NOTES SHARING APP IN ANDROID

By:

MUDANNA ADITYA

NIKHIL RANJAN NAYAK

JYOTI RANJAN MOHAPATRA

BAIBHAB SWAIN

December 10th, 2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute of Technical Education and Research

SIKSHA 'O' ANUSANDHAN UNIVERSITY

Bhubaneswar, Odisha, India

ACKNOWLEDGEMENT

It is matter of great pleasure for us to get this opportunity expressing our sincere sense of gratitude to Siksha “O” Anusandhan Deemed to be University. Firstly, we would like to express our heartily thanks to Institute of Technical Education and Research for providing lab facility and other relevant facilities. My guide **Mrs. Sipra Sahoo** was the main force behind all these efforts. Because of her/his valuable suggestions and proper guidance for this project. We express our sincere thanks to the Computer Science & Information Technology department HOD **Prof (Dr.) Debahuti Mishra** who had allowed us to use facilities of the institute. We are also thankful to all those who have helped us in this endeavor either directly or indirectly especially all our teachers. At last we would like to express a big thank you to all friends and all known & unknown person who had helped us directly or indirectly.

Place: Odisha, Bhubaneswar

CERTIFICATE

This is to certify that the project report titled “**Note Sharing App Using Android**” being submitted by Muddana Aditya, Nikhil Ranjan Nayak, Jyoti Ranjan Mohapatra, Baibhab Swain of CSE-F to the institute of Technical Research and Education, Siksha ‘O’ Anusandhan University, Bhubaneswar for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Technology is a record of Original confide work carried out by them under my/our supervision and guidance. The project work, in my/our opinion, has reached the requisite standard fulfilling the requirements for the degree of Bachelor of Technology.

The application developed for this project work have not been submitted in part or full to any other University or Institute for the award of any degree or diploma.

Dr. Saumendra Pattnaik

Dept. Of Computer Science & Engineering

ITER, SOA Deemed to be University

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Contents

1. Synopsys of the Project

2. Questionnaire

3. Analysis of the Survey

4. SRS Document

4.1 Introduction

4.1.1 Purpose

4.1.2 Scope

4.1.3 Definition, Acronyms and Abbreviations

4.2. The Overall Description

4.2.1 Hardware Interfaces

4.2.2 Software Interfaces

4.2.3 Product Functions

4.2.4 User Characteristics

4.3. Requirements

4.3.1 User Interfaces

4.3.2 Software Interfaces

4.3.3 Hardware Interfaces

4.3.4 Functional Requirements

4.3.5 Nonfunctional Requirements

4.3.6 Maintainability

4.3.7 Portability

5. UML DIAGRAMS

5.1 Use Case diagram

5.2 Class diagram

5.3 Sequence diagram

6. UI (User-Interface)

6.1 Layout Designs

7. Testing

8. Program Code

9. Conclusion and Future Work

10. References

1. Synopsys of the Project

Students often face problem while collecting notes and it interrupts their learning process. Our application will provide a platform for the students where they can come together for collaborative learning. Students can share their notes/materials (both handwritten and in document-PPT, PDF, DOC, etc. format) which students from any branch or semester can access for free. In our platform users can upload their notes giving a short description and some keywords related to the subject, branch and semester. Other users can search the notes using those keywords which will provide them filtered results. Each user will have an account through they will access the application. The platform is planned to be a LEARN | SHARE | EDUCATE platform which aims to increase effectiveness and conceptual clarity of students.

2. Questionnaire

1. Have you ever faced problem for collecting notes during exams?

- a. Yes
- b. No
- c. Not Interested to Answer

2. Do you maintain your notes regularly?

- a. Yes
- b. No

3. Do you think this app will help you save your time during exams?

- a. Yes
- b. No
- c. No Opinion

3. Analysis of the Survey

We found that more than 90% students are facing this problem and have to waste their time on collecting notes as they are not maintaining notes during exam. 80% of the students found the app to be helpful for them if implemented to save their time. This app can help their score more as it will reduce their time wastage.

4. SRS Document

4.1 Introduction

4.1.1 Purpose

At present the notes are only maintained by some particular students those who are serious. Later on, during exam time every student are in need of notes and it gets difficult for them even on daily basis.

4.1.2 Scope

We aim to extend this app beyond Note Sharing to promote collaborations not only between students but also between teachers and management by adding different features such as notice board, reviews, feedbacks etc.

4.2 The Overall Description

4.2.1 Product Perspective

ShareNotes is an application for our university. The application offers various operations like viewing notes, downloading and uploading notes which can be accessed by others. These specifications are conveniently grouped and developed specifically for use on your device. All the operations can be performed when an internet connection is available. Admin of the application can register prospective university students to fully experience all services provided.

4.2.2. Hardware Interfaces

1. Internet Connection
2. Android Device

4.2.3. Software Interfaces

In order to work the code expectedly, software should be built accurately and without bugs.

- Ubuntu, Windows 10
- Android Studio

4.2.4. Product Functions

1. Register: The app has a registration feature where the user is required to enter name, email id, password, branch, section, semester and pass-out year.
2. Login: After registration the app allows students to login after verifying email id and password from firebase.
3. Share: The app shares notes and assignments among the students (sharing of assignments or notes take place among students belonging to same branch) through listing
4. Upload/Download: The app allows its users to download and upload their notes and assignments.
5. Space Optimization: The notes are automatically deleted after 18 months to save server cost.

4.2.5. User Characteristics

There can be various types of users for the app based on their profession. The user can be a student who will act as client. The other side may contain users developing, analysing and testing the app.

Student: Student can use this app to upload, share and download the notes to save their time during exams.

Business Analyst: Person considered as the general advisor and analysing expert who is responsible for finalizing the requirements and consider all the main factors related to the project.

Team Lead: A team leader is the person who provides the instructions and information to a group of individuals i.e. team, who are responsible for achieving a certain goal.

UI Designers: People who design the application with mock-ups and User interface. They use the requirements of the project and design the user interface for the system.

Software Developers: People who develop the code according to the design/layout and make the application work with respect to the client requirement. They also maintain the existing code of the application if any issue comes in the application.

Quality Analyst: Users who are responsible for testing the application and ensure that the developed application is intact with the requirements and issue-free. In addition, the code is feasible and working in all scenarios and conditions.

Database Engineer: The person who deals with the entire database of the system and handles any glitches in the database when any update/changes in the development occur.

4.3. Requirements

4.3.1. User Interface

User interface of the application rely on the layout and the performance. UI Designer is the responsible person for creating creative and ease-of-use user interface.

4.3.2. Functional Requirements

- The requirements are, a set of inputs given in proper syntaxes.
- Capacitive touch screen should be available.
- Minimum 512MB RAM so that the application doesn't crash
- Minimum Storage of 100MB to store downloaded files

4.3.3. Non- Functional Requirements

- Availability: This app is available for all android devices. Soon it will be available on Play Store
- Maintainability: This app doesn't require any maintenance as the server is hosted on Firebase provided for free by Google.

- Usability: It is easy to use the UI/UX as it is extremely responsive interpretable.
- Optimization: The storage is optimized and the notes gets deleted after 18 months automatically.
- Efficiency: The file compressing algorithm used is extremely optimised and hence consumes less storage

4.3.4. Logical Database Requirements

The database is managed automatically by Google.

4.3.5. Design Constraints

Design constraints for Mobile version (Android) are layout, dimensions of the device, content of page, menu bars, colour formatting of contents, navigating through the pages.

4.3.6. Maintainability

It has low maintenance for a developer as the server is scaled automatically by Firebase. The storage is also optimized at regular interval to reduce the problems related to storage usage.

4.3.7. Portability:

Since, Now-a-days Electronic device are not limited to adults or particular age group. Teenagers also allowed to use smart phones without constraints. So, our app is a smart phone app which can easily carried out from one place to another without missing any important details or notifications to parents.

5. UML Diagrams

5.1. Use Case Diagrams

Figure 1

5.2. Class Diagram

Figure 2

5.3. Sequence Diagram

Figure 3

Figure 4

File Upload Flow

Figure 4

List Notes Flow

Figure 5

6. User Interface:

6.1. Layout Designs:

6.1.1. Login Page

This page asks the user to login using email and password.

The image shows a mobile app interface for the 'ShareNotes' application. At the top, there is a blue header bar with the text 'ShareNotes'. Below the header, the screen is white. In the center, there are two input fields: 'Email' and 'Password', each with a small icon to its left. Below these fields is a red button labeled 'LOGIN'. At the bottom, there is a link that says 'Don't have an account? Register here'.

6.1.2. Create User Profile

The image shows a mobile app interface for the 'ShareNotes' application, specifically the registration page. At the top, there is a blue header bar with the text 'ShareNotes'. Below the header, the screen is white. On the left side, there are four input fields: 'Name', 'Email ID', 'Password', and a red button labeled 'REGISTER'. On the right side, there are four input fields: 'Select Branch', 'Enter Section', 'Select Semester', and 'Enter Passout Year', followed by a red button labeled 'START'. At the bottom, there is a link that says 'Already have an account? Login Here'.

This page asks the user to create a profile by taking their name, email, password, branch, section, semester, pass out year.

6.1.3. Home Page

This is the home page of the app which gives the users many options where they can interact with. It consists of a text area where all the notes are displayed. It also consists of a floating button which is used to upload documents.

6.1.4. Upload Page

This page asks the user to upload a notes document by taking the subject branch, section, semester and teacher name.

7. Testing

7.1 Unit Testing

Throughout the course of developing this application, after each new feature was added or changes were made I performed unit testing. Whenever I added something, I ran the app through an emulator on my laptop as well as on both of the android devices I have. I made use of both the android monitor and logcat when testing the app and this was key in helping me identify and eradicate errors. The gradle console also helped with error detection.

7.2. Backwards Compatibility Testing

The app was developed on android sdk version 29 with the minimum sdk version set at 28. The app has been tested on a number of different versions and multiple devices. Most of the features can be used from low versions and upwards. The only concern I would have is if the user's device does not have the hardware required for the step counter to work.

8. Program Code

CompleteDetailsActivity.java

```
package com.soaiter.noteshare;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.UserProfileChangeRequest;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;

import java.util.HashMap;
import java.util.Map;

public class CompleteDetailsActivity extends AppCompatActivity {
```

```

private Spinner usrBranch, usrSemester;
private EditText usrSection, usrPassYr;
private Button startBtn;
private FirebaseFirestore db;
private FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_complete_details);
    final String usrName = getIntent().getStringExtra("user_name");
    final String usrEmail = getIntent().getStringExtra("user_email");
    setupUIViews();

    db = FirebaseFirestore.getInstance();

    startBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (validate()){
                /*
                String temp
                =usrBranch.getSelectedItem().toString()+usrSection.getText().toString()+usrSe
                mester.getSelectedItem().toString()+usrPassYr.getText().toString();
                Toast.makeText(getApplicationContext(),temp,Toast.LENGTH_SHORT).show();
                */
                String uid = user.getId();
                UserProfileChangeRequest profileUpdate = new
                UserProfileChangeRequest.Builder().setDisplayName(usrName).build();
                user.updateProfile(profileUpdate);
                Map<String,Object> user = new HashMap<>();
                user.put("UserName",usrName);
                user.put("UserEmail",usrEmail);
                user.put("UserBranch",usrBranch.getSelectedItem().toString());
                user.put("UserSection",usrSection.getText().toString());
                user.put("UserSemester",usrSemester.getSelectedItem().toString());
                user.put("UserPassoutYear",usrPassYr.getText().toString());
                user.put("uid",uid);

                db.collection("users").add(user).addOnSuccessListener(new
                OnSuccessListener<DocumentReference>() {
                    @Override
                    public void onSuccess(DocumentReference documentReference) {
                        Log.d("userdetail","DocumentSnapshot added with ID: " +
                        documentReference.getId());
                        startActivity(new Intent(CompleteDetailsActivity.this,Homepage.class));
                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override

```

```

public void onFailure(@NonNull Exception e) {
    Log.w("Userdetail","Error",e);
}
});
}
}
});

}

private void setupUIViews(){
    usrBranch = (Spinner)findViewById(R.id.usrBranch);
    usrSection = (EditText)findViewById(R.id.usrSection);
    usrSemester = (Spinner)findViewById(R.id.usrSemester);
    usrPassYr = (EditText)findViewById(R.id.usrPassYr);
    startBtn = (Button)findViewById(R.id.startBtn);

    ArrayAdapter<CharSequence> branchAdapter =
    ArrayAdapter.createFromResource(this,R.array.branchNames,
    android.R.layout.simple_spinner_item);
    branchAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    usrBranch.setAdapter(branchAdapter);

    ArrayAdapter<CharSequence> semesterAdapter =
    ArrayAdapter.createFromResource(this,R.array.semesterNames,
    android.R.layout.simple_spinner_item);
    semesterAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    usrSemester.setAdapter(semesterAdapter);

}

private Boolean validate(){
    Boolean result = false;
    String branch = usrBranch.getSelectedItem().toString();
    String section = usrSection.getText().toString();
    String semester = usrSemester.getSelectedItem().toString();
    String passsYear = usrPassYr.getText().toString();

    if (branch.equals("Select Branch") && section.isEmpty() &&
    semester.equals("Select Semester") && passsYear.isEmpty())
    Toast.makeText(this,"Please Enter All The
    Details",Toast.LENGTH_SHORT).show();
    else
    result = true;

    return result;
}

```

```
}
```

Homepage.java

```
package com.soaiter.noteshare;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.text.method.LinkMovementMethod;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.data.DataBufferRef;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.CollectionReference;
import com.google.firebase.firestore.DocumentChange;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.firestore.Query;
import com.google.firebase.firestore.QuerySnapshot;

import java.util.ArrayList;
import java.util.List;

import javax.annotation.Nullable;

public class Homepage extends AppCompatActivity {

    private RecyclerView noteList;
    private FirebaseFirestore db;
    private Button download;
    private TextView urldemo;
    private String branch, semester;
```

```

private NoteListAdapter noteListAdapter;
private List<Notes> notesList;
private FirebaseAuth.getInstance().getCurrentUser();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_homepage);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    notesList = new ArrayList<>();
    noteListAdapter = new NoteListAdapter(notesList);
    noteList = (RecyclerView)findViewById(R.id.noteList);
    download =(Button)findViewById(R.id.download);
    urldemo =(TextView)findViewById(R.id.urldemo);
    db = FirebaseFirestore.getInstance();
    noteList.setLayoutManager(new LinearLayoutManager(Hompage.this));
    noteList.setAdapter(noteListAdapter);

    String uid = user.getId();

    db.collection("users").whereEqualTo("uid",uid.toString()).get().addOnCompleteListener
    Listener(new OnCompleteListener<QuerySnapshot>() {
    @Override
    public void onComplete(@NonNull Task<QuerySnapshot> task) {
        if (task.isSuccessful()){
            for (DocumentSnapshot document : task.getResult()) {
                //Toast.makeText(Hompage.this, document.get("UserBranch").toString(),
                Toast.LENGTH_SHORT).show();
                branch = document.get("UserBranch").toString();
                semester = document.get("UserSemester").toString();
                Toast.makeText(Hompage.this,branch,Toast.LENGTH_SHORT).show();
                Toast.makeText(Hompage.this,semester,Toast.LENGTH_SHORT).show();
                db.collection("notes").document(branch).collection(semester).addSnapshotListe
                ner(new EventListener<QuerySnapshot>() {
                @Override
                public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable
                FirebaseFirestoreException e) {
                    if (e != null){
                        Log.d("Test","Error: "+e.getMessage());
                    }
                    for (DocumentChange doc: queryDocumentSnapshots.getDocumentChanges()){
                        if(doc.getType() == DocumentChange.Type.ADDED){
                            Notes note = doc.getDocument().toObject(Notes.class);
                            notesList.add(note);

                            noteListAdapter.notifyDataSetChanged();

```

```

}
}
}
});
}
}else{
Toast.makeText(Hompage.this, "error", Toast.LENGTH_SHORT).show();
}
}
});

```

```

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Toast.makeText(Hompage.this,"Clicked",Toast.LENGTH_SHORT).show();
Intent intent = new Intent(Hompage.this, Upload.class);
startActivity(intent);
}
});

```

```

}

```

MainActivity.java

```

package com.soaiter.noteshare;

import android.app.ProgressDialog;
import android.content.Intent;
import android.support.annotation.NonNull;
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 com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

```

```

public class MainActivity extends AppCompatActivity {

    private EditText usrEmail, usrPswd;
    private Button loginBtn;
    private TextView regLink;
    private FirebaseAuth firebaseAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setupUIViews();
        firebaseAuth = FirebaseAuth.getInstance();
        final ProgressDialog progressBar = new ProgressDialog(this);
        progressBar.setCancelable(true); //you can cancel it by pressing back button
        progressBar.setMessage("Logging In ...");
        progressBar.setProgressStyle(ProgressDialog.STYLE_SPINNER);

        loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (validate()){
                    progressBar.show();
                    final String email = usrEmail.getText().toString().trim();
                    final String password = usrPswd.getText().toString().trim();

                    firebaseAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener
                    (MainActivity.this, new OnCompleteListener<AuthResult>() {
                        @Override
                        public void onComplete(@NonNull Task<AuthResult> task) {
                            progressBar.dismiss();
                            if (task.isSuccessful()){
                                Log.d("SignIn Process", "Sucess");
                                FirebaseUser user = firebaseAuth.getCurrentUser();
                                startActivity(new Intent(MainActivity.this,Homepage.class));
                            }
                            else{
                                Log.w("SignIn Process",task.getException());
                                Toast.makeText(MainActivity.this,"Failed",Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                }
                else{
                    Toast.makeText(MainActivity.this,"Bahut Error",Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}

```



```

regLink.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
startActivity(new Intent(MainActivity.this , RegistrationActivity.class));
}
});
}

```

```

private void setupUIViews(){
usrEmail = (EditText)findViewById(R.id.usrEmail);
usrPswd = (EditText)findViewById(R.id.usrPswd);
loginBtn = (Button)findViewById(R.id.loginBtn);
regLink = (TextView)findViewById(R.id.regLink);
}

```

```

private Boolean validate(){
Boolean result = false;
String email = usrEmail.getText().toString();
String password = usrPswd.getText().toString();

```

```

if (password.isEmpty() && email.isEmpty())
Toast.makeText(this,"Please Enter All The
Deatils",Toast.LENGTH_SHORT).show();
else
result = true;

```

```

return result;
}
}

```

NoteListAdapter.java

```

package com.soaiter.noteshare;

```

```

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;

```

```

import java.util.List;

```

```

public class NoteListAdapter extends
RecyclerView.Adapter<NoteListAdapter.ViewHolder> {

```

```

public List<Notes> notesList;
public String url;
private Context context;

public NoteListAdapter(List<Notes> notesList){
    this.notesList = notesList;
}

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
    View view =
        LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.list_note,viewGr
        oup,false);
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull final ViewHolder viewHolder, int i) {

    viewHolder.noteSubject.setText(notesList.get(i).getNoteSubject());
    viewHolder.noteBranch.setText(notesList.get(i).getNoteBranch());
    viewHolder.noteSemester.setText(notesList.get(i).getNoteSemester());
    viewHolder.urldemo.setText(notesList.get(i).getDownloadURL());
    final String url = viewHolder.urldemo.getText().toString();
    viewHolder.download.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Intent.ACTION_VIEW);
            intent.setData(Uri.parse(url));
            context.startActivity(intent);
        }
    });

}

@Override
public int getItemCount() {
    return notesList.size();
}

public class ViewHolder extends RecyclerView.ViewHolder{

    View mView;

    public TextView noteSubject, noteBranch, noteSemester,urldemo;
    public Button download;

```

```

public ViewHolder(@NonNull View itemView) {
    super(itemView);

    mView = itemView;
    context = itemView.getContext();

    noteSubject = (TextView)mView.findViewById(R.id.noteSubject);
    noteBranch = (TextView)mView.findViewById(R.id.noteBranch);
    noteSemester = (TextView)mView.findViewById(R.id.noteSemester);
    download = (Button)mView.findViewById(R.id.download);
    urldemo = (TextView)mView.findViewById(R.id.urldemo);
}
}
}

```

Notes.java

```

package com.soaiter.noteshare;

public class Notes {

    String noteSubject, noteSemester, noteBranch, downloadURL;

    public Notes(){

    }

    public Notes(String noteSubject, String noteSemester, String noteBranch,
String downloadURL){
        this.noteSubject = noteSubject;
        this.noteSemester = noteSemester;
        this.noteBranch = noteBranch;
        this.downloadURL = downloadURL;
    }

    public String getNoteSubject() {
        return noteSubject;
    }

    public void setNoteSubject(String noteSubject) {
        this.noteSubject = noteSubject;
    }

    public String getNoteSemester() {
        return noteSemester;
    }

    public void setNoteSemester(String noteSemester) {
        this.noteSemester = noteSemester;
    }
}

```

```

}

public String getNoteBranch() {
    return noteBranch;
}

public void setNoteBranch(String noteBranch) {
    this.noteBranch = noteBranch;
}

public String getDownloadURL() {
    return downloadURL;
}

public void setDownloadURL(String downloadURL) {
    this.downloadURL = downloadURL;
}

```

RegistrationActivity.java

```

package com.soaiter.noteshare;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class RegistrationActivity extends AppCompatActivity {

    private EditText usrName, usrEmail, usrPswd;
    private Button regBtn;
    private TextView loginLink;
    private FirebaseAuth firebaseAuth;

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

```

```

firebaseAuth = FirebaseAuth.getInstance();

regBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
if (validate()){
//update on db
final String user_name = usrName.getText().toString();
final String user_email = usrEmail.getText().toString().trim();
String user_password = usrPswd.getText().toString().trim();

firebaseAuth.createUserWithEmailAndPassword(user_email,user_password).addOnCo
mpleteListener(new OnCompleteListener<AuthResult>() {
@Override
public void onComplete(@NonNull Task<AuthResult> task) {
if (task.isSuccessful()) {
Toast.makeText(RegistrationActivity.this, "Registration Successful",
Toast.LENGTH_SHORT).show();
Intent intent = new Intent(RegistrationActivity.this,
CompleteDetailsActivity.class);
intent.putExtra("user_name",user_name);
intent.putExtra("user_email",user_email);
startActivity(intent);
}
else{
Toast.makeText(RegistrationActivity.this, "Registration Failed",
Toast.LENGTH_SHORT).show();
}
}
});
}
}
});

loginLink.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
startActivity(new Intent(RegistrationActivity.this , MainActivity.class));
}
});
}

private void setupUIViews(){
usrName = (EditText)findViewById(R.id.usrName);
usrEmail = (EditText)findViewById(R.id.usrEmail);
usrPswd = (EditText)findViewById(R.id.usrPswd);
regBtn = (Button)findViewById(R.id.regBtn);
loginLink = (TextView)findViewById(R.id.loginLink);
}

```

```

private Boolean validate(){
Boolean result = false;
String name = usrName.getText().toString();
String email = usrEmail.getText().toString();
String password = usrPswd.getText().toString();

if (name.isEmpty() && password.isEmpty() && email.isEmpty())
Toast.makeText(this,"Please Enter All The
Deatils",Toast.LENGTH_SHORT).show();
else
result  = true;

return result;
}
}

```

Upload.java

```

package com.soaiter.noteshare;

import android.app.ProgressDialog;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.provider.OpenableColumns;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.io.File;
import java.util.HashMap;

```

```

import java.util.Map;

public class Upload extends AppCompatActivity {

    private Spinner noteBranch , noteSemester;
    private EditText noteSubject , noteTeacher , noteSection;
    private FirebaseStorage storage= FirebaseStorage.getInstance();
    private FirebaseFirestore db = FirebaseFirestore.getInstance();
    private FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    private StorageReference filepath;
    private Button selectNote , upload;
    private TextView selectedNote;
    private Uri uri;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_upload);
        setupUIViews();
        final ProgressDialog progressBar = new ProgressDialog(this);
        progressBar.setCancelable(true);//you can cancel it by pressing back button
        progressBar.setMessage("File uploading ...");
        progressBar.setProgressStyle(ProgressDialog.STYLE_SPINNER);

        selectNote.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent();
                intent.setAction(Intent.ACTION_GET_CONTENT);
                intent.setType("application/pdf");
                startActivityForResult(intent,1212);
            }
        });

        upload.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (validate()){
                    progressBar.show();
                    filepath.putFile(uri).addOnSuccessListener(new
                    OnSuccessListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                            filepath.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                                @Override
                                public void onSuccess(Uri uri) {
                                    Uri downloadUrl = uri;
                                    String name = user.getDisplayName();

```

```

Map<String,Object> note = new HashMap<>();
note.put("noteSubject",noteSubject.getText().toString());
note.put("noteBranch",noteBranch.getSelectedItem().toString());
note.put("noteSemester",noteSemester.getSelectedItem().toString());
note.put("noteTeacher",noteTeacher.getText().toString());
note.put("downloadURL",downloadUrl.toString());
note.put("studentName",user.getDisplayName().toString());

db.collection("notes").document(noteBranch.getSelectedItem().toString()).collection(noteSemester.getSelectedItem().toString()).add(note).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
@Override
public void onSuccess(DocumentReference documentReference) {
Toast.makeText(Upload.this, " Added Success" , Toast.LENGTH_SHORT).show();
startActivity(new Intent(Upload.this,Homepage.class));
}
}).addOnFailureListener(new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception e) {
Toast.makeText(Upload.this, " Upload Failed" , Toast.LENGTH_SHORT).show();
}
});
}
});
progressBar.dismiss();
Toast.makeText(Upload.this, " Upload Success" , Toast.LENGTH_SHORT).show();
}
}).addOnFailureListener(new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception e) {
Toast.makeText(Upload.this,"Upload Failure", Toast.LENGTH_SHORT).show();
}
});
}
}
});
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
switch (requestCode) {
case 1212:
if (resultCode == RESULT_OK) {
// Get the Uri of the selected file
uri = data.getData();
filepath =
storage.getReference().child(noteBranch.getSelectedItem().toString().trim()).

```



```

child(noteSemester.getSelectedItem().toString().trim()).child(uri.getLastPath
Segment());
String uriString = uri.toString();
File myFile = new File(uriString);
String path = myFile.getAbsolutePath();
String displayName = null;

if (uriString.startsWith("content://")) {
Cursor cursor = null;
try {
cursor = Upload.this.getContentResolver().query(uri, null, null, null, null);
if (cursor != null && cursor.moveToFirst()) {
displayName =
cursor.getString(cursor.getColumnIndex(OpenableColumns.DISPLAY_NAME));
}
} finally {
cursor.close();
}
} else if (uriString.startsWith("file://")) {
displayName = myFile.getName();
}
selectedNote.setText(displayName);
}
break;

}
super.onActivityResult(requestCode, resultCode, data);

}

```

```

private void setupUIViews(){
noteBranch = (Spinner)findViewById(R.id.noteBranch);
noteSemester = (Spinner)findViewById(R.id.noteSemester);
noteSubject = (EditText)findViewById(R.id.noteSubject);
noteTeacher = (EditText)findViewById(R.id.noteTeacher);
noteSection = (EditText)findViewById(R.id.noteSection);
selectNote = (Button)findViewById(R.id.selectNote);
selectedNote = (TextView)findViewById(R.id.selectedNote);
upload = (Button)findViewById(R.id.upload);

ArrayAdapter<CharSequence> branchAdapter =
ArrayAdapter.createFromResource(this,R.array.branchNames,
android.R.layout.simple_spinner_item);
branchAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown
n_item);
noteBranch.setAdapter(branchAdapter);
}

```

```

ArrayAdapter<CharSequence> semesterAdapter =
ArrayAdapter.createFromResource(this,R.array.semesterNames,
android.R.layout.simple_spinner_item);
semesterAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropd
own_item);
noteSemester.setAdapter(semesterAdapter);
}

private Boolean validate(){
Boolean result = false;
String Branch = noteBranch.getSelectedItem().toString().trim();
String Semester = noteSemester.getSelectedItem().toString().trim();
String Subject = noteSubject.getText().toString();
String Section = noteSection.getText().toString();
String SelectedNote = selectedNote.getText().toString();

if (Branch.equals("SelectBranch") && Section.isEmpty() &&
Semester.equals("SelectSemester") && Subject.isEmpty() &&
SelectedNote.equals(""))
Toast.makeText(this,"Please Enter All The
Details",Toast.LENGTH_SHORT).show();
else
result = true;

return result;
}
}

```

9. Conclusion

Students often face problem while collecting notes and it interrupts their learning process.

Our application will provide a platform for the students where they can come together for collaborative learning.

It is an easy to host application that can be swiftly implemented using firebase.

This app can be easily scaled with very less cost to promote collaborative learning

Future Work

We aim to extend this app beyond Note Sharing to promote collaborations not only between students but also between teachers and management by adding different features such as notice board, reviews, feedbacks etc.

References

<https://www.tutorialspoint.com/android/index.htm>

<https://developer.android.com/guide>