

|  |
| --- |
| Software Engineering |
| Term project (Documentation) |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 제출일 | 2020.07.01 |  | 전공 | 전기공학과  소프트웨어학과  소프트웨어학과 |
| 과목 | 소프트웨어공학 |  | 학번 | 201636417  201835519  201835461 |
| 담당교수 | 정옥란 |  | 이름 | 심우석  전수환  박한음 |

Table of content

1. **Introduction**

1.1 Purpose of the system

1.2 Scope of the system

1.3 Objectives and success criteria of the project

1.4 Overview

1. **Requirement analysis**

2.1 Functional Requirements

2.2 Nonfunctional Requirements

1. **System modeling**

3.1 Scenario

3.2 Interaction models

3.3 Structure models

1. **Architectural Design**
2. **Implementation**

5.1 Development Environment

5.2 Code

1. **Testing**

6.1 Configuration Management

6.2 Unit Testing

6.3 Test Case Generation

6.4 Test Coverage analysis

6.5 Performance Testing

1. **Glossary**
2. **Introduction**

**1.1 Purpose of the system**

We thought we needed a library management program to keep a lot of books in the library and make them available to users efficiently.

This program should be readily available to both library users and administrators. For example, a user should be able to check the location of a book, and whether the book to be borrowed is available for loan.

Administrators will need to check the location of the book to return the returned book to its original location, also check overdue books for use by other users. Also, if we include the function to read or study books in the reading room, both users and administrators can be more convenient libraries

**1.2 Scope of the system**

We aim to create a library management program that is available to both small and large libraries. In a small library, the program is used as it is, and in a large library, the interoperability of the program is enhanced through database linkage with other libraries.

For example, you tried to borrow a book called "ABC" from the library closest to your house, but there may be no books in the library in front of your house. At that time, you can find a library close to the book or check whether the book is in a specific library

**1.3 Objectives and success criteria of the project**

The first goal is to create programs that can be used in small libraries. And the second goal is to connect databases to enable interoperability between multiple libraries.

And the code of the program should be designed without error and there should be no safety problem when the program is executed.

**1.4 Overview**

People who use the program are divided into users and administrators. And the use section is divided into library and reading room.

Through the program, the user checks if the book is available for loan now, Where the book is located, and you can search how many books are currently available for loan. You can also borrow or return books through a program or administrators.

The administrator will enter the book's unique number into the program and check how many books are currently overdue.

In addition, the user can use a program such as which seat is empty in the reading room, reservation of a seat, and the like.

**2. Requirement**

**2.1 Functional Requirements**

**1) Sign up and Login (customer and administrator)**

Administrators can log in with a set ID without having to sign up for membership.

Customers can create their own account through membership, and after log in, can use the application.

**2) Member Information Management (administrator)**

After logging in to the program, administrators can view member information.

**3) Reserve seat of reading room (customer)**

If the customer wants to use the reading room, customer can choose the available seats and reserve them.

**4) Add and Delete books (administrator)**

Administrators can change the data in the book database by adding or deleting books in the library. When adding a book, they must also enter the book information, such as the book name, book type, year of publication, and unique number of the book.

**5) Loan and Return books (customer)**

Customer can loan or return books. When borrowing/returning, it is entered and managed in the book database.

**6) Search books (customer)**

Customer can search for the books they want through keywords such as book type, year of publication. In addition, they can get the location, information of books by searching.

**2.2 Nonfunctional Requirements**

**Usability**

It will organize a simple UI so that users can see the program functions. The program will be implemented in the form of Java-enabled applications in Android studios.

First, the window that appears in the application is different by separating the customer and the manager through membership and login. When customers log in, they can use the functions of searching, lending, returning, and booking seat. And when administrators log in, they can add or delete books and check the users.

It also manages program storage by connecting databases.

**Reliability**

The programs used by users and administrators in this program are separate from each other. Users should be able to process and book quickly at any time when searching for a book they want. And only the administrator can check the user's information. Other users cannot see. Book changes are properly reflected in the database without errors.

**Performance**

In this program, the system functions of the implementation process are very important. You have to show the information about the book you're looking for in less than two seconds. The list of books searched must be visible within five seconds. Also, when a user borrows and returns a book, and when an administrator adds and deletes a book, the book database must immediately reflect the changes.

**3. System modeling**

**3.1 Scenario**

Before the user logs in to the app, sign-up is executed within the app. When signing up for a membership, the information entered is recorded in its own DB and the registration is completed. After completing the membership, enter the main screen and search for books in the library using the desired category. After searching, apply for a book loan, select a seat in the library and make a reservation.

When a manager logs in to an app with a set ID and password, the administrator mode starts. Press the Add Book button on the main screen to check the list of books currently registered and enter the new book into the app to register. Also, click the Delete Book button to check the list of books currently registered, and select and delete existing books. Finally, click the Manage button to see a list of people currently enrolled in the program.

**3.2 Interaction models**

**텍스트, 지도이(가) 표시된 사진

자동 생성된 설명**

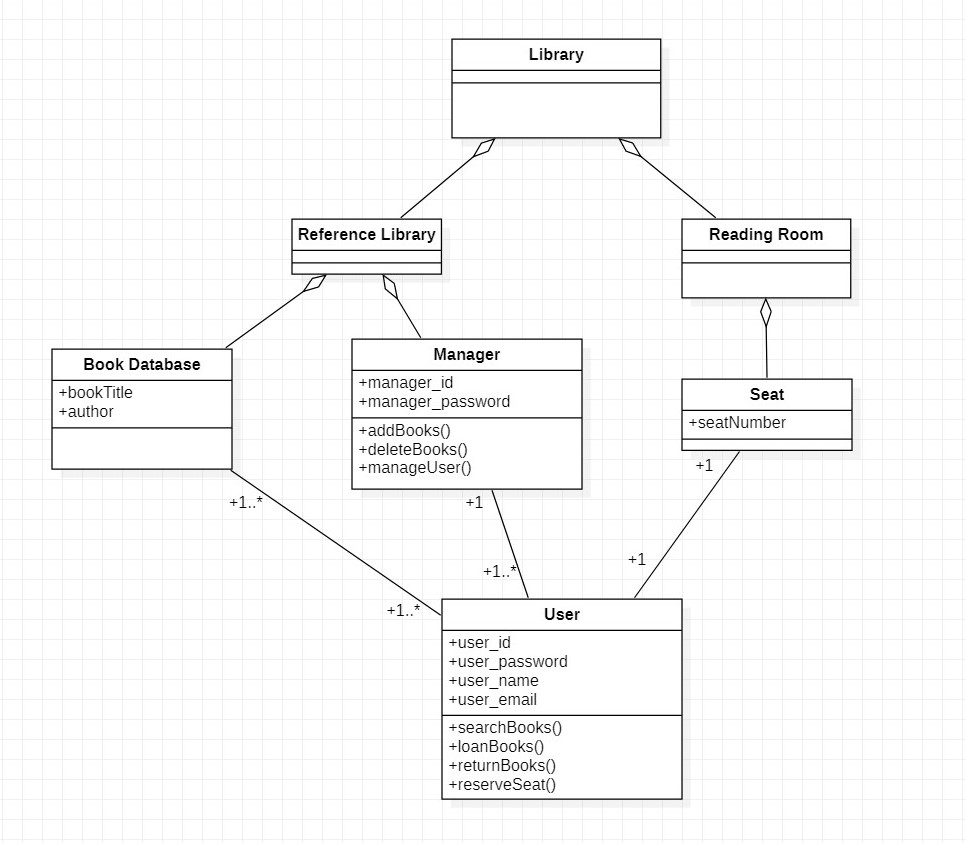
The above figure shows the model of the book management system divided by the use-case diagram. Users perform information verification, book search and reservation with a management system, and managers manage, search and delete and add book lists for users. In addition, information on each book and user is stored in the database.

지도, 테이블, 책상, 전체이(가) 표시된 사진

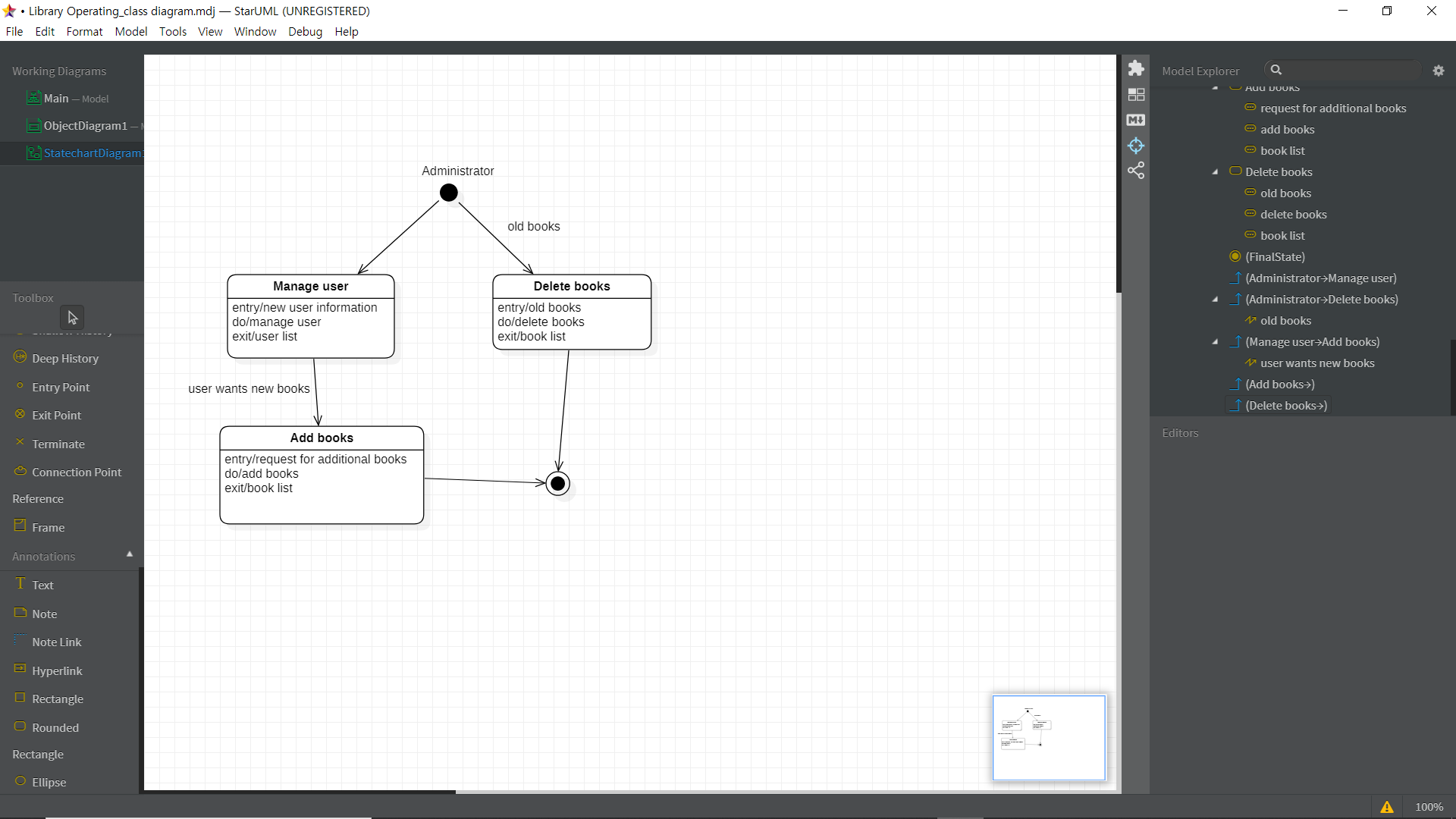
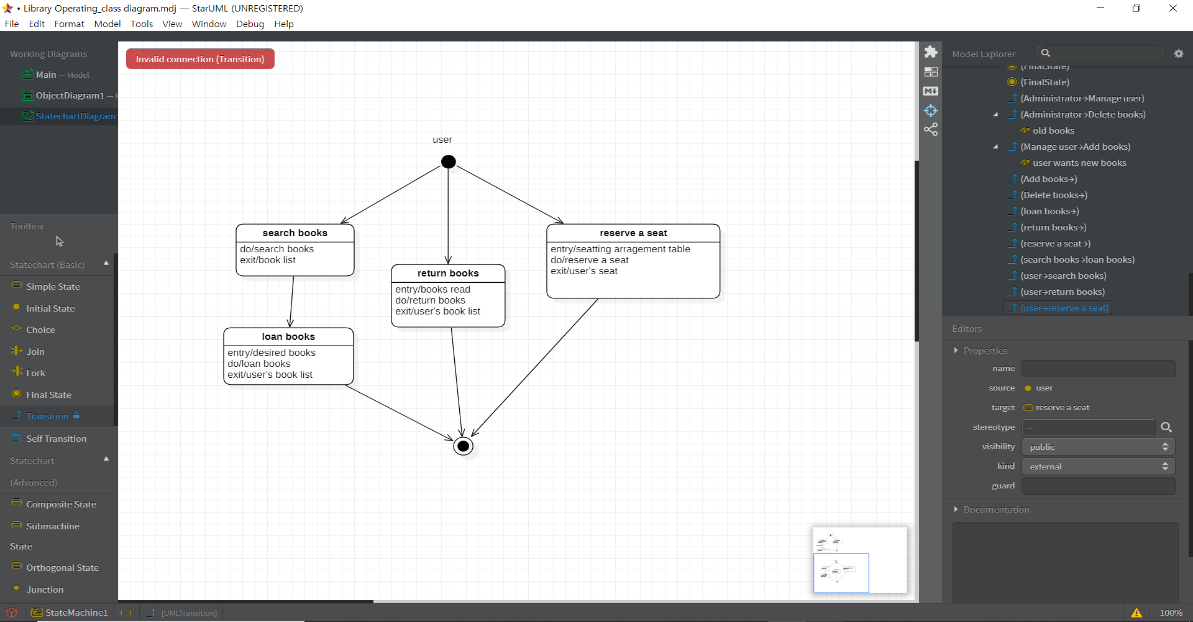
자동 생성된 설명

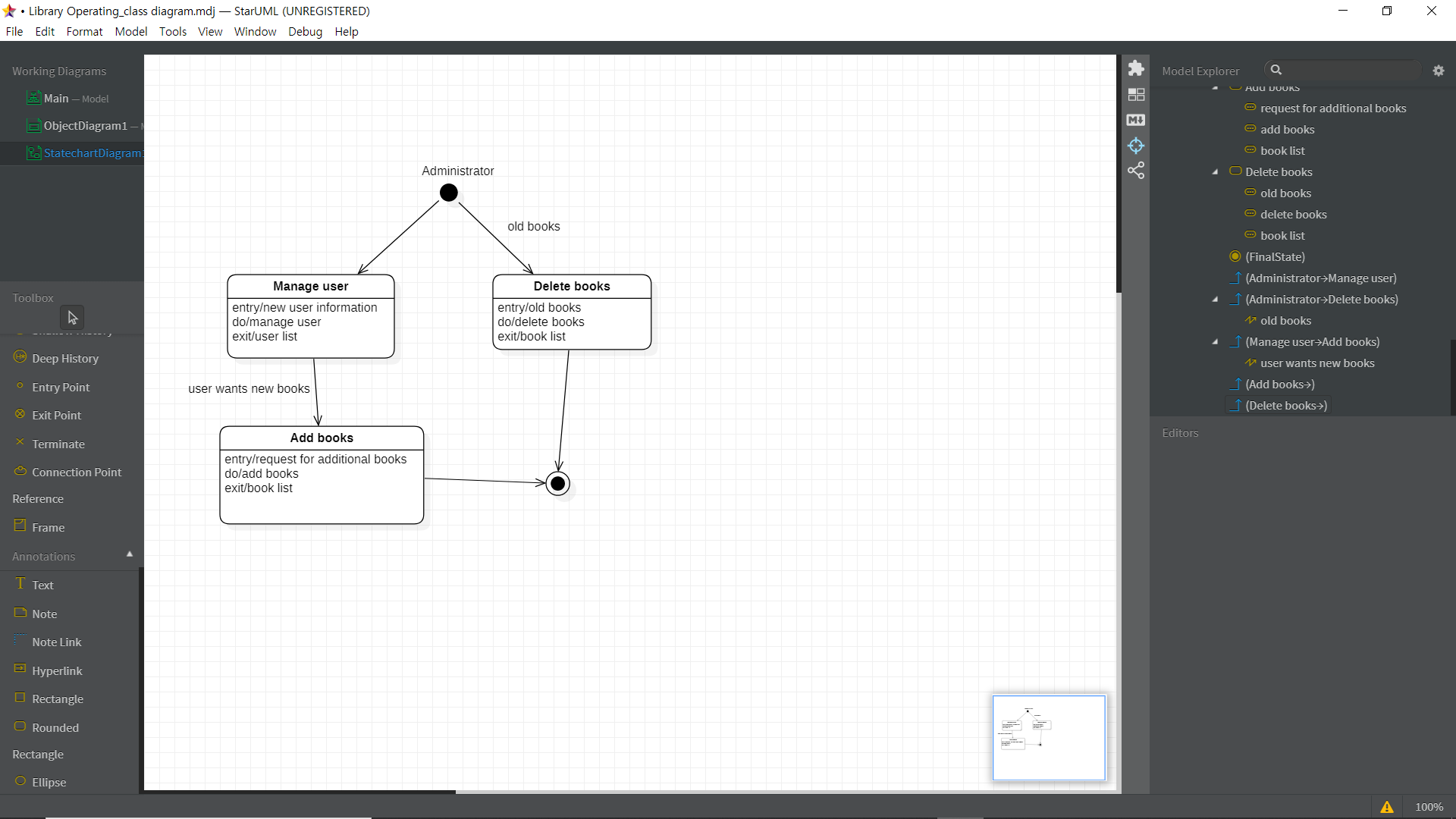
The above figure shows the book management system model as an order diagram. Users find books, obtain information from books, apply for loans, and administrators access the book list to delete or add books

**3.3 Structure models**

****

The above figure shows a structure model created using the class diagram. Attributes and behaviors were organized around managers and users.



The above two figures show the roles of both the user and the manager using a status diagram.

**4. Architecture design**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

As shown in the figure above, layered architecture patterns are designed to enhance security. In the library operation app, layer was divided into configuration services, application services, and utility services to assign roles suitable for each service.

스크린샷이(가) 표시된 사진

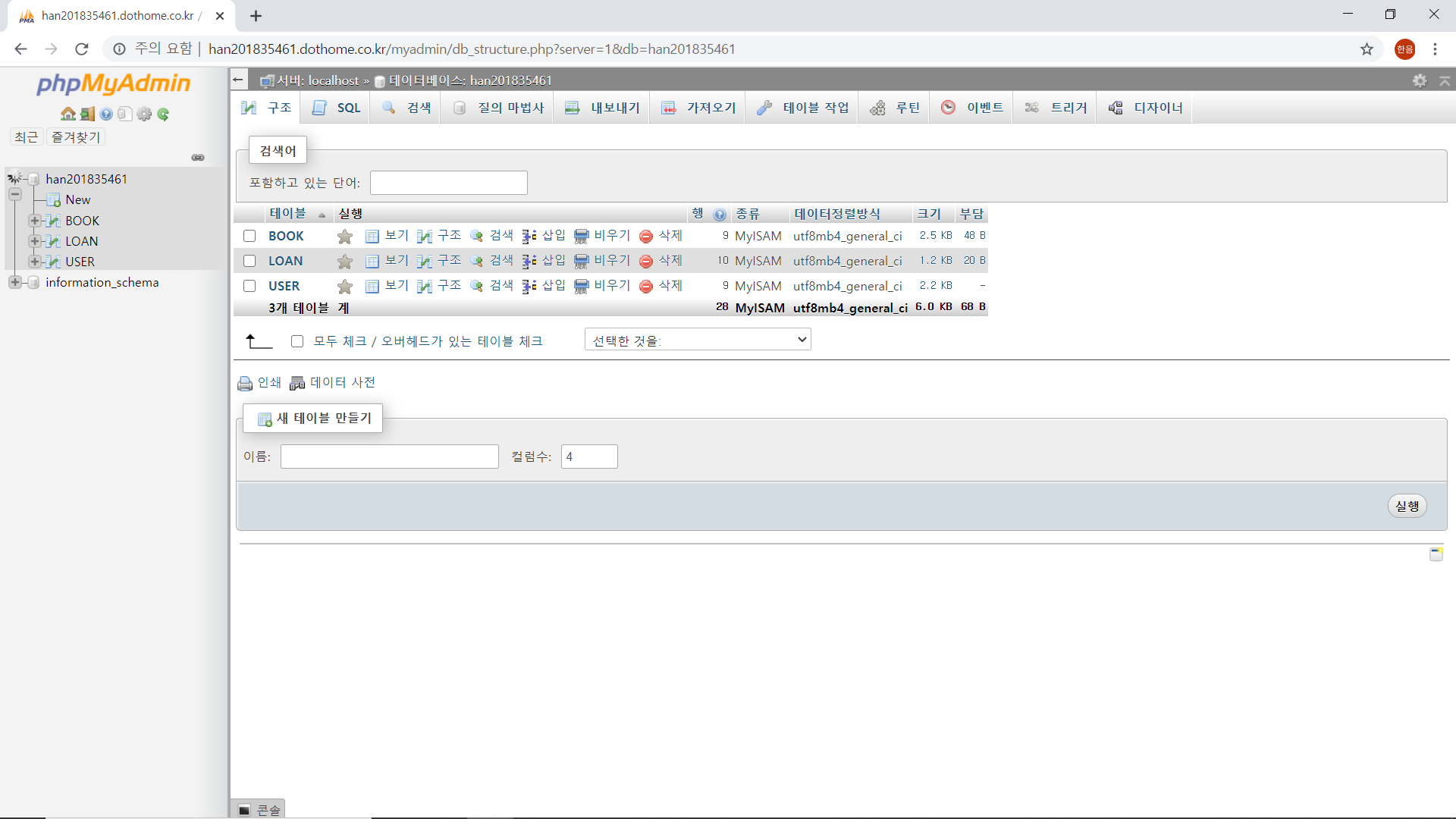
자동 생성된 설명

The operational program be layered on top of the introduction and several users using the app, app programs and services database and the library. Use a database and a web server interface with the services that design.

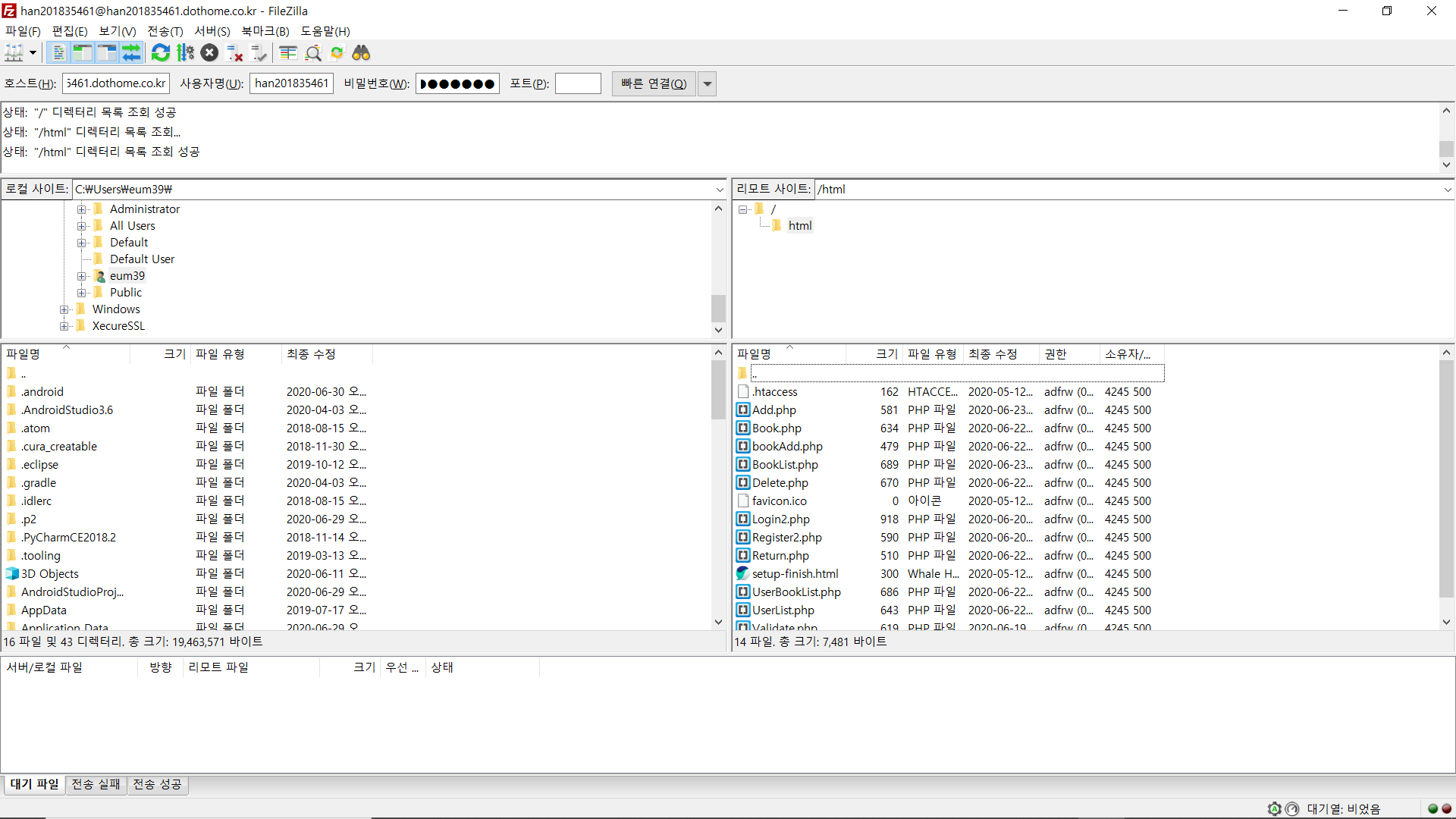
**5. Implementation**

**5.1 Development Enviroment**

We used the Android studio to make an application. The language used Java. And to implement the library program, we need a database, but Android studios don't support database drivers for security reasons, so we decided to build our own web server. We have received domains from sites that offer free hosting services to build Web servers. We built a database from that domain address and made tables for users, books, and loan lists. And we created a php-type web server program file to access the database. Then we installed a program called FileZilla and put in a php file so that the application and the web server could be connected.



This picture is our domain site.



This picture is a FileZilla program with a php file.

**5.2 Code**

## AndroidManifest.xml

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.serverconnection">   <uses-permission android:name="android.permission.INTERNET" />  <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />   <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round"  android:supportsRtl="true"  android:theme="@style/AppTheme"  android:usesCleartextTraffic="true">  <activity android:name=".LoginActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />   <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  <activity android:name=".RegisterActivity"/>  <activity android:name=".MainActivity"/>  <activity android:name=".MainActivity2"/>  <activity android:name=".AddActivity"/>  </application>  </manifest> |

## Build.gradle (:app)

|  |
| --- |
| apply plugin: 'com.android.application'  android **{** compileSdkVersion 29  buildToolsVersion "29.0.3"   defaultConfig **{** applicationId "com.example.serverconnection"  minSdkVersion 16  targetSdkVersion 29  versionCode 1  versionName "1.0"   testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  **}** buildTypes **{** release **{** minifyEnabled false  proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  **}  }  }** dependencies **{** implementation fileTree(dir: 'libs', include: ['\*.jar'])   implementation 'androidx.appcompat:appcompat:1.1.0'  implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  implementation 'com.android.volley:volley:1.1.1'  testImplementation 'junit:junit:4.12'  androidTestImplementation 'androidx.test.ext:junit:1.1.1'  androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0' **}** |

## AddActivity.java //새로운 도서를 추가하는 activity

|  |
| --- |
| package com.example.serverconnection;  import android.content.Intent; import android.os.Bundle; import android.view.MotionEvent; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity;  import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.toolbox.Volley;  import org.json.JSONException; import org.json.JSONObject;  public class AddActivity extends AppCompatActivity {   private EditText idText, nameText, typeText, yearText;  private Button addBtn;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_add*);   idText = findViewById(R.id.*idText*);  nameText = findViewById(R.id.*nameText*);  typeText = findViewById(R.id.*typeText*);  yearText = findViewById(R.id.*yearText*);   addBtn = findViewById(R.id.*addBtn*);  addBtn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  int bookID = Integer.*parseInt*(idText.getText().toString());  String bookName = nameText.getText().toString();  String bookType = typeText.getText().toString();  int bookYear = Integer.*parseInt*(yearText.getText().toString());   Response.Listener<String> responseListener = new Response.Listener<String>() {  @Override  public void onResponse(String response) {  try {  JSONObject jsonObject = new JSONObject(response);  boolean success = jsonObject.getBoolean("success");  if(success) {  Toast.*makeText*(getApplicationContext(), "Add Book Completed", Toast.*LENGTH\_SHORT*).show();  }  else {  Toast.*makeText*(getApplicationContext(), "Add Book Failed", Toast.*LENGTH\_SHORT*).show();  return;  }  } catch (JSONException e) {  e.printStackTrace();  }  }  };   AddRequest addRequest = new AddRequest(bookID, bookName, bookType, bookYear, responseListener);  RequestQueue queue = Volley.*newRequestQueue*(AddActivity.this);  queue.add(addRequest);   }  });   }  } |

## AddAdapter.java

|  |
| --- |
| package com.example.serverconnection;  import android.content.Context; import android.view.View; import android.view.ViewGroup; import android.widget.BaseAdapter; import android.widget.TextView;  import androidx.fragment.app.Fragment;  import java.util.List;  public class AddAdapter extends BaseAdapter {   private Context context;  private List<Book> bookList;  private Fragment parent;   public AddAdapter (Context context, List<Book> bookList, Fragment parent) {  this.context = context;  this.bookList = bookList;  this.parent = parent;  }  @Override  public int getCount() {  return bookList.size();  }   @Override  public Object getItem(int i) {  return bookList.get(i);  }   @Override  public long getItemId(int i) {  return i;  }   @Override  public View getView(final int i, View view, ViewGroup viewGroup) {  View v = View.*inflate*(context, R.layout.*booklist*, null);  TextView bookID = (TextView) v.findViewById(R.id.*id*);  TextView bookName = (TextView) v.findViewById(R.id.*name*);  TextView bookType = (TextView) v.findViewById(R.id.*type*);  TextView bookYear = (TextView) v.findViewById(R.id.*year*);   bookID.setText(bookList.get(i).getBookID()+"");  bookName.setText(bookList.get(i).getBookName());  bookType.setText(bookList.get(i).getBookType());  bookYear.setText(bookList.get(i).getBookYear()+"");   return v;  } } |

## AddRequest.java

|  |
| --- |
| package com.example.serverconnection;  import com.android.volley.AuthFailureError; import com.android.volley.Response; import com.android.volley.toolbox.StringRequest;  import java.util.HashMap; import java.util.Map;  public class AddRequest extends StringRequest {   //서버 URL 설정 (PHP 파일 연동)http://han201835461.dothome.co.kr/myadmin  final static private String *URL* = "http://han201835461.dothome.co.kr/Add.php";  private Map<String, String> map;   public AddRequest(int bookID, String bookName, String bookType, int bookYear, Response.Listener<String> listener) {  super(Method.*POST*, *URL*, listener, null);   map = new HashMap<>();  map.put("bookID", bookID+"");  map.put("bookName", bookName);  map.put("bookType", bookType);  map.put("bookYear", bookYear+"");  }   @Override  protected Map<String, String> getParams() throws AuthFailureError {  return map;  } } |

## AddFragment.java

|  |
| --- |
| package com.example.serverconnection;  import android.app.AlertDialog; import android.content.Intent; import android.net.Uri; import android.os.AsyncTask; import android.os.Bundle; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.EditText; import android.widget.ListView; import android.widget.Spinner; import android.widget.Toast;  import androidx.fragment.app.Fragment;  import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.toolbox.Volley;  import org.json.JSONArray; import org.json.JSONException; import org.json.JSONObject;  import java.io.BufferedReader; import java.io.InputStream; import java.io.InputStreamReader; import java.net.HttpURLConnection; import java.net.URL; import java.net.URLEncoder; import java.util.ArrayList; import java.util.List;  public class AddFragment extends Fragment {   private ListView bookListView;  private AddAdapter adapter;  private List<Book> bookList;  private Button addButton;   @Override  public void onActivityCreated (Bundle b) {  super.onActivityCreated(b);   super.onActivityCreated(b);  bookListView = (ListView) getView().findViewById(R.id.*bookListView*);  bookList = new ArrayList<Book>();  adapter = new AddAdapter(getContext().getApplicationContext(), bookList, this);  bookListView.setAdapter(adapter);   addButton = getView().findViewById(R.id.*addButton*);  addButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent(AddFragment.this.getActivity(), AddActivity.class);  startActivity(intent);  }  });   new AddFragment.BackgroundTask().execute();   }   class BackgroundTask extends AsyncTask<Void, Void, String>  {  String target;   @Override  protected void onPreExecute() {  try {  target="http://han201835461.dothome.co.kr/Book.php";  } catch (Exception e) {  e.printStackTrace();  }  }   @Override  protected String doInBackground(Void... voids) {  try {  URL url = new URL(target);  HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();  InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = new BufferedReader (new InputStreamReader(inputStream));  String temp;  StringBuilder stringBuilder = new StringBuilder();  while((temp = bufferedReader.readLine()) != null)  {  stringBuilder.append(temp + "\n");  }  bufferedReader.close();  inputStream.close();  httpURLConnection.disconnect();  return stringBuilder.toString().trim();  } catch (Exception e) {  e.printStackTrace();  }   return null;  }   @Override  public void onProgressUpdate(Void... values) { super.onProgressUpdate();}   @Override  public void onPostExecute(String result) {  try {  bookList.clear();  JSONObject jsonObject = new JSONObject(result);  JSONArray jsonArray = jsonObject.getJSONArray("response");   int count = 0;  int bookID;  String bookName;  String bookType;  int bookYear;   while(count < jsonArray.length())  {  JSONObject object = jsonArray.getJSONObject(count);  bookID = object.getInt("bookID");  bookName = object.getString("bookName");  bookType = object.getString("bookType");  bookYear = object.getInt("bookYear");   Book book = new Book(bookID, bookName, bookType, bookYear);  bookList.add(book);  count++;  }   adapter.notifyDataSetChanged();  } catch (Exception e) {  e.printStackTrace();  }  }  }   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  return inflater.inflate(R.layout.*fragment\_add*,container,false);  } } |

## AddBookRequest.java

|  |
| --- |
| package com.example.serverconnection;  import com.android.volley.AuthFailureError; import com.android.volley.Response; import com.android.volley.toolbox.StringRequest;  import java.util.HashMap; import java.util.Map;  public class AddBookRequest extends StringRequest {   //서버 URL 설정 (PHP 파일 연동)http://han201835461.dothome.co.kr/myadmin  final static private String *URL* = "http://han201835461.dothome.co.kr/bookAdd.php";  private Map<String, String> map;   public AddBookRequest(String userID, String bookID, Response.Listener<String> listener) {  super(Method.*POST*, *URL*, listener, null);   map = new HashMap<>();  map.put("userID", userID);  map.put("bookID", bookID);  }   @Override  protected Map<String, String> getParams() throws AuthFailureError {  return map;  } } |

## Book.java

|  |
| --- |
| package com.example.serverconnection;  public class Book {  int bookID;  String bookName;  String bookType;  int bookYear;   public int getBookID() {  return bookID;  }   public void setBookID(int bookID) {  this.bookID = bookID;  }   public String getBookName() {  return bookName;  }   public void setBookName(String bookName) {  this.bookName = bookName;  }   public String getBookType() {  return bookType;  }   public void setBookType(String bookType) {  this.bookType = bookType;  }   public int getBookYear() {  return bookYear;  }   public void setBookYear(int bookYear) {  this.bookYear = bookYear;  }   public Book(int bookID, String bookName, String bookType, int bookYear) {  this.bookID = bookID;  this.bookName = bookName;  this.bookType = bookType;  this.bookYear = bookYear;  } } |

## BookFragment.java

|  |
| --- |
| package com.example.serverconnection;  import android.app.AlertDialog; import android.os.AsyncTask; import android.os.Bundle; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ListView; import android.widget.Spinner;  import androidx.fragment.app.Fragment;  import org.json.JSONArray; import org.json.JSONObject;  import java.io.BufferedReader; import java.io.InputStream; import java.io.InputStreamReader; import java.io.UnsupportedEncodingException; import java.net.HttpURLConnection; import java.net.MalformedURLException; import java.net.URL; import java.net.URLEncoder; import java.util.ArrayList; import java.util.List;  public class BookFragment extends Fragment {   private ArrayAdapter typeAdapter;  private Spinner typeSpinner;  private ArrayAdapter yearAdapter;  private static Spinner *yearSpinner*;   private String bookType = "";  private String bookYear = "";   private ListView bookListView;  private BookListAdapter adapter;  private List<Book> bookList;   @Override  public void onActivityCreated (Bundle b) {  super.onActivityCreated(b);   typeSpinner = (Spinner) getView().findViewById(R.id.*bookTypeSpinner*);  *yearSpinner* = (Spinner) getView().findViewById(R.id.*bookYearSpinner*);   typeAdapter = ArrayAdapter.*createFromResource*(getActivity(), R.array.*type*, android.R.layout.*simple\_spinner\_dropdown\_item*);  typeSpinner.setAdapter(typeAdapter);   yearAdapter = ArrayAdapter.*createFromResource*(getActivity(), R.array.*year*, android.R.layout.*simple\_spinner\_dropdown\_item*);  *yearSpinner*.setAdapter(yearAdapter);   bookListView = (ListView) getView().findViewById(R.id.*bookListView*);  bookList = new ArrayList<Book>();  adapter = new BookListAdapter(getContext().getApplicationContext(), bookList, this);  bookListView.setAdapter(adapter);   Button searchButton = (Button)getView().findViewById(R.id.*searchButton*);  searchButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  new BackgroundTask().execute();  }  });  }   class BackgroundTask extends AsyncTask<Void, Void, String>  {  String target;   @Override  protected void onPreExecute() {  try {  target="http://han201835461.dothome.co.kr/BookList.php?bookType=" + URLEncoder.*encode*(typeSpinner.getSelectedItem().toString(), "UTF-8") +  "&bookYear=" + URLEncoder.*encode*(*yearSpinner*.getSelectedItem().toString().substring(0,4), "UTF-8");  } catch (Exception e) {  e.printStackTrace();  }  }   @Override  protected String doInBackground(Void... voids) {  try {  URL url = new URL(target);  HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();  InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = new BufferedReader (new InputStreamReader(inputStream));  String temp;  StringBuilder stringBuilder = new StringBuilder();  while((temp = bufferedReader.readLine()) != null)  {  stringBuilder.append(temp + "\n");  }  bufferedReader.close();  inputStream.close();  httpURLConnection.disconnect();  return stringBuilder.toString().trim();  } catch (Exception e) {  e.printStackTrace();  }   return null;  }   @Override  public void onProgressUpdate(Void... values) { super.onProgressUpdate();}   @Override  public void onPostExecute(String result) {  try {  bookList.clear();  JSONObject jsonObject = new JSONObject(result);  JSONArray jsonArray = jsonObject.getJSONArray("response");   int count = 0;  int bookID;  String bookName;  String bookType;  int bookYear;   while(count < jsonArray.length())  {  JSONObject object = jsonArray.getJSONObject(count);  bookID = object.getInt("bookID");  bookName = object.getString("bookName");  bookType = object.getString("bookType");  bookYear = object.getInt("bookYear");   Book book = new Book(bookID, bookName, bookType, bookYear);  bookList.add(book);  count++;  }   if(count==0)  {  AlertDialog dialog;  AlertDialog.Builder builder = new AlertDialog.Builder(BookFragment.this.getActivity());  dialog = builder.setMessage("There is no book that meets the criteria.").setPositiveButton("check",null).create();  dialog.show();  }  adapter.notifyDataSetChanged();  } catch (Exception e) {  e.printStackTrace();  }  }  }   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  return inflater.inflate(R.layout.*fragment\_book*,container,false);  } } |

## BookListAdapter.java

|  |
| --- |
| package com.example.serverconnection;  import android.app.AlertDialog; import android.content.Context; import android.content.Intent; import android.view.View; import android.view.ViewGroup; import android.widget.BaseAdapter; import android.widget.Button; import android.widget.TextView; import android.widget.Toast;  import androidx.fragment.app.Fragment; import androidx.fragment.app.FragmentTransaction;  import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.toolbox.Volley;  import org.json.JSONException; import org.json.JSONObject;  import java.util.List;  public class BookListAdapter extends BaseAdapter {   private Context context;  private List<Book> bookList;  private Fragment parent;   public BookListAdapter (Context context, List<Book> bookList, Fragment parent) {  this.context = context;  this.bookList = bookList;  this.parent = parent;  }  @Override  public int getCount() {  return bookList.size();  }   @Override  public Object getItem(int i) {  return bookList.get(i);  }   @Override  public long getItemId(int i) {  return i;  }   @Override  public View getView(final int i, View view, ViewGroup viewGroup) {  View v = View.*inflate*(context, R.layout.*book*, null);  TextView bookType = (TextView) v.findViewById(R.id.*type*);  TextView bookName = (TextView) v.findViewById(R.id.*name*);  TextView bookYear = (TextView) v.findViewById(R.id.*year*);   bookType.setText(bookList.get(i).getBookType());  bookName.setText(bookList.get(i).getBookName());  bookYear.setText(bookList.get(i).getBookYear()+"");   v.setTag(bookList.get(i).getBookID());   Button loanButton = (Button) v.findViewById(R.id.*loanBtn*);  loanButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  String userID = MainActivity.*userID*;  Response.Listener<String> responseListener = new Response.Listener<String>() {  @Override  public void onResponse(String response) {  try {  JSONObject jsonObject = new JSONObject(response);  boolean success = jsonObject.getBoolean("success");  if(success) {  AlertDialog.Builder builder = new AlertDialog.Builder(parent.getActivity());  AlertDialog dialog = builder.setMessage("You loan the book!").setPositiveButton("check", null).create();  dialog.show();  }  else {  AlertDialog.Builder builder = new AlertDialog.Builder(parent.getActivity());  AlertDialog dialog = builder.setMessage("loan failure").setPositiveButton("check", null).create();  dialog.show();  }  } catch (JSONException e) {  e.printStackTrace();  }  }  };  AddBookRequest addBookRequest = new AddBookRequest(userID, bookList.get(i).getBookID()+"", responseListener);  RequestQueue queue = Volley.*newRequestQueue*(parent.getActivity());  queue.add(addBookRequest);   }  });   return v;  } } |

## DeleteAdapter.java

|  |
| --- |
| private Context context; private List<Book> bookList; private Fragment parent;  public DeleteAdapter(Context context, List<Book> bookList, Fragment parent) {  this.context = context;  this.bookList = bookList;  this.parent = parent; }  @Override public int getCount() {  return bookList.size(); }  @Override public Object getItem(int i) {  return bookList.size(); }  @Override public long getItemId(int i) {  return i; }  @Override public View getView(final int i, View view, ViewGroup viewGroup) {  View v = View.*inflate*(context, R.layout.*book\_delete*, null);  TextView bookID = (TextView) v.findViewById(R.id.*id*);  TextView bookType = (TextView) v.findViewById(R.id.*type*);  TextView bookName = (TextView) v.findViewById(R.id.*name*);  TextView bookYear = (TextView) v.findViewById(R.id.*year*);   bookID.setText(bookList.get(i).getBookID()+"");  bookType.setText(bookList.get(i).getBookType());  bookName.setText(bookList.get(i).getBookName());  bookYear.setText(bookList.get(i).getBookYear()+"");   v.setTag(bookList.get(i).getBookID());   Button deleteButton = (Button) v.findViewById(R.id.*deleteBtn*);  deleteButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Response.Listener<String> responseListener = new Response.Listener<String>() {  @Override  public void onResponse(String response) {  try {  JSONObject jsonObject = new JSONObject(response);  boolean success = jsonObject.getBoolean("success");  if(success) {  AlertDialog.Builder builder = new AlertDialog.Builder(parent.getActivity());  AlertDialog dialog = builder.setMessage("The book deleted!").setPositiveButton("check", null).create();  dialog.show();  bookList.remove(i);  notifyDataSetChanged();  }  else {  AlertDialog.Builder builder = new AlertDialog.Builder(parent.getActivity());  AlertDialog dialog = builder.setMessage("Delete failure").setNegativeButton("retry", null).create();  dialog.show();  }  } catch (JSONException e) {  e.printStackTrace();  }  }  };  DeleteRequest deleteRequest = new DeleteRequest(bookList.get(i).getBookID()+"", bookList.get(i).getBookName(), bookList.get(i).getBookType(), bookList.get(i).getBookYear()+"", responseListener);  RequestQueue queue = Volley.*newRequestQueue*(parent.getActivity());  queue.add(deleteRequest);  }  });   return v; } |

## DeleteFragment.java

|  |
| --- |
| package com.example.serverconnection;  import android.os.AsyncTask; import android.os.Bundle; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ListView;  import androidx.fragment.app.Fragment;  import org.json.JSONArray; import org.json.JSONObject;  import java.io.BufferedReader; import java.io.InputStream; import java.io.InputStreamReader; import java.net.HttpURLConnection; import java.net.URL; import java.net.URLEncoder; import java.util.ArrayList; import java.util.List;  public class DeleteFragment extends Fragment {   private ListView bookListView;  private DeleteAdapter adapter;  private List<Book> bookList;   @Override  public void onActivityCreated (Bundle b) {  super.onActivityCreated(b);   bookListView = (ListView) getView().findViewById(R.id.*deleteListView*);  bookList = new ArrayList<Book>();  adapter = new DeleteAdapter(getContext().getApplicationContext(), bookList, this);  bookListView.setAdapter(adapter);   new DeleteFragment.BackgroundTask().execute();  }   class BackgroundTask extends AsyncTask<Void, Void, String>  {  String target;   @Override  protected void onPreExecute() {  try {  target="http://han201835461.dothome.co.kr/Book.php";  } catch (Exception e) {  e.printStackTrace();  }  }   @Override  protected String doInBackground(Void... voids) {  try {  URL url = new URL(target);  HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();  InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = new BufferedReader (new InputStreamReader(inputStream));  String temp;  StringBuilder stringBuilder = new StringBuilder();  while((temp = bufferedReader.readLine()) != null)  {  stringBuilder.append(temp + "\n");  }  bufferedReader.close();  inputStream.close();  httpURLConnection.disconnect();  return stringBuilder.toString().trim();  } catch (Exception e) {  e.printStackTrace();  }   return null;  }   @Override  public void onProgressUpdate(Void... values) { super.onProgressUpdate();}   @Override  public void onPostExecute(String result) {  try {  bookList.clear();  JSONObject jsonObject = new JSONObject(result);  JSONArray jsonArray = jsonObject.getJSONArray("response");   int count = 0;  int bookID;  String bookName;  String bookType;  int bookYear;   while(count < jsonArray.length())  {  JSONObject object = jsonArray.getJSONObject(count);  bookID = object.getInt("bookID");  bookName = object.getString("bookName");  bookType = object.getString("bookType");  bookYear = object.getInt("bookYear");   bookList.add(new Book(bookID, bookName, bookType, bookYear));  count++;  }  adapter.notifyDataSetChanged();  } catch (Exception e) {  e.printStackTrace();  }  }  }   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  return inflater.inflate(R.layout.*fragment\_delete*,container,false);  } } |

## DeleteRequest.java

|  |
| --- |
| package com.example.serverconnection;  import com.android.volley.AuthFailureError; import com.android.volley.Response; import com.android.volley.toolbox.StringRequest;  import java.util.HashMap; import java.util.Map;  public class DeleteRequest extends StringRequest {   final static private String *URL* = "http://han201835461.dothome.co.kr/Delete.php";  private Map<String, String> map;   public DeleteRequest(String bookID, String bookName, String bookType, String bookYear, Response.Listener<String> listener) {  super(Method.*POST*, *URL*, listener, null);   map = new HashMap<>();  map.put("bookID", bookID);  map.put("bookName", bookName);  map.put("bookType", bookType);  map.put("bookYear", bookYear);  }   @Override  protected Map<String, String> getParams() throws AuthFailureError {  return map;  } } |

## LoginActivity.java

|  |
| --- |
| package com.example.serverconnection;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast;  import androidx.appcompat.app.AlertDialog; import androidx.appcompat.app.AppCompatActivity;  import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.toolbox.Volley;  import org.json.JSONException; import org.json.JSONObject;  public class LoginActivity extends AppCompatActivity {   private EditText idText, passwordText;  private Button loginButton, registerButton;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_login);   registerButton = findViewById(R.id.*registerButton*);  registerButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Intent intent = new Intent(LoginActivity.this, RegisterActivity.class);  startActivity(intent);  }  });   idText = findViewById(R.id.*idText*);  passwordText = findViewById(R.id.*passwordText*);   loginButton = findViewById(R.id.*loginButton*);  loginButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  final String userID = idText.getText().toString();  String userPassword = passwordText.getText().toString();   Response.Listener<String> responseListener = new Response.Listener<String>() {  @Override  public void onResponse(String response) {  try {  JSONObject jsonObject = new JSONObject(response);  boolean success = jsonObject.getBoolean("success");  if (success) {  String userID = jsonObject.getString("userID");  String userPassword = jsonObject.getString("userPassword");   if(userID.equalsIgnoreCase("manager") && (userPassword.equalsIgnoreCase("manager")))  {  Toast.*makeText*(getApplicationContext(), "WELCOME", Toast.*LENGTH\_SHORT*).show();  Intent intent = new Intent(LoginActivity.this, MainActivity2.class);  intent.putExtra("userID",userID);  intent.putExtra("userPassword",userPassword);  startActivity(intent);  }  else  {  Toast.*makeText*(getApplicationContext(), "WELCOME", Toast.*LENGTH\_SHORT*).show();  Intent intent = new Intent(LoginActivity.this, MainActivity.class);  intent.putExtra("userID", userID);  intent.putExtra("userPassword",userPassword);  startActivity(intent);  }   } else {  Toast.makeText(getApplicationContext(), "Login Failed", Toast.LENGTH\_SHORT).show();  return;  }  } catch (JSONException e) {  e.printStackTrace();  }  }  };   LoginRequest loginRequest = new LoginRequest(userID, userPassword, responseListener);  RequestQueue queue = Volley.newRequestQueue(LoginActivity.this);  queue.add(loginRequest);  }  });  }  } |

## LoginRequest.java

|  |
| --- |
| package com.example.serverconnection;  import com.android.volley.AuthFailureError; import com.android.volley.Response; import com.android.volley.toolbox.StringRequest;  import java.util.HashMap; import java.util.Map;  public class LoginRequest extends StringRequest {   //서버 URL 설정 (PHP 파일 연동)  final static private String *URL* = "http://han201835461.dothome.co.kr/Login2.php";  private Map<String, String> map;   public LoginRequest(String userID, String userPassword, Response.Listener<String> listener) {  super(Method.*POST*, *URL*, listener, null);   map = new HashMap<>();  map.put("userID", userID);  map.put("userPassword", userPassword);  }   @Override  protected Map<String, String> getParams() throws AuthFailureError {  return map;  } } |

## MainActivity.java

|  |
| --- |
| package com.example.serverconnection;  import android.content.Intent; import android.content.pm.ActivityInfo; import android.os.AsyncTask; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.LinearLayout; import android.widget.TextView; import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity; import androidx.fragment.app.FragmentManager; import androidx.fragment.app.FragmentTransaction;  import java.net.URLEncoder;  public class MainActivity extends AppCompatActivity {   public static String userID;  public static int bookID;  private TextView userName;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  setRequestedOrientation(ActivityInfo.SCREEN\_ORIENTATION\_UNSPECIFIED);   userID = getIntent().getStringExtra("userID");   final Button bookButton = (Button)findViewById(R.id.bookButton);  final Button loanBookButton = (Button)findViewById(R.id.loanBookButton);  final Button seatButton = (Button)findViewById(R.id.seatButton);   final LinearLayout notice = (LinearLayout)findViewById(R.id.notice);   bookButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  notice.setVisibility(View.*GONE*);  bookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimaryDark*));  loanBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  seatButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));   FragmentManager fragmentManager = getSupportFragmentManager();  FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  fragmentTransaction.replace(R.id.*fragment*, new BookFragment());  fragmentTransaction.commit();  }  });   loanBookButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  notice.setVisibility(View.*GONE*);  bookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  loanBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimaryDark*));  seatButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));   FragmentManager fragmentManager = getSupportFragmentManager();  FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  fragmentTransaction.replace(R.id.*fragment*, new UserBookFragment());  fragmentTransaction.commit();  }  });   seatButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  notice.setVisibility(View.*GONE*);  bookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  loanBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  seatButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimaryDark*));   FragmentManager fragmentManager = getSupportFragmentManager();  FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  fragmentTransaction.replace(R.id.*fragment*, new SeatFragment());  fragmentTransaction.commit();  }  });   }    //뒤로 가기 버튼을 두 번 눌러 앱 종료하기  private long lastTimeBackPressed;   @Override  public void onBackPressed() {  if(System.currentTimeMillis() - lastTimeBackPressed < 1500)  {  finish();  return;  }  Toast.makeText(this,"Press the back button one more time to exit.", Toast.LENGTH\_SHORT).show();  lastTimeBackPressed = System.currentTimeMillis();  }  } |

## MainActivity2.java

|  |
| --- |
| package com.example.serverconnection;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.LinearLayout; import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity; import androidx.fragment.app.FragmentManager; import androidx.fragment.app.FragmentTransaction;  public class MainActivity2 extends AppCompatActivity {   public static int *bookID*;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main2*);   *bookID* = getIntent().getIntExtra("bookID", *bookID*);   final Button addBookButton = (Button)findViewById(R.id.*addBookButton*);  final Button deleteBookButton = (Button)findViewById(R.id.*deleteBookButton*);  final Button manageButton = (Button)findViewById(R.id.*manageButton*);   final LinearLayout notice = (LinearLayout)findViewById(R.id.*notice*);   addBookButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  notice.setVisibility(View.*GONE*);  addBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimaryDark*));  deleteBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  manageButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));   FragmentManager fragmentManager = getSupportFragmentManager();  FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  fragmentTransaction.replace(R.id.*fragment*, new AddFragment());  fragmentTransaction.commit();  }  });   deleteBookButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  notice.setVisibility(View.*GONE*);  addBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  deleteBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimaryDark*));  manageButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));   FragmentManager fragmentManager = getSupportFragmentManager();  FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  fragmentTransaction.replace(R.id.*fragment*, new DeleteFragment());  fragmentTransaction.commit();  }  });   manageButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  notice.setVisibility(View.*GONE*);  addBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  deleteBookButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimary*));  manageButton.setBackgroundColor(getResources().getColor(R.color.*colorPrimaryDark*));   FragmentManager fragmentManager = getSupportFragmentManager();  FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  fragmentTransaction.replace(R.id.*fragment*, new manageFragment());  fragmentTransaction.commit();  }  });  }    public void mOnPopupClick(View v){  //Invoke pop-up (Activity) with time record  Intent intent = new Intent(this, AddActivity.class);  startActivityForResult(intent, 1);  }   //뒤로 가기 버튼을 두 번 눌러 앱 종료하기  private long lastTimeBackPressed;   @Override  public void onBackPressed() {  if(System.*currentTimeMillis*() - lastTimeBackPressed < 1500)  {  finish();  return;  }  Toast.*makeText*(this,"Press the back button one more time to exit.", Toast.*LENGTH\_SHORT*).show();  lastTimeBackPressed = System.*currentTimeMillis*();  }  } |

## Manage.java

|  |
| --- |
| package com.example.serverconnection;  public class Manage {  String userID;  String userPassword;  String userName;  int userAge;   public String getUserID() {  return userID;  }   public void setUserID(String userID) {  this.userID = userID;  }   public String getUserPassword() {  return userPassword;  }   public void setUserPassword(String userPassword) {  this.userPassword = userPassword;  }   public String getUserName() {  return userName;  }   public void setUserName(String userName) {  this.userName = userName;  }   public int getUserAge() {  return userAge;  }   public void setUserAge(int userAge) {  this.userAge = userAge;  }   public Manage(String userID, String userPassword, String userName, int userAge) {  this.userID = userID;  this.userPassword = userPassword;  this.userName = userName;  this.userAge = userAge;  } } |

## manageFragment.java

|  |
| --- |
| package com.example.serverconnection;  import android.app.AlertDialog; import android.os.AsyncTask; import android.os.Bundle; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ListView; import android.widget.Spinner;  import androidx.fragment.app.Fragment;  import org.json.JSONArray; import org.json.JSONObject;  import java.io.BufferedReader; import java.io.InputStream; import java.io.InputStreamReader; import java.net.HttpURLConnection; import java.net.URL; import java.net.URLEncoder; import java.util.ArrayList; import java.util.List;  public class manageFragment extends Fragment {  private ListView manageListView;  private ManageListAdapter adapter;  private List<Manage> manageList;   @Override  public void onActivityCreated (Bundle b) {  super.onActivityCreated(b);  manageListView = (ListView) getView().findViewById(R.id.*manageListView*);  manageList = new ArrayList<Manage>();  adapter = new ManageListAdapter(getContext().getApplicationContext(), manageList, this);  manageListView.setAdapter(adapter);   new manageFragment.BackgroundTask().execute();  }   class BackgroundTask extends AsyncTask<Void, Void, String>  {  String target;   @Override  protected void onPreExecute() {  try {  target="http://han201835461.dothome.co.kr/UserList.php";  } catch (Exception e) {  e.printStackTrace();  }  }   @Override  protected String doInBackground(Void... voids) {  try {  URL url = new URL(target);  HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();  InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = new BufferedReader (new InputStreamReader(inputStream));  String temp;  StringBuilder stringBuilder = new StringBuilder();  while((temp = bufferedReader.readLine()) != null)  {  stringBuilder.append(temp + "\n");  }  bufferedReader.close();  inputStream.close();  httpURLConnection.disconnect();  return stringBuilder.toString().trim();  } catch (Exception e) {  e.printStackTrace();  }   return null;  }   @Override  public void onProgressUpdate(Void... values) { super.onProgressUpdate();}   @Override  public void onPostExecute(String result) {  try {  manageList.clear();  JSONObject jsonObject = new JSONObject(result);  JSONArray jsonArray = jsonObject.getJSONArray("response");   int count = 0;  String userID;  String userPassword;  String userName;  int userAge;   while(count < jsonArray.length())  {  JSONObject object = jsonArray.getJSONObject(count);  userID = object.getString("userID");  userPassword = object.getString("userPassword");  userName = object.getString("userName");  userAge = object.getInt("userAge");   Manage manage = new Manage(userID, userPassword, userName, userAge);  manageList.add(manage);  count++;  }   adapter.notifyDataSetChanged();  } catch (Exception e) {  e.printStackTrace();  }  }  }   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  return inflater.inflate(R.layout.*fragment\_manage*,container,false);  } } |

## ManageListAdapter.java

|  |
| --- |
| package com.example.serverconnection; import android.app.AlertDialog; import android.content.Context; import android.content.Intent; import android.view.View; import android.view.ViewGroup; import android.widget.BaseAdapter; import android.widget.Button; import android.widget.TextView; import android.widget.Toast;  import androidx.fragment.app.Fragment; import androidx.fragment.app.FragmentTransaction;  import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.toolbox.Volley;  import org.json.JSONException; import org.json.JSONObject;  import java.util.List;  public class ManageListAdapter extends BaseAdapter {   private Context context;  private List<Manage> manageList;  private Fragment parent;   public ManageListAdapter (Context context, List<Manage> manageList, Fragment parent) {  this.context = context;  this.manageList = manageList;  this.parent = parent;  }  @Override  public int getCount() {  return manageList.size();  }   @Override  public Object getItem(int i) {  return manageList.get(i);  }   @Override  public long getItemId(int i) {  return i;  }   @Override  public View getView(final int i, View view, ViewGroup viewGroup) {  View v = View.*inflate*(context, R.layout.*manage*, null);  TextView userID = (TextView) v.findViewById(R.id.*userID*);  TextView userPassword = (TextView) v.findViewById(R.id.*userPassword*);  TextView userName = (TextView) v.findViewById(R.id.*userName*);  TextView userAge = (TextView) v.findViewById(R.id.*userAge*);   userID.setText(manageList.get(i).getUserID());  userPassword.setText(manageList.get(i).getUserPassword());  userName.setText(manageList.get(i).getUserName());  userAge.setText(manageList.get(i).getUserAge()+"");   return v;  } } |

## RegisterActivity.java

|  |
| --- |
| package com.example.serverconnection;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast;  import androidx.appcompat.app.AppCompatActivity;  import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.toolbox.Volley;  import org.json.JSONException; import org.json.JSONObject;  public class RegisterActivity extends AppCompatActivity {   private EditText idText, passwordText, nameText, ageText;  private Button registerButton;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_register*);   idText = findViewById(R.id.*idText*);  passwordText = findViewById(R.id.*passwordText*);  nameText = findViewById(R.id.*nameText*);  ageText = findViewById(R.id.*ageText*);   registerButton = findViewById(R.id.*registerButton*);  registerButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  String userID = idText.getText().toString();  String userPassword = passwordText.getText().toString();  String userName = nameText.getText().toString();  int userAge = Integer.*parseInt*(ageText.getText().toString());   Response.Listener<String> responseListener = new Response.Listener<String>() {  @Override  public void onResponse(String response) {  try {  JSONObject jsonObject = new JSONObject(response);  boolean success = jsonObject.getBoolean("success");  if(success) {  Toast.*makeText*(getApplicationContext(), "Registration Completed", Toast.*LENGTH\_SHORT*).show();  Intent intent = new Intent(RegisterActivity.this, LoginActivity.class);  startActivity(intent);  }  else {  Toast.*makeText*(getApplicationContext(), "Registration Failed", Toast.*LENGTH\_SHORT*).show();  return;  }  } catch (JSONException e) {  e.printStackTrace();  }  }  };   RegisterRequest registerRequest = new RegisterRequest(userID, userPassword, userName, userAge, responseListener);  RequestQueue queue = Volley.*newRequestQueue*(RegisterActivity.this);  queue.add(registerRequest);   }  });  } } |

## RegisterRequest.java

|  |
| --- |
| package com.example.serverconnection;  import com.android.volley.AuthFailureError; import com.android.volley.Response; import com.android.volley.toolbox.StringRequest;  import java.util.HashMap; import java.util.Map;  public class RegisterRequest extends StringRequest {   //서버 URL 설정 (PHP 파일 연동)http://han201835461.dothome.co.kr/myadmin  final static private String *URL* = "http://han201835461.dothome.co.kr/Register2.php";  private Map<String, String> map;   public RegisterRequest(String userID, String userPassword, String userName, int userAge, Response.Listener<String> listener) {  super(Method.*POST*, *URL*, listener, null);   map = new HashMap<>();  map.put("userID", userID);  map.put("userPassword", userPassword);  map.put("userName", userName);  map.put("userAge", userAge+"");  }   @Override  protected Map<String, String> getParams() throws AuthFailureError {  return map;  } } |

## ReturnRequest.java

|  |
| --- |
| package com.example.serverconnection;  import com.android.volley.AuthFailureError; import com.android.volley.Response; import com.android.volley.toolbox.StringRequest;  import java.util.HashMap; import java.util.Map;  public class ReturnRequest extends StringRequest {   final static private String *URL* = "http://han201835461.dothome.co.kr/Return.php";  private Map<String, String> map;   public ReturnRequest(String userID, String bookID, Response.Listener<String> listener) {  super(Method.*POST*, *URL*, listener, null);   map = new HashMap<>();  map.put("userID", userID);  map.put("bookID", bookID);  }   @Override  protected Map<String, String> getParams() throws AuthFailureError {  return map;  } } |

## SeatFragment.java

|  |
| --- |
| package com.example.serverconnection;  import android.os.Bundle; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ImageButton; import android.widget.ListView; import android.widget.Spinner; import android.widget.Toast;  import androidx.annotation.Nullable; import androidx.fragment.app.Fragment;  import java.util.ArrayList;  public class SeatFragment extends Fragment {   private ImageButton S1, S2, S3, S4, S5, S6, S7, S8, S9;   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  return inflater.inflate(R.layout.*fragment\_seat*,container,false);  }  @Override  public void onActivityCreated (Bundle b) {  super.onActivityCreated(b);   S1 = (ImageButton)getView().findViewById(R.id.*seat1*);  S1.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 1 reservation completed", Toast.*LENGTH\_LONG*).show();  S1.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S2 = (ImageButton)getView().findViewById(R.id.*seat2*);  S2.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 2 reservation completed", Toast.*LENGTH\_LONG*).show();  S2.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S3 = (ImageButton)getView().findViewById(R.id.*seat3*);  S3.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 3 reservation completed", Toast.*LENGTH\_LONG*).show();  S3.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S4 = (ImageButton)getView().findViewById(R.id.*seat4*);  S4.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 4 reservation completed", Toast.*LENGTH\_LONG*).show();  S4.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S5 = (ImageButton)getView().findViewById(R.id.*seat5*);  S5.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 5 reservation completed", Toast.*LENGTH\_LONG*).show();  S5.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S6 = (ImageButton)getView().findViewById(R.id.*seat6*);  S6.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 6 reservation completed", Toast.*LENGTH\_LONG*).show();  S6.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S7 = (ImageButton)getView().findViewById(R.id.*seat7*);  S7.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 7 reservation completed", Toast.*LENGTH\_LONG*).show();  S7.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S8 = (ImageButton)getView().findViewById(R.id.*seat8*);  S8.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 8 reservation completed", Toast.*LENGTH\_LONG*).show();  S8.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });   S9 = (ImageButton)getView().findViewById(R.id.*seat9*);  S9.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Toast.*makeText*(getActivity(), "Number 9 reservation completed", Toast.*LENGTH\_LONG*).show();  S9.setBackgroundColor(getResources().getColor(R.color.*colorGray*));  }  });  } } |

## UserBookAdapter.java

|  |
| --- |
| package com.example.serverconnection;  import android.app.AlertDialog; import android.content.Context; import android.content.Intent; import android.os.AsyncTask; import android.view.View; import android.view.ViewGroup; import android.widget.BaseAdapter; import android.widget.Button; import android.widget.TextView; import android.widget.Toast;  import androidx.fragment.app.Fragment; import androidx.fragment.app.FragmentTransaction;  import com.android.volley.RequestQueue; import com.android.volley.Response; import com.android.volley.toolbox.Volley;  import org.json.JSONArray; import org.json.JSONException; import org.json.JSONObject;  import java.io.BufferedReader; import java.io.InputStream; import java.io.InputStreamReader; import java.net.HttpURLConnection; import java.net.URL; import java.net.URLEncoder; import java.util.List;  public class UserBookAdapter extends BaseAdapter {   private Context context;  private List<Book> bookList;  private Fragment parent;  private String userID = MainActivity.*userID*;   public UserBookAdapter (Context context, List<Book> bookList, Fragment parent) {  this.context = context;  this.bookList = bookList;  this.parent = parent;  }  @Override  public int getCount() {  return bookList.size();  }   @Override  public Object getItem(int i) {  return bookList.get(i);  }   @Override  public long getItemId(int i) {  return i;  }   @Override  public View getView(final int i, View view, ViewGroup viewGroup) {  View v = View.*inflate*(context, R.layout.*book\_return*, null);  TextView bookID = (TextView) v.findViewById(R.id.*id*);  TextView bookType = (TextView) v.findViewById(R.id.*type*);  TextView bookName = (TextView) v.findViewById(R.id.*name*);  TextView bookYear = (TextView) v.findViewById(R.id.*year*);   bookID.setText(bookList.get(i).getBookID()+"");  bookType.setText(bookList.get(i).getBookType());  bookName.setText(bookList.get(i).getBookName());  bookYear.setText(bookList.get(i).getBookYear()+"");   v.setTag(bookList.get(i).getBookID());   Button returnButton = (Button) v.findViewById(R.id.*returnBtn*);  returnButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  Response.Listener<String> responseListener = new Response.Listener<String>() {  @Override  public void onResponse(String response) {  try {  JSONObject jsonObject = new JSONObject(response);  boolean success = jsonObject.getBoolean("success");  if(success) {  AlertDialog.Builder builder = new AlertDialog.Builder(parent.getActivity());  AlertDialog dialog = builder.setMessage("You return the book!").setPositiveButton("check", null).create();  dialog.show();  bookList.remove(i);  notifyDataSetChanged();  }  else {  AlertDialog.Builder builder = new AlertDialog.Builder(parent.getActivity());  AlertDialog dialog = builder.setMessage("Return failure").setNegativeButton("retry", null).create();  dialog.show();  }  } catch (JSONException e) {  e.printStackTrace();  }  }  };  ReturnRequest returnBookRequest = new ReturnRequest(userID, bookList.get(i).getBookID()+"", responseListener);  RequestQueue queue = Volley.*newRequestQueue*(parent.getActivity());  queue.add(returnBookRequest);  }  });   return v;  }  } |

## UserBookFragment.java

|  |
| --- |
| package com.example.serverconnection;  import android.app.AlertDialog; import android.os.AsyncTask; import android.os.Bundle; import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ListAdapter; import android.widget.ListView; import android.widget.Spinner;  import androidx.fragment.app.Fragment;  import org.json.JSONArray; import org.json.JSONObject;  import java.io.BufferedReader; import java.io.InputStream; import java.io.InputStreamReader; import java.net.HttpURLConnection; import java.net.URL; import java.net.URLEncoder; import java.util.ArrayList; import java.util.List;  public class UserBookFragment extends Fragment {   private ListView userBookListView;  private UserBookAdapter adapter;  private List<Book> bookList;   @Override  public void onActivityCreated (Bundle b) {  super.onActivityCreated(b);   userBookListView = (ListView) getView().findViewById(R.id.*userBookListView*);  bookList = new ArrayList<Book>();  adapter = new UserBookAdapter(getContext().getApplicationContext(), bookList, this);  userBookListView.setAdapter(adapter);  new BackgroundTask().execute();  }   class BackgroundTask extends AsyncTask<Void, Void, String>  {  String target;   @Override  protected void onPreExecute() {  try {  target="http://han201835461.dothome.co.kr/UserBookList.php?userID=" + URLEncoder.*encode*(MainActivity.*userID*, "UTF-8");  } catch (Exception e) {  e.printStackTrace();  }  }   @Override  protected String doInBackground(Void... voids) {  try {  URL url = new URL(target);  HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();  InputStream inputStream = httpURLConnection.getInputStream();  BufferedReader bufferedReader = new BufferedReader (new InputStreamReader(inputStream));  String temp;  StringBuilder stringBuilder = new StringBuilder();  while((temp = bufferedReader.readLine()) != null)  {  stringBuilder.append(temp + "\n");  }  bufferedReader.close();  inputStream.close();  httpURLConnection.disconnect();  return stringBuilder.toString().trim();  } catch (Exception e) {  e.printStackTrace();  }   return null;  }   @Override  public void onProgressUpdate(Void... values) { super.onProgressUpdate();}   @Override  public void onPostExecute(String result) {  try {  JSONObject jsonObject = new JSONObject(result);  JSONArray jsonArray = jsonObject.getJSONArray("response");   int count = 0;  int bookID;  String bookName;  String bookType;  int bookYear;   while(count < jsonArray.length())  {  JSONObject object = jsonArray.getJSONObject(count);  bookID = object.getInt("bookID");  bookName = object.getString("bookName");  bookType = object.getString("bookType");  bookYear = object.getInt("bookYear");   bookList.add(new Book(bookID, bookName, bookType, bookYear));  count++;  }  adapter.notifyDataSetChanged();  } catch (Exception e) {  e.printStackTrace();  }  }  }   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  return inflater.inflate(R.layout.*fragment\_userbook*,container,false);  } } |

## ADD.php //도서 추가(관리자)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $bookID = $\_POST["bookID"];  $bookName = $\_POST["bookName"];  $bookType = $\_POST["bookType"];  $bookYear = $\_POST["bookYear"];  $statement = mysqli\_prepare($con, "INSERT INTO BOOK VALUES (?,?,?,?)");  mysqli\_stmt\_bind\_param($statement, "issi", $bookID, $bookName, $bookType, $bookYear);  mysqli\_stmt\_execute($statement);  $response = array();  $response["success"] = true;      echo json\_encode($response);  ?> |

## Book.php //도서 리스트 확인(관리자)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $bookID = $\_GET["bookID"];  $bookName = $\_GET["bookName"];  $bookType = $\_GET["bookType"];  $bookYear = $\_GET["bookYear"];    $result = mysqli\_query($con, "SELECT \* FROM BOOK");  $response = array();  while($row = mysqli\_fetch\_array($result)){  array\_push($response, array("bookID"=>$row[0], "bookName"=>$row[1], "bookType"=>$row[2], "bookYear"=>$row[3]));  }    echo json\_encode(array("response"=>$response), JSON\_UNESCAPED\_UNICODE);  mysqli\_close($con);  ?> |

## bookAdd.php //도서 대출(사용자\_손님)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $userID = $\_POST["userID"];  $bookID = $\_POST["bookID"];  0  $statement = mysqli\_prepare($con, "INSERT INTO LOAN VALUES (?, ?)");  mysqli\_stmt\_bind\_param($statement, "si", $userID, $bookID);  mysqli\_stmt\_execute($statement);    $response = array();  $response["success"] = true;    echo json\_encode($response);  ?> |

## BookList.php //도서 검색(사용자\_손님)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $bookID = $\_GET["bookID"];  $bookName = $\_GET["bookName"];  $bookType = $\_GET["bookType"];  $bookYear = $\_GET["bookYear"];    $result = mysqli\_query($con, "SELECT \* FROM BOOK WHERE bookType = '$bookType' AND bookYear= '$bookYear'");  $response = array();  while($row = mysqli\_fetch\_array($result)){  array\_push($response, array("bookID"=>$row[0], "bookName"=>$row[1], "bookType"=>$row[2], "bookYear"=>$row[3]));  }    echo json\_encode(array("response"=>$response), JSON\_UNESCAPED\_UNICODE);  mysqli\_close($con);  ?> |

## Delete.php //도서 삭제(관리자)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $bookID = $\_POST["bookID"];  $bookName = $\_POST["bookName"];  $bookType = $\_POST["bookType"];  $bookYear = $\_POST["bookYear"];  $statement = mysqli\_prepare($con, "DELETE FROM BOOK WHERE bookID = '$bookID' AND bookName = '$bookName' AND bookType = '$bookType' AND bookYear = '$bookYear'");  mysqli\_stmt\_bind\_param($statement, "issi", $bookID, $bookName, $bookType, $bookYear);  mysqli\_stmt\_execute($statement);  $response = array();  $response["success"] = true;      echo json\_encode($response);  ?> |

## Login2.php //로그인(관리자, 사용자\_손님)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $userID = $\_POST["userID"];  $userPassword = $\_POST["userPassword"];    $statement = mysqli\_prepare($con, "SELECT \* FROM USER WHERE userID = ? AND userPassword = ?");  mysqli\_stmt\_bind\_param($statement, "ss", $userID, $userPassword);  mysqli\_stmt\_execute($statement);  mysqli\_stmt\_store\_result($statement);  mysqli\_stmt\_bind\_result($statement, $userID, $userPassword, $userName, $userAge);  $response = array();  $response["success"] = false;    while(mysqli\_stmt\_fetch($statement)) {  $response["success"] = true;  $response["userID"] = $userID;  $response["userPassword"] = $userPassword;  $response["userName"] = $userName;  $response["userAge"] = $userAge;  }  echo json\_encode($response);  ?> |

## Register2.php //회원가입(사용자\_손님)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $userID = $\_POST["userID"];  $userPassword = $\_POST["userPassword"];  $userName = $\_POST["userName"];  $userAge = $\_POST["userAge"];  $statement = mysqli\_prepare($con, "INSERT INTO USER VALUES (?,?,?,?)");  mysqli\_stmt\_bind\_param($statement, "sssi", $userID, $userPassword, $userName, $userAge);  mysqli\_stmt\_execute($statement);  $response = array();  $response["success"] = true;      echo json\_encode($response);  ?> |

## Return.php //도서 반납(사용자\_손님)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $userID = $\_POST["userID"];  $bookID = $\_POST["bookID"];  $statement = mysqli\_prepare($con, "DELETE FROM LOAN WHERE userID = '$userID' AND bookID = '$bookID'");  mysqli\_stmt\_bind\_param($statement, "si", $userID, $bookID);  mysqli\_stmt\_execute($statement);  $response = array();  $response["success"] = true;      echo json\_encode($response);  ?> |

## UserBookList.php //도서 대출 목록(사용자\_손님)

|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $userID = $\_GET["userID"];    $result = mysqli\_query($con, "SELECT BOOK.bookID, BOOK.bookName, BOOK.bookType, BOOK.bookYear FROM USER, BOOK, LOAN WHERE USER.userID = '$userID' AND USER.userID = LOAN.userID AND LOAN.bookID = BOOK.bookID");  $response = array();  while($row = mysqli\_fetch\_array($result)){  array\_push($response, array("bookID"=>$row[0], "bookName"=>$row[1], "bookType"=>$row[2], "bookYear"=>$row[3]));  }    echo json\_encode(array("response"=>$response), JSON\_UNESCAPED\_UNICODE);  mysqli\_close($con);  ?> |

## UserList.php //사용자 관리(관리자)

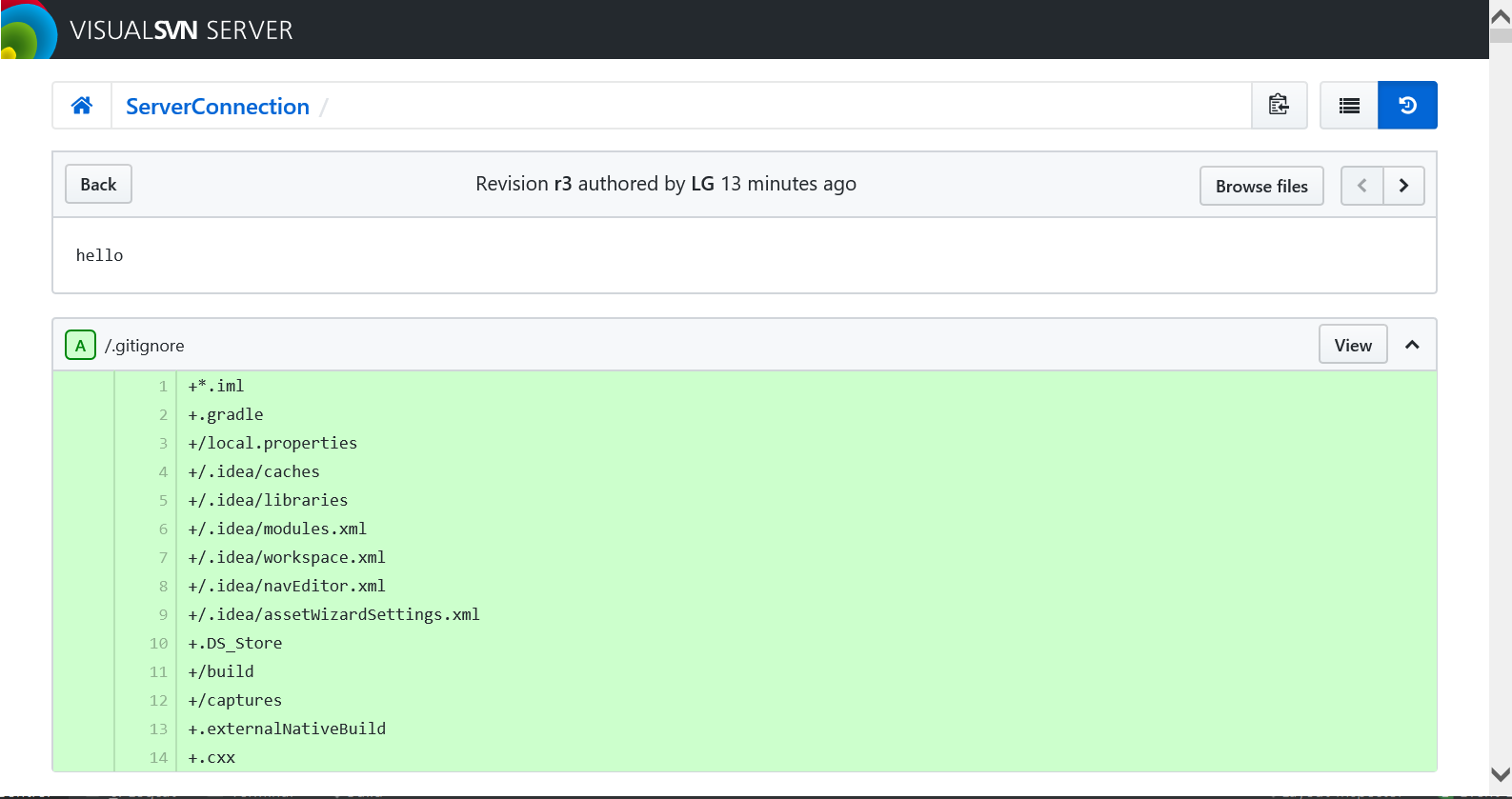
|  |
| --- |
| <?php  $con = mysqli\_connect("localhost", "han201835461", "han8776!", "han201835461");  mysqli\_query($con,'SET NAMES utf8');  $userID = $\_GET["userID"];  $userPassword = $\_GET["userPassword"];  $userName = $\_GET["userName"];  $userAge = $\_GET["userAge"];    $result = mysqli\_query($con, "SELECT \* FROM USER");  $response = array();  while($row = mysqli\_fetch\_array($result)){  array\_push($response, array("userID"=>$row[0], "userPassword"=>$row[1], "userName"=>$row[2], "userAge"=>$row[3]));  }    echo json\_encode(array("response"=>$response), JSON\_UNESCAPED\_UNICODE);  mysqli\_close($con);  ?> |

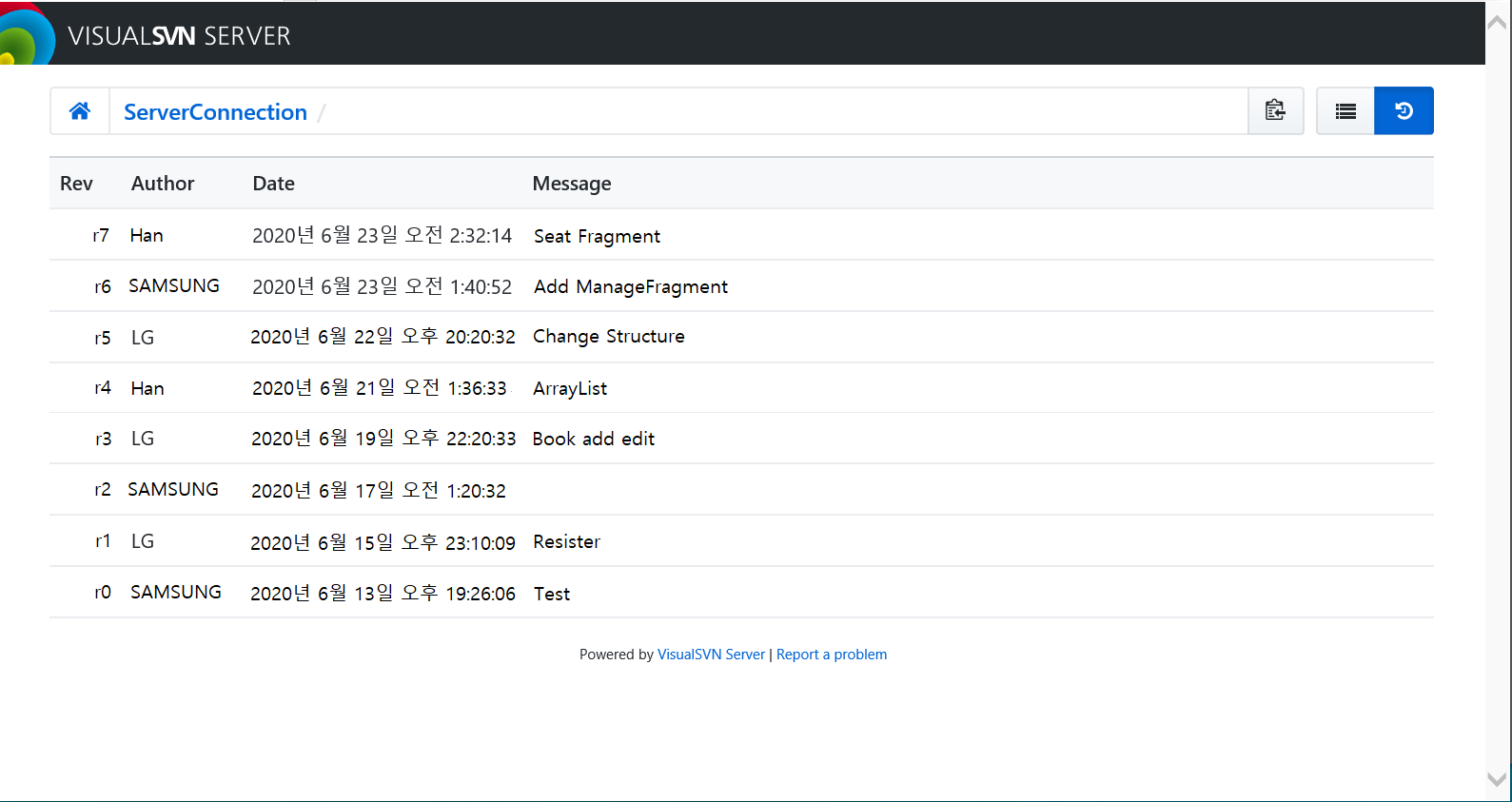
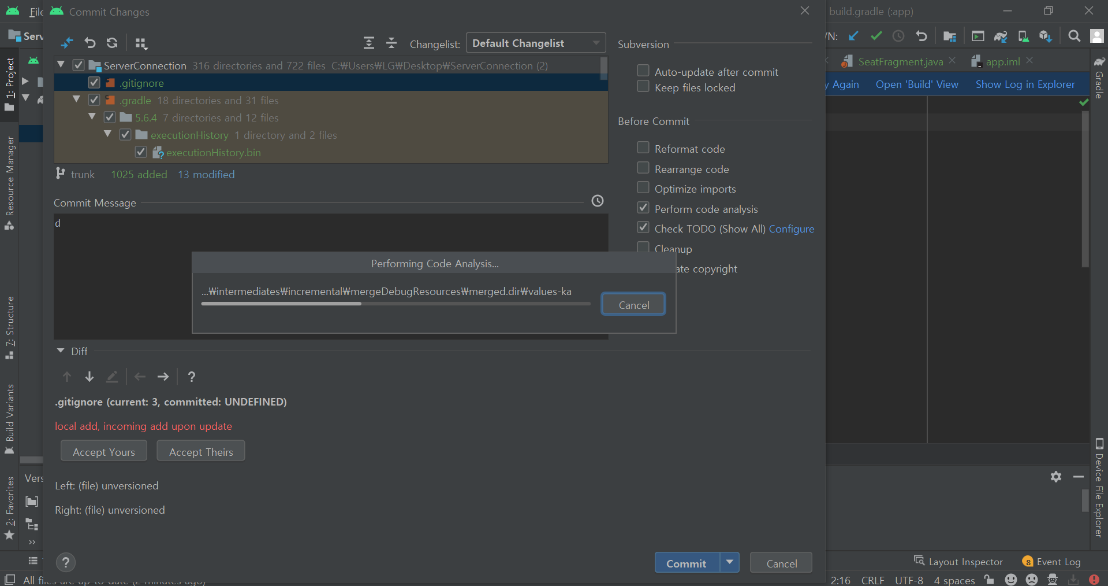
**6. Testing**

**6.1 Configuration Management**

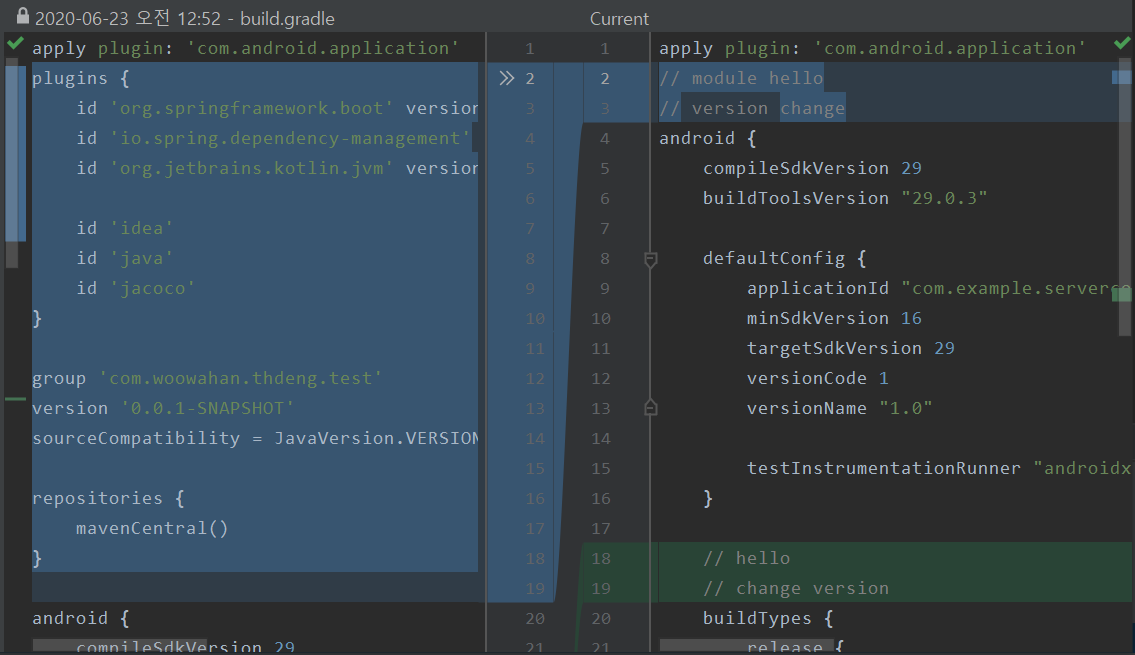
Configuration management for the development environment was set by Android Studio 4.0 and Android 10. A program called Subversion was used for configuration management. After building the server using the VisualSVN server program, SVN data was stored on the server.

To save the data of configuration management, we did a work called Commit, so we did it once after each individual work.

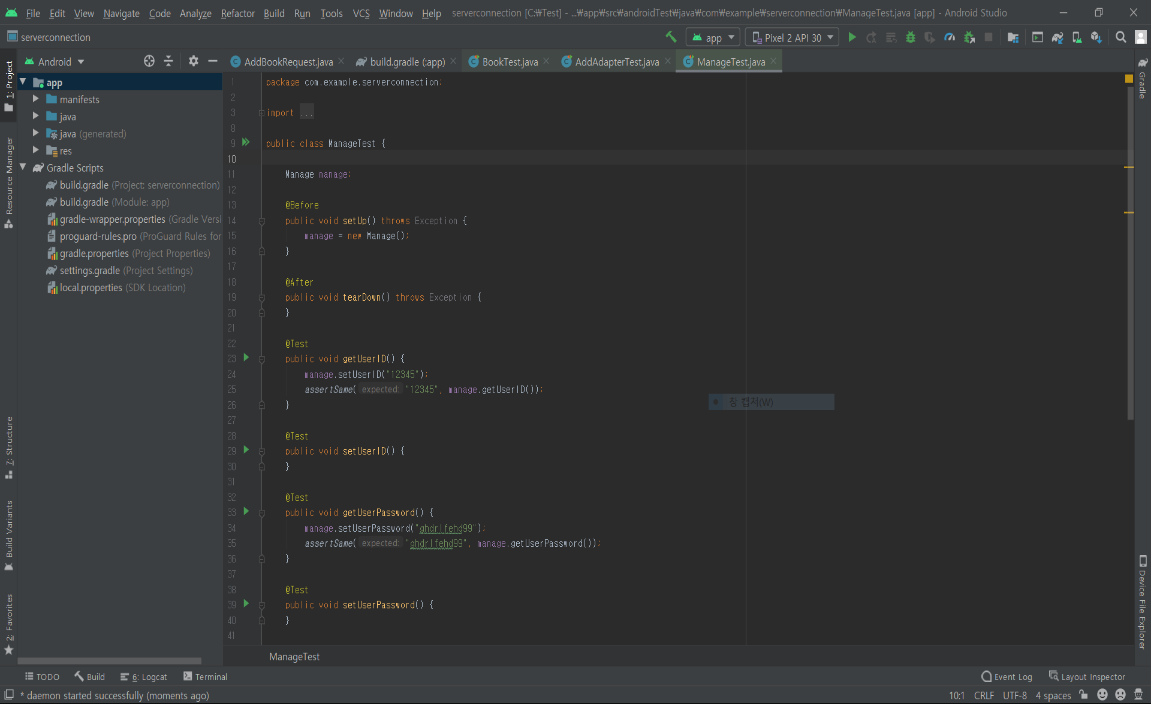
(VisualSVN server)

(Upload time of data stored in SVN server)

(Commit)

It shows the change of Code line and Base line.

**6.2 Unit test**

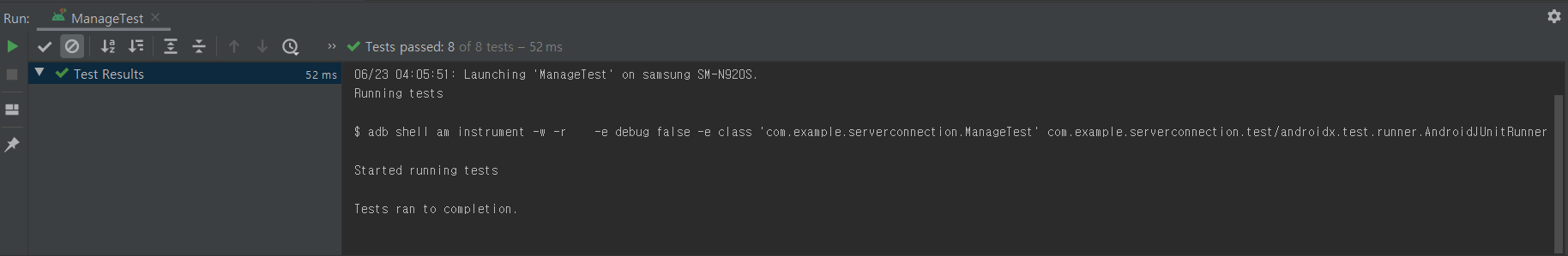
Unit test was Junit built into Android Studio. After writing test code for each function, the result was checked and corrected.

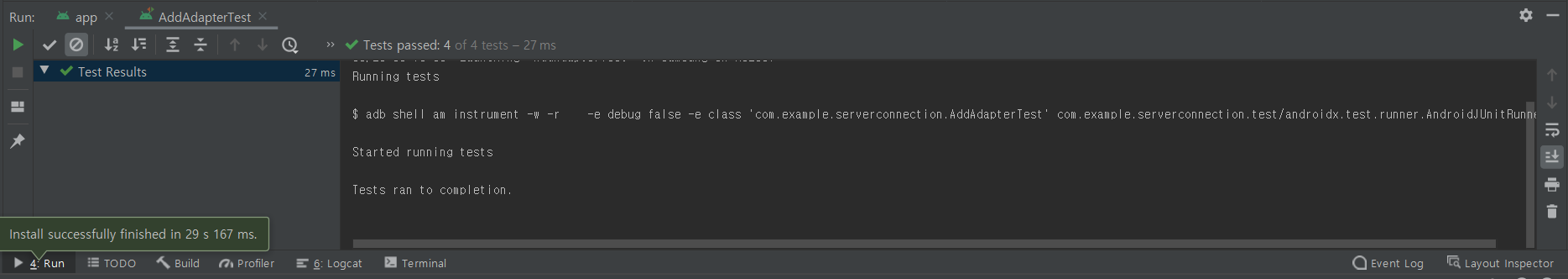
The writing in the red box are test code. Create and run a test code for the function of the class to be tested.

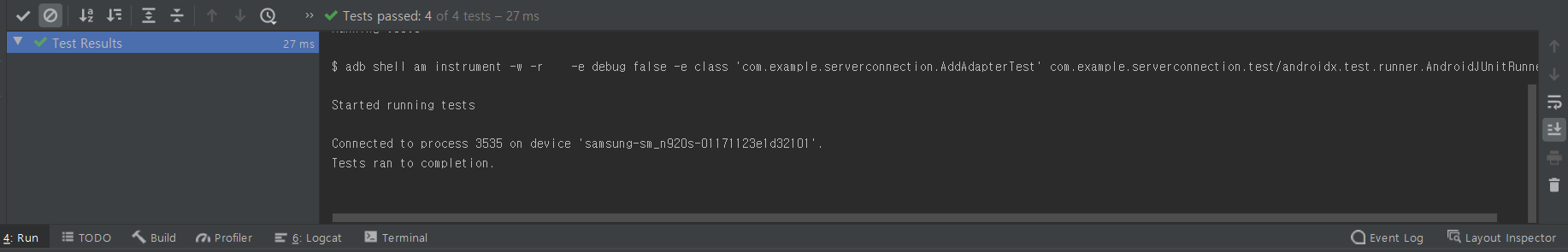
스크린샷이(가) 표시된 사진

자동 생성된 설명🡪Result

We tried to run the test code and the test results came out successfully.

Manage Test

Book Test

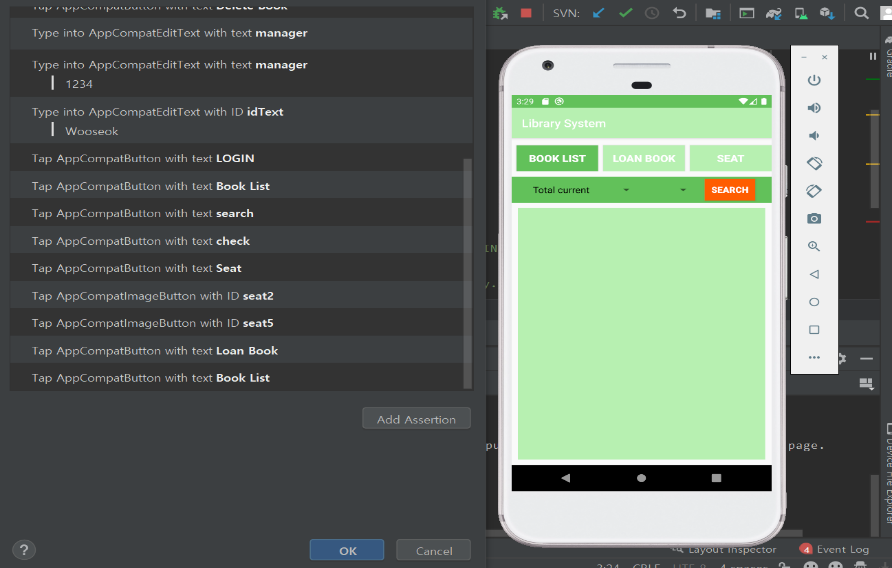
Add adapter Test

**6.3 Test case generation**

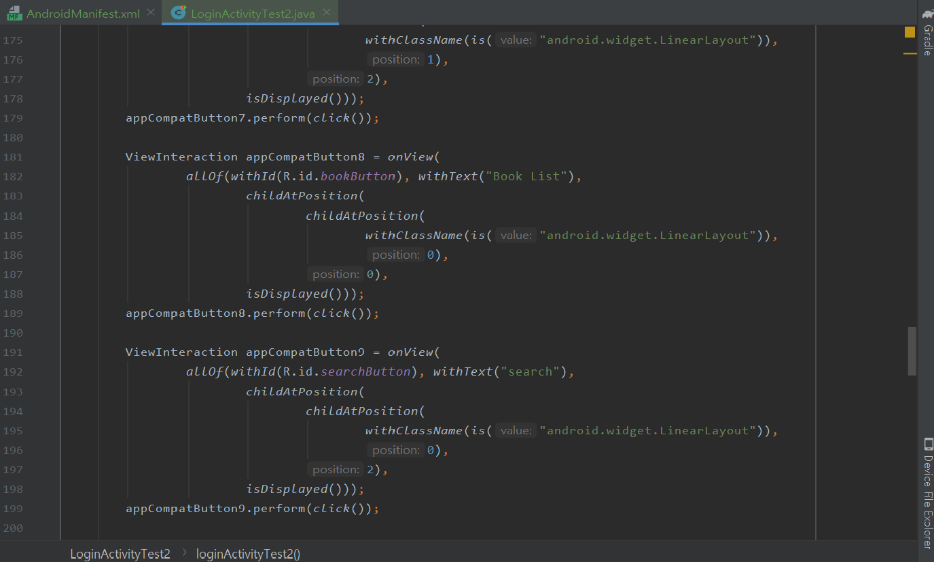
For test case generation, we used a feature called Espresso built into Android Studio. For example, 1. Login – 2. Register – 3. Main screen – 4. (User Admission) Book Search – 5. Book loan – 6. Reading room. As We said, we ran an automated test to see if the successive actions worked.

This process was executed every time an Activity was added, and it was taken after checking if there was a problem.

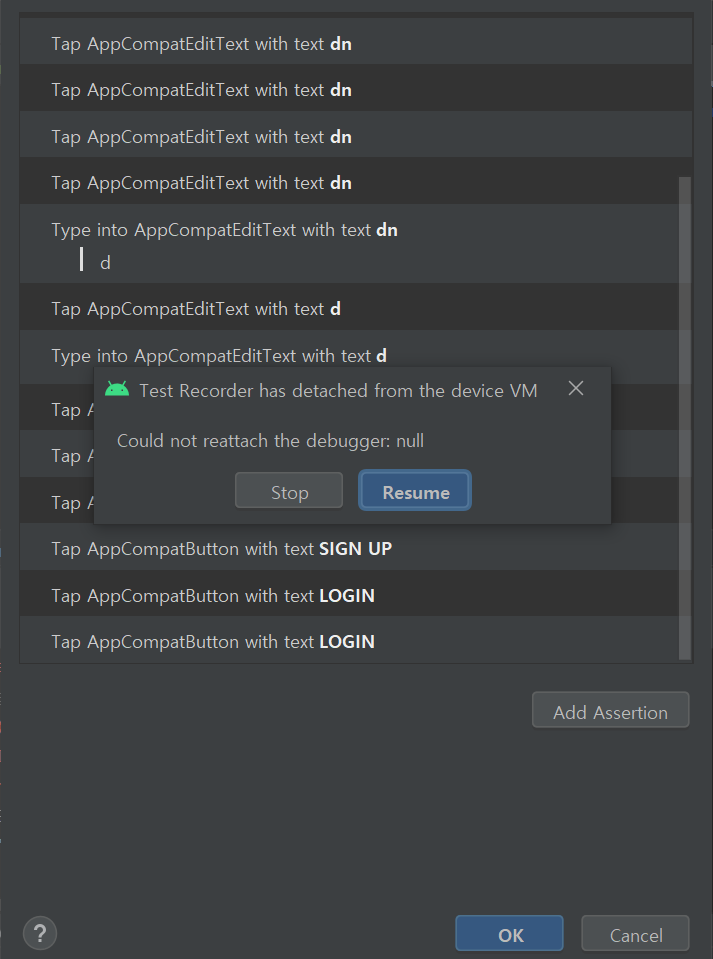
You can enter the code directly, or you can automatically execute what I selected through the UI. Also, since the results were reported, it could be used more conveniently than the unit test.



(We had test without entering the code directly through the UI.)



(Directly entered code to check if functions are executed continuously in a specific situation.

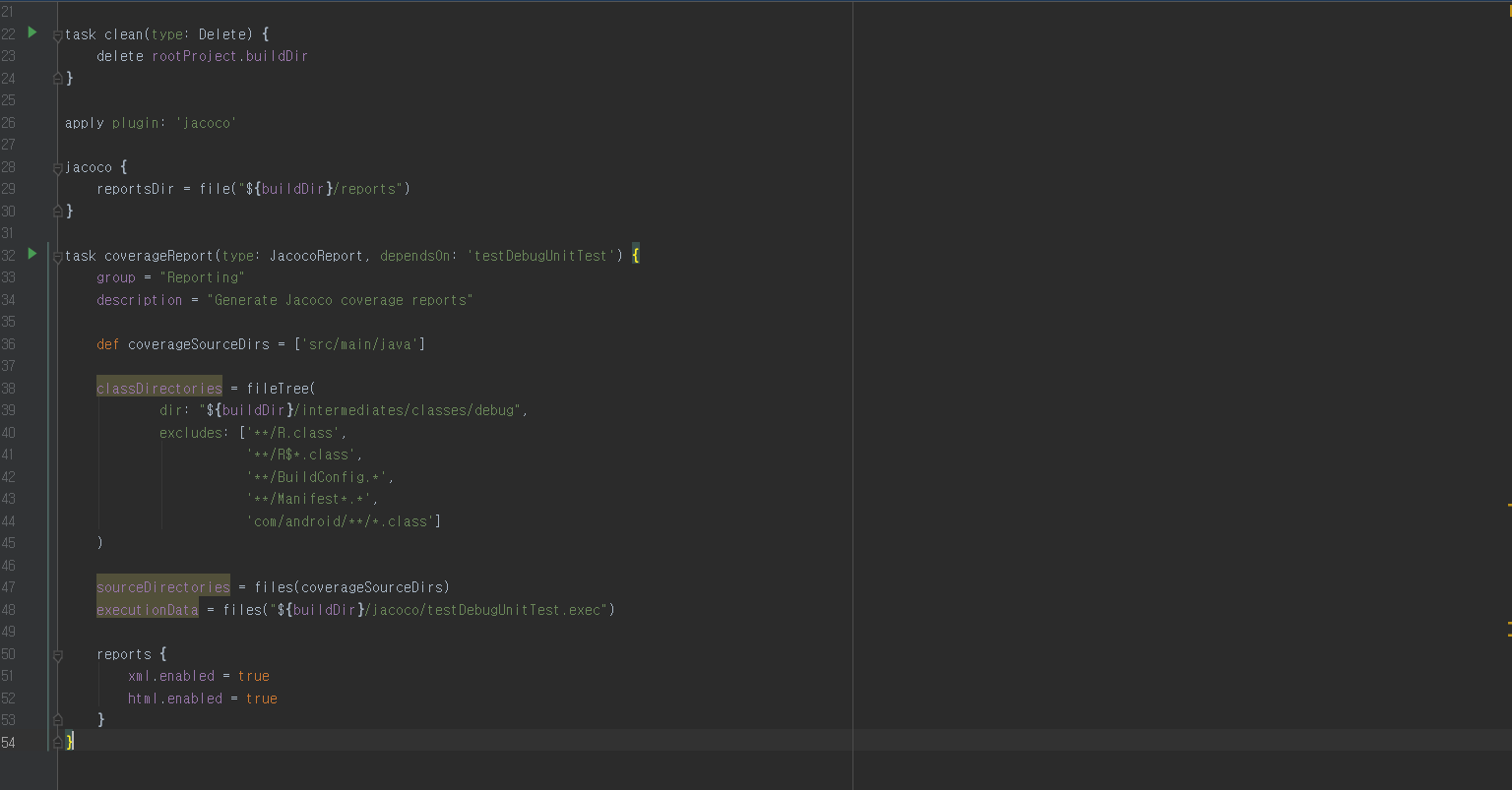
(Notified when there is an error in a specific function or activity)

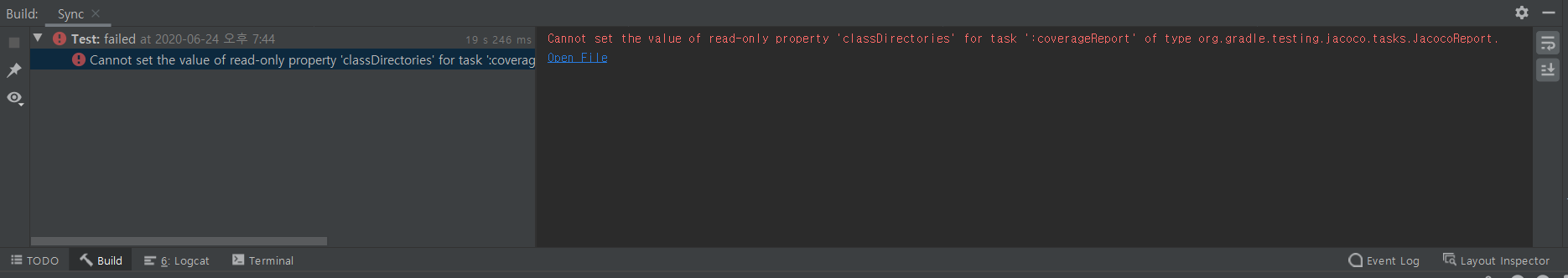
**6.4 Test**

**coverage analysis**

For test coverage analysis, JaCoCo, a tool built into Android Studio, was used.

Despite continuous attempts, we have not been able to confirm the exact result value due to the continuous error. But We could roughly figure out how much code coverage is.



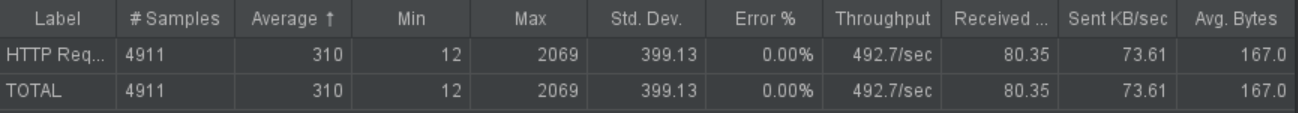
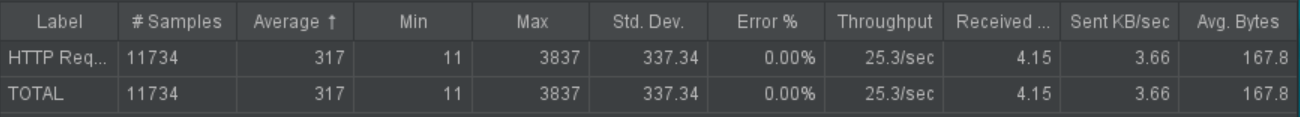
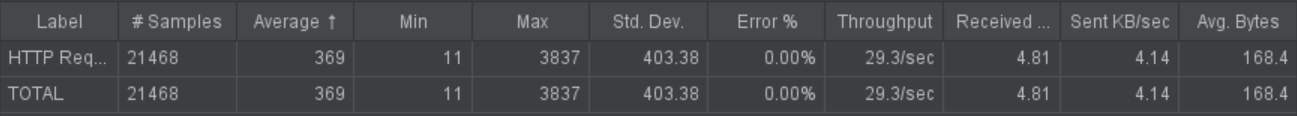


(Result – Although the results were not as good as expected and there was an error, we found it important to analyze how much code testing has been done by trying this test)

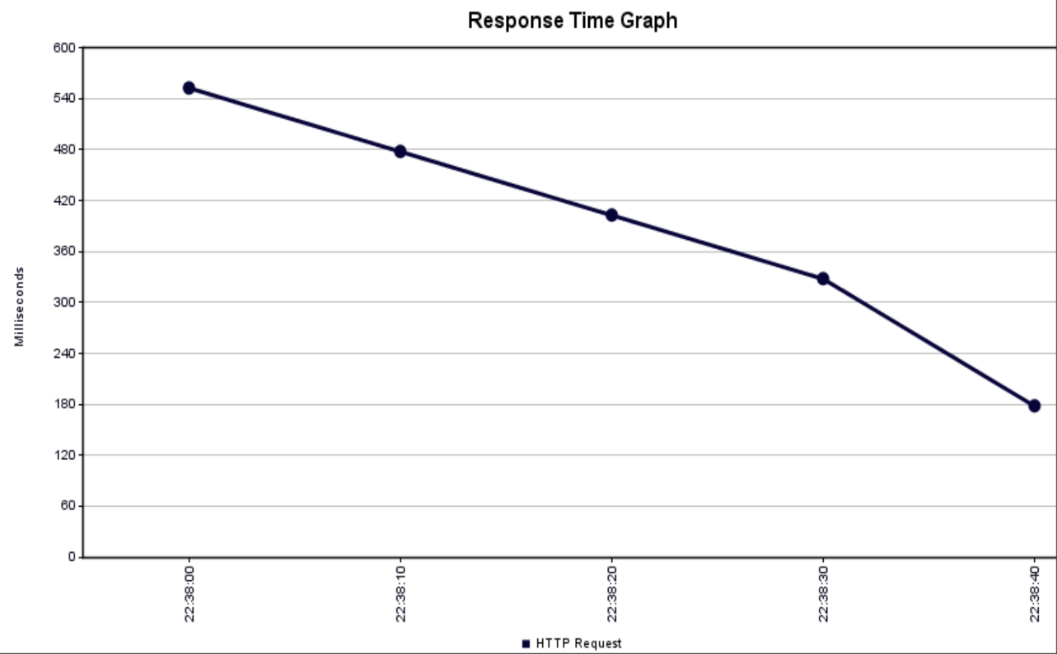
**6.4 Performance Testing**

For performance testing, we used a program called Jmeter. When we ran the application, we experimented with some situations where traffic could hit the server. So, some experiments have confirmed that the program works reliably even for simultaneous use by many people.

In addition, after installing the program on the phone, it worked directly and tested. We focused on checking whether each function works properly and whether there are any problems with the interface. When we tried, all the functions were executed in 1 seconds.

1. When 5000 administrators simultaneously add / remove books.
2. When 5000 users / administrators try to log in at the same time.
3. When 10000 users / administrators try to log in at the same time.

Looking at Throughput, as the number of users increased, the time increased, but there was no problem in using Error as 0%.

4. It is a graph of the response time when 10000 members sign up.

It takes time immediately, but gradually decreases, so the server is used stably.

**7. Glossary**

Unit test : unit testing is a method by which individual units of source code are tested to determine if they are fit for use

Test case generator : test case generators synthesize full test cases.

Configuration management : configuration management is a method is used to manage change

Database : a database is an organized collection of data, generally stored and accessed electronically from a computer system.

Android Studio : it is the official integrated development environment (IDE) for Android and Android-only applications (apps) production.

UI : (User Interface) is a physical or virtual medium created for the purpose of temporary or permanent access to communicate between a person (user) and objects or systems, especially machines and computer programs.

Response time : in technology, response time is the time a system or functional unit takes to react to a given input.

Performance testing : is the process of determining the speed, responsiveness and stability of a computer, network, software program or device under a workload.

Layered Architecture pattern: This pattern is also called the n-tier architecture pattern. This can be used in structured programs that can be divided into groups of submodules.

Domain: It refers to a computer address on the Internet in numbers in English that is easy to understand. System, organization, type of organization, country name. The domain address we use is <http://han201835461.dothome.co.kr/myadmin/>.

FileZilla: It is an FTP client program to upload and download files via FTP. It is developed as an open source and can be used for free.