



---

## CONTENT REPOSITORY APPLICATION USING ANDROID

**Bhargav D. Kamble<sup>a</sup>, Mohamadjaid J. Momin<sup>a</sup>, Abdulrahaman S. Nayakawadi<sup>a</sup>, Akshay B. Powar<sup>a</sup>, Mrs. Sandhya P. Mohite<sup>b</sup>**

<sup>a</sup>Student, Sanjay Ghodawat Polytechnic, Atigre, 416118, India

<sup>b</sup>Guide, Sanjay Ghodawat Polytechnic, Atigre, 416118, India.

---

### ABSTRACT

Nowadays, a great deal of people use smart phones. Particularly, Android based smart phones become more and more prevalent in the smart phone markets. Most people use their smart phones longer than other devices. As a result, users frequently share contents and the needs of file sharing via smart phones have been increased. As a result, the need for file sharing for mobiles has increased. File sharing has emerged as one of the most important feature in mobiles. Sharing files or storing data on the internet for personal or professional use, also called content repository.

It allows users to easily collaborate and share written document, spreadsheet, or PDF and find content within the system. This paper presents an overview of the online file sharing in android phones and incorporates with Firebase Realtime Database which can be used to Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.

**Keywords:** File, Sharing, Android, Firebase, Content Repository.

---

## 1. INTRODUCTION

### 1.1 Background

The performance of mobile devices, especially smart phones, has been enhanced considerably for the last few years. Nowadays, a great deal of people use smart phones. Particularly, Android based smart phones become more and more prevalent in the smart phone markets. Most people use their smart phones longer than other devices. As a result, users frequently share contents and the needs of file sharing via smart phones have been increased. There is a various kind of file sharing applications out there in the internet. Such as Send anywhere, Sharely, WhatsApp, telegram etc. since telegram is known as good security holder among all still it got some disadvantages. As a file sharing app carries a lot of important features, it should hold a good security for the files for the users, easy to use, file should be transferred within minimal time etc.

### 1.2 Environment

Most mobile file sharing is performed either on the local or on the web. The local are quite reliable, but users have to be together to share files. The web based file sharing applications have some disadvantages like files can't be viewed before downloading, having a long account creation process, password protected file sharing etc. In order to overcome these problems, we propose a content repository application for the Android devices.

### 1.3 Use of Project

- a) Share files over the Internet.
- b) Share files with extra information like title description.
- c) View Basic type files before downloading.
- d) Password Protected file sharing

### 1.4 Purpose

- a) To create specific app for file sharing.
- b) Reduce human Efforts

- c) .More Understandable file sharing.
- d) Credential less account.
- e) Create account in just 2 steps.
- f) View files before downloading.

---

## 2. LITERATURE REVIEW

### 2.1 DESING AND IMPLEMENTATION OF A FILE SHARING APPLICATION FOR ANDROID.

Use of paper is increasing day by day. To make 1 ton of paper, many big and small trees are being cut down. About 20,000,000 trees are cut down annually for virgin paper, which are used for production of books. Over 30 million acres of forest are destroyed annually. The entire process of creating paper is damaging the environment. It starts off with a tree being cut down and ends its life by being burned, emitting carbon dioxide into the atmosphere. Chopping down trees decreases the amount of oxygen in the ecosystem, thus adversely affecting all other lifeforms. To avoid these circumstances and reduce the use of paper this android application can be used. [8]

### 2.2 BLOCK CODING

Over the last few years, there has been a drastic change in information technology. This includes the various ways in which files can be shared and stored. Cloud computing is publicized as the next major step for all forms of typical information technology use. From businesses, to nonprofit organizations, to single users, there seems to be various applications which can use cloud computing to offer better, faster, and smarter computing. Android Operating System is a relatively a mobile Operating System which has been steadily taking over more and more market stake. Easy to use, easy to develop for, and open source, it has picked up a following of developers who want to create content for the masses. This project aims to combine the two, building a cloud based application for Android, offering users the power of cloud computing in the palm of their hand for file sharing and collaboration.

### 2.3 ANDROID-BASED CHAT APPLICATION USING FIREBASE

Today, paperless (and even virtual) offices are taking file sharing even further. Internet users are communicating through sharing entire folders of information online, and trusting these online platforms as their primary means of document storage. During the Internet's infancy, before it was named the "Internet" it was referred to as ARPANET and file sharing was a practice reserved only for the most tech understanding of computer users. File sharing was also really considered more file transferring, as it usually consisted of manually transferring files with a technological medium like a floppy disc. [7]

### 2.4 GETTING STARTED WITH SKETCHWARE

Sketchware is a FREE Scratch based integrated development environment (IDE) for developing real android apps. Sketchware uses block language like Scratch, an innovative programming language invented by MIT that transforms the complex language of text-based coding into visual, dragand- drop building blocks. Scratch is such a simple and understandable language, even little children can learn to develop; however, Sketchware can translate your project into Java and XML source codes. among mobile devices [2] [4] [6]. Block-based programming emerged as the best solution to the above problem. The idea is to create a code in a way that is both visual (simple) and similar to traditional text-based coding (powerful). All the developer has to do is to connect visual "building blocks" in a logical way. Block-based coding is a form of programming language where the developer issues instructions by dragging and dropping blocks. This helps to prevent syntax errors and developers do not have to memorize syntax to write code. For example, in text-based language, missing a semicolon will result in an error message or prevent the program from running at all. These types of errors are eliminated with block-based coding. Block coding is just as powerful and useful as textbased language, but easier to use and understand. Having to drag and drop commands is easier than typing it character-by-character with a keyboard [2] [4] [6].

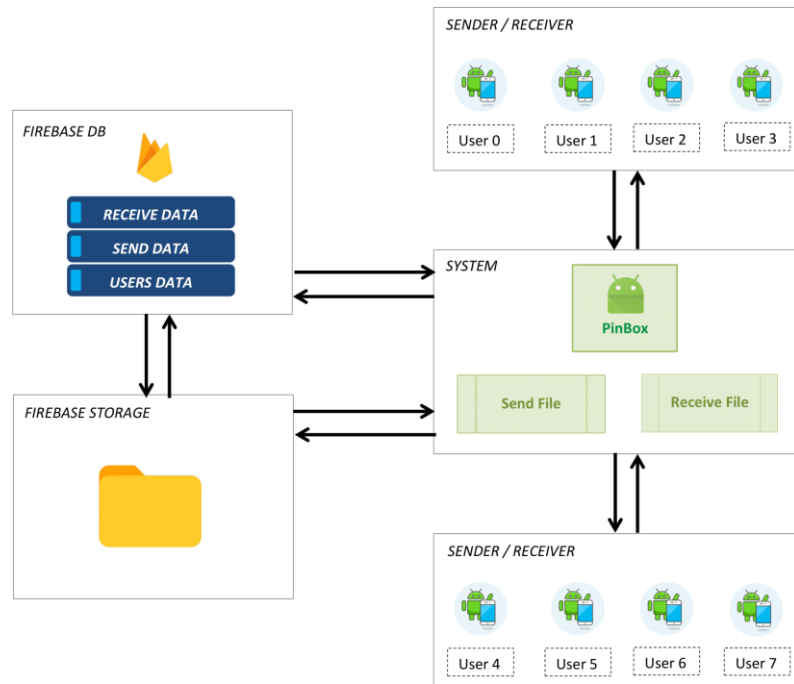
### 2.5 LEARN ABOUT BUILD PRODUCTS.

Firebase is a NoSQL database that uses connections which allows the costumers to store and recuperate the data from the database [1] [5].

- **Firestore cloud:** For taking care of the data, for instance, text, pictures, accounts developing the establishment would be irksome and exorbitant for an adolescent archict so the firebase gives the establishment of cloud storage. [1] [5]
- **Steady Database:** It is a cloud-encouraged NoSQL informational collection. Beside the approval, cloud organization and steady data bases firebase also aoffers a help for crash uncovering, etc [1] [5].

---

## 3. SYSTEM ARCHITECTURE



**Fig. 3.1 – System Architecture.**

### 3.1 MODULE 01: AUTHENTICATION

Authentication is a first step in most of the applications, from where authorized users only can use the system. For this system also authentication is installed but without taking any credentials from user like email, phone etc. User can create account by just entering name, username and password. By using this type of authentication user can create account in just 2 minutes, not like lengthy methods like email login, verification etc. So every user is firstly navigated to sign up or login page from where user can go forward to next process.

### 3.2 MODULE 02: RECEIVE BOX AND SEND BOX

After the successful authentication next user is navigated to the main screen from where user can receive or send the files to other users. There are two types of boxes visible in this module which are:

- RECEIVE BOX this is a simple and attractive view in which received files are displayed to the user.
- SEND BOX this is a simple and attractive view in which to whom files were sent and file information like seen, downloaded or not etc. is visible. Also from this view user is able to do file sharing.

### 3.3 MODULE 03: RECEIVED OR SENT FILE VIEW

In this view user is able to see received or sent file. This view includes sender or receiver name, date, file title, description and live preview feature. The most important feature of this is the Live Preview, through which the user can view the file without downloading it, even if he wants to download it.

### 3.4 MODULE 04: FILE SHARING

In this module user has to answer the 3 or 4 questions then only user can share the files. Let's see which are those questions:

- Username User has to enter the username of target person to whom file is being shared. This function is simplified for user, so user must know 2 or 3 letters of target user and afterwards system shows target user suggestions to the user.
- Title Through the title, the user can tell the front what kind of file it is, which is an important feature.
- Description What's even better is that the user can describe the file in detail like a live guide.
- Downloadable and Password Protection Through this function the user can tell whether the next one can download this file or not and can also send the file by adding the password which is a type of one security.

### 3.5 ADVANCED FILE VISIBILITIES

Through this function the sender can know whether the user has viewed or downloaded the file or not. This function is implemented in the send box, which should be done right there. This is also an important feature that clears up a lot of confusions between sender and the receiver.

### 3.6 LIVE PREVIEW

This feature can only run on basic file types, through which you can cover the file without downloading it, which is a time-saving as well as memory-saving feature. Time saving because there is no need to download the file nor need a separate third party app to view it. And on this basis the user can know whether there is a need to download this file or not.

### 3.7 FILE MANAGER

The app also has a built in file manager, through which the user can select and share files. User can share only one file at a time and can share with only one user

## 4. SYSTEM CONFIGURATION

### 4.1 Software Configuration

- OPERATING SYSTEM : ANDROID V5.0 OR HIGHER.
- API LEVEL : MIN - 21 & MAX - 29
- FRONT-END LANGUAGES : XML. BACK-END
- LANGUAGES : JAVA. DATABASE : FIREBASE.

### 4.2 Hardware Configuration

- RAM : 1GB MINIMUM.
- STORAGE : 1GB MINIMUM.
- OTHER : INTERNET CONNECTION.

## 5. UML DIAGRAMS

### 5.1 Flowchart

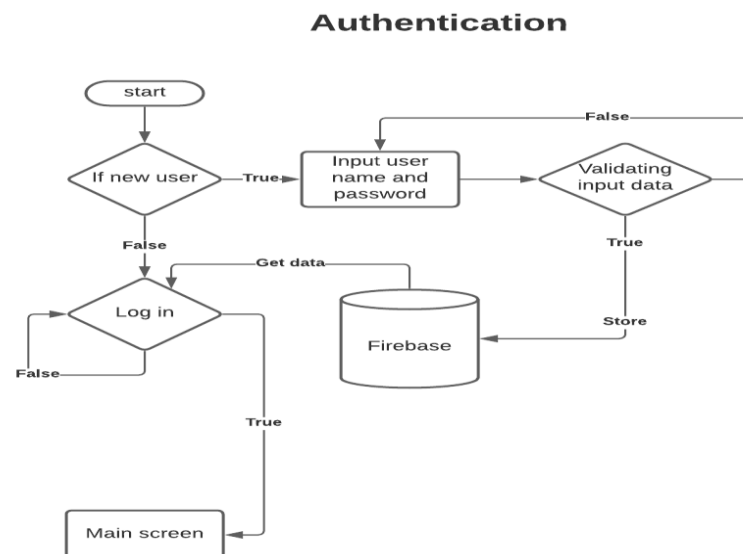


Fig. 5.1.1 – Authentication Process.

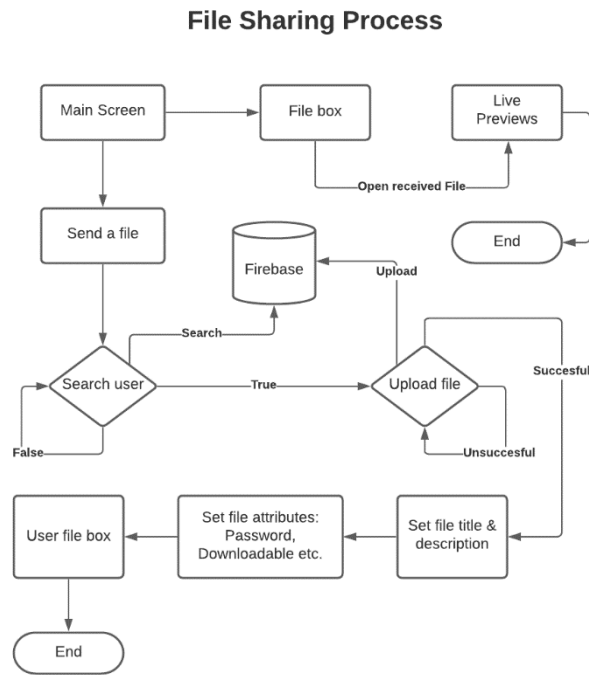


Fig. 5.1.2 – File Sharing Process.

5.2 Data Flow

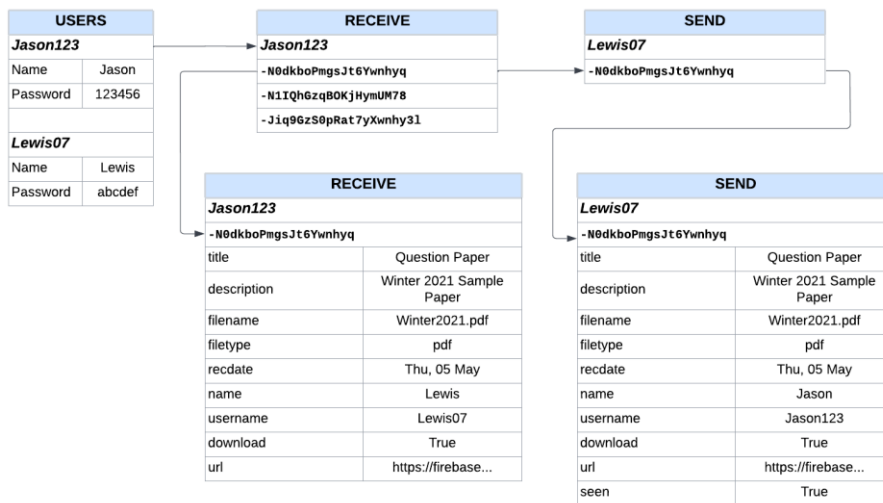
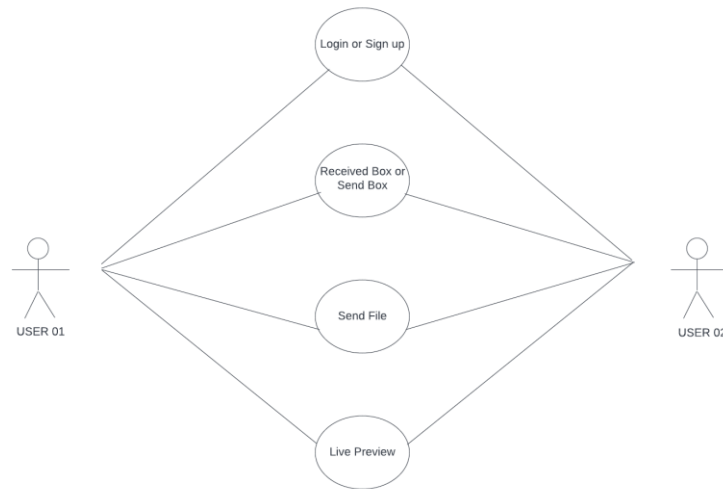


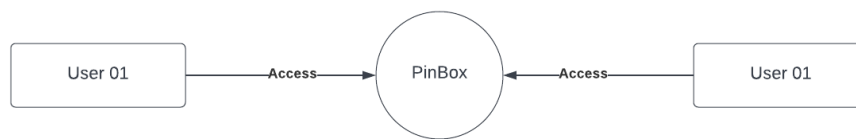
Fig. 5.2.1 – Data Flow.

**5.3 Use Case Diagram**

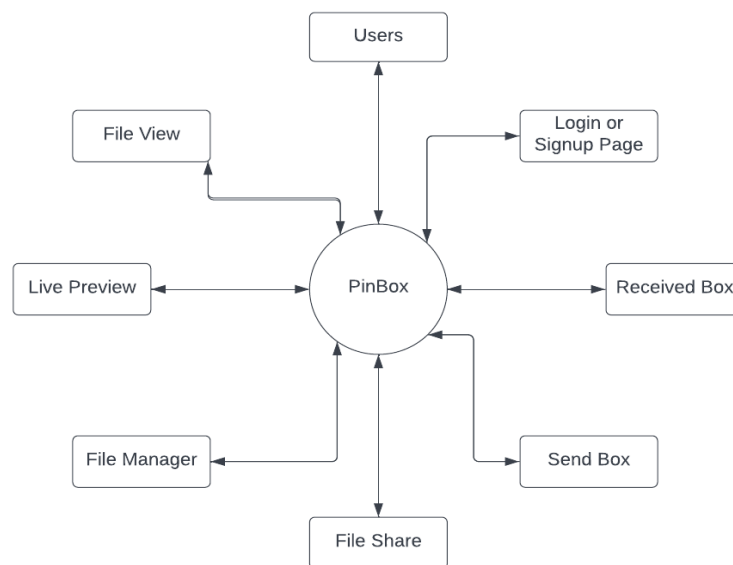


**Fig. 5.3.1 – Use Case Diagram.**

**5.4 Data Flow Diagrams**



**Fig. 5.4.1 – DFD Level 0.**



**Fig. 5.4.2 –DFD Level 1.**

## 6. SYSTEM CODING

### 6.1 File Sharing Module

```

_storage_upload_success_listener = new OnCompleteListener<Uri>() {
    @Override
    public void onComplete(Task<Uri> _param) {
        final String _downloadUrl = _param.getResult().toString();
        map.put("url", _downloadUrl);
        key = database.push().getKey();
        database.child(edittext.getText().toString()).child(key).updateChildren(map);

        map.remove("username");
        map.remove("name");
        map.put("username", edittext.getText().toString());
        map.put("name", receiverName);
        database.child(sp.getString("username", "")).child(key).updateChildren(map);

        pd.dismiss();
        Toast t = Toast.makeText(getApplicationContext(), "The file
        \".concat(Uri.parse(getIntent().getStringExtra("file"))
        .getLastPathSegment().concat("\") has been successfully sent to
        ").concat(edittext.getText().toString()), Toast.LENGTH_LONG);
        t.show();
        i.setClass(getApplicationContext(), HomeActivity.class);
        startActivity(i);
        finish();
    }
};

```

### 6.2 Target Username Suggestion

```

database.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot _dataSnapshot) {
        try {
            for (DataSnapshot _data : _dataSnapshot.getChildren()) {
                if(_data.getKey().equals(sp.getString("username", null))){
                    continue;
                }
                if(!users.contains(_data.getKey())){
                    users.add(_data.getKey().toString());
                }
            }
            catch (Exception _e) {
                _e.printStackTrace();
            }
        }
    }
    @Override
    public void onCancelled(DatabaseError _databaseError) {
    }
});
edittext.setAdapter(new ArrayAdapter<String>(getBaseContext(),
android.R.layout.simple_list_item_, users));

```

### 6.3 Network Module

```

public void _networkError() {
    d.setTitle(Html.fromHtml("<b><font color=#F>Network
    Error</font></b>"));
    d.setMessage(Html.fromHtml("<font>You are not connected to the

```

```

internet.</font>"));
d.setPositiveButton(
Html.fromHtml("<font color=#FB>Retry</font>")
, new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface _dialog, int _which) {
i.setClass(getApplicationContext(), MainActivity.class);
startActivity(i);
finish();
}
});
d.setNegativeButton(
Html.fromHtml("<font color=#FB>Exit</font>")
, new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface _dialog, int _which) {
finishAffinity();
}
});
d.setCancelable(false);
d.create().show();}

```

## 7. RESULT OF PROJECT



Fig. 6.1 – Splash Screen.

Fig. 6.2 – Sign Up Screen.

Fig. 6.3 – Log In Screen.



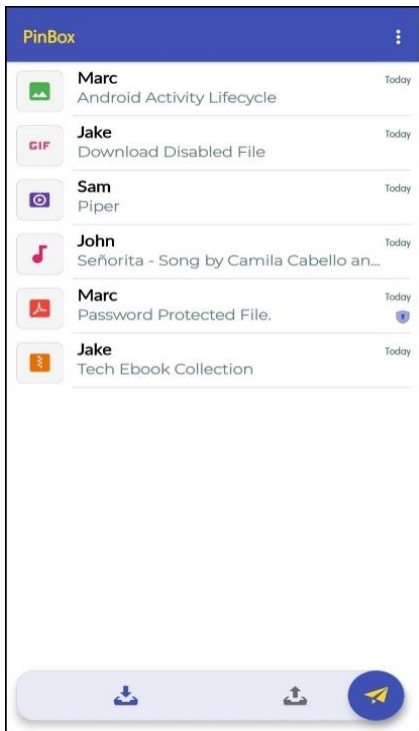


Fig. 6.4 – Received File Box.



Fig. 6.5 – Sended File Box.

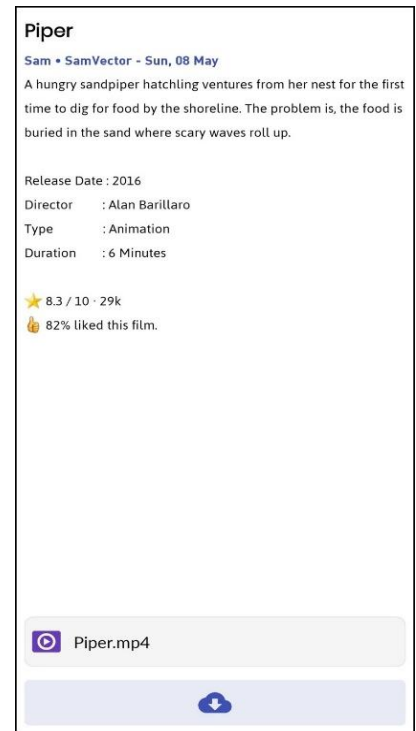


Fig. 6.6 – Received File View

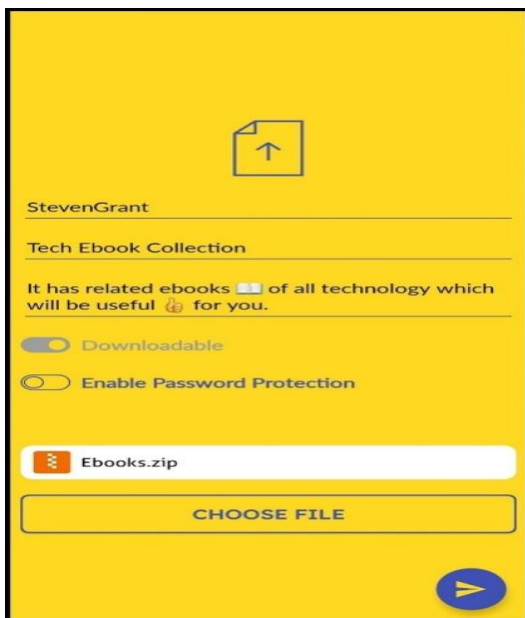


Fig. 6.1 – Send File View.

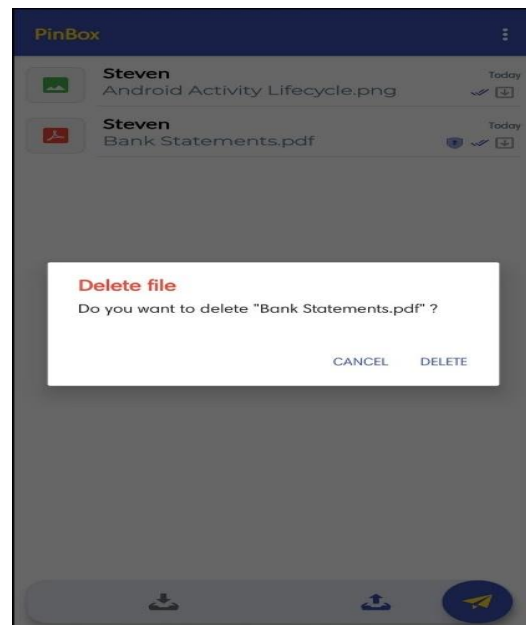


Fig. 6.2 – Delete File View.

---

## 8. CONCLUSION

This system has been made to overcome the shortcomings of the previous system, which has happened to a great extent. Like live preview for basic files, sharing file which is binded with title and description, password protected file sharing etc.

### Future Enhancements

- Better Accessibility.
- fast search and retrieval of documents.
- Live previews for most of file types.
- Encrypted file sharing.
- Transferring large files quicker.
- Continuous file sharing, even with poor connectivity.

### REFERENCES

---

- [1] Android Developers, "Build Your First Android App in Java". Available: <https://developer.android.com/codelabs/build-your-first-android-app0>
- [2] Sketchware Developers, "Getting Started". Available: <https://docs.sketchware.io/docs/gettingstarted.html>
- [3] Lahiri Kolli, M. Swathi, Kirankumar, K. Shree Krishna, Dr. H Girisha, "Document Sharing Android Project". IJERCSE 4, April 2018.
- [4] Sanskar Shukla, Subhash Chandra Gupta and Praveen Mishra, "Android-Based Chat Application Using Firebase". January 27-19, 2021.
- [5] Sung Park. "Must-read Guide for New Sketchware Developers". Available: <https://medium.com/sketchware/must-read-guide-for-new-sketchware-developers-231aa9e10358>
- [6] Firebase Developers. "Learn about build products". Available: <https://firebase.google.com/docs/build>
- [7] Codejig Developers, "Block Coding". Available: <https://www.codejig.com/en/blockbasedcoding/>
- [8] Chrisantus Oden, "Design and Implementation of a File Sharing Application for Android". Available: <https://www.projecttopics.org/design-implementation-file-sharing-applicationandroid.html>
- [9] Neon Projects Developers, "Android Paper Free Document Sharing App Project". Available: <https://nevonprojects.com/android-paper-free-document-sharing-app-project>