ANDROID DATA RECOVERY TOOL

By

CAPT SAFIULLAH SHAH

CAPT USAMA HAMEED TARAR

CAPT KAMRAN SAEED

Supervised by:

MAJ AMMAR HASSAN (Dept. of IS)

Submitted to the faculty of Department of Information Security,
Military College of Signals, National University of Sciences and Technology, Islamabad,
in partial fulfillment for the requirements of B.E Degree in Information Security.
June 2024

In the name of Allah, the Most Beneficent, the Most Merciful

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

“Android Data Recovery Tool”

is carried out by

Capt Safiullah Shah

Capt Usama Hameed Tarar

Capt Kamran Saeed

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Information Security in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Maj Ammar Hassan

Department of IS

Supervisor

Date: ____________

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our sincere appreciation goes to our supervisor, Maj Ammar Hassan for bearing with us and generously providing their valuable time and advice, which greatly contributed to our project's and academic success.

Our heartfelt gratitude goes to our group members for their tireless work and collaboration in bringing this project to life.

Plagiarism Certificate (Turn it in Report)

This thesis has 7% similarity index. Turn it in report endorsed by Supervisor is attached.

____

Capt Safiullah Shah

____

Capt Usama Hameed Tarar

____

Capt Kamran Saeed

____

Signature of Supervisor

ABSTRACT

In today's digital age, since cellphones are now considered essential personal and business data storage devices in the digital age, data loss is a major problem for consumers everywhere. This problem is made worse by the dearth of effective and user-friendly data recovery apps for Android smartphones, which puts users at risk of permanently losing important data. Our senior project, The Android Recovery Tool, offers a creative approach in response to this urgent problem.

Our project's main goal is to provide consumers the tools they need to easily and confidently recover lost data. Regardless of customers' level of technical expertise, the Android Recovery Tool streamlines the data recovery process with its user-friendly interface and easy connection with personal Androids. Moreover, our approach places a high priority on creating reliable recovery algorithms that can retrieve a variety of data kinds, such as documents, movies, and images.

The goal of our project is to provide an effective and dependable data recovery solution designed especially for Android devices through thorough study, development, and testing. Our works will conclude in a user-comprehensible result that gives users the capacity to safeguard their digital assets and lower the effects of data loss situations.

In summary, it is an effort to create something in the middle of feature-heavy resource hungry desktop-based data recovery tools which are very user-unfriendly. This is an effort to bridge the gap with a tool that requires very less resources and is extremely user-friendly.

Table of Contents

Chapter 1: Introduction

The growth of mobile phones corresponded with the 21st century's technological boom. In recent years, there has been a notable surge in the manufacture, development, and utilization of smartphones [1]. As a result, Google's Android operating system leads the competition: in 2015, the market shares of the Android operating system increased to 81.2%, while that of rival systems, such as Windows, IOs, and others, was 15,8%, 2,2%, and 0.8%, respectively.

But as more and more data is created and kept on Android devices, there is a growing chance that some of it will be lost due to software bugs or inadvertent deletion. This problem is made worse by the dearth of effective and user-friendly data recovery apps for Android smartphones, which puts users at risk of permanently losing important data. Although several software tools exist to address this requirement but many of them lack the efficiency, reliability or user-availability required to successfully recover lost or deleted data. This disparity in the market highlights the importance of developing a Android-based Android data recovery tool that will not only simplify the recovery process but will also improves the chances of successful data retrieval.

The proposed Android Data Recovery Tool aims to focus on these challenges by relying on advanced algorithms, intuitive user interface design, and strong performance. By thorough research, development, and testing this project will deliver a reliable and efficient data recovery result specifically for Android devices. The result of these efforts will be a user-friendly application that will allows users to protect their digital assets and reduce the impact of data loss incidents.

The main goal this thesis is to organize and examine the making of an properly named "Android Data Recovery Tool”. The grouping of cutting-edge techniques, a user-friendly interface, and strong performance, Android Data Recovery Tool presents a complete answer to the problems related to Android data recovery.

To meet the immediate need for applicable and reliable data recovery answers in the Android ecosystem. This thesis presents the development of Android Data Recovery Tool for android devices.

This goal will be achieved by the review of existing literature on Android data recovery techniques and methods. This literature review will provide understandings into the industrial background of data recovery tools.

File System Analysis: This method can locate and recover deleted files by scanning the file system of android device. When searching for deleted files that are still present on the storage medium. The file system analysis uses the file allocation table (FAT) or other metadata structures. File system analysis tools mostly dig up files by using their metadata information and rebuild the directory structure[2].

Data Carving: Data carving files can be restored from storage medium by using their primary data patterns or file signatures. This technique is useful for recovering somewhat erased or fragmented files because it does not depend on the file system metadata. Data carving tools extract and rebuild files by scanning the raw data on the storage device and identifying file signatures despite of the file system structure of the files.

Root Access and Custom Recovery: Rooting an Android smartphone gives users elevated privileges to the operating system which overcomes the limitations and gain access to system files. Users carry out sophisticated data recovery activities which include backing up and restoring system images and recovering removed contents from the internal storage of device or external SD card by using recovery software, like TWRP (Team Win Recovery Project) [3].

Backup and Restore: Data loss can be protected on Android devices by regularly backing up your data to online services or external storage. Users can do backups of their data which can include contacts, messages, pictures, videos and app data. This can be done by using the built-in backup tools of the Android operating system or third-party backup programs. Users can recover lost or deleted information by restoring their data from backups in the event of data loss.

Third-Party Data Recovery Software: For Android smartphones, there are numerous third-party data recovery software programs that include functions including file scanning, file previewing, and file recovery. These software programs recover a variety of file formats, such as images, movies, documents, and communications, by combining file system inspection, data carving, and sophisticated algorithms. To search for and retrieve deleted data from an Android device's internal storage or external SD card, users can connect their devices to a computer and launch data recovery software.

1.1 Overview

The availability of Android smartphones all around the world emphasizes the importance of the personal and professional data present on these devices. However, users face serious difficulties if they unintentionally lose such data as a result of device faults or accidental deletion. Our project offers a novel solution—a ANDROID-based Android data recovery tool—to tackle this pressing problem.

The Android system just updates and deletes the file table entry when an Android file is destroyed; it does not really wipe the data stored on the Android storage. Because the system can store new files in the space occupied by deleted ones, it is quick and effective. Android data recovery software may therefore search the Android memory for lost files and seek for deleted file data, and if such data is found [4], it can restore deleted files using the deleted file space.

This thesis project recommends creating an Android-based software program called the "Android data recovery tool" that is explicitly developed to retrieve deleted data from Android smartphones. The "Android data recovery tool" will help users who have deleted their storage or by mistake lost data.

1.2 Problem Statement

Android smartphones are very common in today's digital world and they are acting as important storage devices for both personal and business data. But the dependence on these devices has also brought a serious problem. This shortcoming develops a number of important concerns:

1.2.1 **Limited Data Recovery Success:**

Dr. Fone cannot give always assurance of successful recovery of all types of data, most importantly in situations of serious device damage or data encryption.

1.2.1 **Complex Pricing Structure**:

Dr. Fone's pricing mechanism for many modules can be problematic which can cause confusion or unexpected expenses for certain functions.

1.2.2 **Compatibility difficulties:**

Users may occasionally have compatibility difficulties with the most recent operating systems or device models, prompting upgrades or changes for optimal operation.

1.2.3 **Dependence on Regular Updates:**

Dr. Fone's ability may depend on regular updates to continue with new device technology and software changes which can cause disturbance in service or functioning if the problem is not catered for.

1.2.4 Recovery Scope:

Disk Drill version of Windows can recover data only from Android microSD cards and not from the internal memory of the Android mobile devices.

1.2.5 Limitation on Recovery Features:

Some recovery features are restricted to specific Android versions

1.2.6 Security Risks of Online Data Recovery:

One disadvantage of using online data recovery software is the greater likelihood of data loss or theft. When data recovery activities are carried out online, sensitive information is transported across the internet, leaving it vulnerable to interception by malicious actors[5].

1.2 Statistics

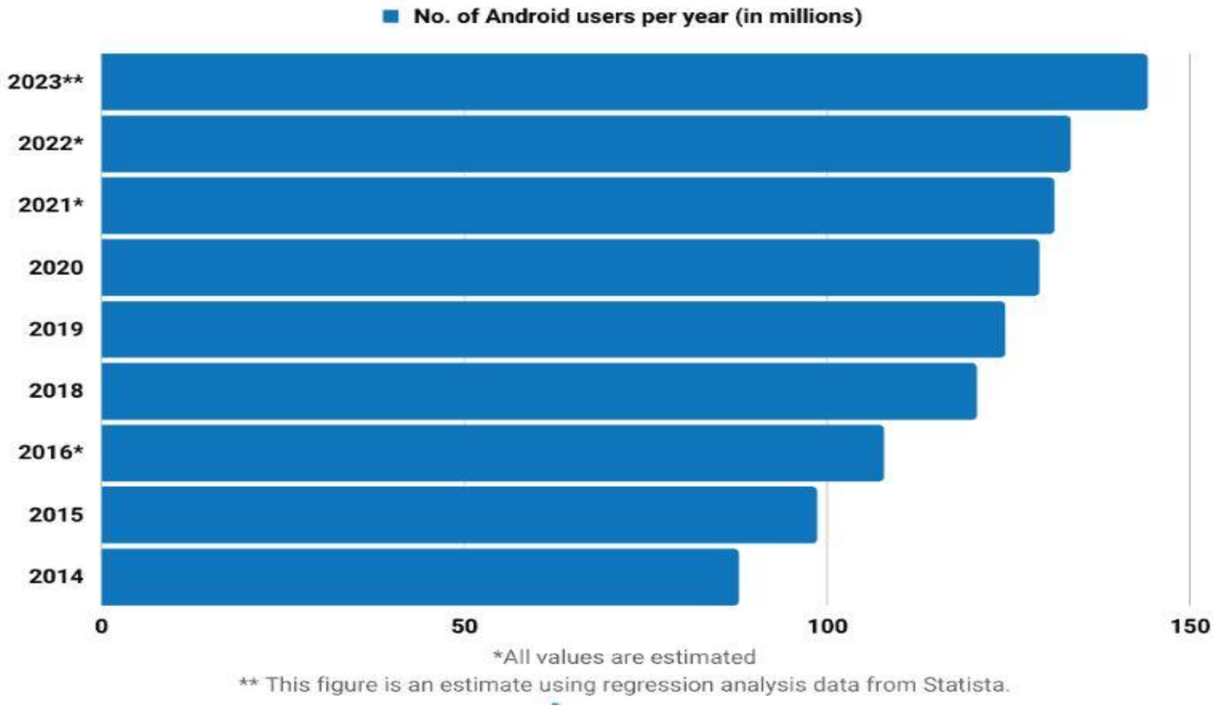


Figure 1: Android Market Share as of February 2023

- Android ruled globally with a **70.94% market percentage** in mobile working systems in 2023 as of now. The Android OS Market Share in the **United States has reached 42.60%**.

- Android telephones dominate the worldwide marketplace share with about **72% of cellular marketplace shares as of January 2024**

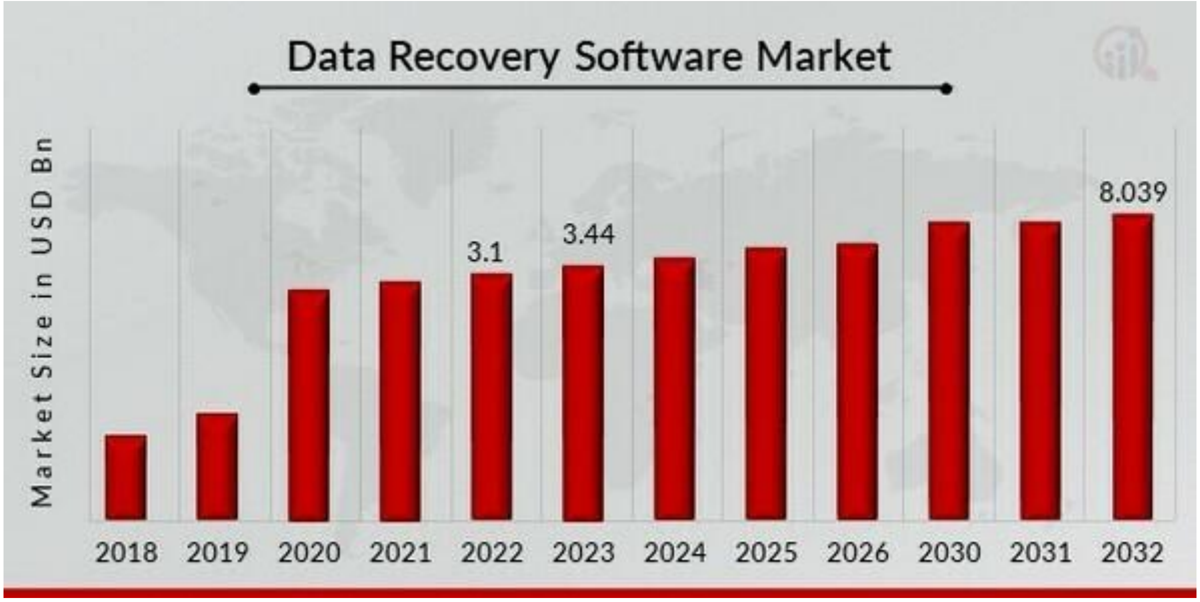


Figure 2: Data Recovery Market

- Data Recovery Software Market Size was valued at USD 3.1 Billion in 2022 [6]
- It includes ANDROID based and Android based software's both

1.3 Proposed Solution

To solve the issues stated in the problem statement, we propose developing a powerful Windows-based Android data recovery tool.

1.3.1 Enhanced Data Recovery Success:

Our program will use innovative algorithms and approaches to maximize data recovery success, even when the device is severely damaged or encrypted. We hope to obtain higher recovery rates than existing methods by rigorously testing and optimizing them[7].

1.3.2 Cost-Free Solution:

Because our software is built in-house for internal use within the organization, there will be no charges associated with its use. Unlike commercial solutions with convoluted price systems, our indigenous tool will be supplied to employees for free, ensuring access to all important functionalities without incurring any costs.

1.3.3 Seamless Compatibility:

Our tool will priorities compatibility with the most recent operating systems and device models, reducing compatibility issues for consumers. We will proactively monitor industry trends and provide regular updates to ensure optimal performance across several platforms.

1.3.4 Efficient Update Management:

This software is designed in-house by giving us complete management on its production and upgrades. This inherent flexibility will allow us to produce active update management procedures and confirming that our result will continue to operate optimally without requiring external upgrades.

1.3.5 **Expanded Recovery Scope**:

This tool will recover data from both Android microSD cards and internal device memory. This extensive recovery opportunity will allow users to restore data from all storage areas on their Android devices[8].

1.3.6 **Enhanced Recovery Features:**

Our tool will have a wide range of recovery options with minimum limitations depending on Android versions. This will ensure that users can access all recovery options despite of their device's Android version.

1.3.7 **Enhanced Security with Offline Functionality**:

This software is designed for internal use within the organization and offline functions. It will automatically improve the security by minimizing the requirement to communicate sensitive data on the Internet. By keeping the data recovery app fully offline we will limit the chance of malicious actors trying to interfere with the data of company which will protect its confidentiality and integrity. This offline function will also improve the user privacy by saving all recovery procedures within the network of company environment which add protection against any data loss or theft[9].

1.3.8 Methodology:

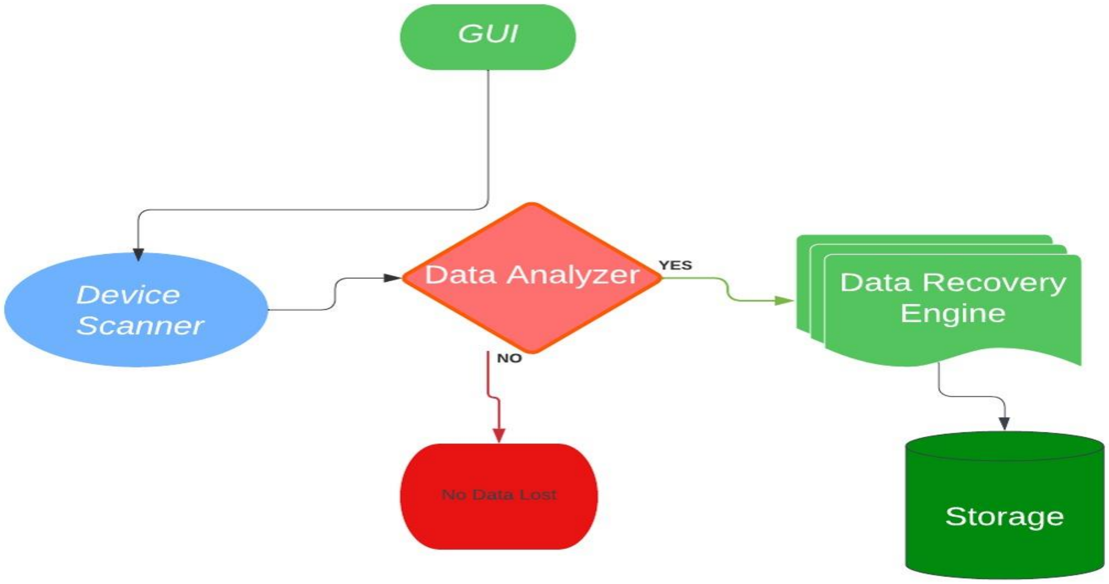
1. **Data Analysis and Scanning:** The Android Data Recovery Tool perform a complete inspection of the storage on the device by using the advance algorithm. This involves detecting and recovering lost data by scanning both external SD cards and internal storage.

2. **File System Reconstruction:** This application detect the fragmented or partly overwritten files using highly developed file system reconstruction algorithms which makes it possible to restore data that might not be accessible otherwise.

3. **Data Carving:** This app values the signature patterns and again reconstruct them from raw data blocks using data carving techniques to recover fragmented.

4. **User Interaction and Feedback:** The utility recognizes signature patterns and reform them from raw blocks of data using data carving techniques to recover fragmented or partially erased files.

1.3.9 Architecture Diagram:



This architecture diagram explains the main mechanism of the Android Data Recovery Tool and their connections:

1. **Graphical User Interface (GUI):**
    - Provides a user-friendly interface for interacting with the tool.
    - Allows users to initiate and monitor the data recovery process.
    - Allow to view recoverable file and customize recovery options.
2. **Device Scanner:**
    - Responsible for detecting and establishing a connection with Android devices.
    - Scans the device's storage to identify lost or deleted data and prepares it for analysis.

3. **Data Analyzer:**
    - It analyzes the examined data to identify the recoverable files and evaluate their integrity.
    - The advanced algorithms is used to analyze file structures, metadata and content to determine whether the data can be recovered.
4. **Data Recovery Engine:**
    - Implements recovery algorithms and procedures to retrieve lost or deleted data.
    - Utilizes the analyzed data to reconstruct files, repair corrupted data, and recover data fragments.
5. **Storage:**
    - Manages temporary storage for scanned and analyzed data.
    - Stores recovered data temporarily before transferring it to the user's designated location.

These parts together streamline the data recovery procedure and give users a simple and efficient way to get their lost data back from Android devices.

1.3 Working Principle

The Android Data Recovery Tool work based on an widespread collection of principles which are created to guarantee effective data recovery from Android devices. The following steps explain how the tool functions:

i. **Data Scanning and Analysis:**

   a. Identification of the device is the first step and after doing it the application will use the data scanner component for scanning the storage of device.

   b. After that the file system of device is scanned thoroughly by looking into the files and storage partitions.

   c. After that the data analyzer will analyze the data in the real-time to find possible recoverable files basing on the file signatures, metadata and other differential features.

   d. Advanced algorithms are used to analyze file structures, detect the data fragments and evaluate the integrity of recoverable data.

ii. **Recovery Procedure Initiation:**

   a. After the completion of scanning and analysis part is completed the data recovery engine part will start its work of recovery process.

   b. In this part the recovery engine will try to find and after that rank the recoverable files which can be restored basing on the results of analysis.

   c. The different methods are used in the recovery engine to recover the lost data pieces which include file carving, metadata reconstruction, and content extraction.

iii. **File Reconstruction and Integrity Verification:**

a. The recovery engine then reassembles the data fragments which are recovered into complete files.

b. After reassembling the data fragments the checksum verification and error correction techniques are used to confirm the recovered files to check the data integrity.

c. File validation procedures are used to reduce the likelihood of data loss and to confirm the reliability of recovered data.

iv. **Data Transfer and Storage:**

a. The restored data is shifted from temporary storage location to the user specified location once the recovery process is completed.

b. After that users can assign the local storage or cloud storage services as the destination for recovered files using the options which are provided by the tool.

c. The confidentiality and integrity of recovered data is protected both by secure transfer protocols and encryption algorithm used during the data transmission.

v. **User Feedback and Interaction:**

a. The Graphical User Interface (GUI) component will give the user real-time response and communication choices during the process of recovery.

b. The recovery process is visible to users for clarity including the quantity of files that have been found, recovered and are still bring recovered.

c. Flexibility and control is provided to customers by interactive features over the data recovery process by enabling them to stop, resume or end it.

1.4 Objectives

1.4.1 General Objectives:

- Create and develop a local Android-based Android data recovery tool that can be used to recover different kinds of data from Android devices.
- Verify that the product is working with a diversity of Android device types and versions. Use of reliable algorithms to maintain data integrity when scanning, analyzing and restoring information.
- Create and include an user-friendly, intuitive user interface to facilitate seamless interaction.

1.4.2 Academic Objectives:

- Review the literature which is already in presence considering the data recovery methods specially as it connects to Android devices.
- Share the findings through talks and publishing in academic journals to further understanding in the topic.
- Working together with the other researchers to share knowledge and improve understanding of Android data recovery.

1.5 **Scope**

The scope of project include the designing, creating and deploying an Android-based Android data recovery solution. The primary objective of tool will be to restore different kinds of data that are normally kept on Android smartphones such as papers, audio files, and video files. To meet the requirement of a wide range of users the development efforts will be pointed toward confirming that the tool is compatible with a wide range of Android devices and operating system versions.

The scope also highlights the data security and integrity by incorporating strong algorithms for data scanning, analysis and recovery. To increase the flexibility of tool the project will also examine the likelihood of adding facilities. This would make data recovery possible from specially created android applications to be more efficient.

1.6 Deliverables

- The main product of this project will be an Android-based Android data recovery program with large data restoration resources and user-friendly interface.
- There will be thorough documentation included with this program that will describe how to install, handle and troubleshoot it.
- Comprehensive video tutorial series, providing step-by-step guidance, feature demonstrations, and helpful tips for users of all technical levels.
- The project deliverables will also contain thorough testing results that will verify the functionality of the product on various Android smartphone models.

1.7 Relevant Sustainable Development Goals

The development and completion of the Android Recovery Tool project support with several Sustainable Development Goals (SDGs) outlined by the United Nations. These goals represent a global call to address pressing social, economic, and environmental challenges. The Android Recovery Tool contributes to the following SDGs:

Goal 9: Industry, Innovation, and Infrastructure

By giving an innovative solution for data recovery from Android devices the Android Recovery Tool supports Goal 9 by promoting technological innovation and enhancing infrastructure development. The tool enable access to advanced technology and contributes to the growth of the digital economy.

Goal 16: Peace, Justice, and Strong Institutions. The Android Recovery Tool supports Goal 16 by fostering trust and confidence in digital technologies and institutions.

1.8 Structure of Thesis

- Chapter 2 contains the literature review, background and Existing solutions and their drawbacks
- Chapter 3 is detailed design and architecture and it basically lists down the use cases, design rationale, architecture and other design components of the Android Data Recovery Tool
- Chapter 4 introduces detailed evaluation and analysis of the code
- Chapter 5 contains the proposed solution, future direction and conclusion of the project
- Chapter 6 is References and it lists the references of the resources used

Chapter 2: Literature Review

A review of the existing literature reveals several key trends and challenges in the field of Android data recovery. Over the years, there has been a sharp increase in the use of smartphones as a mobile phone. In 1973, Motorola released the first mobile phone, the DynaTAC 8000X, which was created by Martin Cooper (Morum de L. Simão, Caús Sícoli, Peotta de Melo, Deus, & de Sousa Junior, 2011) [10]. Mobile phones have grown increasingly potent over time and are now an essential part of our life. The creation of software-based data recovery tools designed especially for Android smartphones is one well-known field of study. To recover lost or erased data from device storage, these technologies usually include file system analysis, data carving techniques, and algorithmic approaches.

- Industrial Background
- Existing solutions and their drawbacks
- Research Papers

2.1 Industrial background

Smartphones are become commonplace instruments for productivity, entertainment, and communication in today's digital world. Of all the mobile operating systems available, Android—created by Google—is the most popular worldwide. Millions of consumers worldwide have chosen Android as their operating system because of its open-source design, wide range of device compatibility, and robust app store. The widespread use of Android devices and the important role they play in storing personal and professional data have a straight effect on the industrial environment of Android data recovery.

2.1.1 Proliferation of Android Devices:

The use of exponential expansion in Android device across a wide range of demographics and geographic locations describes the industrial establishment of Android data recovery. Android smartphones are designed to accommodate a wide range of customers with different choices. They range from flagship models with cutting-edge abilities to budget-friendly mobiles. Android devices have been widely promoted due to their price and ease of access which has resulted in a large user base that includes individuals, corporations, government agencies and educational institutions.

2.1.2 Storage of Critical Data:

Android smartphones are important storage devices for different kind of data such as contacts, messages, documents, videos, images, and apps. Users save sensitive personal data, work-related information, priceless memories, and critical papers on their devices. The need of protecting data from loss or illegal access is highlighted by the growing convergence of personal and professional life on mobile devices. Therefore, the need to create reliable solutions that can recover a variety of data types from Android devices has molded the industrial history of Android data recovery[11].

2.1.3 Risks of Data Loss:

Even while Android devices are convenient and useful, there is always a chance that users will lose their data for a variety of reasons, such as theft, physical damage, virus attacks, hardware malfunctions, software errors, and inadvertent deletion. Critical data loss can have far-reaching effects, from financial ramifications and privacy concerns to inconvenience and lost productivity. Minimizing the dangers related to data loss events is fundamental to ensuring the soft functioning and durability of Android devices.

2.1.4 Demand for Reliable Recovery Solutions:

The increase in the use of Android devices has guided to a rise in the need for reliable and effective data recovery programs designed especially for the Android devices. Customers want guarantee that in the case of an unexpected data loss their priceless information should be recovered safely. The concept of effective data recovery applications has difficulties due to the complexity of the architecture of Android devices, the different number of file systems and most importantly the use of commercial data storage formats.

2.1.5 Technological Advancements:

Technological developments in mobile hardware, software and data recovery techniques have an impression on the industrial environment of Android data recovery. By each passing day the generation of smartphones bringing smartphones with improved performance and sophisticated security mechanisms. The data recovery solutions have to keep up to date in order to cater them. The continuous development of Android data recovery techniques is helped by developments in forensic analysis techniques, device scanning technology and data recovery algorithms.

2.2 Existing solutions and their drawbacks

There are already a number of methods available in the ground of Android data recovery to deal with the difficulties involved in getting lost or erased data back from Android devices. The goal of these solutions is to give people choices for recovering their important data back. This section inspects some of the options available on the market and examines the drawbacks of each.:

2.2.1 Dr Fone from Wondershare (Windows & Mac):

Dr. Fone from Wondershare is a flexible software solution for Windows and Mac users that incorporates thorough tools for data recovery and device management. Dr. Fone's simple interface and strong abilities allow users to easily recover lost data from different devices including smartphones and tablets[13].

Drawbacks:

- **Limited Data Recovery Success:** Dr. Fone procedure cannot always assure the successful recovery of all types of data specially in case of serious device damage or data encryption.
- **Complex Pricing Structure**: The mechanism of pricing for Dr. Fone's many modules is very complicated and difficult which is one of the reasons to cause confusion and unexpected expenses for those looking for certain functions.
- **Compatibility difficulties:** Users can also have compatibility difficulties with the most recent operating systems of their devices or changes for optimal operation
- **Dependence on Regular Updates:** Dr. Fone's effectiveness may be dependent on regular updates of product to keep up with the new device technology and software

changes. This can cause disruptions in service and functioning if not handled immediately.

2.2.2 Disk Drill (Windows & Mac):

Disk Drill is a thorough data recovery program that can recover the lost data from a different devices including Android smartphones and tablets. The Mac version of software can recover lost data from internal memory or an SD card, however the Windows version is limited to SD card recovery only[14].

Drawbacks:

- **Recovery Scope:** Windows version of the Disk Drill can recover data only from Android microSD cards, not from the internal memory of the Android mobile devices.
- **High Subscription:** The pricing of Disk Drill can be complicated, potentially causing confusion or unexpected expenses for those looking for certain functions.

2.2.3 Tenor share UltData for Android (Windows & Mac):

Tenor share UltData for Android is designed to help Windows and Mac users recover lost data without root. All versions of the Android operating system from version 5 up are fully supported, but some recovery features are restricted to specific Android versions.

Drawbacks:

- **Limitation on Recovery Features:** Some recovery features are restricted to specific Android versions
- **High Prices:** Tenorshare UltData offer its services at very high prices.

- **Security Risks of Online Data Recovery:** One disadvantage of using online data recovery software is the greater likelihood of data loss or theft. When data recovery activities are carried out online, sensitive information is transported across the internet, leaving it vulnerable to interception by malicious actors.

CHAPTER 3: DESIGN AND DEVLOPMENT

SYSTEM OVERVIEW

Pakistan is a state where we are still dependent on proprietary software for carrying out data recovery of android mobiles. This proprietary software has now turned out to be very costly and at times unreliable which is why we have now come up with a solution. Android data recovery tool looks forward to providing the users with a tool comprising all the functions which are already present in a proprietary software at a cheaper cost in comparison to the charges of the expensive proprietary software.

To enable effective data recovery from Android devices, the Android Data Recovery Software has a modular architecture. The modular architecture of the Android Data Recovery Software enables effective data recovery while offering consumers a smooth and user-friendly interface. The software's components allow users to easily and reliably restore important data from their Android smartphones[15].

3.1 SYSTEM ARCHITECTURE

3.1.2 Architectural Design

The Android Data Recovery Tool's architectural design can be seen by the fact that the user interface acts like the first point of interaction among the user and the software. The data recovery engine, which is the main component in charge of managing the recovery process, which is reached by the user interface at the initial stage of the entire process. The Engine contacts the File System Access feature that obtains instant access to an Android device's file system in order to ask for data that has to be extracted from the file system. The data processing part gets the returned data from the file system and it will use it to examine and process it after delivering it to the engine. Following that, the main component which is Engine continues on to the Data Processing section, where the information that was collected is reviewed and changed. The processed data remains stored in the relevant storage and transmits a signal back into the data processing system confirming that that data has been successfully preserved. The Engine retrieves the information that was processed to the user interface following the storage process, so terminating the data restoration procedure. A quick overview of how all of the parts of the Android data recovery software communicate is presented by this sequence diagram[16].

3.1.3 **Architecture Diagram**

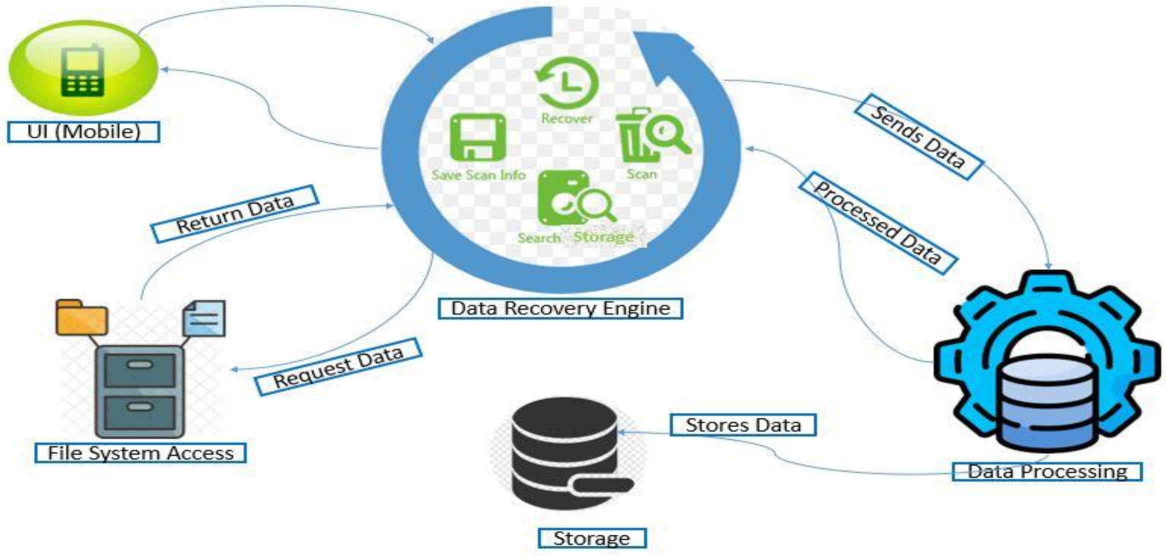


Figure 4: Architecture Diagram of Android Data Recovery Tool

3.2 **DESIGN RATIONALE**

A number of justification for choosing one of the expected designs of architecture.

1. The User Interface (UI) provides links between the user and the software, as presented by the sequence diagram. The method of design summaries what the user wants and a seamless experience of use, that allows users to get started and stop the data recovery operation with ease.
2. The diagram depicts an open architecture with separate sections for the Data Recovery Engine, File System Access, Data Processing, and Storage. This design's adaptation and scalability ensure the inclusion of new features or improvements of those currently present in individual components without properly disrupting the system as a whole.
3. Every component in this sequence diagram has a distinct function in the data restoration procedure. For example, whereas the File System Access module retrieves data from Android device’s file system. The Data Retrieval Engine initializes the mobile device's file system alongside all essential. This kind of design thinking encourages effectiveness and clarity.
4. The sequence diagram outlines the parts that information passes through to show how the procedures needed to successfully recover data; this design principle guarantees that data will be handled appropriately at each stage of its journey, lessening the possibility of corruption or leakage of data that could occur if such factors were not taken into account or factored in during this entire period.

3.3 **COMPONENT DESIGN**

The user and different device modules should be operational at their paramount and cooperating with each other smoothly for the program to work at its best. Let's examine the interactions between each of the additives[17].

3.4 **USE CASE DESCRIPTIONS**

Table 1: Use Case Number 1

Use Case ID	01
Actors	User
Use Case Name	Initiate data recovery process
Description	The user launches the data recovery software on their Android device
Flow	1. The individual opt for the app interface on their smartphones to start the data restoration software 2. The device shows the default person interface and shows details restoration picks 3. To kick off the data recovery practice, the person chooses the "Start Recovery" choice

Table 2: Use Case Number 2

Use Case ID	02
Actors	User
Use Case Name	Select data recovery options
Description	The user customizes the data recovery process by selecting specific options
Flow	1. The application suggests the individual with number of recovery choices, wide-ranging of which file type to recover, after starting the process of scanning 2. The person chooses their desired techniques of restoration in keeping with their very own requirements 3. The user's alternatives inform the software's changes to its scanning settings and filters 4. After verifying their choices, the person initiates the data restoration process

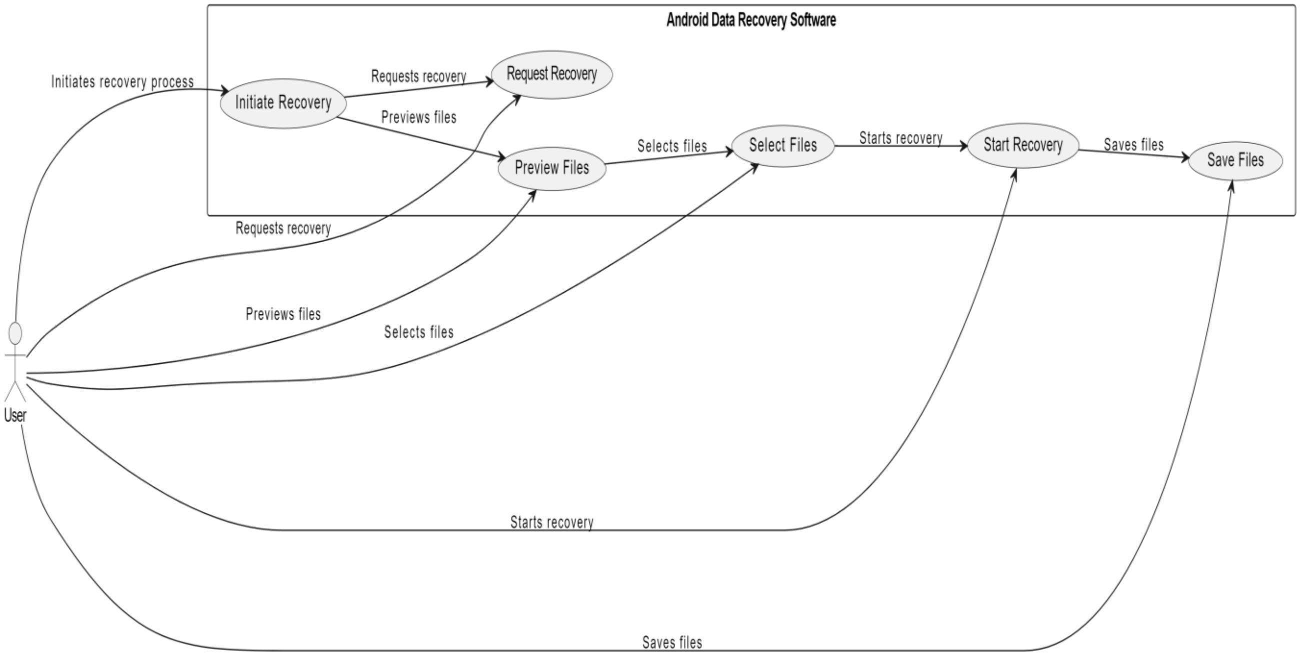
Table 3: Use Case Number 3

Use Case ID	03
Actors	User
Use Case Name	View scan results
Description	The user reviews the results of the data recovery scan
Flow	1. After completing the data recovery scan, the software displays the scan results to the user 2. The user navigates through the scan results, which may include a list of recovered files or directories 3. The software provides detailed information about each recovered item, such as file name, size, and recovery status 4. The user has the option to preview recovered files or directories before proceeding with the recovery process

Table 4: Use Case Number 4

Use Case ID	04
Actors	User
Use Case Name	Recover data
Description	The user selects and recovers specific data items from the scan results
Flow	1.Users may opt for the data types you wish to retrieve while viewing across the scan outcomes 2. The user continues the data recovery processes by acknowledging what they want 3. The picked items of data are retrieved and saved by the program to the designated location 4. The tool signals the user and gives them the choice to see the data that was recovered at the specified location after its completion was successful

3.5 **USE CASE DIAGRAM**



3.6 **HUMAN INTERFACE DESIGN**

3.6.1 **OVERVIEW OF USER INTERFACE**

The user interface (UI) of the Android data recovery mobile software has a very simple display for ease of use to facilitate the users. Those who will use this application will have the choice to select their files options for recovery and will be able to monitor the scan's real time progress. Once the scan is over, viewers may discover restored folders and files, will examine contents, and select what to retrieve. The display also helps in offering options for choosing where to save data and confirming restoration choices.[18].

3.6.2 SCREEN OBJECTS AND ACTIONS

- Buttons
  - Scan Device
  - Select File Types
  - Recover Selected

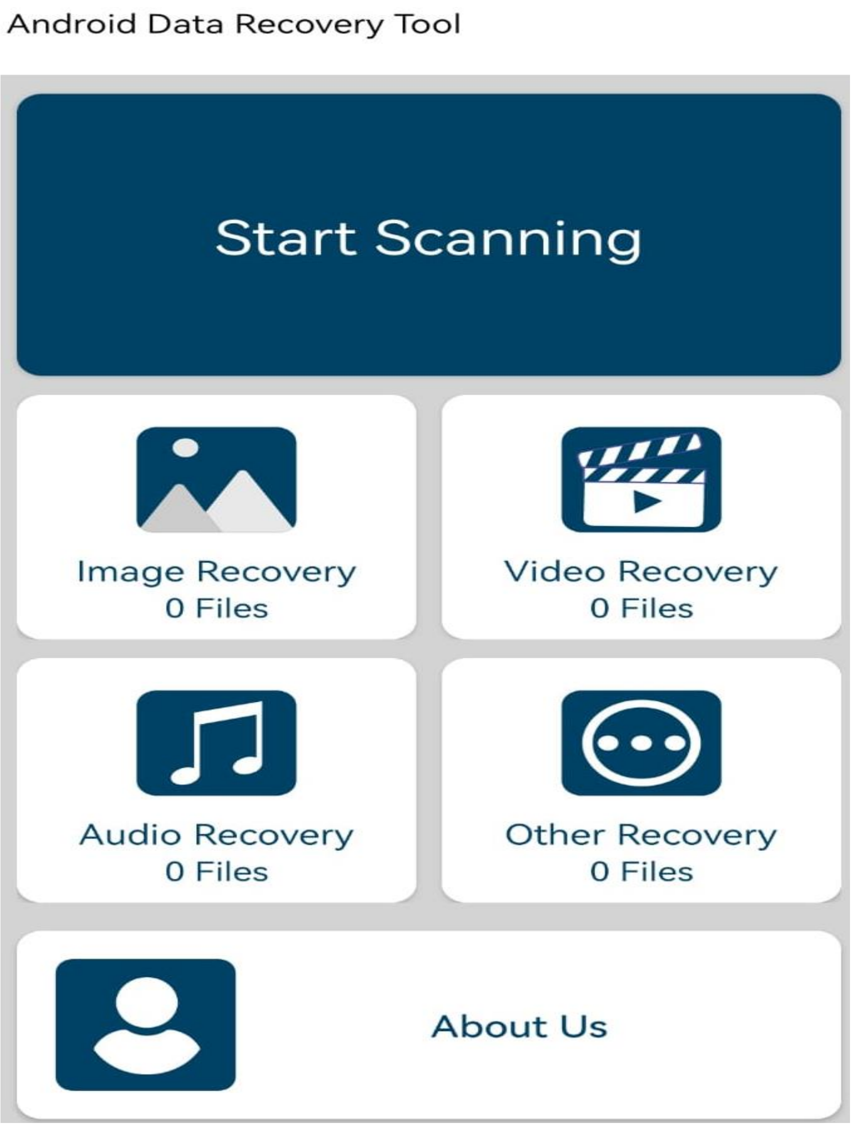
- Labels
  - Progress Indicator
  - File Information Labels

- Icons
  - Application Logo
  - File Type Icons

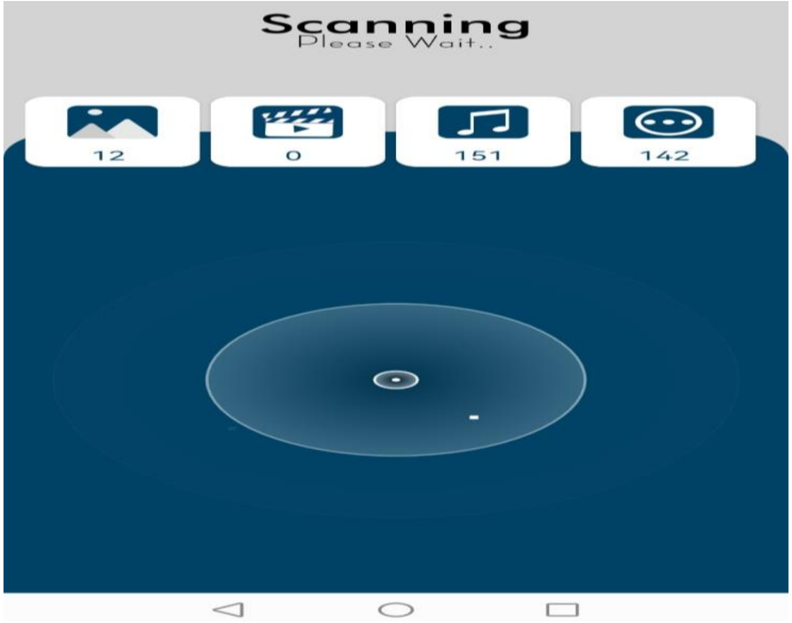
- Scroll
- Select

3.6.3 **Screen Images**

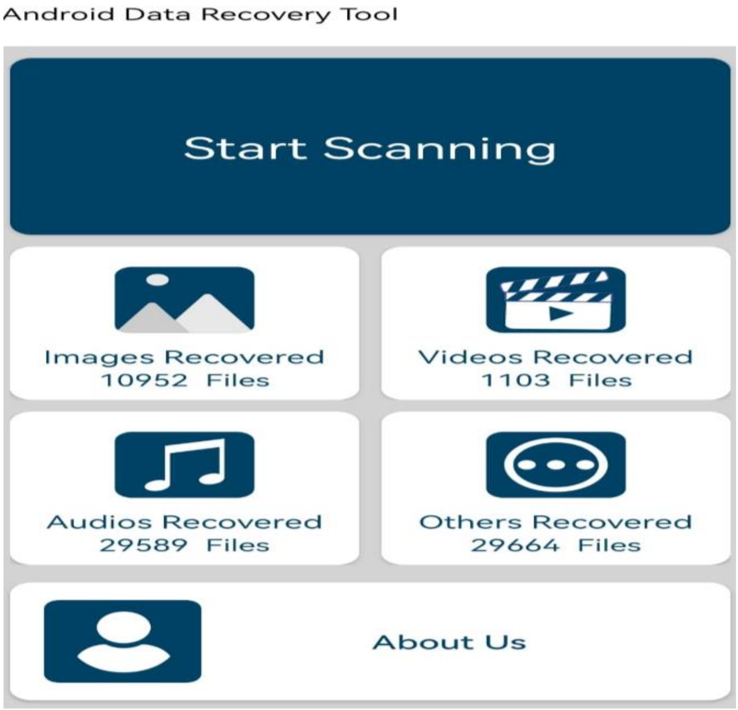
Application Screen



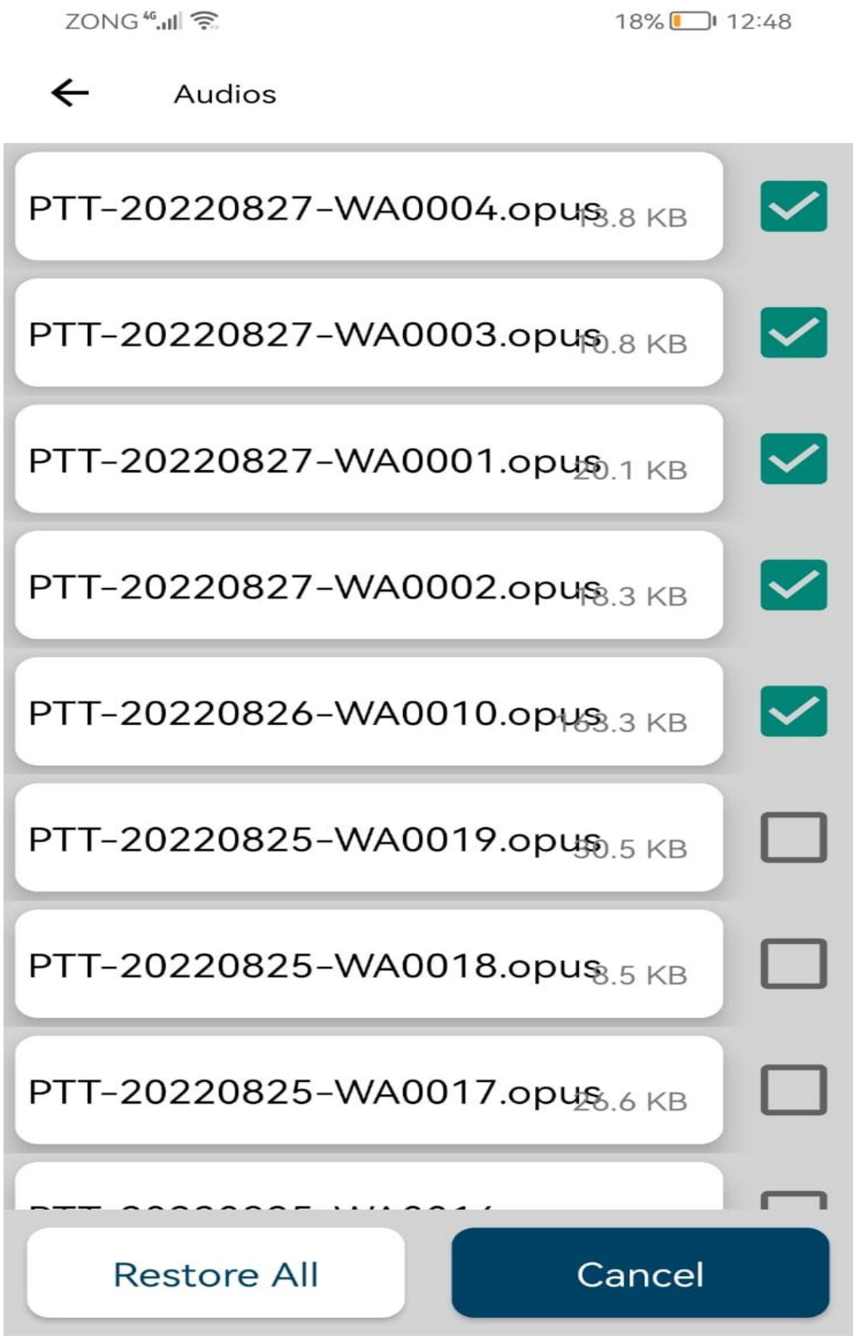
Initializing Scanning



Recovery Process Completion



Selection of File Types



PTT-20220827-WA0004.opus 13.8 KB ☑

PTT-20220827-WA0003.opus 10.8 KB ☑

PTT-20220827-WA0001.opus 20.1 KB ☑

PTT-20220827-WA0002.opus 18.3 KB ☑

PTT-20220826-WA0010.opus 183.3 KB ☑

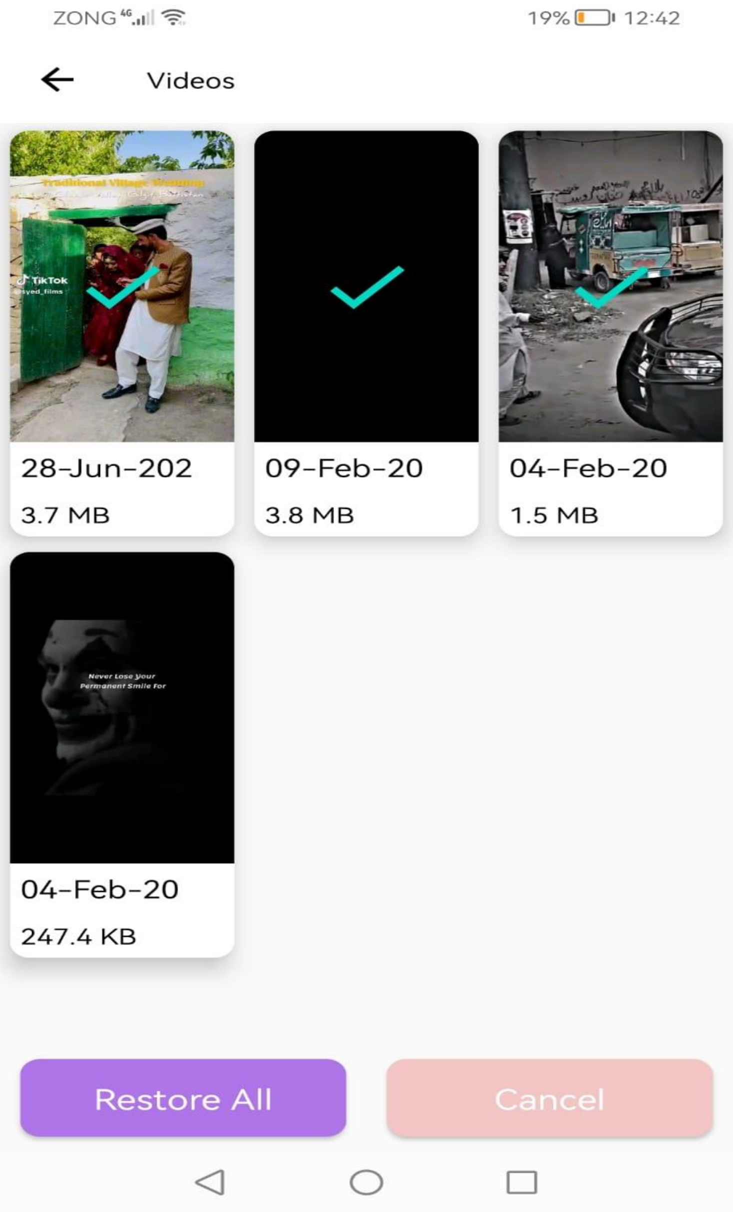
PTT-20220825-WA0019.opus 30.5 KB □

PTT-20220825-WA0018.opus 8.5 KB □

PTT-20220825-WA0017.opus 26.6 KB □

PTT-20220825-WA0016 □

Recovery Selected



ZONG 4G 19% 12:42

Videos

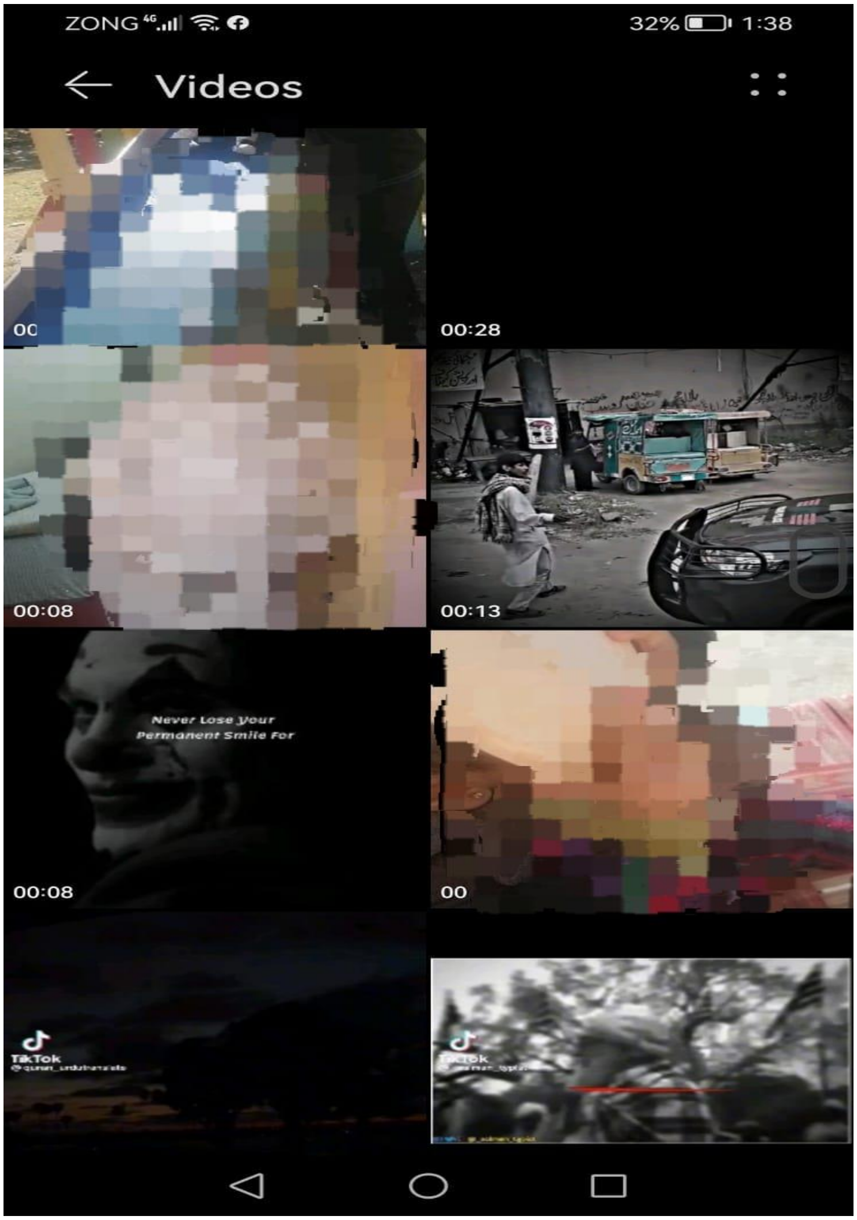
28-Jun-202
3.7 MB

09-Feb-20
3.8 MB

04-Feb-20
1.5 MB

04-Feb-20
247.4 KB

Recovered Data



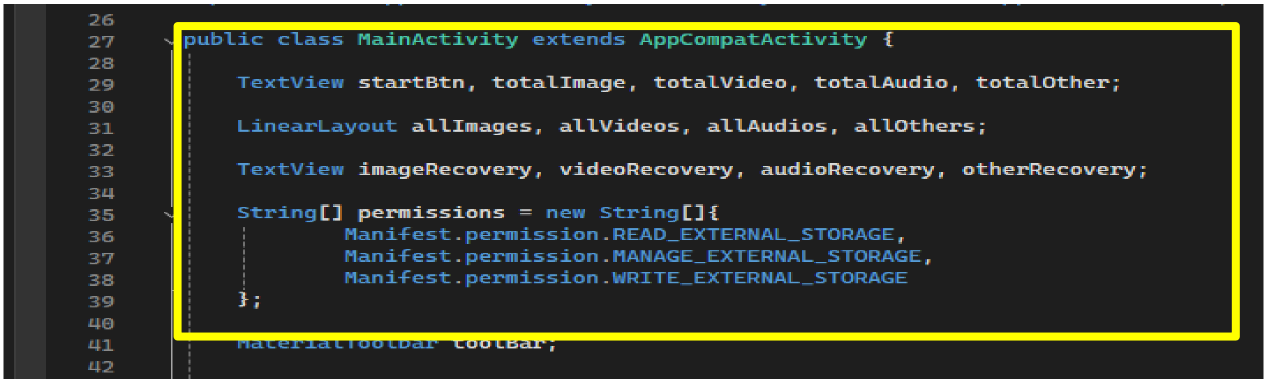
ZONG 4G
32% 1:37
Videos
00:05
00:1
0
TikTok
@syed_films
00:17
00:31
00:30

CHAPTER 4: DETAILED EVALUATION AND ANALYSIS OF THE CODE

4.1 **Application Activities**

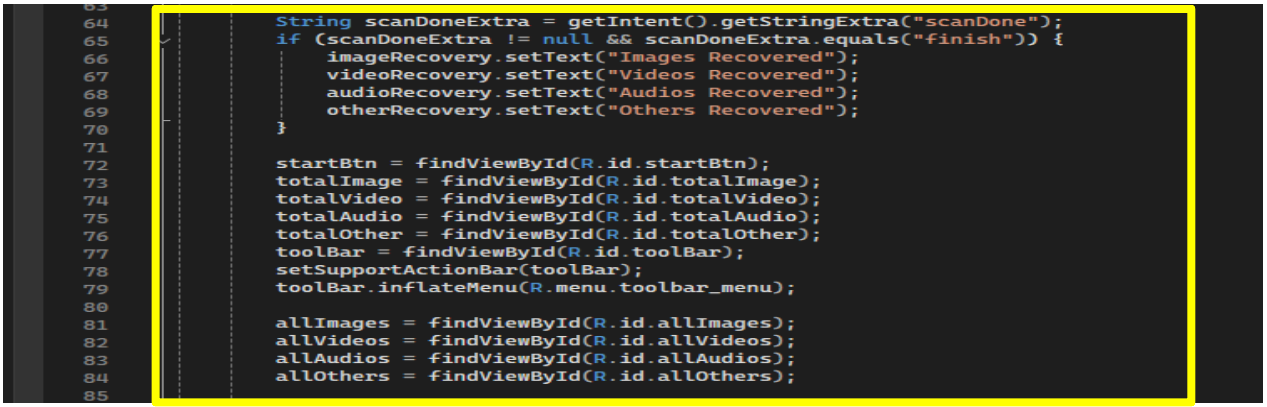
The Application's Activity contains Java files. These specify the various displays and features. Each file represents a unique user interface. These actions include searching for various sorts of files, displaying recovered files, and providing users with options and information panels

4.1.1 **Main Activity**



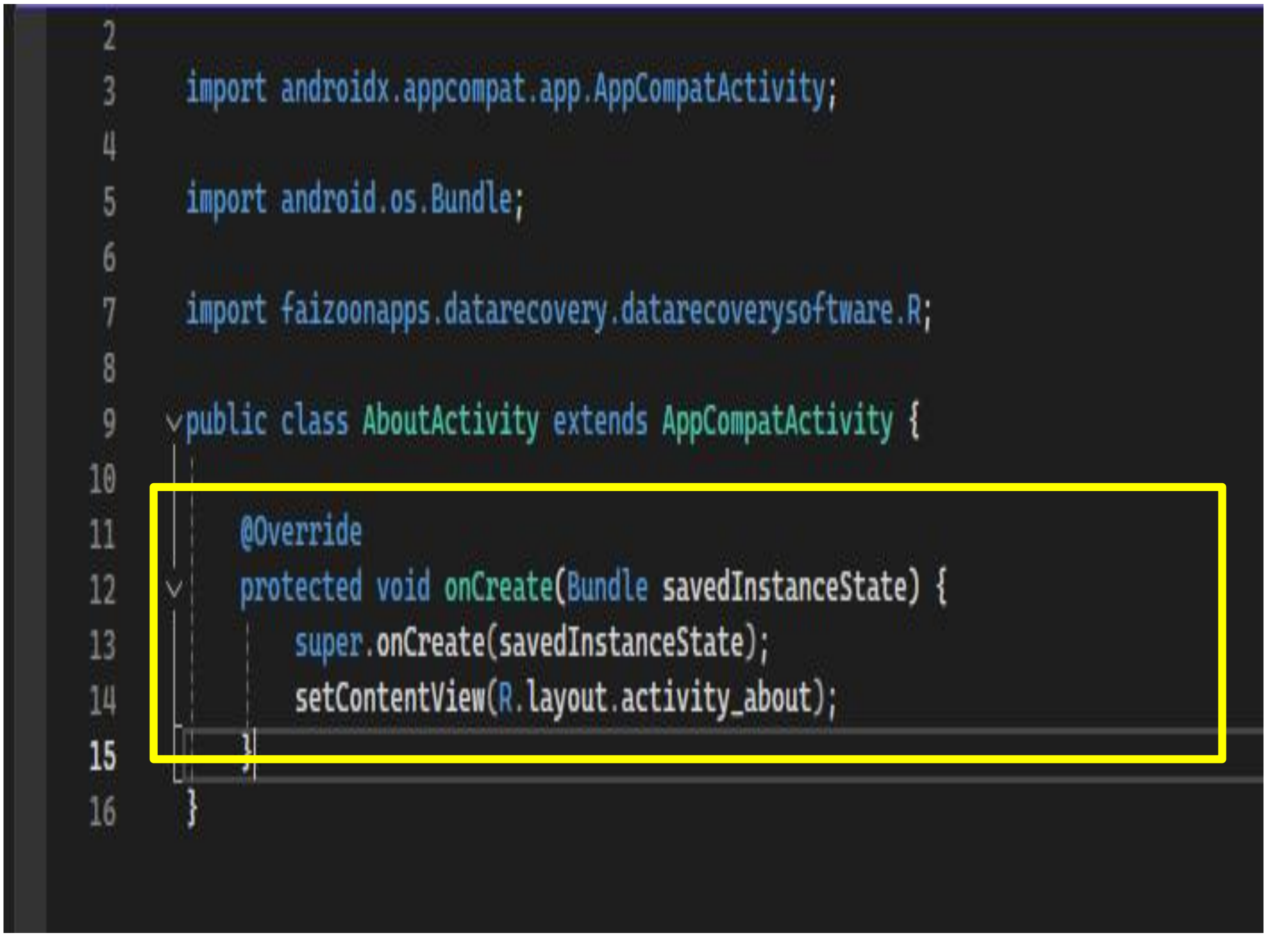
```
public class MainActivity extends AppCompatActivity {

    TextView startBtn, totalImage, totalVideo, totalAudio, totalOther;

    LinearLayout allImages, allVideos, allAudios, allOthers;

    TextView imageRecovery, videoRecovery, audioRecovery, otherRecovery;

    String[] permissions = new String[]{
            Manifest.permission.READ_EXTERNAL_STORAGE,
            Manifest.permission.MANAGE_EXTERNAL_STORAGE,
            Manifest.permission.WRITE_EXTERNAL_STORAGE
    };

    MaterialToolbar toolBar;
```

The data recovery software application's primary activity is represented by the file 'MainActivity.java'. It requests permission to access storage for scanning files. The layout includes buttons for scanning pictures, audios and other media.



```
        String scanDoneExtra = getIntent().getStringExtra("scanDone");
        if (scanDoneExtra != null && scanDoneExtra.equals("finish")) {
            imageRecovery.setText("Images Recovered");
            videoRecovery.setText("Videos Recovered");
            audioRecovery.setText("Audios Recovered");
            otherRecovery.setText("Others Recovered");
        }

        startBtn = findViewById(R.id.startBtn);
        totalImage = findViewById(R.id.totalImage);
        totalVideo = findViewById(R.id.totalVideo);
        totalAudio = findViewById(R.id.totalAudio);
        totalOther = findViewById(R.id.totalOther);
        toolBar = findViewById(R.id.toolBar);
        setSupportActionBar(toolBar);
        toolBar.inflateMenu(R.menu.toolbar_menu);

        allImages = findViewById(R.id.allImages);
        allVideos = findViewById(R.id.allVideos);
        allAudios = findViewById(R.id.allAudios);
        allOthers = findViewById(R.id.allOthers);
```

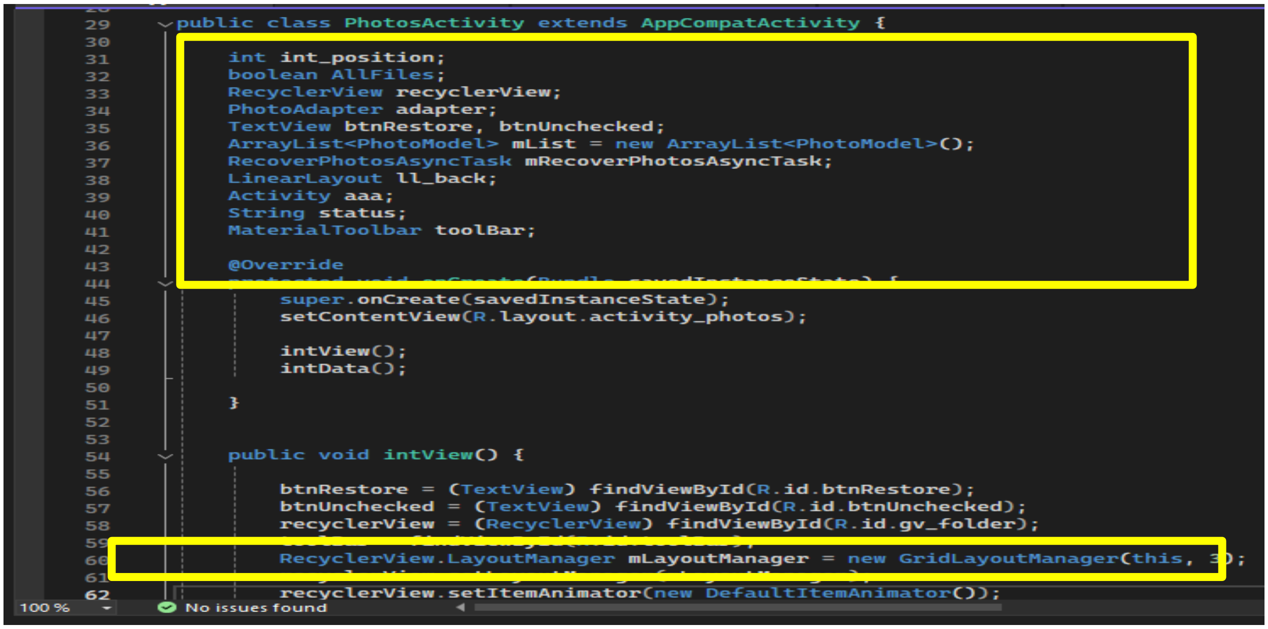
4.1.2 **About Activity**



```

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import faizoonapps.datarecovery.datarecoverysoftware.R;

public class AboutActivity extends AppCompatActivity {

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

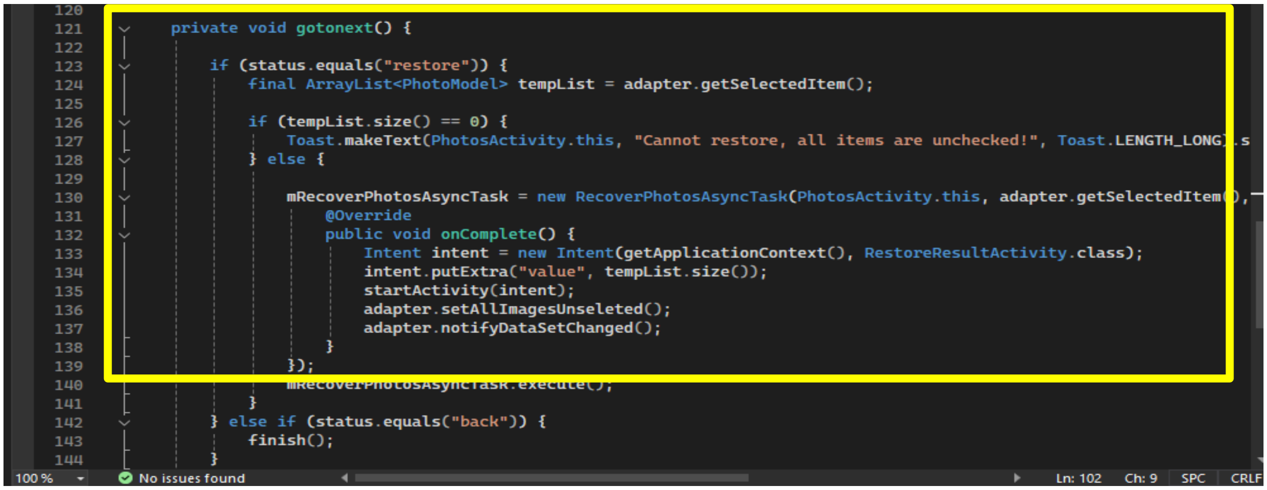
The AboutActivity.java file displays information about the application. Content view is changed to a layout called activity_about.xml. The main focus of this activity is to serve as an informational screen where user may learn about the program, with no further functions beyond selecting its layout.

4.1.3 Photos Activity



```java
public class PhotosActivity extends AppCompatActivity {

    int int_position;
    boolean AllFiles;
    RecyclerView recyclerView;
    PhotoAdapter adapter;
    TextView btnRestore, btnUnchecked;
    ArrayList<PhotoModel> mList = new ArrayList<PhotoModel>();
    RecoverPhotosAsyncTask mRecoverPhotosAsyncTask;
    LinearLayout ll_back;
    Activity aaa;
    String status;
    MaterialToolbar toolBar;

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

        intView();
        intData();

    }


    public void intView() {

        btnRestore = (TextView) findViewById(R.id.btnRestore);
        btnUnchecked = (TextView) findViewById(R.id.btnUnchecked);
        recyclerView = (RecyclerView) findViewById(R.id.gv_folder);
        toolBar = findViewById(R.id.toolBar);
        RecyclerView.LayoutManager mLayoutManager = new GridLayoutManager(this, 3);
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
```

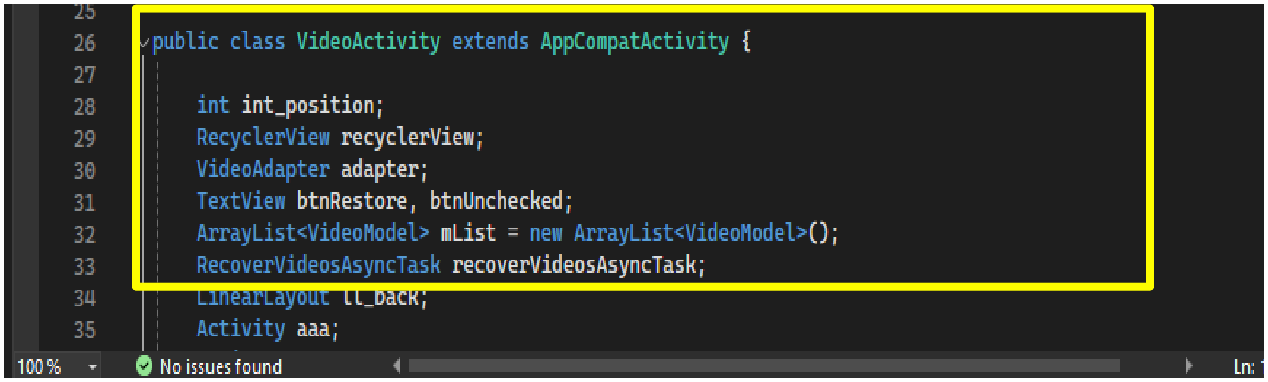
The PhotosActivity.java file displays recovered photographs in the data recovery application. It builds upon "App Compact Activity". The layout contains a Recycler View that displays photos in a grid format.



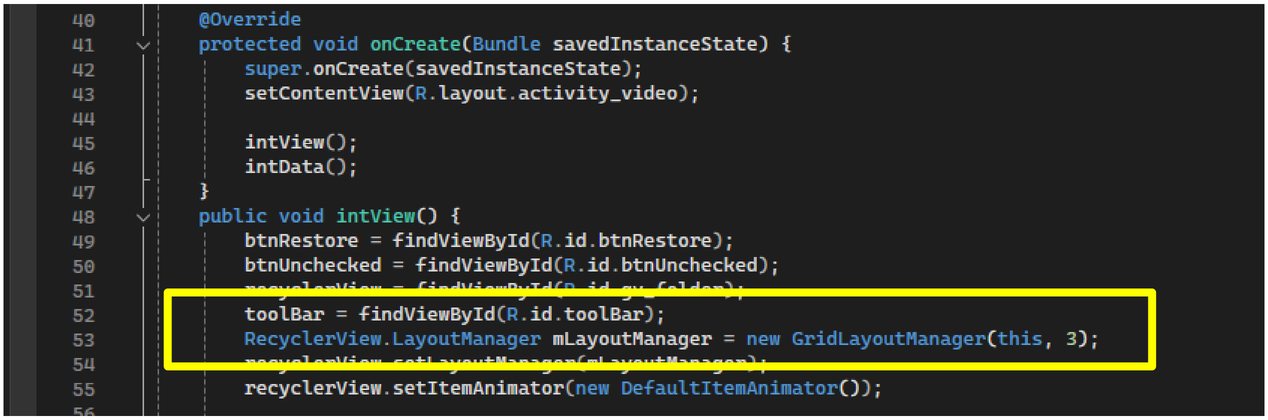
```java
    private void gotonext() {

        if (status.equals("restore")) {
            final ArrayList<PhotoModel> tempList = adapter.getSelectedItem();

            if (tempList.size() == 0) {
                Toast.makeText(PhotosActivity.this, "Cannot restore, all items are unchecked!", Toast.LENGTH_LONG).s
            } else {

                mRecoverPhotosAsyncTask = new RecoverPhotosAsyncTask(PhotosActivity.this, adapter.getSelectedItem(),
                    @Override
                    public void onComplete() {
                        Intent intent = new Intent(getApplicationContext(), RestoreResultActivity.class);
                        intent.putExtra("value", tempList.size());
                        startActivity(intent);
                        adapter.setAllImagesUnseleted();
                        adapter.notifyDataSetChanged();
                    }
                });
                mRecoverPhotosAsyncTask.execute();
            }
        } else if (status.equals("back")) {
            finish();
        }
```

Data is initialized based on the selected album. It also manages the logic for recovering certain photographs and changing the UI accordingly. Users can either recover or uncheck chosen images. Option for the previous screen is also provided through the use of “Tool bar”. Overall, this helps users when processing image recovery.

4.1.4 Video Activity



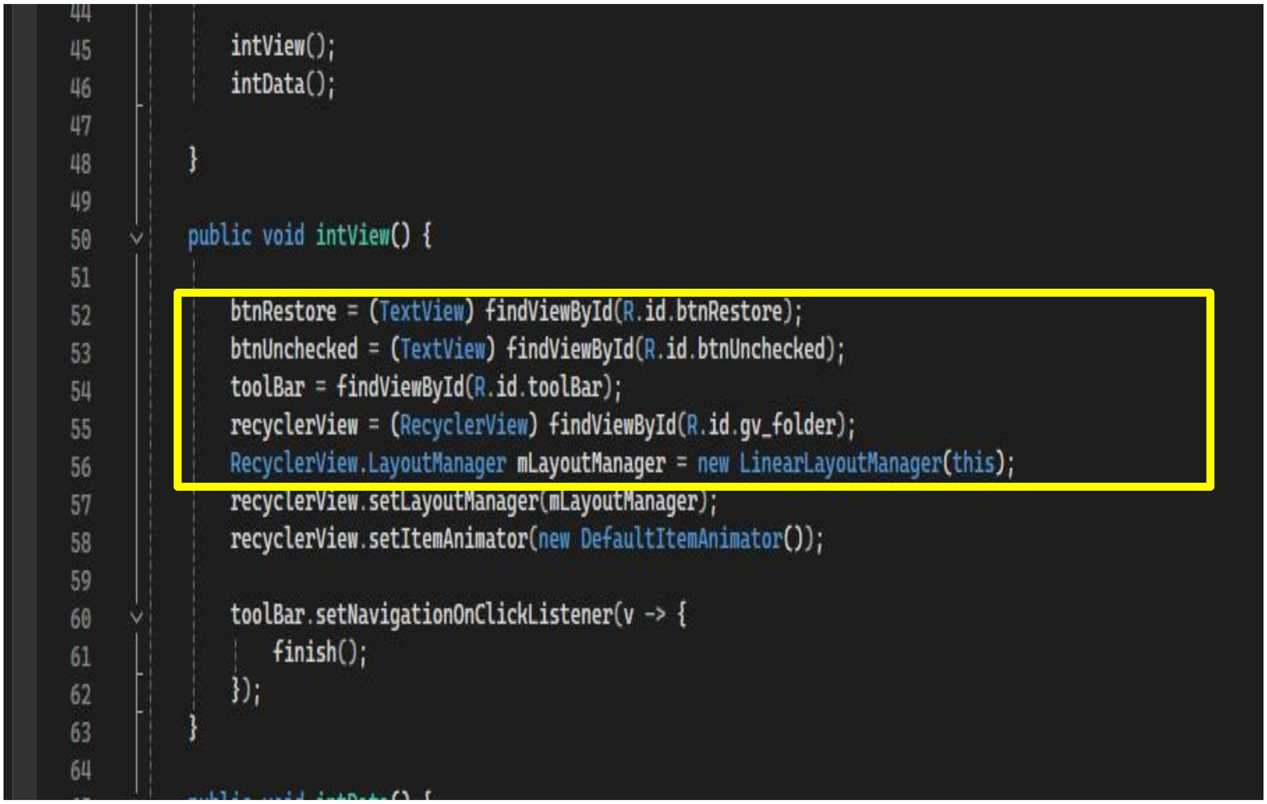
```java
public class VideoActivity extends AppCompatActivity {

    int int_position;
    RecyclerView recyclerView;
    VideoAdapter adapter;
    TextView btnRestore, btnUnchecked;
    ArrayList<VideoModel> mList = new ArrayList<VideoModel>();
    RecoverVideosAsyncTask recoverVideosAsyncTask;
    LinearLayout ll_back;
    Activity aaa;
```



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

        intView();
        intData();
    }
    public void intView() {
        btnRestore = findViewById(R.id.btnRestore);
        btnUnchecked = findViewById(R.id.btnUnchecked);
        recyclerView = findViewById(R.id.gv_folder);
        toolBar = findViewById(R.id.toolBar);
        RecyclerView.LayoutManager mLayoutManager = new GridLayoutManager(this, 3);
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
```

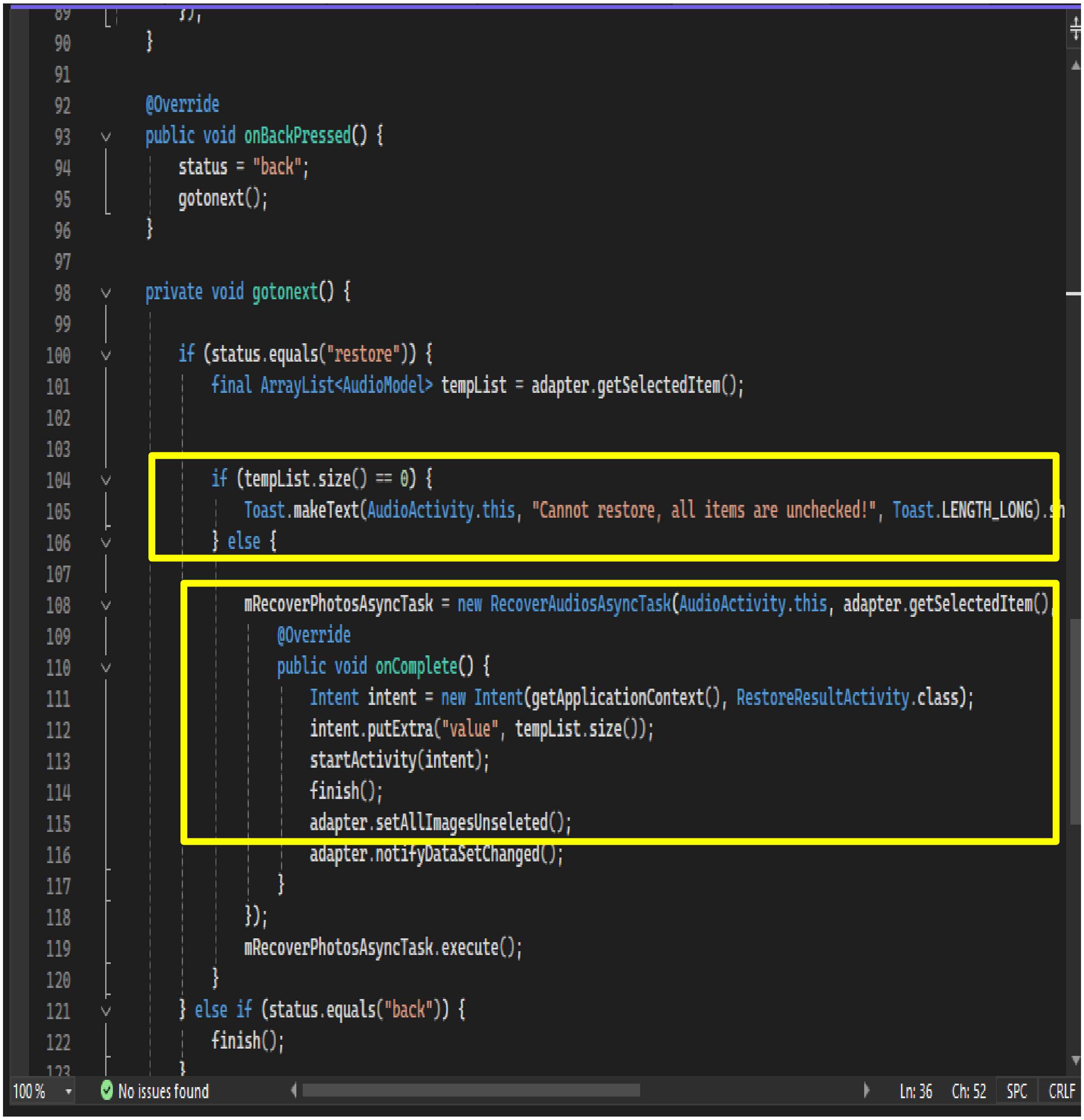
The “Video Activity” file displays recovered Videos in the data recovery application. It is also builds upon “App Compat Activity”, same as “Photos Activity” file. It has the “Recycler View” layout that displays the videos in a grid format. Users of the application can either restore or uncheck selected videos. Data is initialized based on the video album selected. Option for the previous screen is also provided through the use of “Tool bar”.

4.1.5 **Audio Activity**



```
        intView();
        intData();

    }

    public void intView() {

        btnRestore = (TextView) findViewById(R.id.btnRestore);
        btnUnchecked = (TextView) findViewById(R.id.btnUnchecked);
        toolBar = findViewById(R.id.toolBar);
        recyclerView = (RecyclerView) findViewById(R.id.gv_folder);
        RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(this);
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());

        toolBar.setNavigationOnClickListener(v -> {
            finish();
        });
    }

    public void intData() {
```

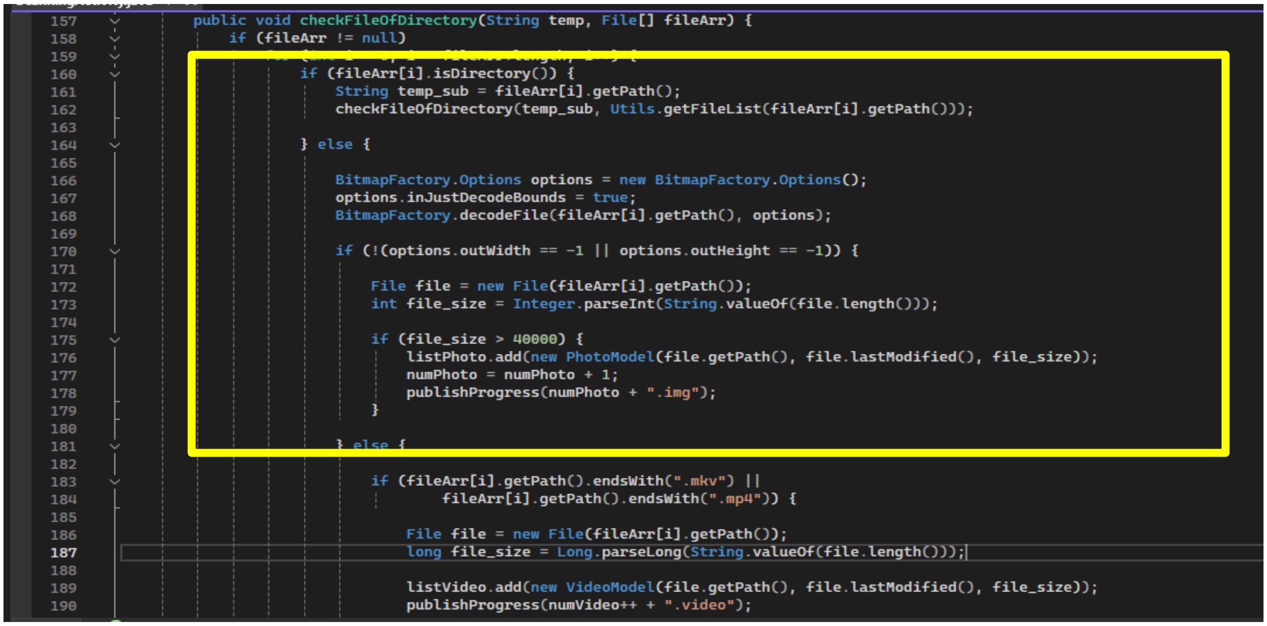
The AudioActivity.java file displays recovered audios in the data recovery application. The layout contains a Recycler View that displays audios in a linear layout Manager.



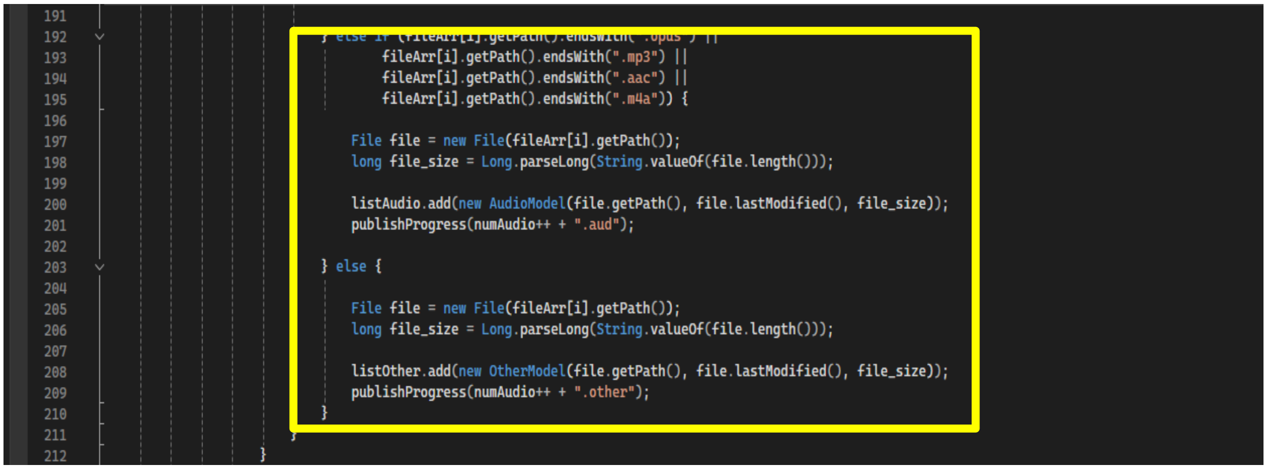
```
        });
    }

    @Override
    public void onBackPressed() {
        status = "back";
        gotonext();
    }

    private void gotonext() {

        if (status.equals("restore")) {
            final ArrayList<AudioModel> tempList = adapter.getSelectedItem();


            if (tempList.size() == 0) {
                Toast.makeText(AudioActivity.this, "Cannot restore, all items are unchecked!", Toast.LENGTH_LONG).sh
            } else {

                mRecoverPhotosAsyncTask = new RecoverAudiosAsyncTask(AudioActivity.this, adapter.getSelectedItem()
                    @Override
                    public void onComplete() {
                        Intent intent = new Intent(getApplicationContext(), RestoreResultActivity.class);
                        intent.putExtra("value", tempList.size());
                        startActivity(intent);
                        finish();
                        adapter.setAllImagesUnseleted();
                        adapter.notifyDataSetChanged();
                    }
                });
                mRecoverPhotosAsyncTask.execute();
            }
        } else if (status.equals("back")) {
            finish();
        }
```

Users can either restore or uncheck chosen audio files. The activity provides options for recovering audio files and managing back navigation.

4.1.6 Scanning Activity

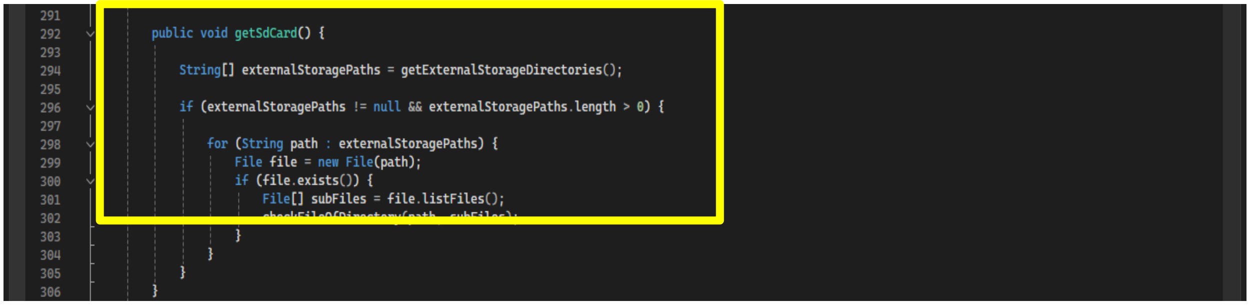


```java
public void checkFileOfDirectory(String temp, File[] fileArr) {
    if (fileArr != null)
        for (int i = 0; i < fileArr.length; i++) {
            if (fileArr[i].isDirectory()) {
                String temp_sub = fileArr[i].getPath();
                checkFileOfDirectory(temp_sub, Utils.getFileList(fileArr[i].getPath()));

            } else {

                BitmapFactory.Options options = new BitmapFactory.Options();
                options.inJustDecodeBounds = true;
                BitmapFactory.decodeFile(fileArr[i].getPath(), options);

                if (!(options.outWidth == -1 || options.outHeight == -1)) {

                    File file = new File(fileArr[i].getPath());
                    int file_size = Integer.parseInt(String.valueOf(file.length()));

                    if (file_size > 40000) {
                        listPhoto.add(new PhotoModel(file.getPath(), file.lastModified(), file_size));
                        numPhoto = numPhoto + 1;
                        publishProgress(numPhoto + ".img");
                    }

                } else {

                    if (fileArr[i].getPath().endsWith(".mkv") ||
                            fileArr[i].getPath().endsWith(".mp4")) {

                        File file = new File(fileArr[i].getPath());
                        long file_size = Long.parseLong(String.valueOf(file.length()));

                        listVideo.add(new VideoModel(file.getPath(), file.lastModified(), file_size));
                        publishProgress(numVideo++ + ".video");
```



```java
                    } else if (fileArr[i].getPath().endsWith(".opus") ||
                            fileArr[i].getPath().endsWith(".mp3") ||
                            fileArr[i].getPath().endsWith(".aac") ||
                            fileArr[i].getPath().endsWith(".m4a")) {

                        File file = new File(fileArr[i].getPath());
                        long file_size = Long.parseLong(String.valueOf(file.length()));

                        listAudio.add(new AudioModel(file.getPath(), file.lastModified(), file_size));
                        publishProgress(numAudio++ + ".aud");

                    } else {

                        File file = new File(fileArr[i].getPath());
                        long file_size = Long.parseLong(String.valueOf(file.length()));

                        listOther.add(new OtherModel(file.getPath(), file.lastModified(), file_size));
                        publishProgress(numAudio++ + ".other");
                    }
                }
            }
```

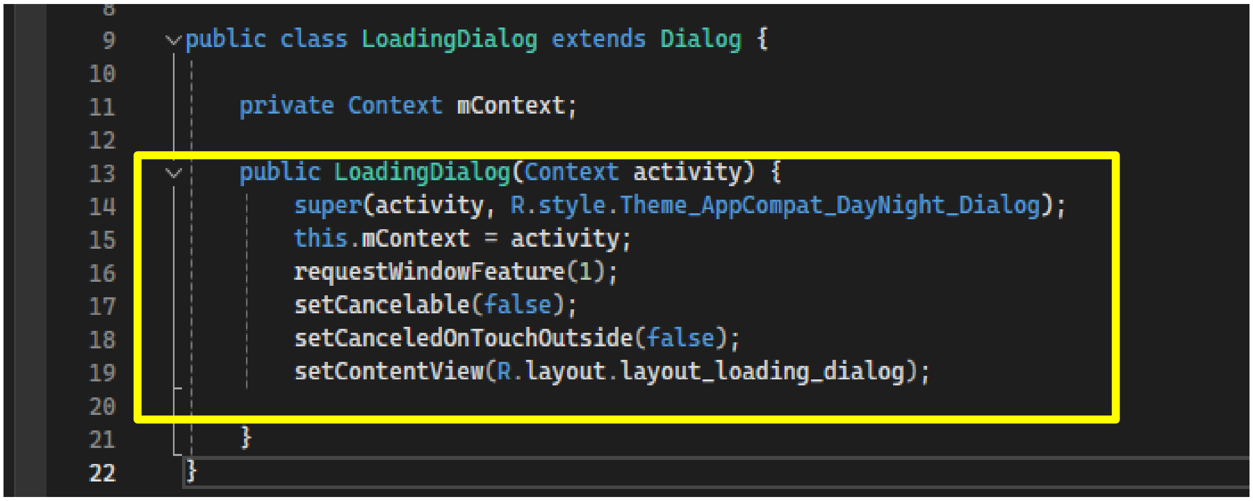
The “Scanning Activity” scans the storage of the targeted Android device for photos, videos and audio files that can be recovered. The scan is performed using Async Tasks which is a class in Android that makes it easy to run background tasks and update the user interface without needing to handle threads and handlers directly. It has been designed to simplify the process of performing asynchronous operations in an Android application. It improves the performance because it can scan the storage directories asynchronously.



```java
    public void getSdCard() {

        String[] externalStoragePaths = getExternalStorageDirectories();

        if (externalStoragePaths != null && externalStoragePaths.length > 0) {

            for (String path : externalStoragePaths) {
                File file = new File(path);
                if (file.exists()) {
                    File[] subFiles = file.listFiles();
                    checkFileOfDirectory(path, subFiles);
                }
            }
        }
    }
```

This method enables a user to scan and process all files in the external storage directories. It scans these directories to identify and categorize files like images, videos, audio files, and others.

4.2 **Loading Dialog**



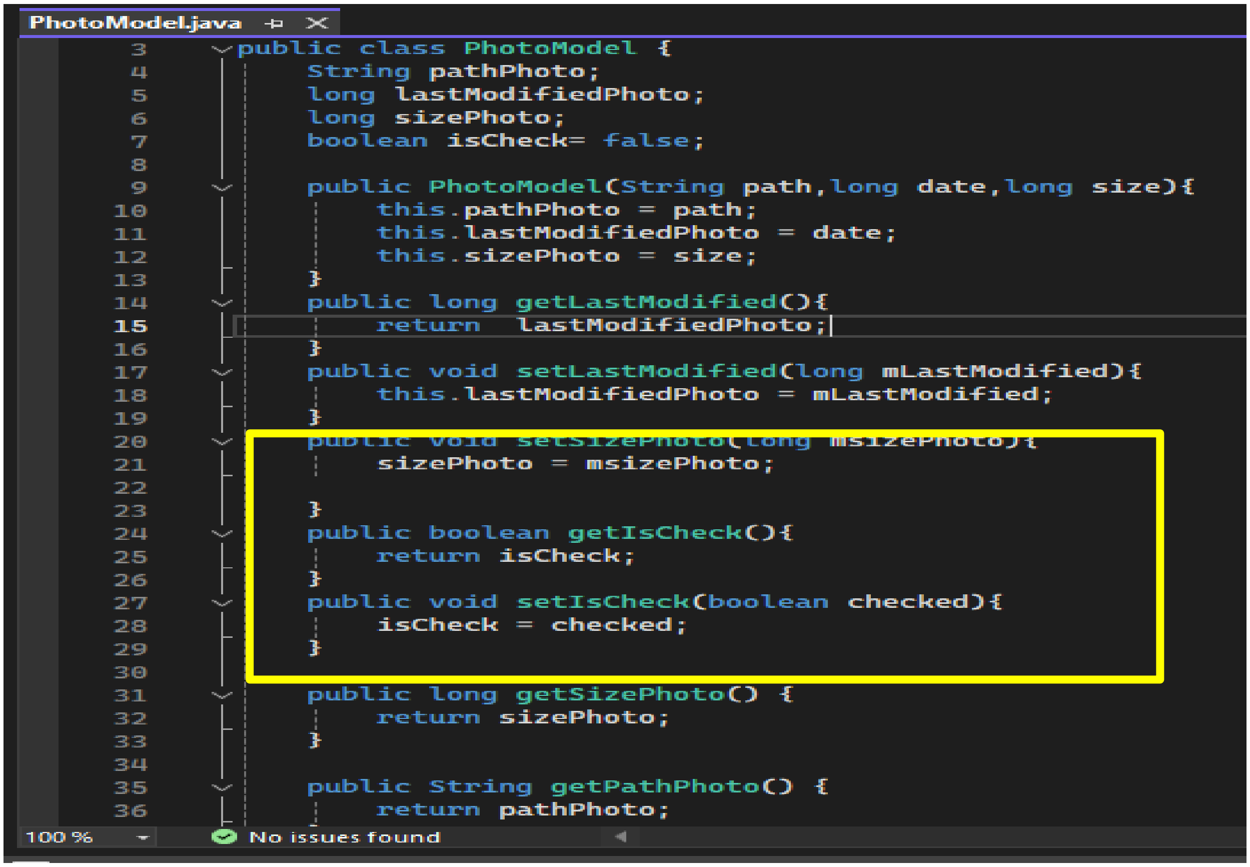
```java
public class LoadingDialog extends Dialog {

    private Context mContext;

    public LoadingDialog(Context activity) {
        super(activity, R.style.Theme_AppCompat_DayNight_Dialog);
        this.mContext = activity;
        requestWindowFeature(1);
        setCancelable(false);
        setCanceledOnTouchOutside(false);
        setContentView(R.layout.layout_loading_dialog);

    }
}
```

The 'Loading Dialog' class plays the main role in displaying loading animation while executing background operations. It is a custom style dialog that is built through “Theme App Compat Day Night Dialog”, which matches the dialog with the application theme. The dialog is so configured that it cannot be cancelled by using the back button or touching outside events. This layout is described in “layout loading dialog.xml”, which includes the loading animation.

4.3 Application Models

In this section, the model classes are listed which are: Photo Model, Video Model, Audio Model. Each model class refers to a type of content, such as images, movies, audio files. These classes help in providing important information such as the file location, latest changed date, size, and a flag indicating if the file is checked.

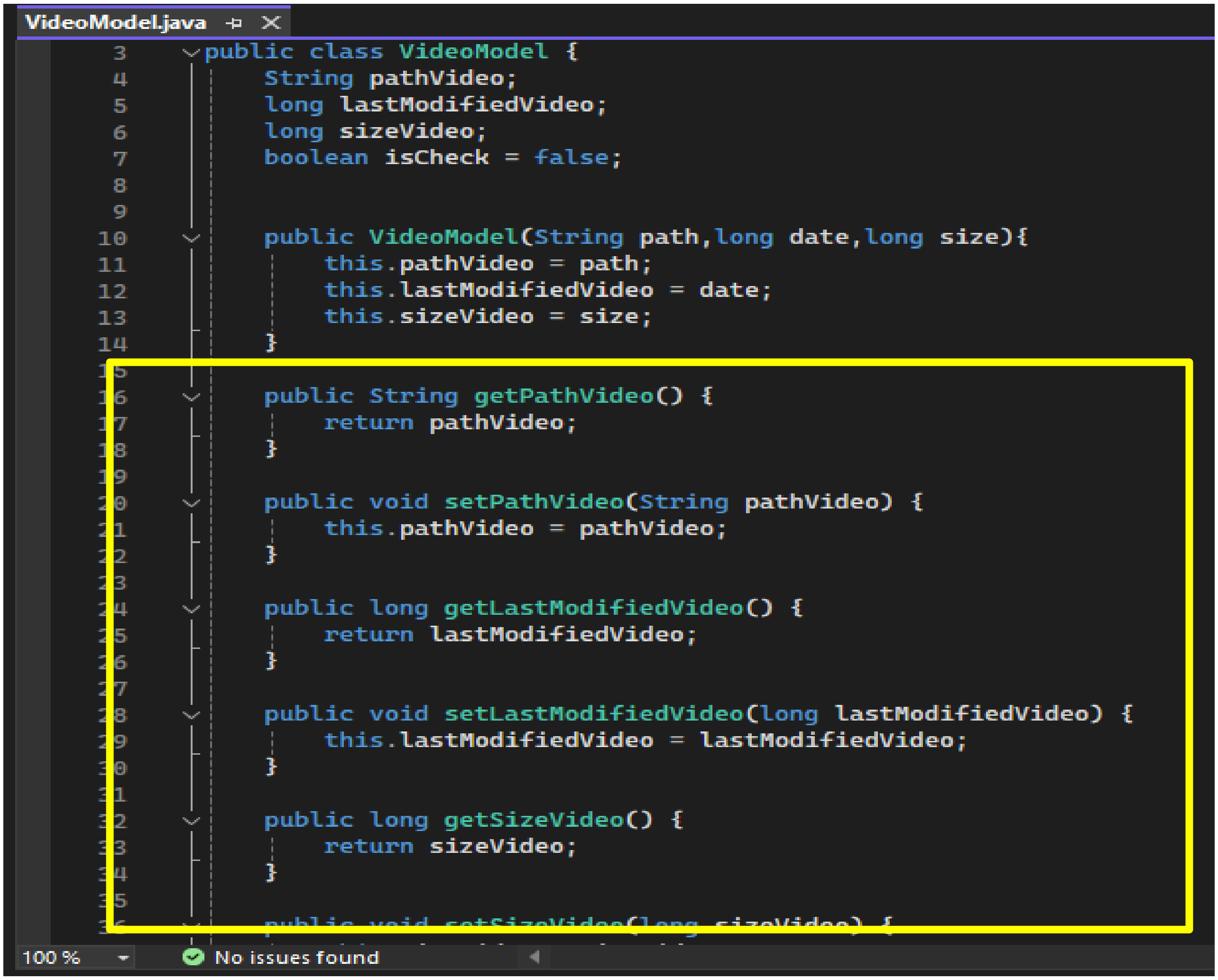
4.3.1 **Photo Model**



```java
public class PhotoModel {
    String pathPhoto;
    long lastModifiedPhoto;
    long sizePhoto;
    boolean isCheck= false;

    public PhotoModel(String path,long date,long size){
        this.pathPhoto = path;
        this.lastModifiedPhoto = date;
        this.sizePhoto = size;
    }
    public long getLastModified(){
        return  lastModifiedPhoto;
    }
    public void setLastModified(long mLastModified){
        this.lastModifiedPhoto = mLastModified;
    }
    public void setSizePhoto(long msizePhoto){
        sizePhoto = msizePhoto;

    }
    public boolean getIsCheck(){
        return isCheck;
    }
    public void setIsCheck(boolean checked){
        isCheck = checked;
    }

    public long getSizePhoto() {
        return sizePhoto;
    }

    public String getPathPhoto() {
        return pathPhoto;
```

This class is used in the Data Recovery Application to represent the details of photographs such as its path, latest edited date, size. The class uses functions like “get Last Modified”, “get Size Video”, “get Is Check”, and “set Is Check” for getting and setting these attributes. Main functionalities that are provided by these utilities are encapsulation and data integrity. This class provides the facility to administrate and manipulate of data within the application.

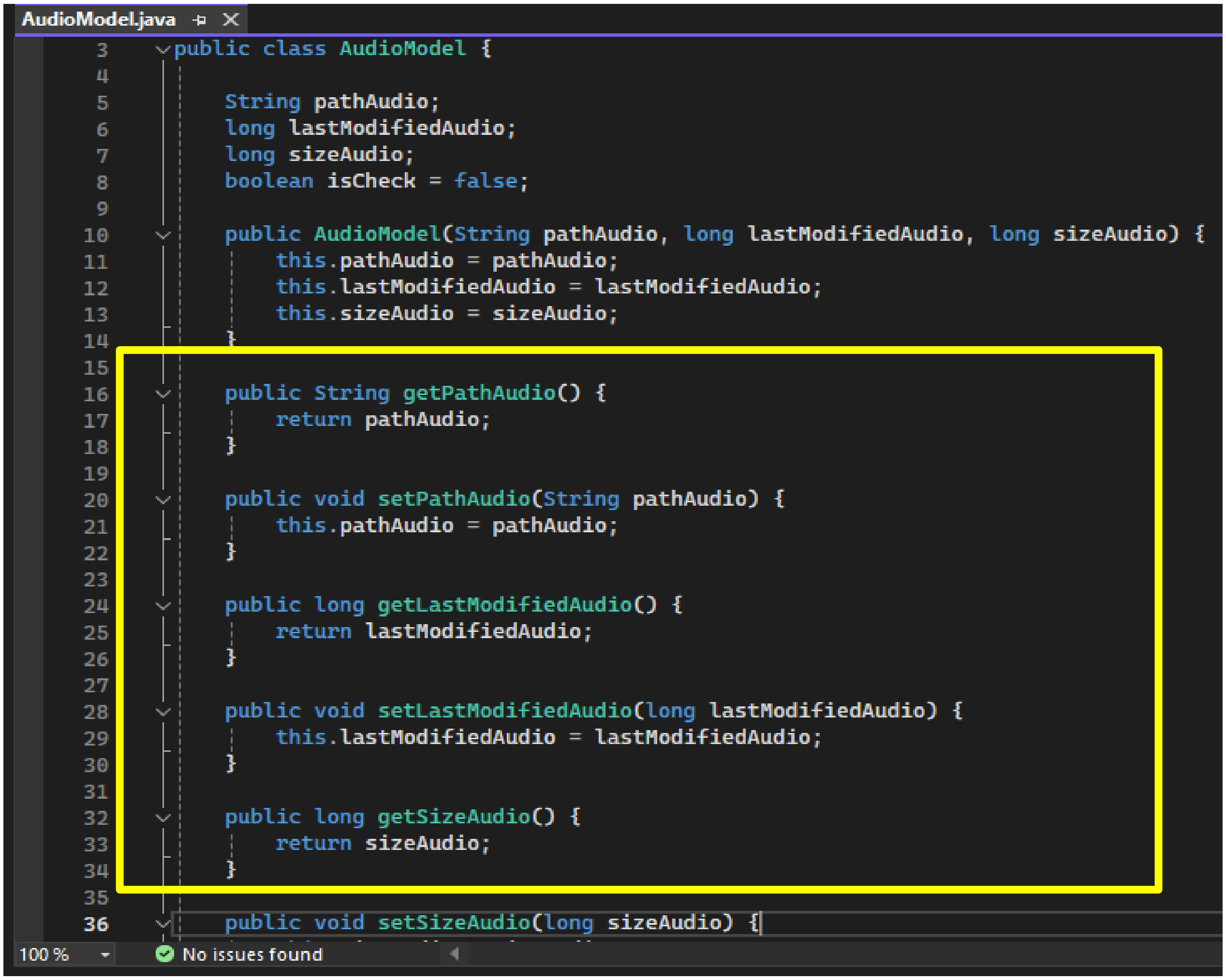
4.3.2 **Video Model**



```java
public class VideoModel {
    String pathVideo;
    long lastModifiedVideo;
    long sizeVideo;
    boolean isCheck = false;


    public VideoModel(String path,long date,long size){
        this.pathVideo = path;
        this.lastModifiedVideo = date;
        this.sizeVideo = size;
    }

    public String getPathVideo() {
        return pathVideo;
    }

    public void setPathVideo(String pathVideo) {
        this.pathVideo = pathVideo;
    }

    public long getLastModifiedVideo() {
        return lastModifiedVideo;
    }

    public void setLastModifiedVideo(long lastModifiedVideo) {
        this.lastModifiedVideo = lastModifiedVideo;
    }

    public long getSizeVideo() {
        return sizeVideo;
    }

    public void setSizeVideo(long sizeVideo) {
```

This class is used in the Data Recovery Application to represent the details of video file such as its path, latest edited date, size. The class uses functions like “get Last Modified”, “get Size Video”, “get Is Check”, and “set Is Check” for getting and setting these attributes. Main functionalities that are provided by these utilities are encapsulation and data integrity. This class provides the facility to administrate and manipulate of video data within the application.

4.3.3 **Audio Model**



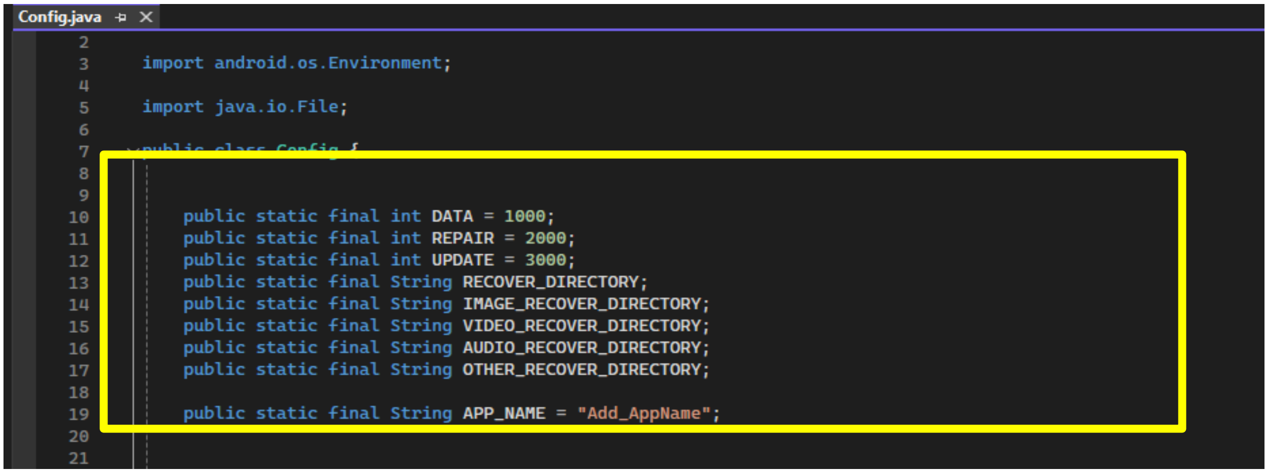
```java
public class AudioModel {

    String pathAudio;
    long lastModifiedAudio;
    long sizeAudio;
    boolean isCheck = false;

    public AudioModel(String pathAudio, long lastModifiedAudio, long sizeAudio) {
        this.pathAudio = pathAudio;
        this.lastModifiedAudio = lastModifiedAudio;
        this.sizeAudio = sizeAudio;
    }

    public String getPathAudio() {
        return pathAudio;
    }

    public void setPathAudio(String pathAudio) {
        this.pathAudio = pathAudio;
    }

    public long getLastModifiedAudio() {
        return lastModifiedAudio;
    }

    public void setLastModifiedAudio(long lastModifiedAudio) {
        this.lastModifiedAudio = lastModifiedAudio;
    }

    public long getSizeAudio() {
        return sizeAudio;
    }

    public void setSizeAudio(long sizeAudio) {
```

This class is used in the Data Recovery Application to represent the details of audio file such as its path, latest edited date, size. The class uses functions like “get Last Modified”, “get Size Audio”, “get Is Check”, and “set Is Check” for getting and setting these attributes. Main functionalities that are provided by these utilities are encapsulation and data integrity. This class provides the facility to administrate and manipulate of audio data within the application.

4.4 Application Utilities

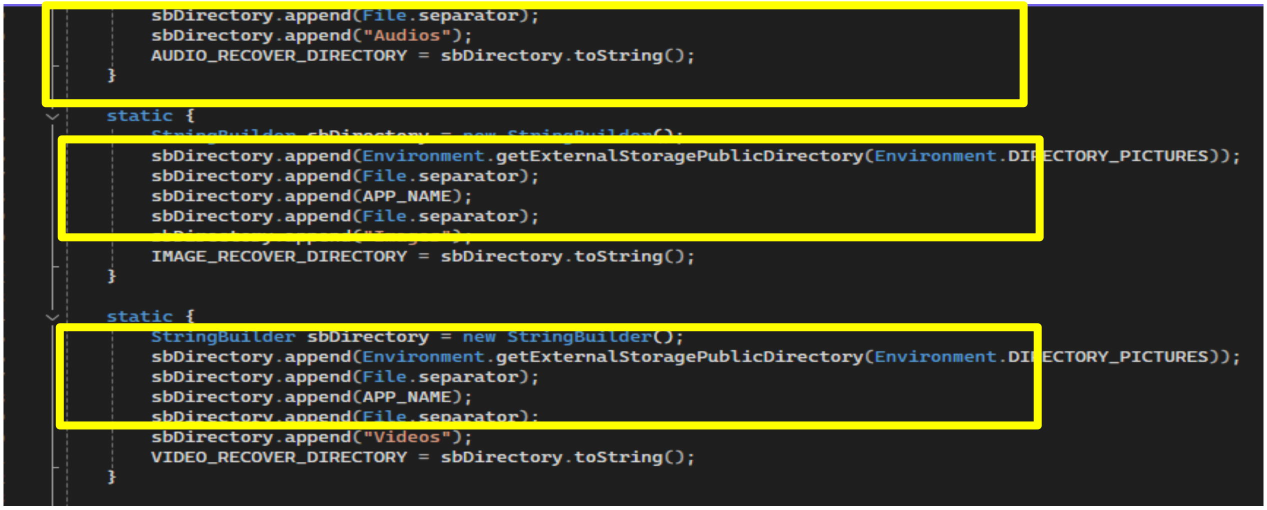
"Application Utilities" package provides utility classes that helps the application to work. These classes streamline routine operations and offer central functionality across many portions of the application.

4.4.1 Configuration



```
import android.os.Environment;

import java.io.File;

public class Config {


    public static final int DATA = 1000;
    public static final int REPAIR = 2000;
    public static final int UPDATE = 3000;
    public static final String RECOVER_DIRECTORY;
    public static final String IMAGE_RECOVER_DIRECTORY;
    public static final String VIDEO_RECOVER_DIRECTORY;
    public static final String AUDIO_RECOVER_DIRECTORY;
    public static final String OTHER_RECOVER_DIRECTORY;

    public static final String APP_NAME = "Add_AppName";

```

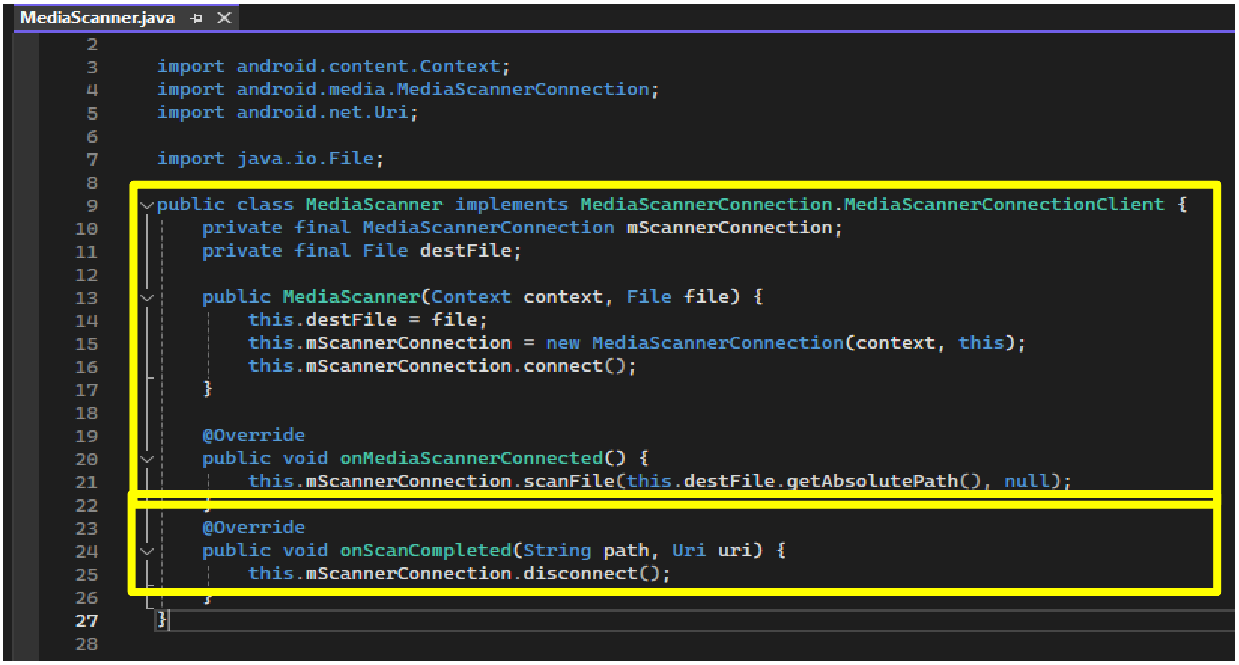
Configuration variables including Data, Repair, Update, Recover Directory, Image Recover Directory, Video Recover Directory and Audio Recover Directory are being used in this Configuration Class that serves their role throughout the Data Recovery Application.



```
            sbDirectory.append(File.separator);
            sbDirectory.append("Audios");
            AUDIO_RECOVER_DIRECTORY = sbDirectory.toString();
        }

        static {
            StringBuilder sbDirectory = new StringBuilder();
            sbDirectory.append(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES));
            sbDirectory.append(File.separator);
            sbDirectory.append(APP_NAME);
            sbDirectory.append(File.separator);
            sbDirectory.append("Images");
            IMAGE_RECOVER_DIRECTORY = sbDirectory.toString();
        }

        static {
            StringBuilder sbDirectory = new StringBuilder();
            sbDirectory.append(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES));
            sbDirectory.append(File.separator);
            sbDirectory.append(APP_NAME);
            sbDirectory.append(File.separator);
            sbDirectory.append("Videos");
            VIDEO_RECOVER_DIRECTORY = sbDirectory.toString();
        }
```

Folders are created such as Photos, Video, Audios within the application's directory. These folders are named using the application's name and its external storage directory paths.

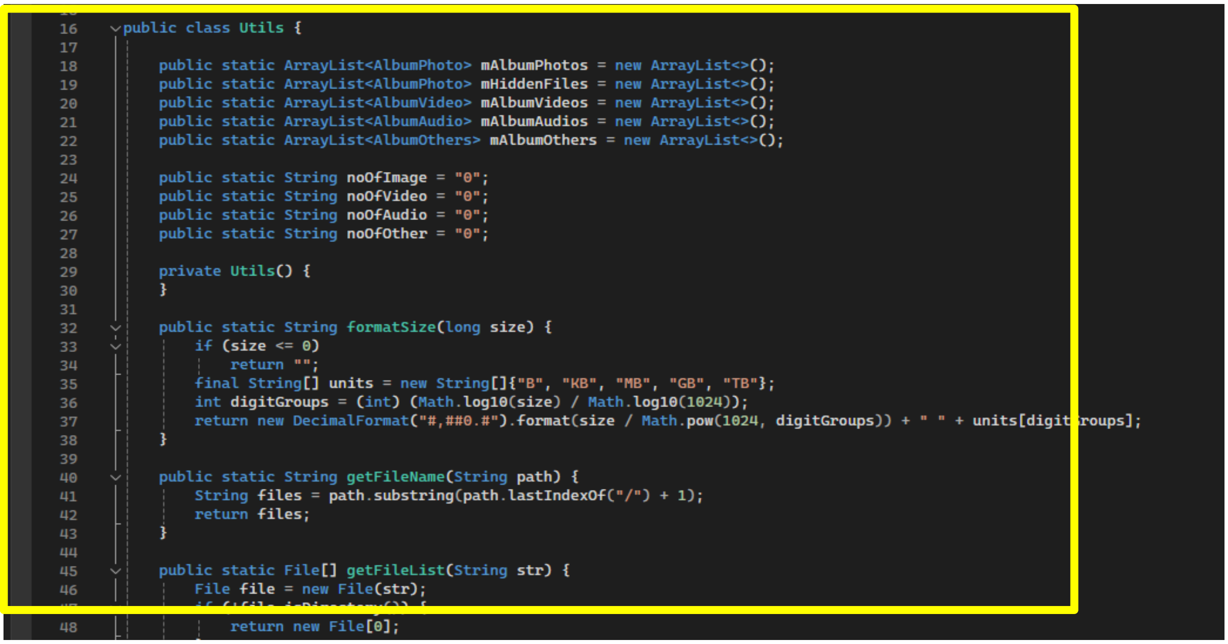
4.4.2 **Media Scanner**



```

import android.content.Context;
import android.media.MediaScannerConnection;
import android.net.Uri;

import java.io.File;

public class MediaScanner implements MediaScannerConnection.MediaScannerConnectionClient {
    private final MediaScannerConnection mScannerConnection;
    private final File destFile;

    public MediaScanner(Context context, File file) {
        this.destFile = file;
        this.mScannerConnection = new MediaScannerConnection(context, this);
        this.mScannerConnection.connect();
    }

    @Override
    public void onMediaScannerConnected() {
        this.mScannerConnection.scanFile(this.destFile.getAbsolutePath(), null);
    }
    @Override
    public void onScanCompleted(String path, Uri uri) {
        this.mScannerConnection.disconnect();
    }
}

```

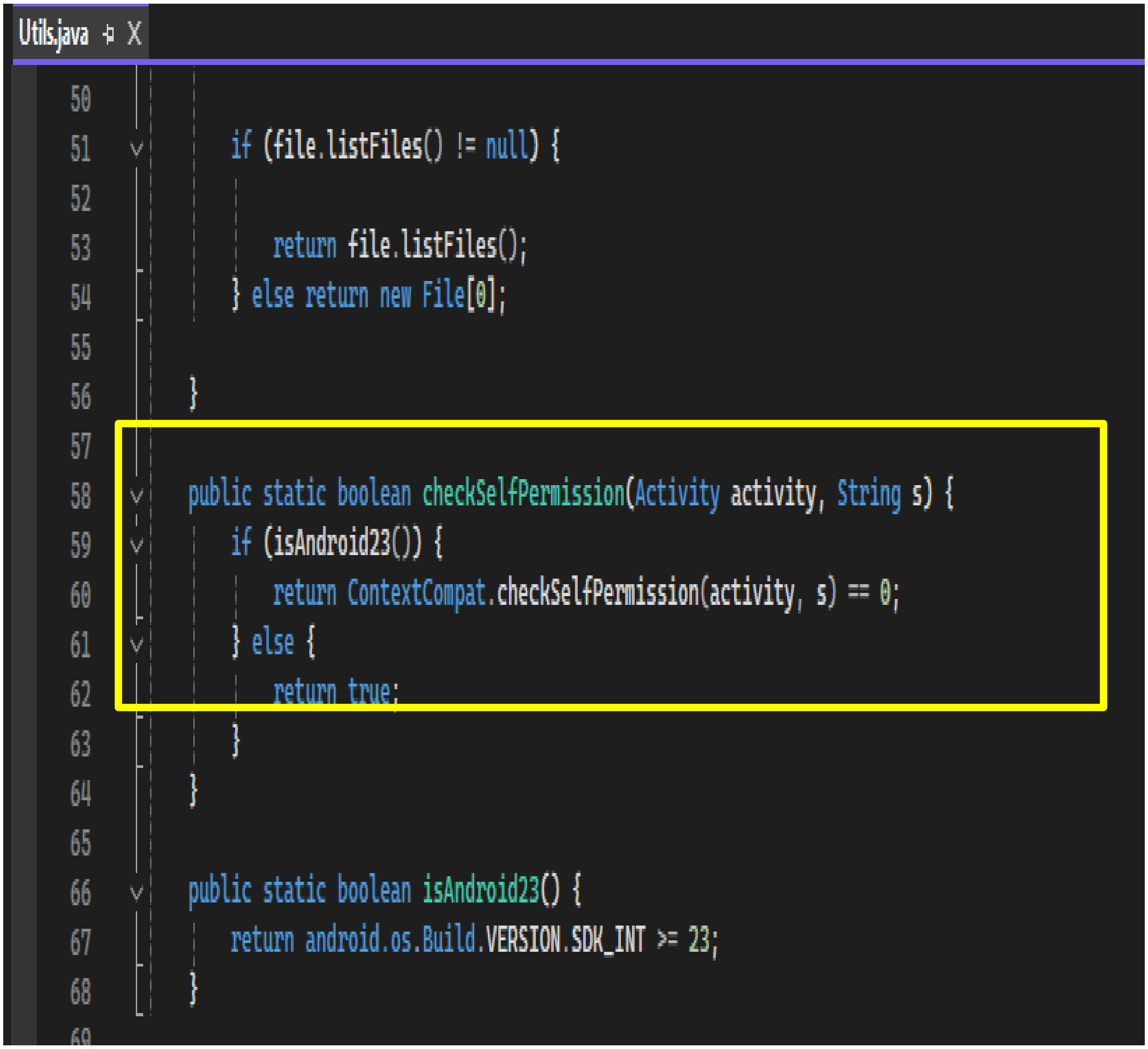
This class scans and updates media files on the device by implementing the "Media Scanner Connection" and "Media Scanner Connection Client" interface is used to manage media scanning events. When the scanning operations need to be performed on a file, it connects the file to the media scanner. Upon connecting, it performs a scan using the “Scan File” method. When the scan is finished, the “on Scan Completed” method is called to disconnect the scanner. This class guarantees that the system has updated or freshly produced the media files and they have been made available for usage in the application.

4.4.3 **Utilities**



```java
public class Utils {

    public static ArrayList<AlbumPhoto> mAlbumPhotos = new ArrayList<>();
    public static ArrayList<AlbumPhoto> mHiddenFiles = new ArrayList<>();
    public static ArrayList<AlbumVideo> mAlbumVideos = new ArrayList<>();
    public static ArrayList<AlbumAudio> mAlbumAudios = new ArrayList<>();
    public static ArrayList<AlbumOthers> mAlbumOthers = new ArrayList<>();

    public static String noOfImage = "0";
    public static String noOfVideo = "0";
    public static String noOfAudio = "0";
    public static String noOfOther = "0";

    private Utils() {
    }

    public static String formatSize(long size) {
        if (size <= 0)
            return "";
        final String[] units = new String[]{"B", "KB", "MB", "GB", "TB"};
        int digitGroups = (int) (Math.log10(size) / Math.log10(1024));
        return new DecimalFormat("#,##0.#").format(size / Math.pow(1024, digitGroups)) + " " + units[digitGroups];
    }

    public static String getFileName(String path) {
        String files = path.substring(path.lastIndexOf("/") + 1);
        return files;
    }

    public static File[] getFileList(String str) {
        File file = new File(str);
        if (!file.isDirectory()) {
            return new File[0];
```

This Class uses static variables to store lists of albums for images, videos and audios. Which allows the real quick access and control of recovered information. This Utils class also contains the “utility methods” which have been used throughout the Data Recovery application for formatting file sizes, extracting file names from paths, and retrieving directory listings.

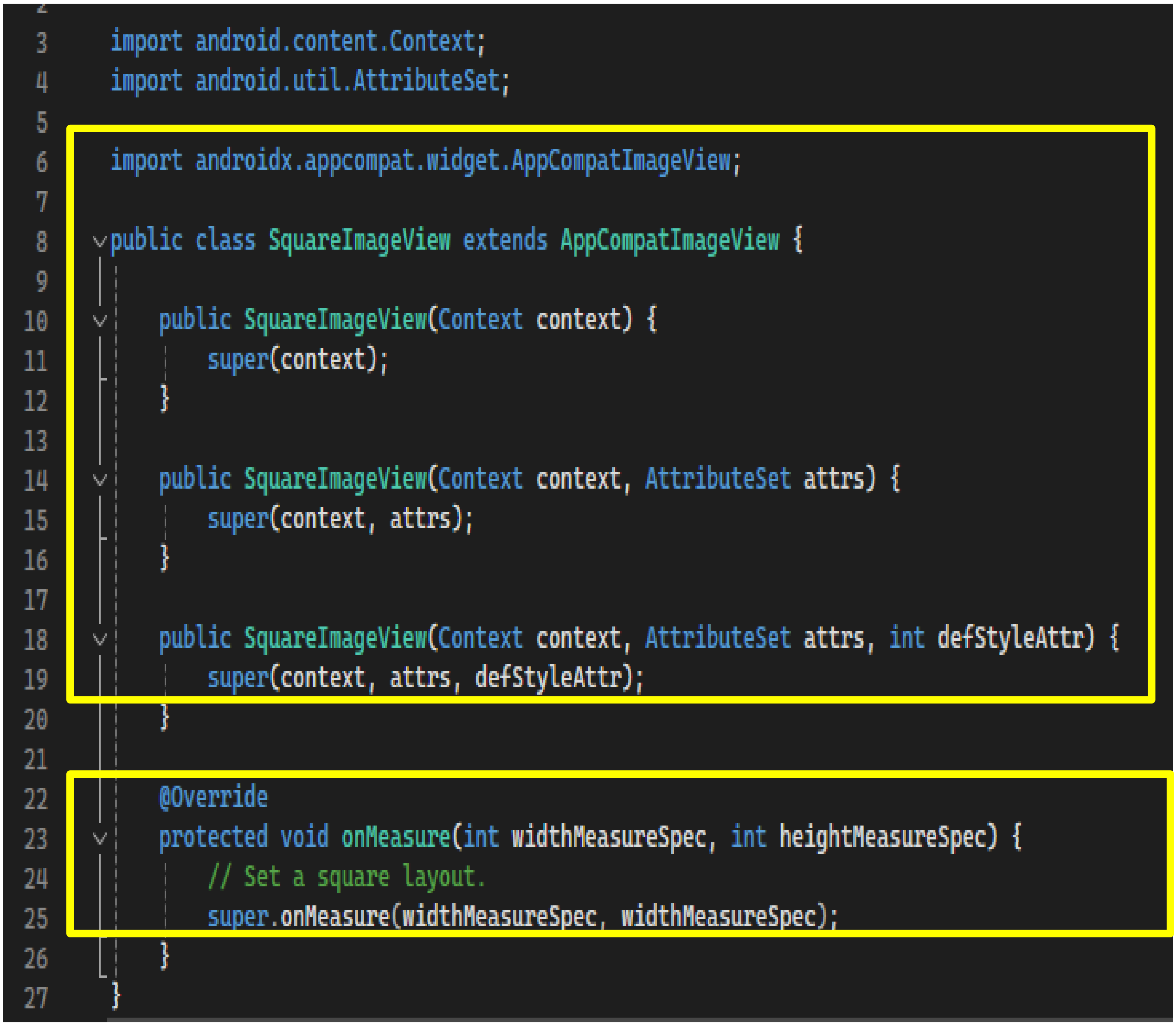


```java

        if (file.listFiles() != null) {

            return file.listFiles();
        } else return new File[0];

    }

    public static boolean checkSelfPermission(Activity activity, String s) {
        if (isAndroid23()) {
            return ContextCompat.checkSelfPermission(activity, s) == 0;
        } else {
            return true;
        }
    }

    public static boolean isAndroid23() {
        return android.os.Build.VERSION.SDK_INT >= 23;
    }
```

The “check Self Permission” is a function that checks permissions for Android 23 (Marshmallow) and higher, whereas “is Android23” function checks the Android version for conditional actions. Overall, this class centralizes common utility functions that is being used across the application, making them easier to organize and manage.

4.5 **Application View**

4.5.1 **Square Image View**



```java
import android.content.Context;
import android.util.AttributeSet;

import androidx.appcompat.widget.AppCompatImageView;

public class SquareImageView extends AppCompatImageView {

    public SquareImageView(Context context) {
        super(context);
    }

    public SquareImageView(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public SquareImageView(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
    }

    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
        // Set a square layout.
        super.onMeasure(widthMeasureSpec, widthMeasureSpec);
    }
}
```

The "Square Image View" class from the "Apps View" package of the Data Recovery application extends the App Compact Image View to produce a square image view. This custom view guarantees that pictures shown within it have a square ratio irrespective of screen size or orientation.

CHAPTER 5: CONCLUSION

In modern life, smartphones—especially phones with the Android operating system—have become increasingly commonplace almost each individual have a phone even a small kid is having a smartphone. This has highlighted the urgent need and requirement for efficient data recovery solutions which can be fully trusted. The danger of data loss via unintentional deletion, software bugs, or other problems increases and it may also be concerning as more data is kept on these devices. In order to solve this prominent issue, this thesis demonstrates the development of a refined, user-friendly Android data recovery tool that is intended to offer trustworthy and efficient data recovery options for Android users.

5.1 **Understanding the Problem**

This thesis emphasizes how dominant and important is the Android operating system is in the worldwide market beating all other operating system in terms of numbers and its uses, with 81.2% of the market in 2015—far more than any of its rivals. [19]. Because of its widespread adoption, a significant quantity of personal and business data is kept on Android smartphones. But doing so also increases the risk of data loss, which can have dangerous repercussions for users and for the businesses as well. Due to difficulties including low recovery success rates, complicated and high price structures, compatibility challenges, and security threats connected with internet recovery tools, existing data recovery solutions frequently fall short of consumers' demands.

5.2 The Proposed Solution

The thesis suggests a powerful mobile-based Android data recovery tool to allay and lessen these worries. This software objective is to increase the success rate of data recovery, guarantee compatibility with a broad spectrum of devices and new versions of operating systems, and improve data security through offline operation in which there will be no requirement if the internet. The following are some of the main characteristics and development techniques used in this tool:

i. **Progressive Algorithms for Data Recovery Mechanism:** The software uses refined processes to examine the Android mobile devices' internal and external storage both which may be an SD card. Even in some situations when there is a mobile device which is damaged or encrypted, these techniques may also try to help but not guaranteed with the identification and recovery of lost or deleted data.

ii. **User-Friendly Interface**: The software is designed in such a way that revolves around and help the user for an easy-to-use graphical user interface (GUI), which enables users to start and stop the data recovery process at any time with ease.

iii. **Data Carving and File System Analysis:** A technique will be used which will identify and find out the basic data patterns or file signatures, the algorithm will also help to retrieve files that have been partially or completely wiped from the mobile phone storage. By looking into the smartphone's file system data structures, file system analysis techniques may be utilized to find out and recover deleted files from the mobile.

iv. **Root Access and Custom Recovery**: The application can perform refined recovery events including backing up of all the mobile data and directly extracting data from SD cards or internal storage by gaining elevated rights through rooting if it is necessary otherwise this option will not be used.

v. **Offline Capabilities for Added Safety:** This application will be functioned without the use of internet which will guarantee the security and integrity of user data, operating the application offline will also lower the danger of data theft or loss during the recovery process.

5.3 Methodology and Implementation

The development process of the Android Data Recovery Tool was comprehensive, involving several key stages to ensure its effectiveness and reliability:

i. **Device Detection and Connection**: The tool's Device Scanner component initiates the process by detecting and securely connecting to the target Android device using industry-standard protocols.

ii. **Data Scanning and Analysis**: The Data Scanner section carefully scans the device's storage, while the Data Analyzer section identifies possibly recoverable files in real-time. Advanced algorithms assess file structures, detect data fragments, and evaluate the integrity of recoverable data.

iii. **Recovery Process Initiation:** To reclaim deleted information, the Data Recovery Engine which is the main component of the software will manage the method of recovery by employing a variety of techniques such file carving, metadata reconstruction, and content extraction.

iv. **File Reconstruction and Integrity Verification:** To maintain integrity of the data in order to avoid data loss, the recovery engine uses improvement for errors and checksum verification algorithms when reconstructing restored data segments into whole files.

v. **Data Relocation and Storage:** when the retrieval process is over, the user will specify a location where the recovered data is sent, it will guarantee for the safe and effective data management.

vi. **User Feedback and Contact**: Real-time information on the data recovery process, which including the discovered, recovered, and awaiting file statuses, are given to users via the user interface.

5.4 Achievements and Deliverables

The main outcome of our project is that, it will be an innovative and efficient Android data recovery tool that will solves the drawbacks of currently available software's in the market.

Important accomplishments of this approach consist of:

i. **Enhanced Data Recovery Success**: The application performs very well from the alternatives in terms of restoration accuracy rates to its unique algorithms and techniques.

ii. **Economical Approach**: The application was created internally and offers customers a free alternative to pricey privately owned software.

iii. **Smooth Interoperability**: The application verifies that it is compatible with the most recent Android smartphones and operating systems, and it updates often to ensure optimal efficiency.

iv. **Broader Recovery Range**: This solution offers a thorough recovery scope since, in contrast to some other solutions, it can recover data from internal device memory as well as SD cards.

v. **Increased Privacy with Offline features**: The sue of this tool will have considerably lower security concerns by functioning offline, guaranteeing the integrity and confidentiality of user data.

5.5 Future Directions

Even if our Android Data Recovery Tool shows a considerable success in the industry of a data recovery for mobiles, studies and research may go ahead further in this field:

i. **Improving Integration**: To increase its compliance, the next version of the program after adding addition features might accept a larger variety of file formats and Android operating system versions.

ii. **Improving Algorithms:** The tool's effectiveness and success percentages might be raised even more by constantly improving its restoration algorithms and its process and procedures.

iii. **Connectivity with Cloud Facilities**: Offering consumers additional choices for data backup and restoration may be achieved by improving capabilities for seamless integration with providers of cloud storage but it may also include online activity.

iv. **User Education**: By creating thorough user manuals and tutorials, users may get the most of the product by learning how to properly utilize it and maximize its advantages.

Concerning the problems that are associated with data loss on Android mobile phones, our software named Android Data Recovery Tool is a major development. Through the utilization of cutting-edge technology, a availability of user-friendly interface, and strong functionality, the software offers its customers all over the world a dependable and skilled resolution. In addition to adding to the body of knowledge on data recovery methods, this thesis has given users a very useful tool that may meaningfully alter how they protect their digital assets placed in their smart phones. The effective design and application of this solution also highlight the significance of ongoing innovation and study in the data recovery space, guaranteeing that the consumers may depend on their digital devices with confidence and not worry about losing important data.

REFERENCES

[1] Dr.E.N.Ganesh ,DEAN VISTAS . Data Recovery on Android systems using Open Software Tools. https://www.researchgate.net/publication/364211170_Data_Recovery_on_Android_systems_using_Open_Software_Tools

[2] Casey, Eoghan, and Golden Richard. "Digital evidence file system analysis." Digital Investigation 3, no. 1 (2006): 4-11. https://rishikeshpansare.files.wordpress.com/2016/02/digital-evidence-and-computer-crime-third-edition.pdf

[3] Rogers, Roger G. "Rootkits on Android: What are they? What do they do?." Journal in Computer Virology and Hacking Techniques 9, no. 4 (2013): 163-175. https://link.springer.com/journal/11416

[4] Akhil C Joy, Arjun Mithun, K Shebix Xavier. Implementing Deletion Recovery on Android. https://www.researchpublish.com/upload/book/Restore%20Bin%20Restoration%20on%20Android%20Device-1510.pdf

[5] Garfinkel, Simson, and David Margrave. "The security of proprietary data recovery tools." In International Conference on Digital Forensics and Cyber Crime, pp. 3-15. Springer, Cham, 2016. https://www.marketresearchfuture.com/reports/data-recovery-software-market-5739

[6] Casey, Eoghan, et al. "An experimental study of the Windows recycle bin." Digital Investigation 5 (2008): S64-S71.

[7] Chevalier, Steven, and Charles S. Patterson. "Where is the data? Why you cannot debate CPU vs. GPU performance without the answer." ACM SIGOPS Operating Systems Review 44, no. 1 (2010): 42-49.

[8] Brian D Carrier. "Digital investigation and forensic analysis of disk storage." Digital Investigation 6, Supplement (2009): S16-S22. https://www.researchgate.net/publication/220542507_Defining_Digital_Forensic_Examination_and_Analysis_Tool_Using_Abstraction_Layers

[9] Khushboo Rathi. Selected Forensic Data Acquisition from Android Devices. https://shsu-ir.tdl.org/server/api/core/bitstreams/74b6c737-359f-41b1-b6db-1482459b6682/content

[10] Baggili, Ibrahim, et al. "Digging for data: identifying unique data on mobile devices." Digital Investigation 9 (2012): S60-S68.

[11] Breitinger, Frank, et al. "Filling the gap: Integrated cyber-physical systems for mobile device and network forensics." In International Conference on Availability, Reliability, and Security, pp. 558-571. Springer, Cham, 2016.

[12] Wondershare. (2023). Dr.Fone - Data Recovery (Android). Wondershare Official Website. Retrieved from Wondershare Dr.Fone. https://drfone.wondershare.com/android-data-recovery.html

[13] Smith, A. (2015). The Economics of Data Recovery Software. Journal of Information Technology Management, 26(3), 45-56.

[14] DATA RESTORATION FOR ANDROID by H. Stefi Sterlin; P. Gearam Balaji; A. Edwin. [online] Available at: https://sci-hub.se/10.1109/ICICES.2017.8070762

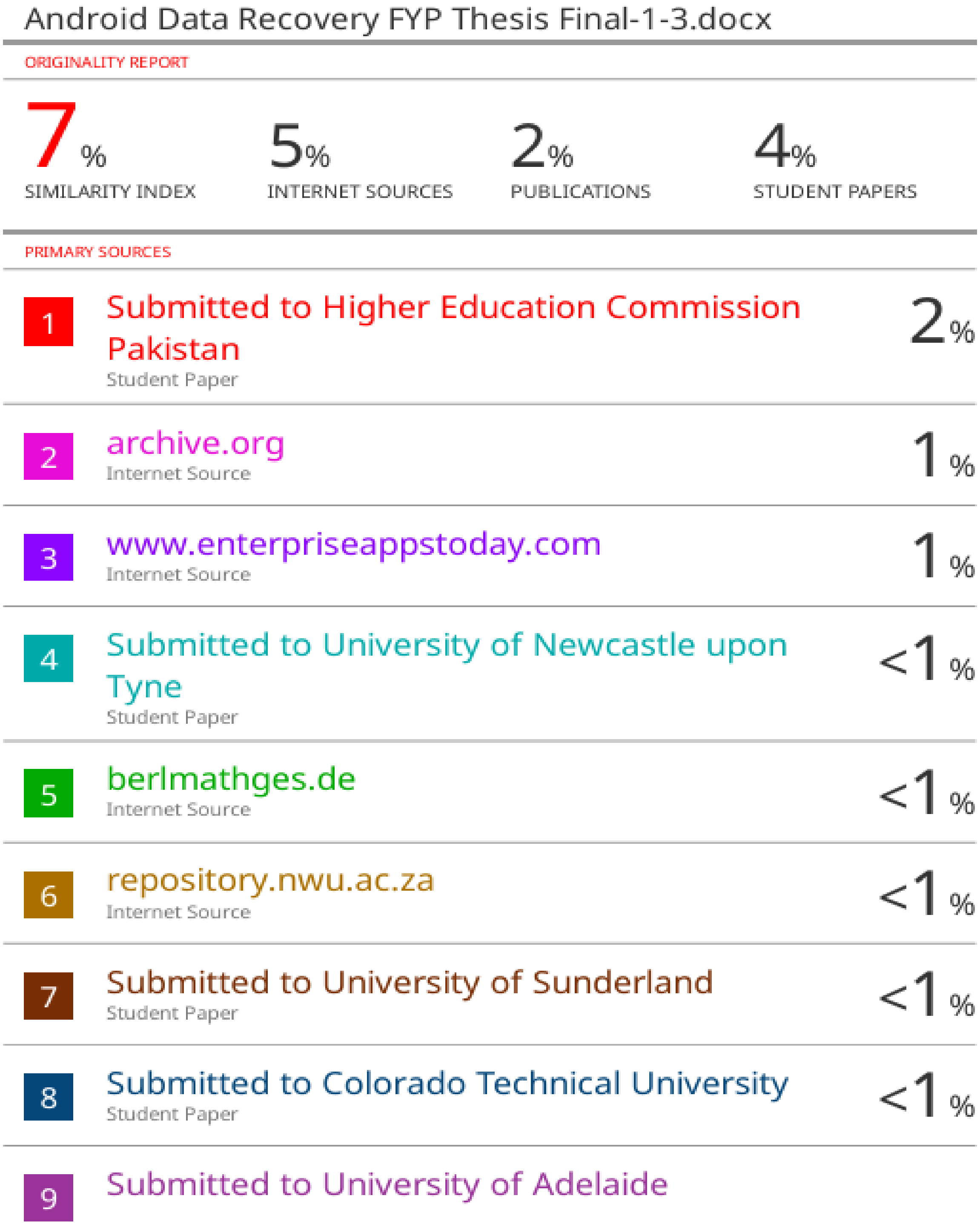
[15] The Value of Software Architecture Recovery for Maintenance by Daniel Link Ramin Moazeni. [online] Available at: https://arxiv.org/abs/1901.07700

[16] Application of Android-based Learning Media Applications by S.Andri.
[online] Available at:https://journal.cattleyadf.org/index.php/jatilima/article/view/389

[17] DATA RESTORATION FOR ANDROID by H. Stefi Sterlin; P. Gearam Balaji; A. Edwin. [online] Available at: https://sci-hub.se/10.1109/ICICES.2017.8070762

[18] Nihar Ranjan Roy; Anshul Kanchan Khanna; Leesha Aneja. Android phone forensic: Tools and techniques. [online] Available at: https://sci-hub.se/10.1109/CCAA.2016.7813792

PLAGIRISM REPORT



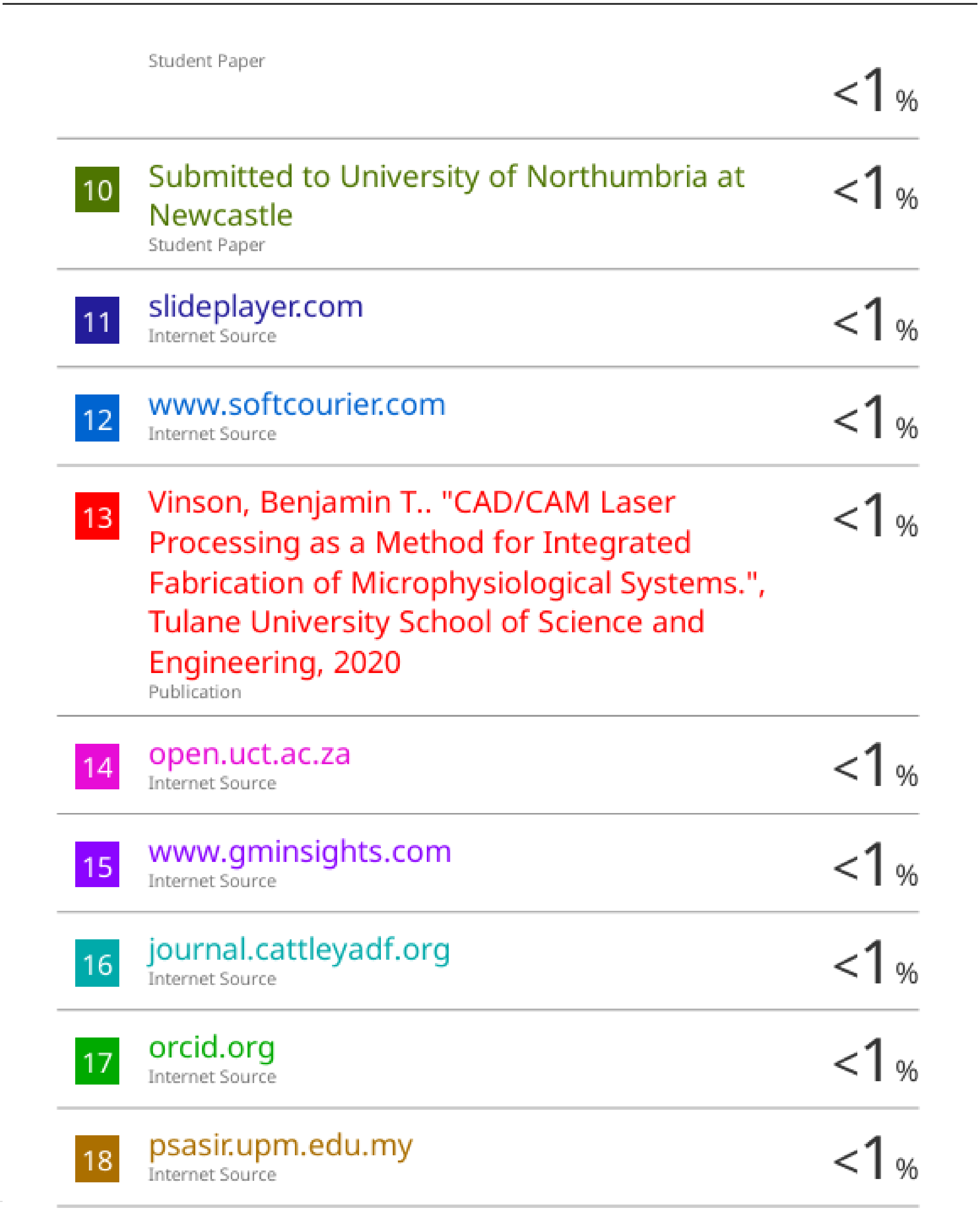
Android Data Recovery FYP Thesis Final-1-3.docx

ORIGINALITY REPORT

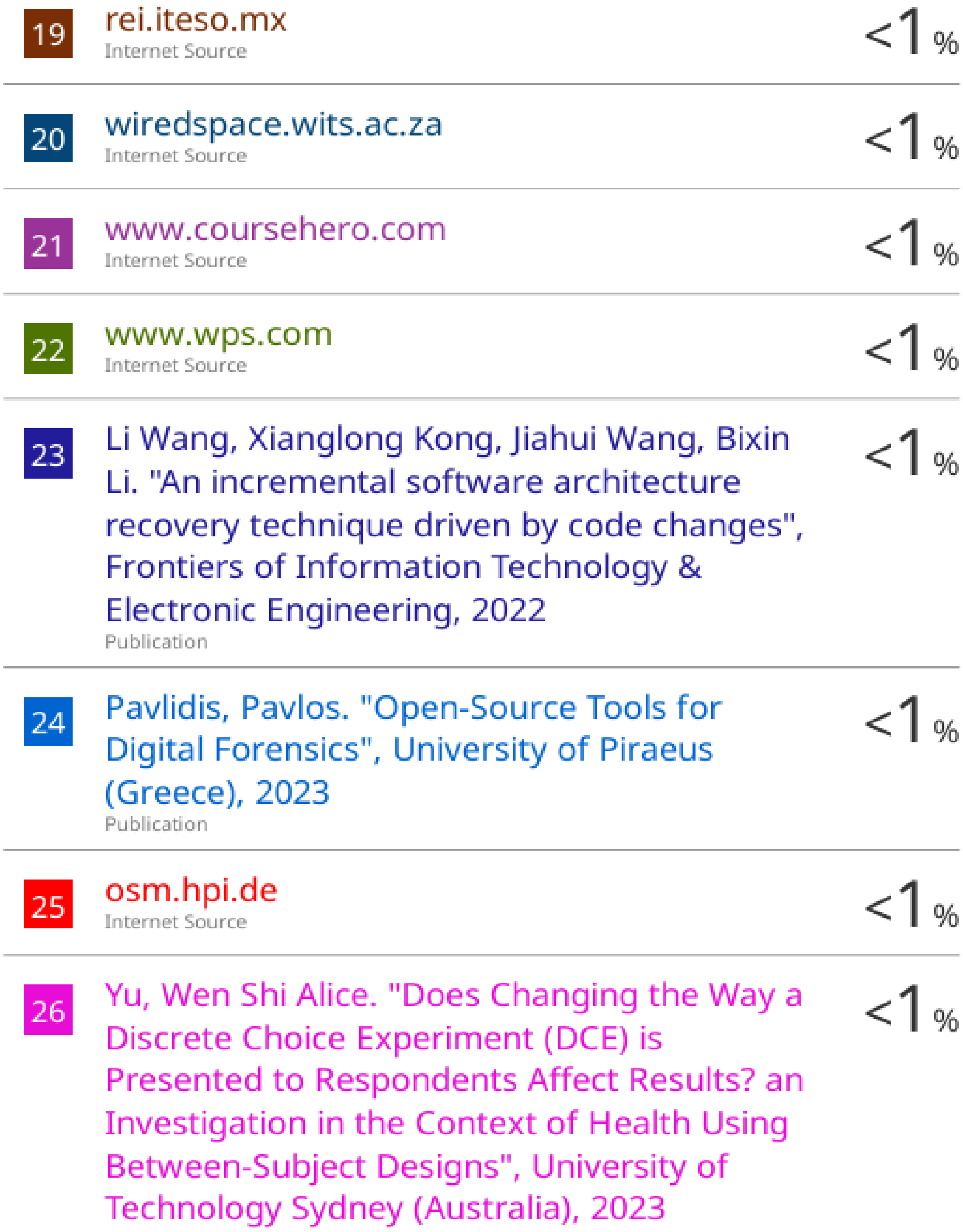
7%	5%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	2%
2	archive.org Internet Source	1%
3	www.enterpriseappstoday.com Internet Source	1%
4	Submitted to University of Newcastle upon Tyne Student Paper	<1%
5	berlmathges.de Internet Source	<1%
6	repository.nwu.ac.za Internet Source	<1%
7	Submitted to University of Sunderland Student Paper	<1%
8	Submitted to Colorado Technical University Student Paper	<1%
9	Submitted to University of Adelaide	



	Student Paper	<1 %
10	Submitted to University of Northumbria at Newcastle Student Paper	<1 %
11	slideplayer.com Internet Source	<1 %
12	www.softcourier.com Internet Source	<1 %
13	Vinson, Benjamin T.. "CAD/CAM Laser Processing as a Method for Integrated Fabrication of Microphysiological Systems.", Tulane University School of Science and Engineering, 2020 Publication	<1 %
14	open.uct.ac.za Internet Source	<1 %
15	www.gminsights.com Internet Source	<1 %
16	journal.cattleyadf.org Internet Source	<1 %
17	orcid.org Internet Source	<1 %
18	psasir.upm.edu.my Internet Source	<1 %



19	rei.iteso.mx Internet Source	<1 %
20	wiredspace.wits.ac.za Internet Source	<1 %
21	www.coursehero.com Internet Source	<1 %
22	www.wps.com Internet Source	<1 %
23	Li Wang, Xianglong Kong, Jiahui Wang, Bixin Li. "An incremental software architecture recovery technique driven by code changes", Frontiers of Information Technology & Electronic Engineering, 2022 Publication	<1 %
24	Pavlidis, Pavlos. "Open-Source Tools for Digital Forensics", University of Piraeus (Greece), 2023 Publication	<1 %
25	osm.hpi.de Internet Source	<1 %
26	Yu, Wen Shi Alice. "Does Changing the Way a Discrete Choice Experiment (DCE) is Presented to Respondents Affect Results? an Investigation in the Context of Health Using Between-Subject Designs", University of Technology Sydney (Australia), 2023	<1 %