A

PROJECT REPORT ON

File Manager and Duplicate Remover for Android Devices

By

Hardik Bagada(CE009)(17CEUSG001)
Mahendrasinh Barad(CE011)(17CEUBG136)
Harshad Chovatiya(CE019)(17CEUOT078)

B.Tech CE Semester-VI
Subject: (CE 621) System Design Practice
Project Title: Swift File Manager

Guided By:
Prof. Pandav K. Patel
Assistant Professor
Department of Computer Engineering

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

CERTIFICATE

This is to certify that the practical work/term work carried out in the subject of **System Design Practice** and recorded in this journal is the bonafide work of

Hardik Bagada(CE009)(17CEUSG001)
Mahendrasinh Barad(CE011)(17CEUBG136)
Harshad Chovatiya(CE019)(17CEUOT078)

Of B.Tech Semester **VI** in the Branch of **Computer Engineering** during the Academic year **2019-2020**.

Prof. Pandav K. Patel
Assistant Professor,
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University, Nadiad

Dr. C.K. Bhensdadia,
Head of Department
Of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University, Nadiad

CONTENTS

Abstract

It is very difficult to manage all the files and folders in an android device when the number of files and folders continuously increases day-to-day.Our application Swift File Manager helps users to manage all of his/her documents and keep them safe without loss of any data. An application also helps users to manage files based on their types.

Due to the long term uses of a device, storage memory of devices get fulled because of some duplicate files present in the device from which the user is unaware. Our application helps users to find all the duplicate files present in the device and also provides the functionality to delete them.

Introduction

Swift File Manager in an android application developed using Dart as a programming language. Users can easily store all of his files and documents and can also easily get them back when needed. The application also provides functionality to display files of any specific type. Through an application, Users can also find out a list of all duplicate files available on the system.

Technology Used:

❏ Flutter

Platform Used:

❏ Windows 10
❏ Ubuntu 18.04 (Bionic)

Tools Used:

❏ Visual Studio code
❏ Git as VCS(Version Control System)

Software Requirements Specification

Users of the system are any android end-users.

End-User :

R.1 Manage Files

R.1.1 Delete File

Input: Prompt to delete file

Description: Users are able to delete files in the file system.

R.1.2 View File Details

Input: Prompt to show details

Output: File properties display

Description: File Details like size, type and permissions are shown.

R.1.3 Share File

Input: Prompt to Share file

Output: Applications from which to share

Description: Share file to different applications from the file manager.

R.1.4 Open File

Input: Open command

Output: Application from which file can be opened

R.1.5 Show Hidden Files

Description: Users can see all the hidden files in the system.

R.2 Manage Folder

R.2.1 Create Folder

Input: Prompt to create folder

Output: Created folder

R.2.2 Delete Folder

Input: Prompt to delete folder

Description: Users can delete the targeted folder.

R.2.3 Read Folder

Description: Users can open a folder from internal and external storage and able to see all the files and folders inside it.

R.3 See Category wise Files

Description: Users can easily find files by clicking on Categories-images, audios, videos, and documents.

R.4 Find Duplicate Files

Description: Users can find out a list of all duplicate files available on the system and users can also able to delete duplicate files.

Non-Functional Requirements:

Performance

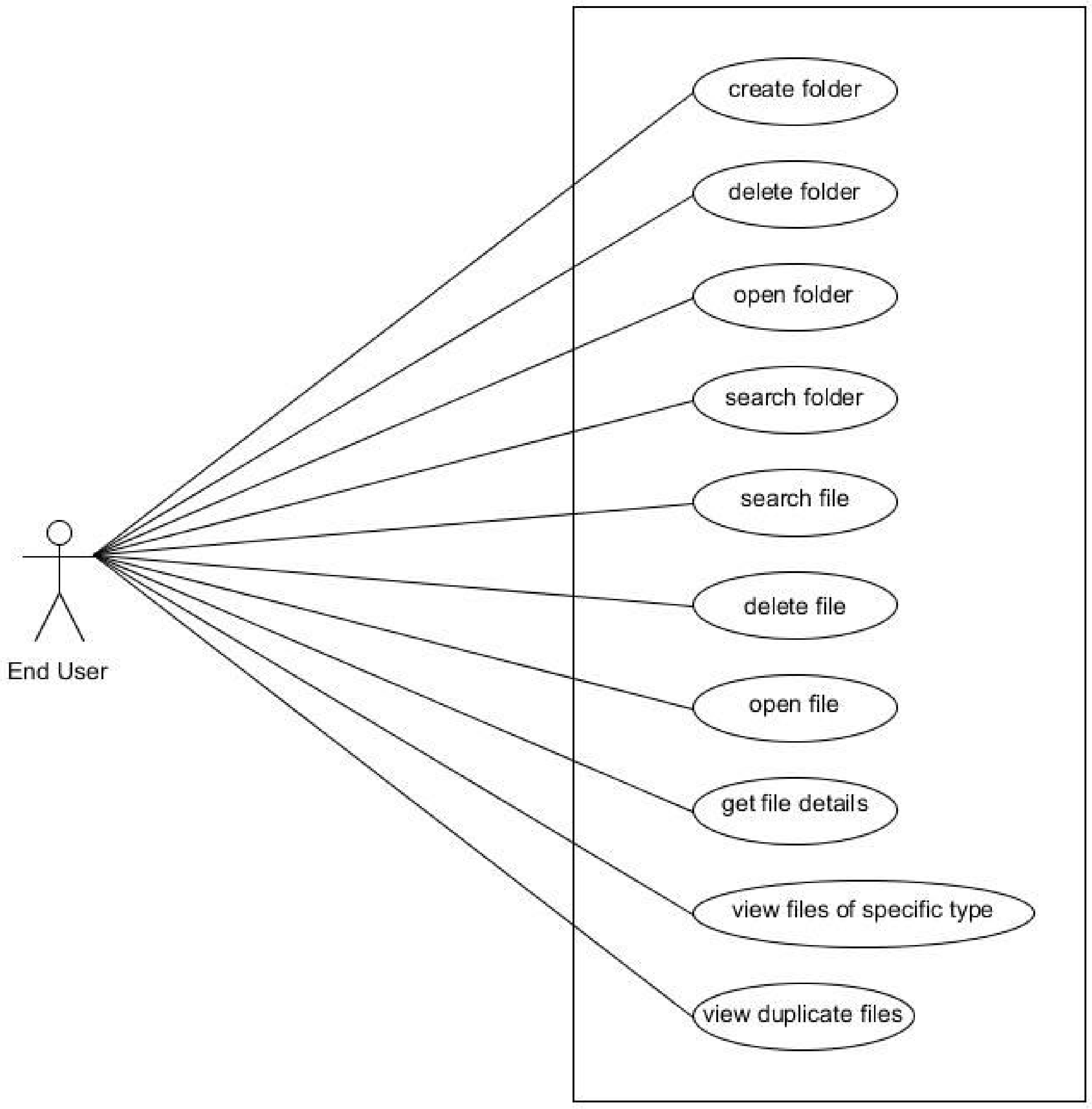
1)It responds to each and every query in less than 1 second

2)It operates with zero downtime.

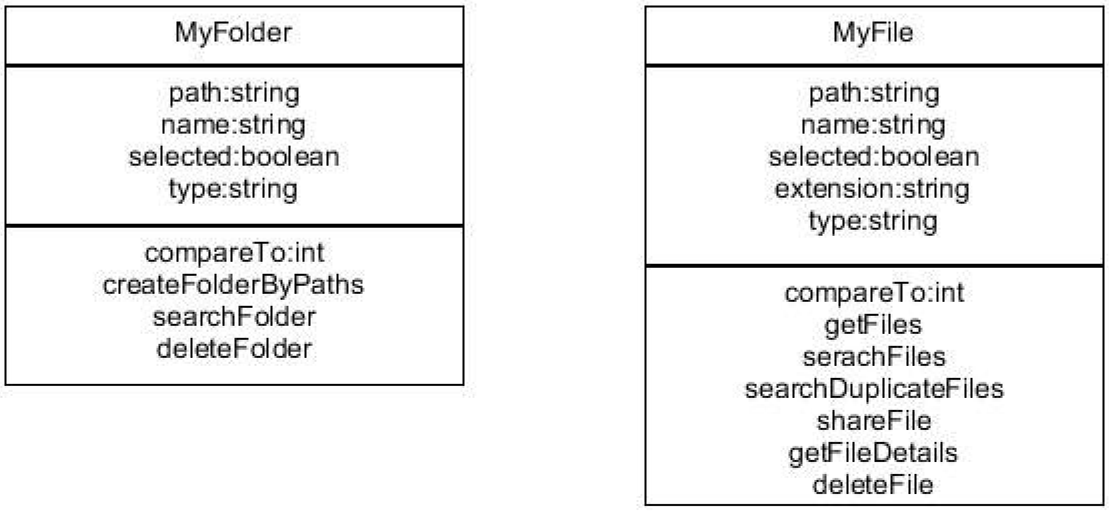
3)Efficient Duplication and search results.

Design

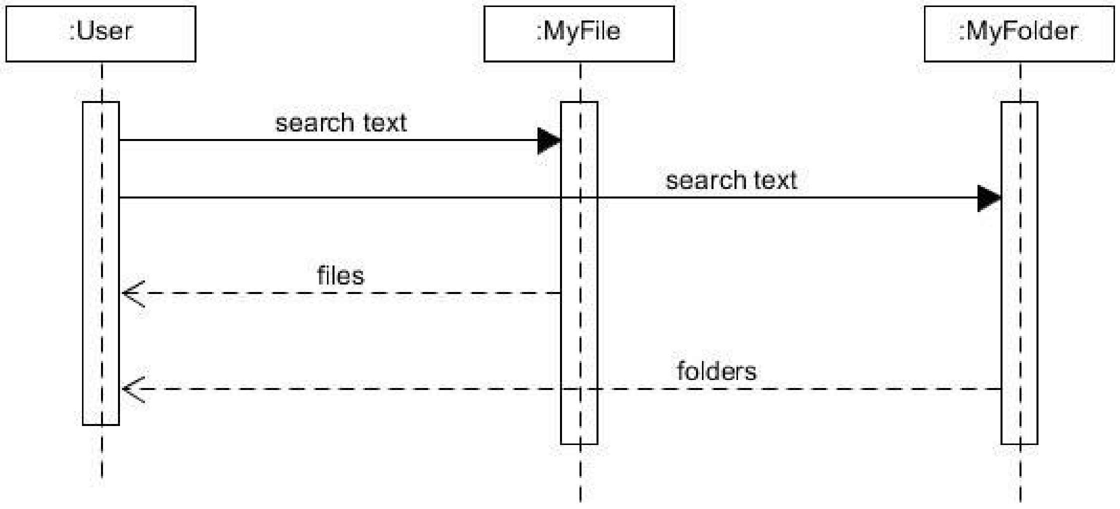
I) Use case Diagram :



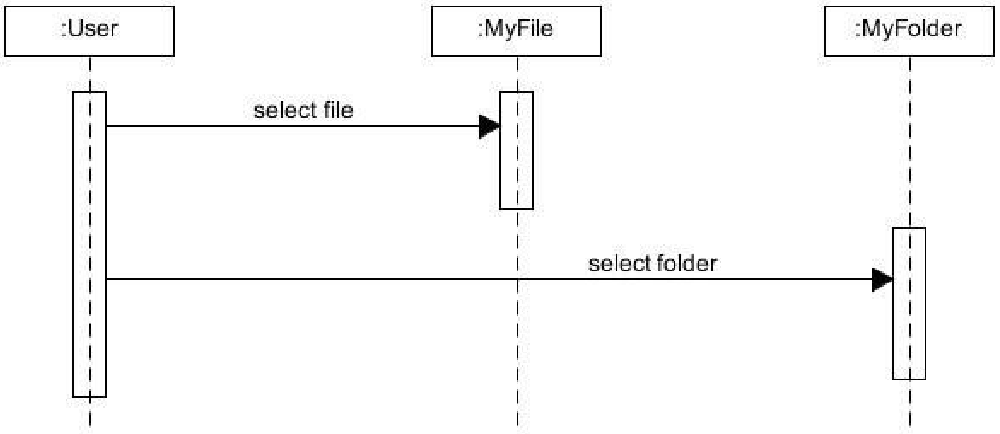
II) Class Diagram :



<u>III) Sequence Diagram :</u>



1) Search file/folder from system



2) Remove file/folder

Implementation Details

- Files/Folder Module
  - In flutter, everything is a widget so for file and folder displays used and FileTile view.
  - A *stateful* widget is dynamic: for example, it can change its appearance in response to events triggered by user interactions or when it receives data.
  - All modules are developed by extending the *stateful* widget.

- Search Module
  - Searching is done using getting all files and matching with the applied keyword.
  - It also counts all files in the system.
  - Searching takes 2 or 3 seconds.

- Category wise File module
  - Segregation of files done by the following steps:
    - Check whether it's a file or not.
    - If file then check if it matches the MIME(Multipurpose Internet Mail Extensions) of respected type or not.
    - Ex. for Image files - image/bmp, image/cis-cod, image/jpeg, image/tiff, image/gif, image/ief, image/png
    - If it matches mime type then add it to the list.
    - Then display using ListTile View and Card.

- Duplication Module:
  - To find duplicate files in the file system, first, find the checksum of file hash using the MD5 Algorithm.

  - MD5 has one important benefit above SHA1, SHA2, and upcoming SHA3: many platforms allow faster implementation of MD5 than other hashes. The performance in this use case varies highly according to the used hash function.

  - Calculating the hash will make the program run slow. It’s better to also **check the file size**.

  - Steps to find duplicate file:
    - Check if **file size** is equal
    - If step 1 passes, check the **file type**,
    - If step 2 passes, check the **hash at last.**
    - If step 3 passes, then add both files to the list.

  - The collision rate for this approach is 2^-89 in million files, which is very small.

Testing

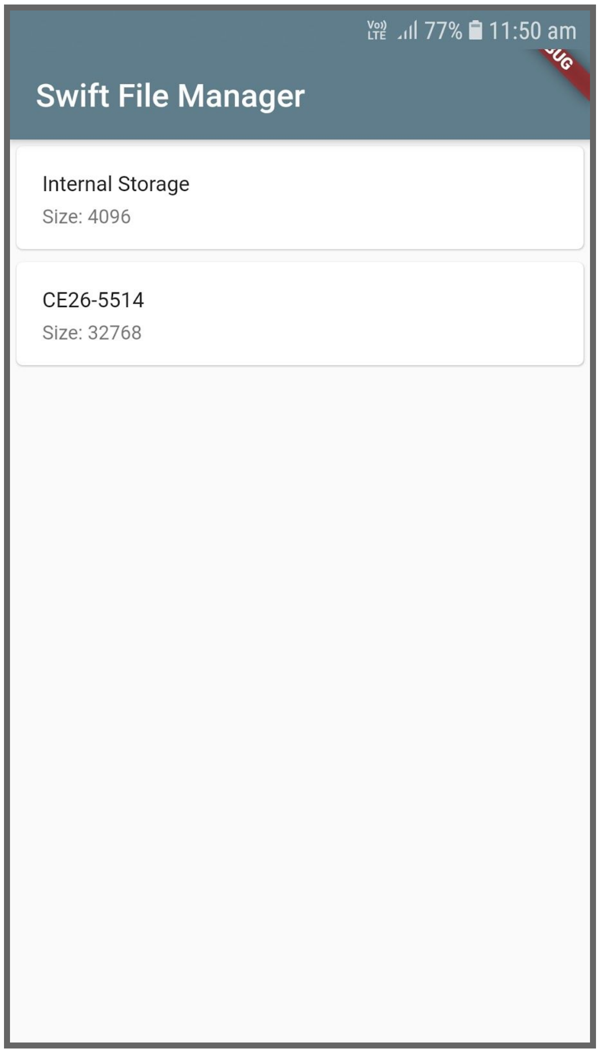
- Entire flutter app is built using widget classes.
- Actually, Entire flutter application is one widget.
- Tests help ensure that your app performs correctly before you publish it, while retaining feature and bug fix velocity.
- a well-tested app has many widget tests, tracked by code coverage.
- **test coverage** is a measure used to describe the degree to which the source code of a program is executed when a particular test suite runs. A program with high test coverage, measured as a percentage, has had more of its source code executed during testing, which suggests it has a lower chance of containing undetected software bugs compared to a program with low test coverage.
- So to test UI framework in flutter application, we have performed widget testing.
- Execution speed of UI testing is quick as per flutter dev documentation.It requires more dependencies.
- Other modules of logic are tested using manual testing.
- Performed test based on given test scenarios in table.

Results of tests(Manual)

No.	Test Scenarios	Expected result	Actual Result	Result
1	Getting Media category wise on user system	Yes	Yes	Pass
2	Search files from system	User should get results in less than 10 seconds after query	User is getting results in 3 seconds after query	Pass
3	Creation of folder in current directory	Yes	Yes	Pass
4	Sharing file	All supported media	All supported media in system can be used to share file	Pass
5	Support external file systems(like OTG, memory card,etc.)	Yes	Yes	Pass

Screenshots

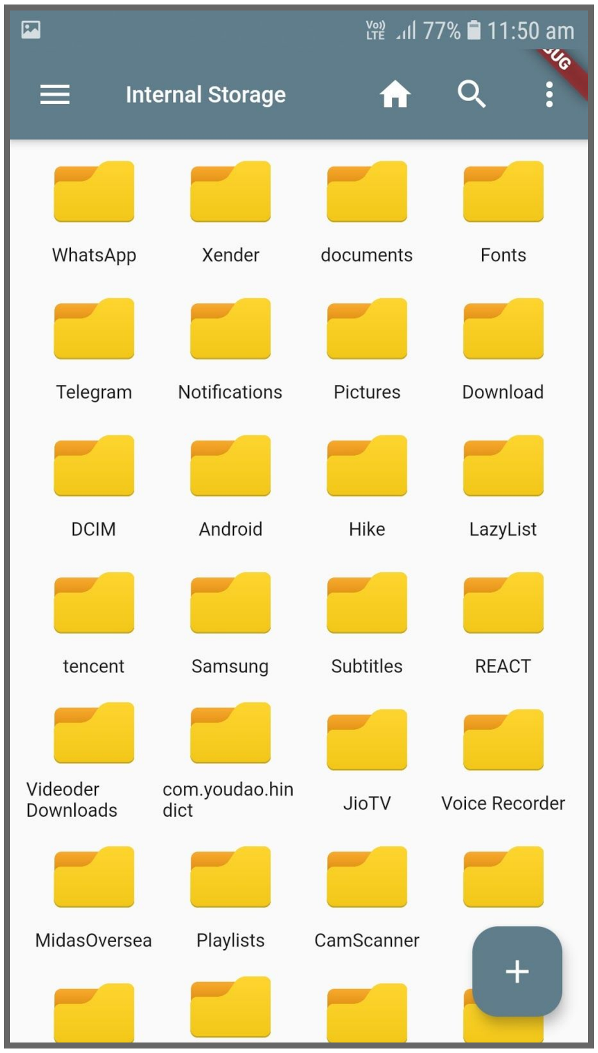
Major Functions Screenshots:



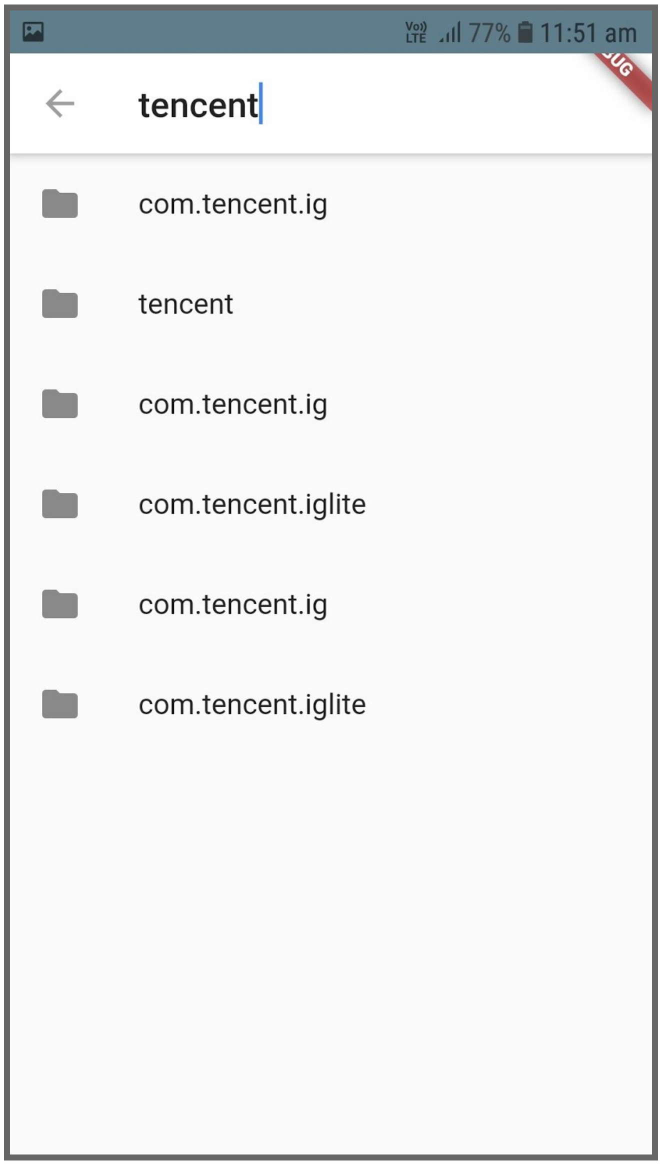
Swift File Manager

Internal Storage
Size: 4096

CE26-5514
Size: 32768



I) Home Screen   II) Internal Storage Screen


tencent

com.tencent.ig

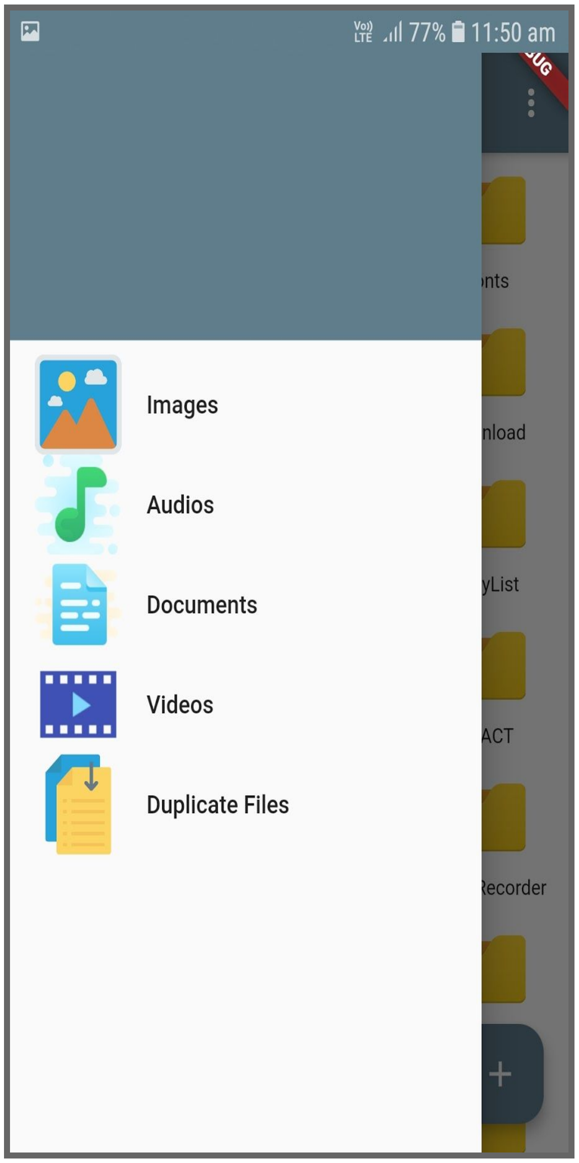
tencent

com.tencent.ig

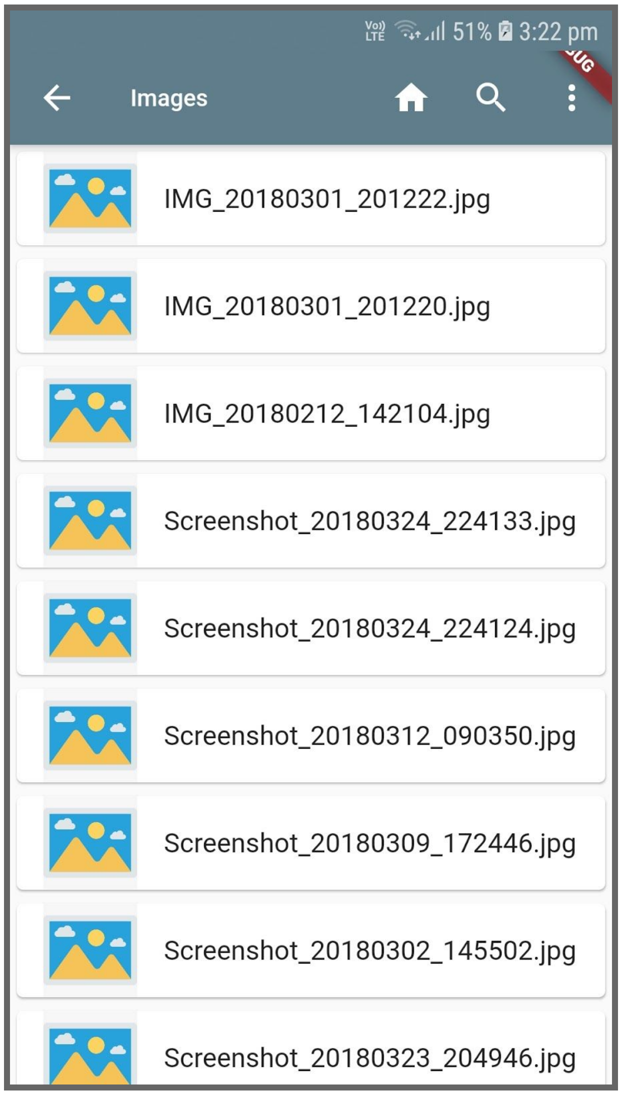
com.tencent.iglite

com.tencent.ig

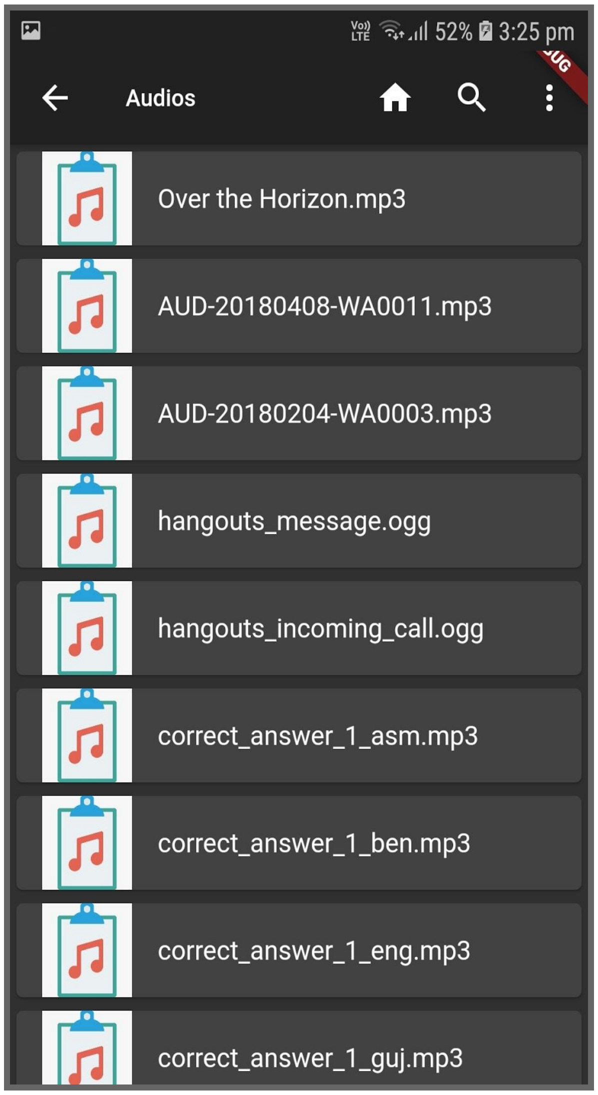
com.tencent.iglite



III) Search Screen   IV) Drawer Screen

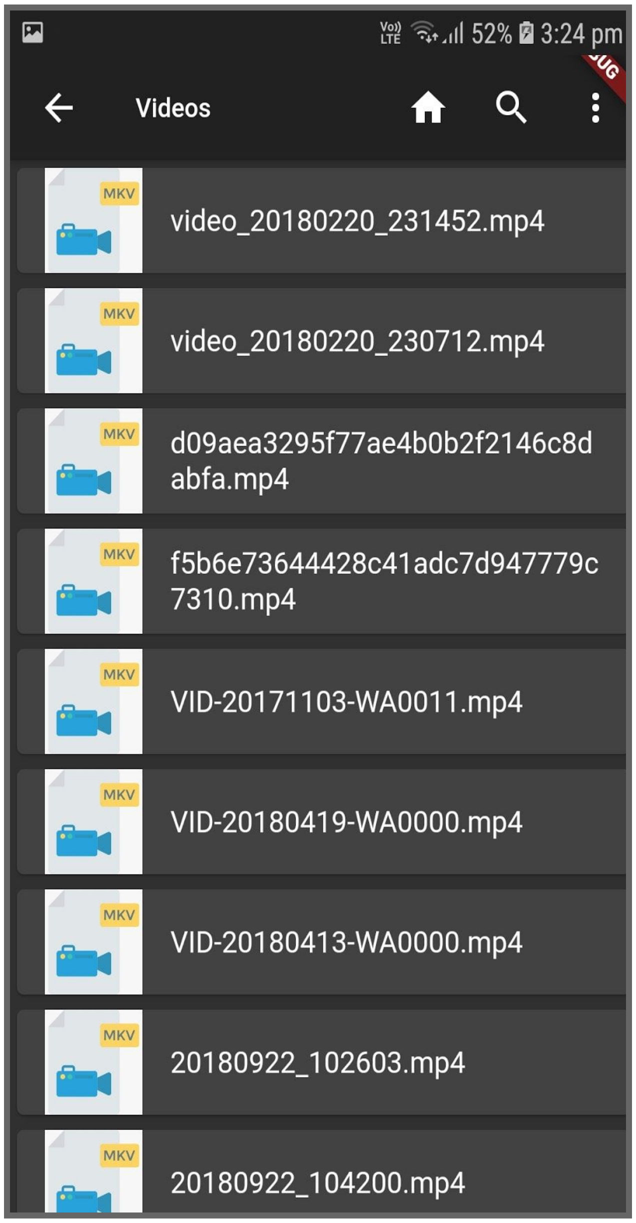


IMG_20180301_201222.jpg
IMG_20180301_201220.jpg
IMG_20180212_142104.jpg
Screenshot_20180324_224133.jpg
Screenshot_20180324_224124.jpg
Screenshot_20180312_090350.jpg
Screenshot_20180309_172446.jpg
Screenshot_20180302_145502.jpg
Screenshot_20180323_204946.jpg

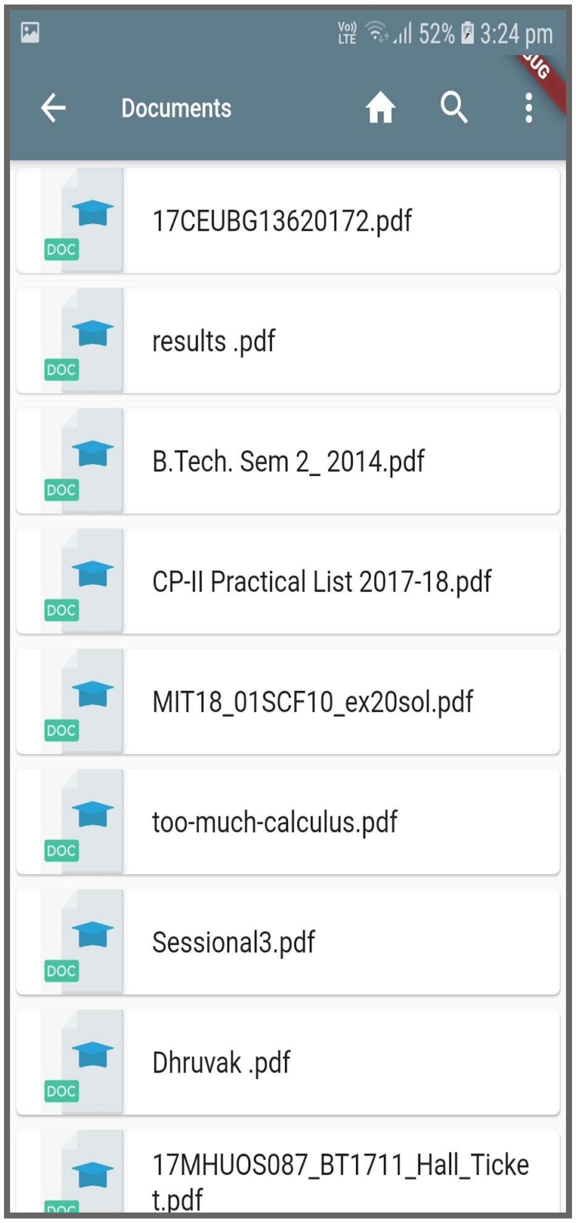


Over the Horizon.mp3
AUD-20180408-WA0011.mp3
AUD-20180204-WA0003.mp3
hangouts_message.ogg
hangouts_incoming_call.ogg
correct_answer_1_asm.mp3
correct_answer_1_ben.mp3
correct_answer_1_eng.mp3
correct_answer_1_guj.mp3

V) Images Screen VI) Audios Screen

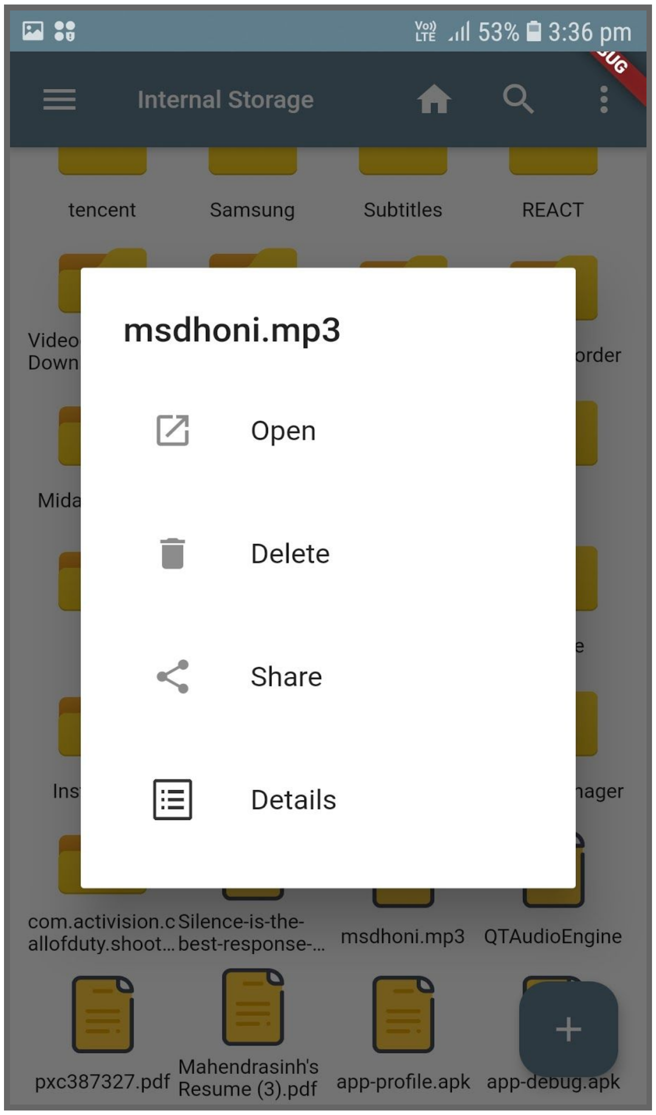


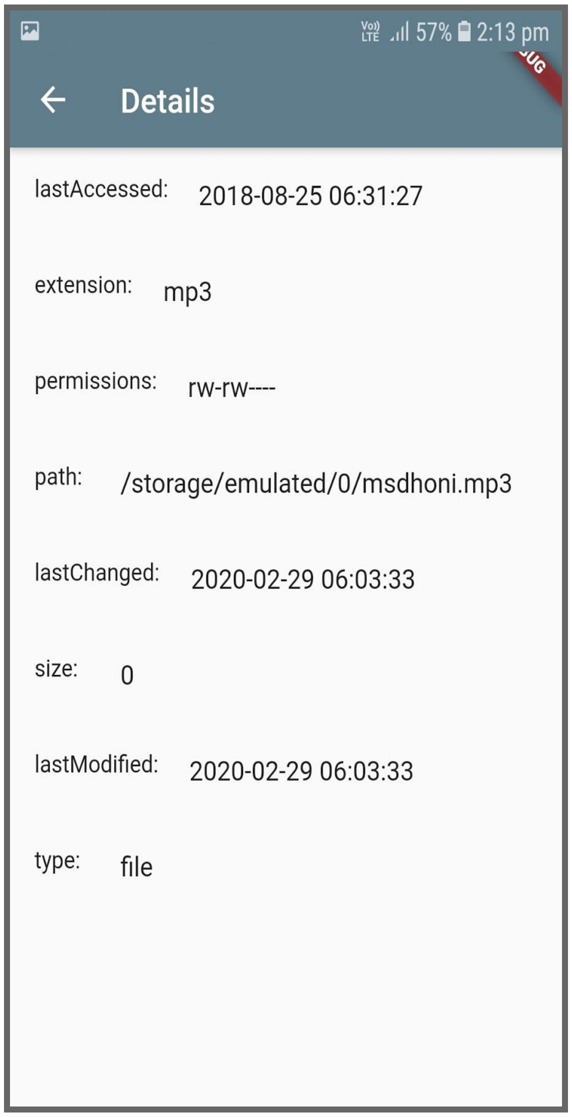
video_20180220_231452.mp4
video_20180220_230712.mp4
d09aea3295f77ae4b0b2f2146c8dabfa.mp4
f5b6e73644428c41adc7d947779c7310.mp4
VID-20171103-WA0011.mp4
VID-20180419-WA0000.mp4
VID-20180413-WA0000.mp4
20180922_102603.mp4
20180922_104200.mp4



17CEUBG13620172.pdf
results .pdf
B.Tech. Sem 2_ 2014.pdf
CP-II Practical List 2017-18.pdf
MIT18_01SCF10_ex20sol.pdf
too-much-calculus.pdf
Sessional3.pdf
Dhruvak .pdf
17MHUOS087_BT1711_Hall_Ticket.pdf

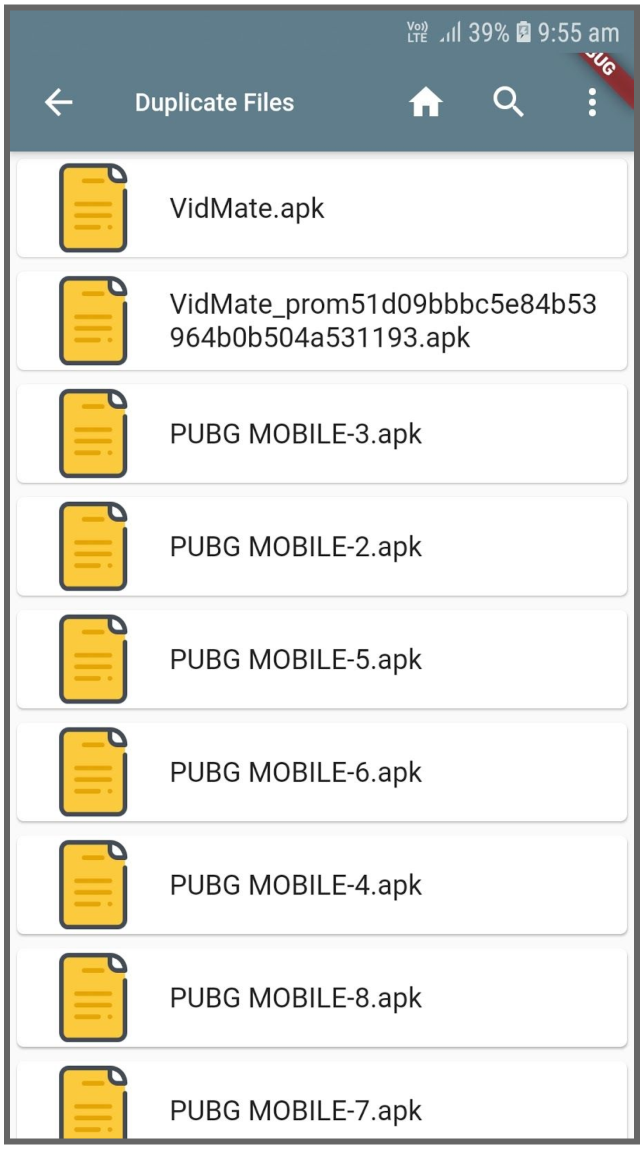
VII)Videos Screen VIII) Documents Screen

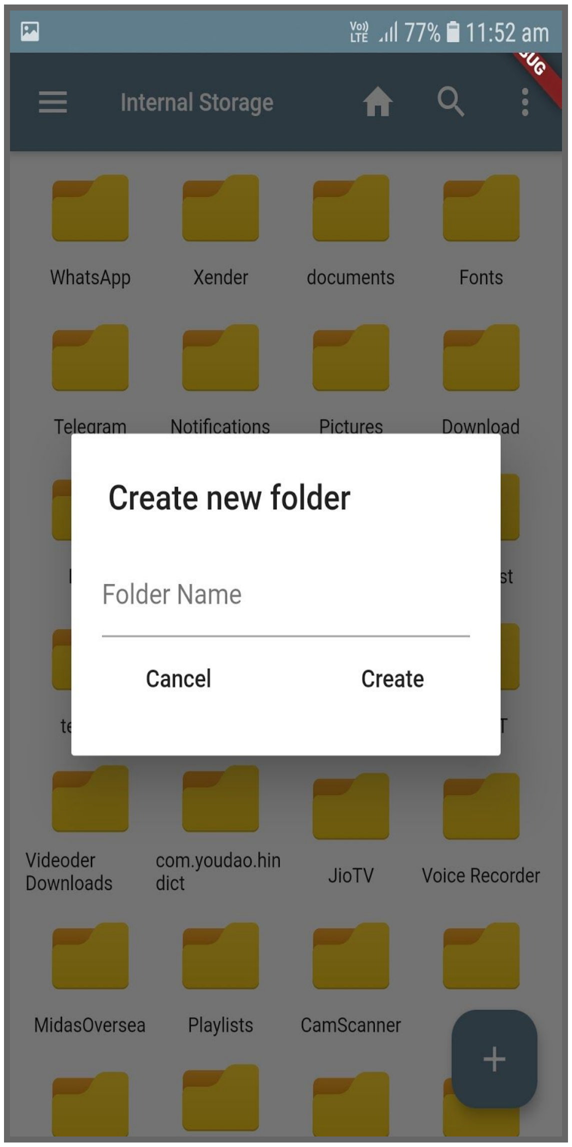




lastAccessed:	2018-08-25 06:31:27
extension:	mp3
permissions:	rw-rw----
path:	/storage/emulated/0/msdhoni.mp3
lastChanged:	2020-02-29 06:03:33
size:	0
lastModified:	2020-02-29 06:03:33
type:	file

IX) File Details Pop-up X) File Details Screen





XI) Duplicate Files Screen XII) Creation of Folder

Conclusion

Users have all the rights to use the basic functionality of the file manager. Users can easily get all the files of specific types and can easily navigate among folders. Users can also find all the duplicate files available in the system and also be able to delete them to make more space on the device.

Limitation and Future Extension

- Android application has lacked in terms of security aspects.

- Limitation
    - Duplication algorithm taking up to 2mins for the system containing more than 10000 files.
    - Move operation not implemented.

- Future Extension
    - Better User Interface
    - Efficient Searching
    - Cache deletion for better optimization
    - Login with third party

BIBLIOGRAPHY

For the successful implementation of this project, we referred to many websites and video tutorials. We developed an android application on visual studio code. We use Github as a project management tool. Mostly we referred to online material for the syntax of Dart.

Reference Websites:

- ❏ https://dart.dev/
- ❏ https://flutter.dev/
- ❏ https://pub.dev/
- ❏ https://stackoverflow.com/
- ❏ https://medium.com/