrl(pdfUrl);
    pdf.setYear(year);
    pdf.setUnitname(unitname);
    pdf.setUnitcode(unitcode);
    pdf.setLecturer(lecturer);
    pdfList.add(pdf);


}

pdfAdapter=new PdfAdapter(Courses3.this,R.layout.listlayout, pdfList);

listView.setAdapter(pdfAdapter);
```

```
                pdfAdapter.notifyDataSetChanged();

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    },

    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
    }
);

RequestQueue request = Volley.newRequestQueue(this);
request.add(stringRequest);

}

//onecompsci fetchpdf
private void getPdfs() {

progressDialog.setMessage("Loading PastPapers... Please Wait");
```

```
progressDialog.show();
StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {


                progressDialog.dismiss();
                try {
                    JSONObject obj = new JSONObject(response);
                    Toast.makeText(Courses3.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();

                    JSONArray jsonArray = obj.getJSONArray("threecompsci");

                    for(int i=0;i<jsonArray.length();i++){

                        //Declaring a json object corresponding to every pdf object in our json Array
                        JSONObject jsonObject = jsonArray.getJSONObject(i);
                        //Declaring a Pdf object to add it to the ArrayList  pdfList
                        Pdf pdf  = new Pdf();
                        String pdfName = jsonObject.getString("PdfName");
                        String pdfUrl = jsonObject.getString("PdfURL");
                        String year = jsonObject.getString("year");
```

```
            String unitname = jsonObject.getString("unitname");
            String unitcode = jsonObject.getString("unitcode");
            String lecturer =  jsonObject.getString("lecturer");
            pdf.setName(pdfName);
            pdf.setUrl(pdfUrl);
            pdf.setYear(year);
            pdf.setUnitname(unitname);
            pdf.setUnitcode(unitcode);
            pdf.setLecturer(lecturer);
            pdfList.add(pdf);



        }

        pdfAdapter=new PdfAdapter(Courses3.this,R.layout.listlayout, pdfList);

        listView.setAdapter(pdfAdapter);

        pdfAdapter.notifyDataSetChanged();

    } catch (JSONException e) {
        e.printStackTrace();
    }
```

```
            }
        },

        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        }
    );

    RequestQueue request = Volley.newRequestQueue(this);
    request.add(stringRequest);

  }

}
```

Course4.java

```
package com.example.lupp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Bundle;import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.ProgressDialog;
import android.support.annotation.StringDef;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
```

```
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.UUID;


public class Courses4 extends AppCompatActivity implements View.OnClickListener {

    private EditText editText;
```

```
    public static final String PDF_FETCH_URL =
"http://192.168.43.247:80/AndroidUpload/getPdfs.php";

    //Image request code

    private int PICK_PDF_REQUEST = 1;

    //storage permission code

    private static final int STORAGE_PERMISSION_CODE = 123;

    //Uri to store the image uri

    private Uri filePath;

    //ListView to show the fetched Pdfs from the server

    ListView listView;

    //button to fetch the intiate the fetching of pdfs.

    Button buttonFetch;

    Button bict2;

    //Progress bar to check the progress of obtaining pdfs

    ProgressDialog progressDialog;

    //an array to hold the different pdf objects

    ArrayList<Pdf> pdfList= new ArrayList<Pdf>();

    //pdf adapter

    PdfAdapter pdfAdapter;


    //menu

    //menu

    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        MenuInflater inflater = getMenuInflater();
```

```
    inflater.inflate(R.menu.menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()){
        case R.id.home:
            Intent intent2 = new Intent(Courses4.this, MainActivity.class);
            startActivity(intent2);
            return true;

        case R.id.admin:
            Intent intent1 = new Intent(Courses4.this, AdminLogin.class);
            startActivity(intent1);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}



@Override
protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_courses4);

//initializing ListView
listView = (ListView) findViewById(R.id.listView);

//initializing buttonFetch
buttonFetch = (Button) findViewById(R.id.compsci);
bict2 = (Button) findViewById(R.id.bict);
//initializing progressDialog

progressDialog = new ProgressDialog(this);

//Setting clicklistener

buttonFetch.setOnClickListener(this);
bict2.setOnClickListener(this);///this crefers to implement onclick in current class



//setting listView on item click listener
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
```

```
            Pdf pdf = (Pdf) parent.getItemAtPosition(position);
            Intent intent = new Intent();
            intent.setAction(Intent.ACTION_VIEW);
            intent.addCategory(Intent.CATEGORY_BROWSABLE);
            intent.setData(Uri.parse(pdf.getUrl()));
            startActivity(intent);

        }
    });


}



/*
 * This is the method responsible for pdf upload
 * We need the full pdf path and the name for the pdf in this method
 * */
```

```
    //method to show file chooser



    //handling the ima chooser activity result
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);


        if (requestCode == PICK_PDF_REQUEST && resultCode == RESULT_OK && data != null && data.getData() != null) {
            filePath = data.getData();


        }
    }



    //Requesting permission



    //This method will be called when the user will tap on allow or deny
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {


        //Checking the request code of our request
```

```
    if (requestCode == STORAGE_PERMISSION_CODE) {


        //If permission is granted

        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

            //Displaying a toast

            Toast.makeText(this, "Permission granted now you can read the storage",
Toast.LENGTH_LONG).show();

        } else {

            //Displaying another toast if permission is not granted

            Toast.makeText(this, "Oops you just denied the permission",
Toast.LENGTH_LONG).show();

        }

    }

}



@Override

public void onClick(View v) {

    if(v==buttonFetch){

        getPdfs();

    }

    if (v == bict2){

        getbict2();

    }
```

```
    }
    //onebict fetchpdfs
    private void getbict2() {


        progressDialog.setMessage("Loading PastPapers... Please Wait");
        progressDialog.show();
        StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {



                        progressDialog.dismiss();
                        try {
                            JSONObject obj = new JSONObject(response);
                            Toast.makeText(Courses4.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();


                            JSONArray jsonArray = obj.getJSONArray("fourbict");


                            for(int i=0;i<jsonArray.length();i++){


                                //Declaring a json object corresponding to every pdf object in our json Array
```

```
    JSONObject jsonObject = jsonArray.getJSONObject(i);
    //Declaring a Pdf object to add it to the ArrayList  pdfList
    Pdf pdf  = new Pdf();
    String pdfName = jsonObject.getString("PdfName");
    String pdfUrl = jsonObject.getString("PdfURL");
    String year = jsonObject.getString("year");
    String unitname = jsonObject.getString("unitname");
    String unitcode = jsonObject.getString("unitcode");
    String lecturer =  jsonObject.getString("lecturer");
    pdf.setName(pdfName);
    pdf.setUrl(pdfUrl);
    pdf.setYear(year);
    pdf.setUnitname(unitname);
    pdf.setUnitcode(unitcode);
    pdf.setLecturer(lecturer);
    pdfList.add(pdf);



}

pdfAdapter=new PdfAdapter(Courses4.this,R.layout.listlayout, pdfList);

listView.setAdapter(pdfAdapter);

pdfAdapter.notifyDataSetChanged();
```

```
                } catch (JSONException e) {
                    e.printStackTrace();
                }


            }
        },

        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        }
    );

    RequestQueue request = Volley.newRequestQueue(this);
    request.add(stringRequest);

}

//onecompsci fetchpdf
private void getPdfs() {
```

```
progressDialog.setMessage("Loading PastPapers... Please Wait");
progressDialog.show();
StringRequest stringRequest = new StringRequest(Request.Method.POST,
PDF_FETCH_URL,

        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {


                progressDialog.dismiss();
                try {
                    JSONObject obj = new JSONObject(response);
                    Toast.makeText(Courses4.this,obj.getString("message"),
Toast.LENGTH_SHORT).show();

                    JSONArray jsonArray = obj.getJSONArray("fourcompsci");

                    for(int i=0;i<jsonArray.length();i++){

                        //Declaring a json object corresponding to every pdf object in our json Array
                        JSONObject jsonObject = jsonArray.getJSONObject(i);
                        //Declaring a Pdf object to add it to the ArrayList  pdfList
                        Pdf pdf  = new Pdf();
                        String pdfName = jsonObject.getString("PdfName");
                        String pdfUrl = jsonObject.getString("PdfURL");
```

```
            String year = jsonObject.getString("year");
            String unitname = jsonObject.getString("unitname");
            String unitcode = jsonObject.getString("unitcode");
            String lecturer =  jsonObject.getString("lecturer");
            pdf.setName(pdfName);
            pdf.setUrl(pdfUrl);
            pdf.setYear(year);
            pdf.setUnitname(unitname);
            pdf.setUnitcode(unitcode);
            pdf.setLecturer(lecturer);
            pdfList.add(pdf);


        }

        pdfAdapter=new PdfAdapter(Courses4.this,R.layout.listlayout, pdfList);

        listView.setAdapter(pdfAdapter);

        pdfAdapter.notifyDataSetChanged();

    } catch (JSONException e) {
        e.printStackTrace();
    }
```

```
            }
        },

        new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        }
    );

    RequestQueue request = Volley.newRequestQueue(this);
    request.add(stringRequest);

  }

}
```

Constant,java

```
package com.example.lupp;

public class Constants {

  public class ServiceType {
```

```
    public static final String BASE_URL = "http://192.168.43.247/lupp/";//folder not db table
    public static final String LOGIN = BASE_URL + "simplelogin.php";
    public static final String ADMINLOGIN = BASE_URL + "adminlogin.php";
    public static final String REGISTER =  BASE_URL + "simpleregister.php";


  }
  // webservice key constants
  public class Params {

    public static final String NAME = "name";
    public static final String HOBBY = "hobby";
    public static final String USERNAME = "username";
    public static final String PASSWORD = "password";
  }
}
```

ParseContent.java

```
package com.example.lupp;

import android.app.Activity;

import org.json.JSONArray;
import org.json.JSONException;
```

```
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;

public class ParseContent {

    private final String KEY_SUCCESS = "status";
    private final String KEY_MSG = "message";
    private final String KEY_AddressList = "addressList";
    private final String KEY_DATA = "Data";
    private ArrayList<HashMap<String, String>> hashMap;
    private Activity activity;
    PreferenceHelper preferenceHelper;

    ArrayList<HashMap<String, String>> arraylist;

    public ParseContent(Activity activity) {
        this.activity = activity;
        preferenceHelper = new PreferenceHelper(activity);

    }

    public boolean isSuccess(String response) {
        try {
```

```
        JSONObject jsonObject = new JSONObject(response);
        if (jsonObject.optString(KEY_SUCCESS).equals("true")) {
            return true;
        } else {

            return false;
        }

    } catch (JSONException e) {
        e.printStackTrace();
    }
    return false;
}

public String getErrorMessage(String response) {
    try {
        JSONObject jsonObject = new JSONObject(response);
        return jsonObject.getString(KEY_MSG);

    } catch (JSONException e) {
        e.printStackTrace();
    }
    return "No data";
}
```

```
    public void saveInfo(String response) {
        preferenceHelper.putIsLogin(true);
        try {
            JSONObject jsonObject = new JSONObject(response);
            if (jsonObject.getString(KEY_SUCCESS).equals("true")) {
                JSONArray dataArray = jsonObject.getJSONArray("data");
                for (int i = 0; i < dataArray.length(); i++) {

                    JSONObject dataobj = dataArray.getJSONObject(i);
                    preferenceHelper.putName(dataobj.getString(Constants.Params.NAME));
                    preferenceHelper.putHobby(dataobj.getString(Constants.Params.HOBBY));
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
}
```

PreferenceHelper.java

```
package com.example.lupp;

import android.content.Context;
```

```
import android.content.SharedPreferences;

public class PreferenceHelper {

    private final String INTRO = "intro";
    private final String NAME = "name";
    private final String HOBBY = "hobby";
    private SharedPreferences app_prefs;
    private Context context;

    public PreferenceHelper(Context context) {
        app_prefs = context.getSharedPreferences("shared",
                Context.MODE_PRIVATE);
        this.context = context;
    }

    public void putIsLogin(boolean loginorout) {
        SharedPreferences.Editor edit = app_prefs.edit();
        edit.putBoolean(INTRO, loginorout);
        edit.commit();
    }
    public boolean getIsLogin() {
        return app_prefs.getBoolean(INTRO, false);
    }
```

```
    public void putName(String loginorout) {
        SharedPreferences.Editor edit = app_prefs.edit();
        edit.putString(NAME, loginorout);
        edit.commit();
    }
    public String getName() {
        return app_prefs.getString(NAME, "");
    }

    public void putHobby(String loginorout) {
        SharedPreferences.Editor edit = app_prefs.edit();
        edit.putString(HOBBY, loginorout);
        edit.commit();
    }
    public String getHobby() {
        return app_prefs.getString(HOBBY, "");
    }

}
```

APPENDIX 3:TEST DATA

Id	Description	Test Data	Expected results	Actual results
1	Student registration	Name:Nancy Username:nancyk Together with registration number and password.	Registration successful. Specific information is stored in lupp table in admin database.	Registration successful
2	Student Log in		Allows student registered into the system	Login successful
3	Librarian login	Name&Password is entered	Allows librarian with a specific name and password	Permission granted
4	Uploading	Librarian uploads pdf	Librarian upload past paper in pdf format	Upload completed successfully
5	Download	Specific student download pdf required.	Student download past papers stored in database	Complete action with