



CMPE 277: Smartphone Application Development

Project Report

Spartan Drive

Submitted To:

Prof. Charles Zhang

Date of Submission

13th Dec 2015

Submitted By

Team 3

Team Members	SJSU ID
Ankit Devani	009987861
Aparna Kansal	009636497
Fareen Magdum	009979125
Mario Vinay	009991904
Rashmi Hadli	010046725
Roshan Srinivas	009874904

Contents

1. INTRODUCTION	3
The Objective	3
The Idea.....	3
2. THE MOTIVATION.....	3
3. DESIGN.....	4
3.1 High Level Design	4
3.2 Component Level Design.....	5
4. TECHNOLOGY CHOICES.....	5
5. FEATURES.....	5
6. TESTING PLAN.....	19
7. LESSONS LEARNT	20
PROJECT CONCEPTS.....	20
8. FUTURE WORK.....	20
9. CONCLUSIONS	21

TABLE OF FIGURES

Sr No	Title	Page no
1	High-Level Design	5
2	Component Design-1	6

1. INTRODUCTION

The term project for Smartphone Application development expected us to apply to the concepts of Mobile application development in either Android or iOS and develop an application that will satisfy a purpose thus conforming to the term projects requirements.

The Objective

The objective of this project is to build a mobile application that interacts with a server in the cloud at the same time implementing and exercising principles and methodologies of mobile application development. The aim is to build a project that will satisfy the needed requirements to suffice all the needed functionalities.

The Idea

These days mobile is gaining a lot of importance as a platform for creating useful applications. Recently, during the Black Friday sale, it was observed that most of the online shopping was done through mobile applications. Such is the impact of mobile world. With the growing use of applications, sharing items is a common phenomenon. With this thought as an inspiration we have chosen the topic “Spartan Drive”. The concept of sharing allows users to share any content they wish with anyone from anywhere. This helps to bridge gaps and bring people closer. Our scope for the project is to provide a platform wherein a user can share upload files/folders with other users through their university email id. We aim to build a mobile application which will allow users to create, view, upload and share files/folders. We have included a number of features which include Uploading, Searching, Deleting, sharing files/folder amongst users using this application. We have tried to develop our app on the lines of Google Drive

2. THE MOTIVATION

We wanted to build a mobile application that will enable us to share files/folders. We carefully studied all the features of Google Drive and tested them. We did a lot of research regarding the current implementation methods used in Google Drive. After careful observation we jotted down a few features which enhance the basic functionality of our application. After which we decided to develop a mobile application for sharing. Using this functionality as the base we went and added a lot many more complex features to enhance the user’s experience. During the course of implementation, we came across a number of interesting facts about Android development and we happened to learn a lot during this project.

3. DESIGN

3.1 High Level Design

We have designed the application using vast technology stack which uses various frameworks for development and integration.

The core services layer is implemented as a library project and the UI will be treated as the main project. Using this kind of structure, we can distinguish between these two layers and it will help us to distribute the work properly.

Various consumer objects are available on the UI Layer, they are responsible for handling the interface. These consumer objects are activities, fragments, views and adapters. On the UI layer we get everything in charge of managing the interface: activities, fragments, views, adapters, etc. All of them are data consumer objects.

The main part is the core services layer and the other helping parts like the beans, clients etc. are used mainly to manage the data.

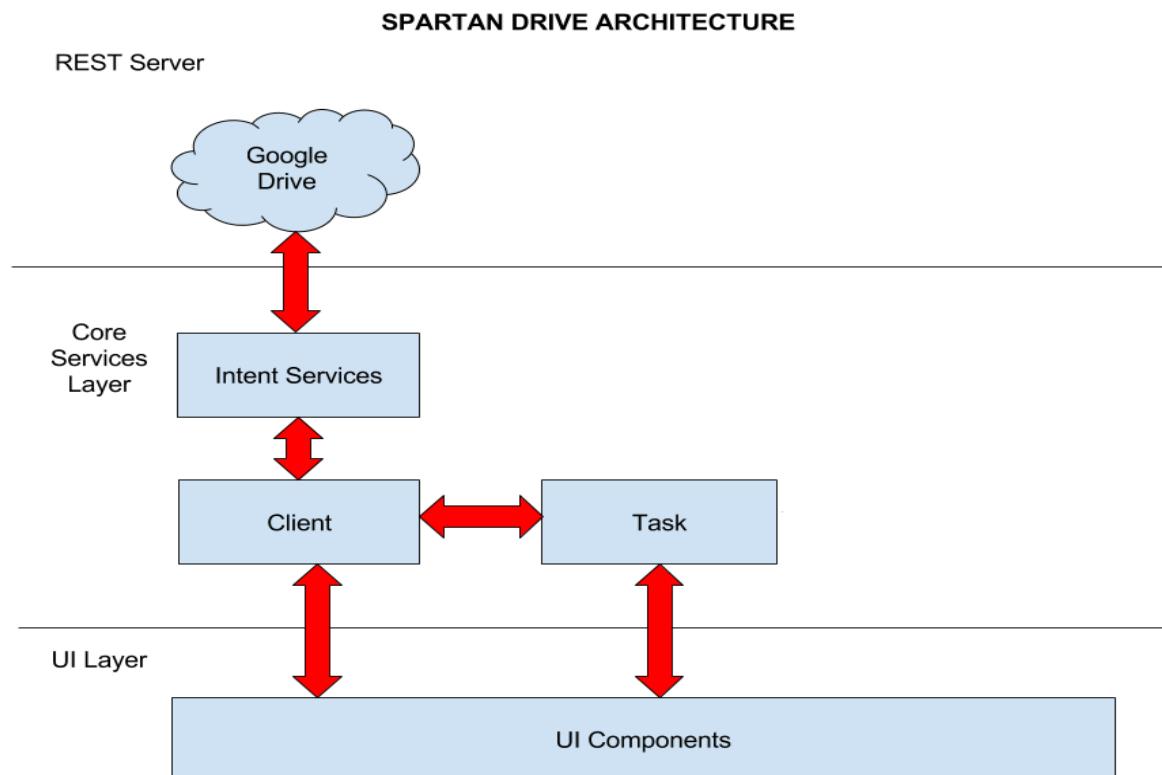


Fig 3.1 High Level Design

3.2 Component Level Design

To implement the features described below we have developed a component level design as below with all the required variables and classes.

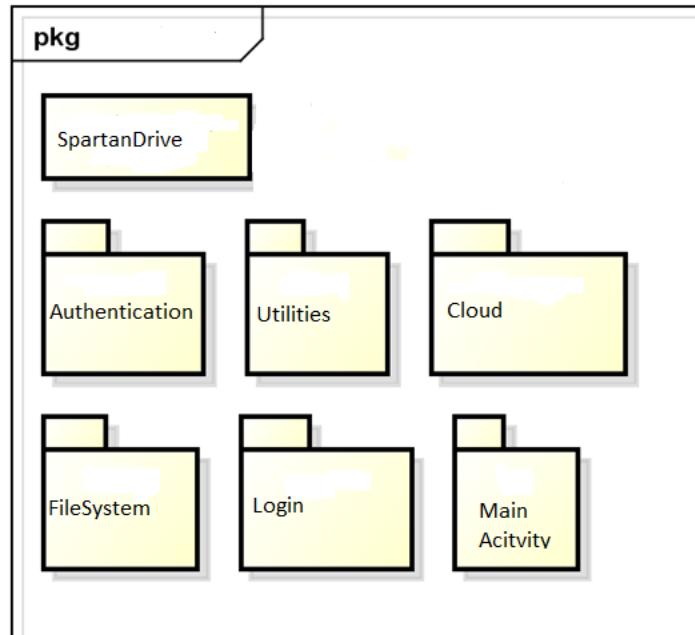


Fig 3.2 Component Design -1

4. TECHNOLOGY CHOICES

The application is divided into two layers Frontend and Backend or as we call them -

- **UI Layer** - Android
- **Core Service Layer** - Java
- **Google API**: For User Login
- **Google Drive** - For storing files

5. FEATURES

We implemented the following features in our application

i. **Login:**

Google Sign-in is integrated with the Spartan Drive App, which gives us a flexibility to use google drive API as google drive is used as a cloud storage.

ii. File Upload:

With this feature user will be able to create a new folder in the root folder as well as, in any folder.

iii. Folder Management:

When the user clicks on the create icon, create a folder where you can mention the title of the folder. The user can even rename the folder and delete any specific folder.

iv. File Information

This feature shows the title of the folder, description like the creation date/time

Different icons are used for each mimetype.

Supported icons are Doc, xls, pdf, ppt, png, txt, jpg .

v. Browse Files

Files are listed in each level. The user will be able to navigate into any level of directory one by one,

Features Implemented after the demo:

i. Browse Files

Some features of the browse file like Can view all supported type. You can email the specific file to anyone. This feature attaches the file to Gmail along with the subject. You can choose to send to anyone.

ii. File Sharing

On clicking the info icon, the user gets a new screen that takes to a new screen. on clicking the “Add people”, a new pop-up window appears to enter the email id. Once you click the “ok” button, the file/folder is shared. This feature works for a single file and a whole folder.

iii. Search:

This feature allows you to search all the files whether owned or shared by name or description

iv. Usage Report

This feature shows the status of the drive usage.

v. Push Notification:

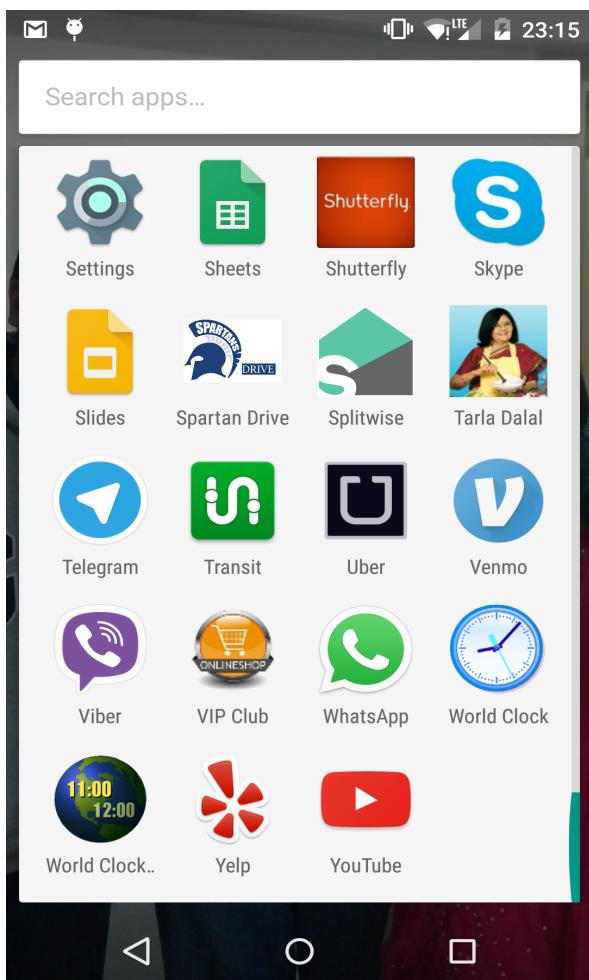
Whenever you share a folder or a file or unshared, a notification message is sent to the recipient's device. This is implemented using parse.com . We used a channel method, where the devices are subscribed to a channel. When an event occurs a notification is sent to the device, if subscribed.

vi. Sign-Out:

This feature signs out the app.

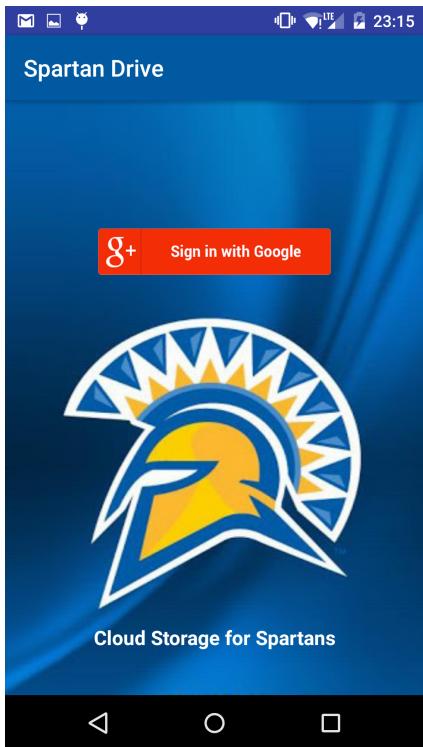
Screenshots:

App icon:

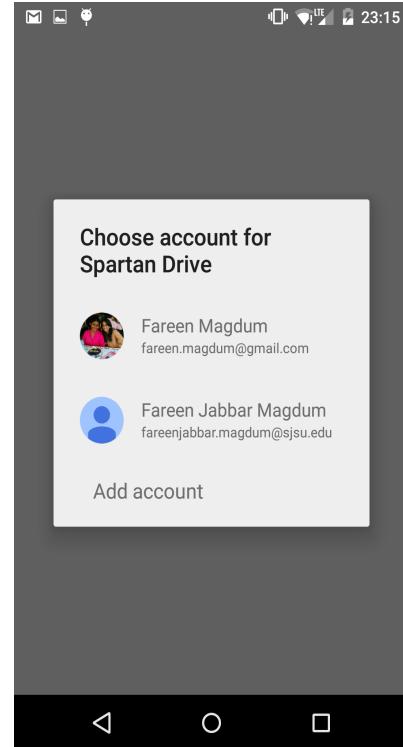


SpartanDrive

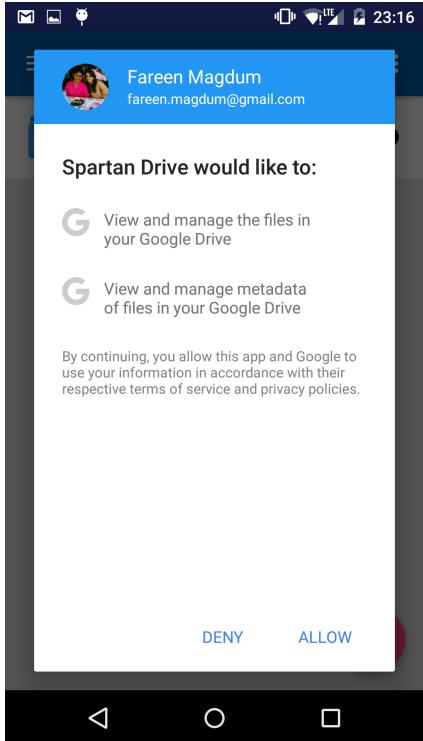
Login Screen



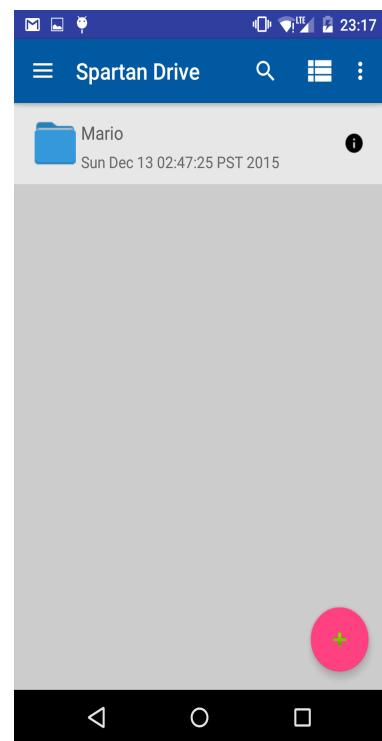
Account Picker:



Permission:

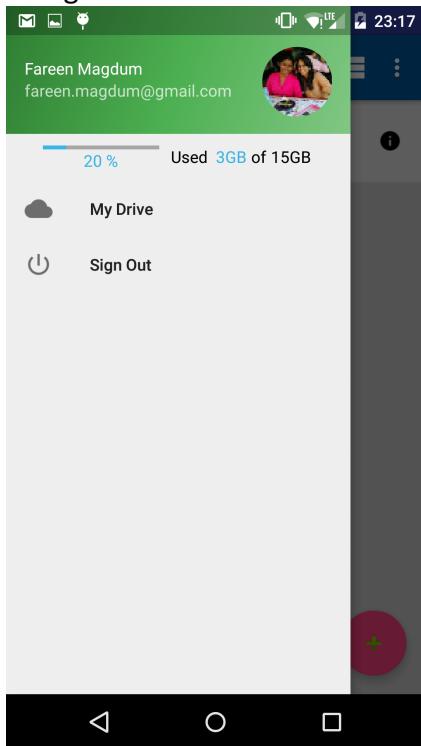


Home Screen

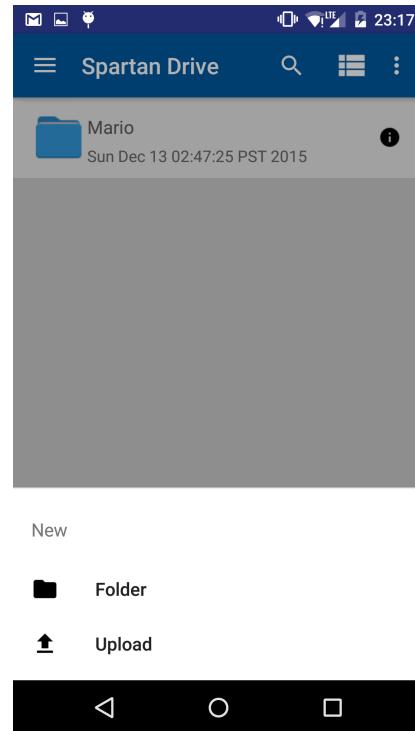


SpartanDrive

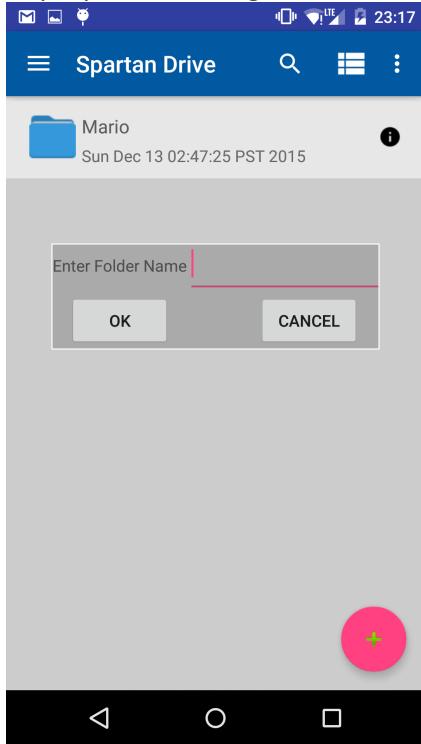
Navigator drawer:



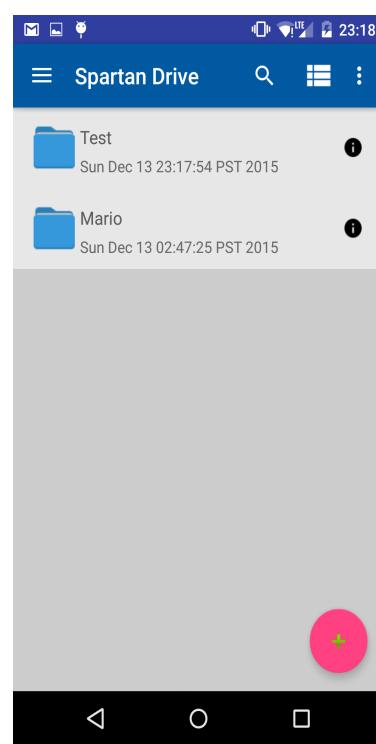
SnackBar for creating folder and upload files:



Pop-up for entering folder name:

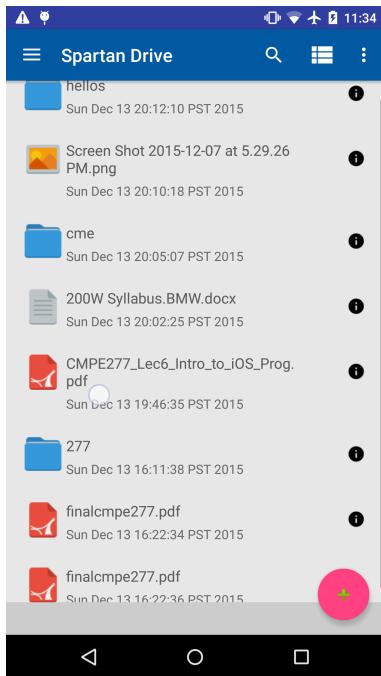


New Folder Created:

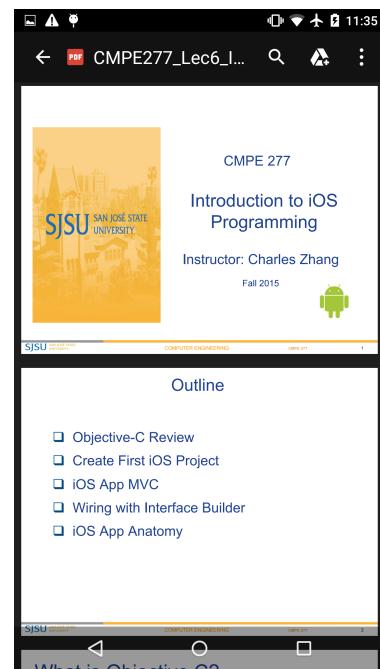


SpartanDrive

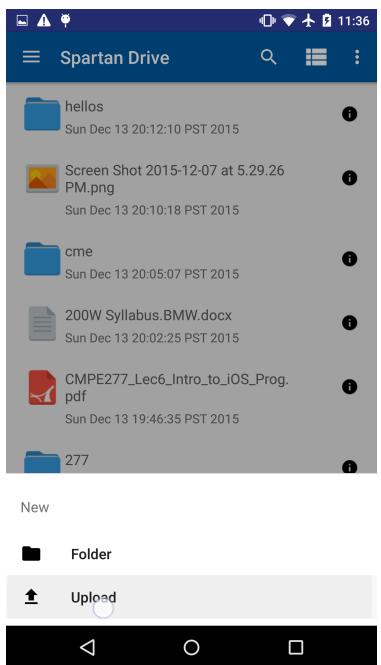
Click on a file to view it:



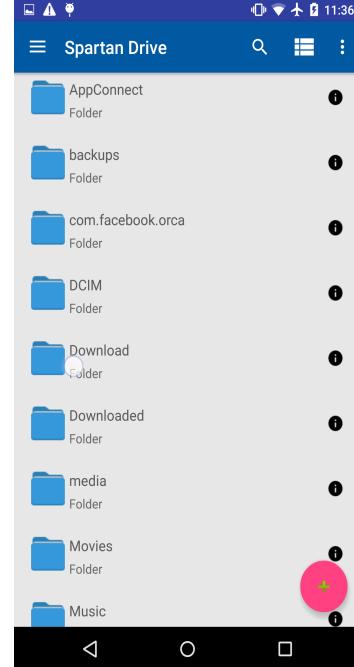
View a file:



Upload a file:

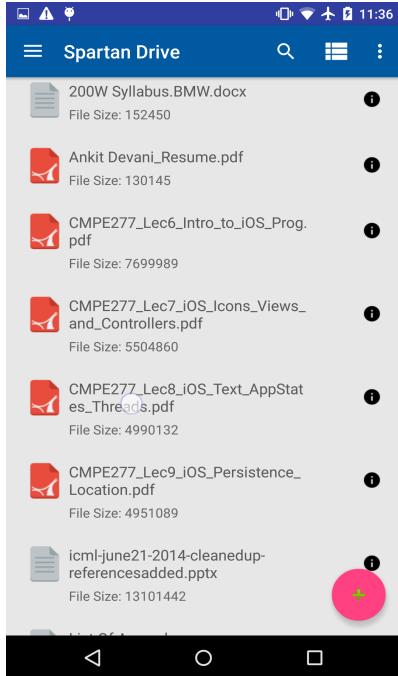


Disk contents:

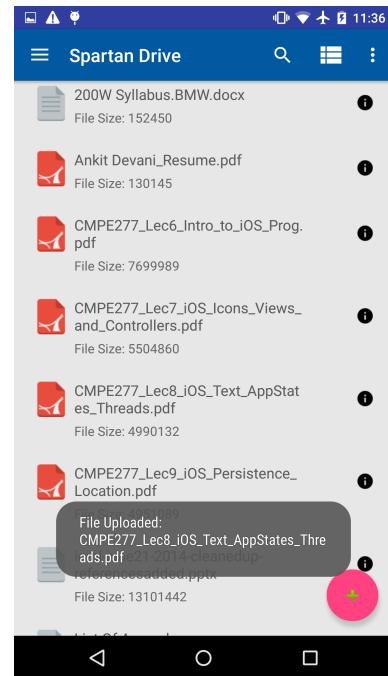


SpartanDrive

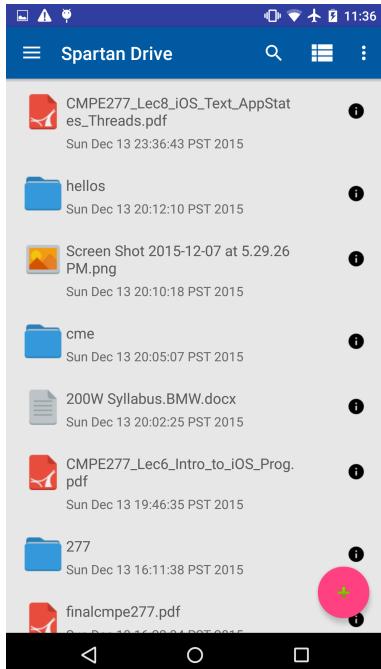
Choose a file to upload:



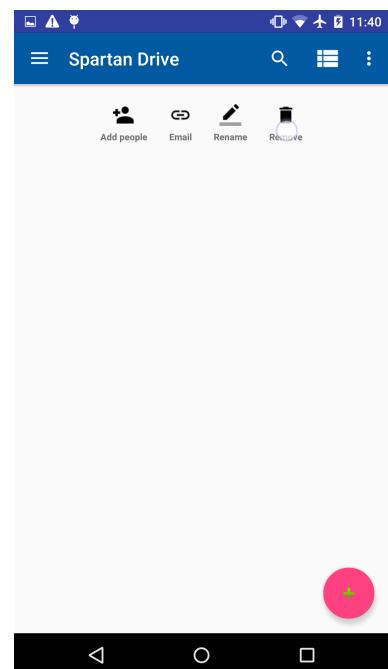
File uploaded:



Uploaded File Visible on Drive:

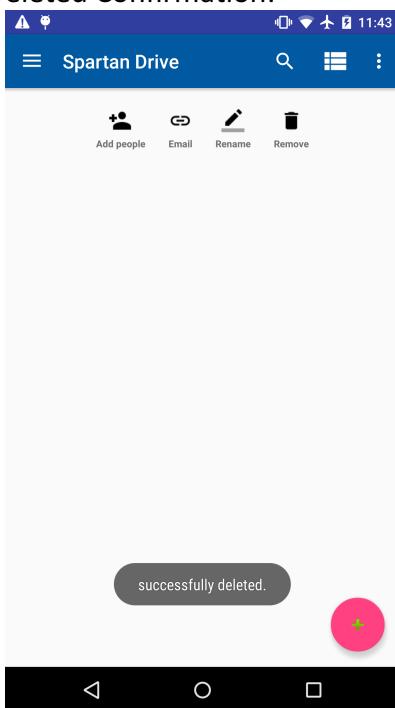


Delete a file:

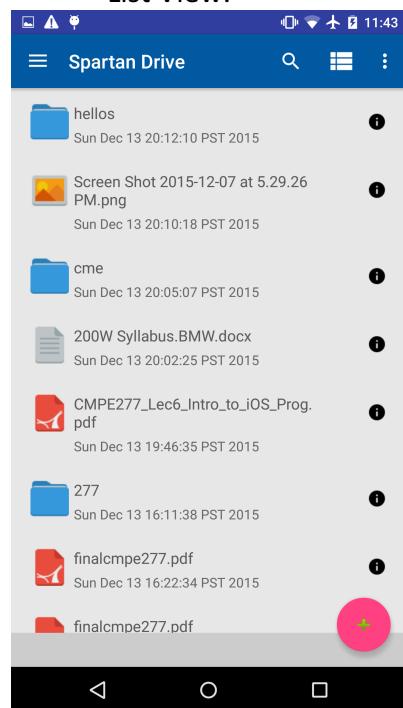


SpartanDrive

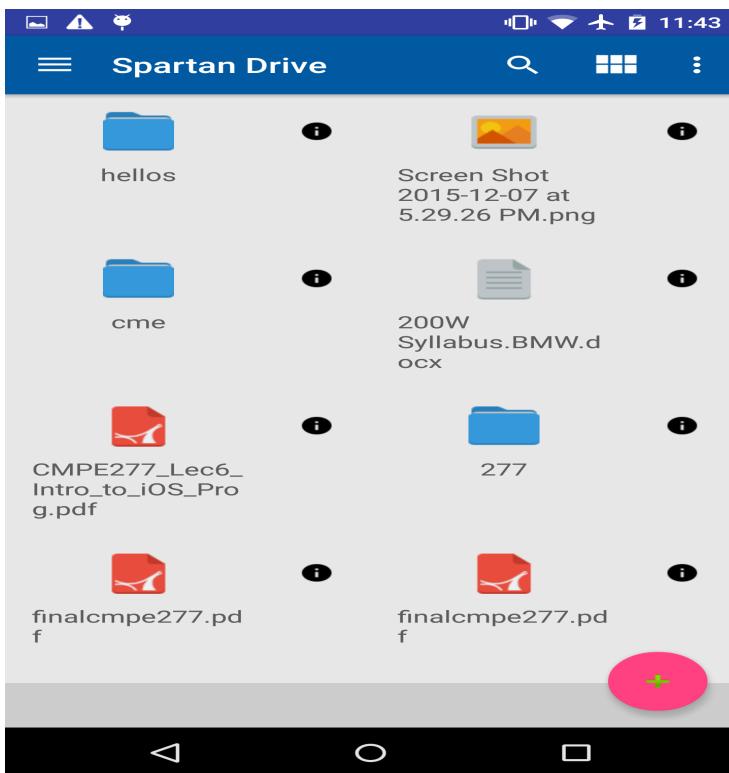
File Deleted Confirmation:



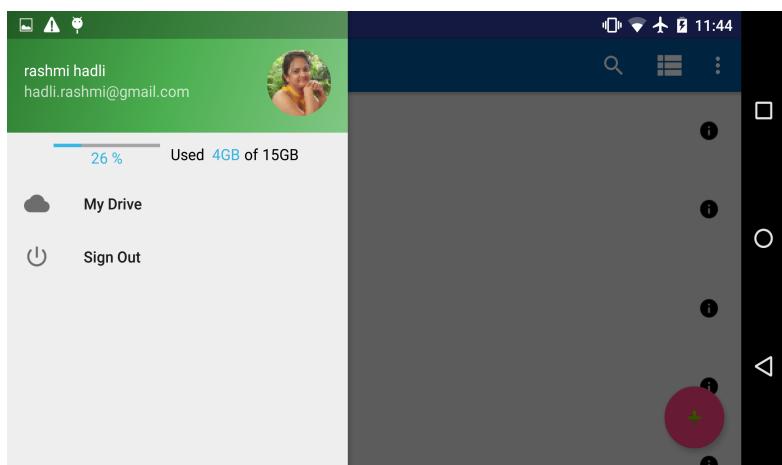
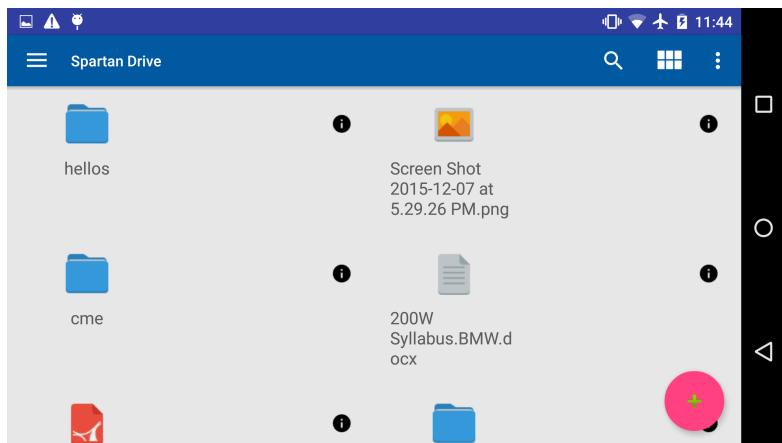
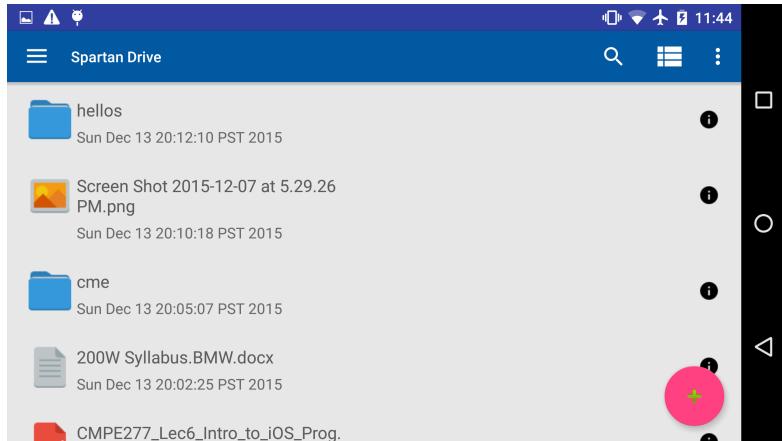
List View:



Grid view:

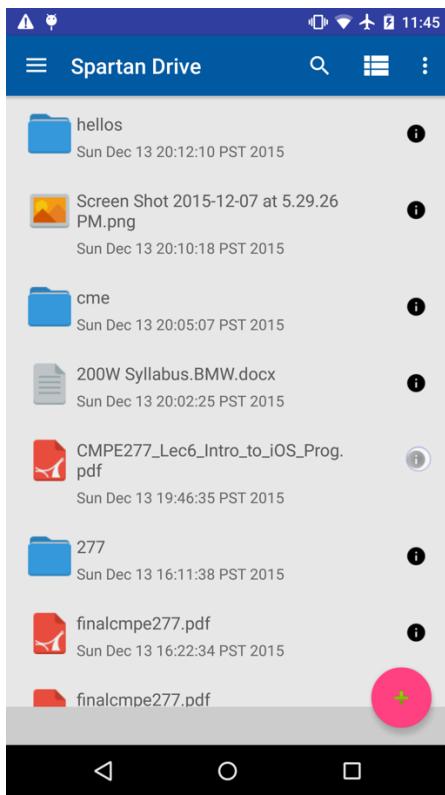


Landscape Mode:

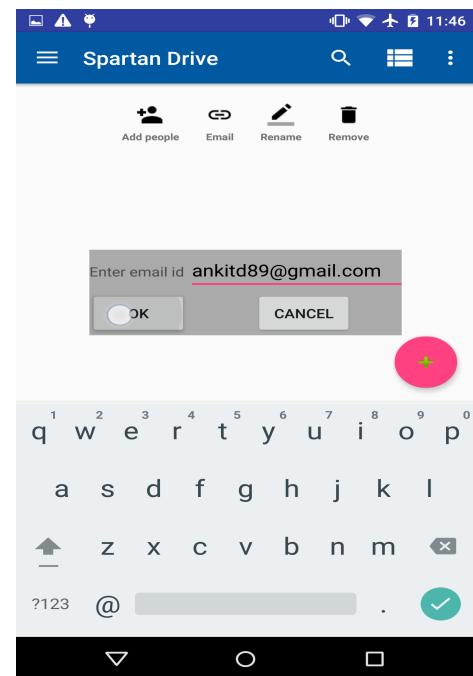
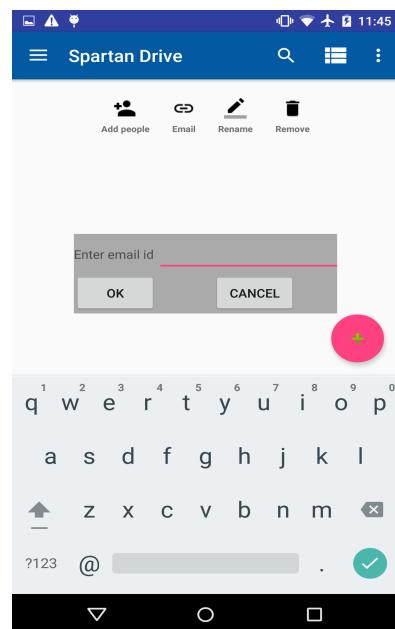
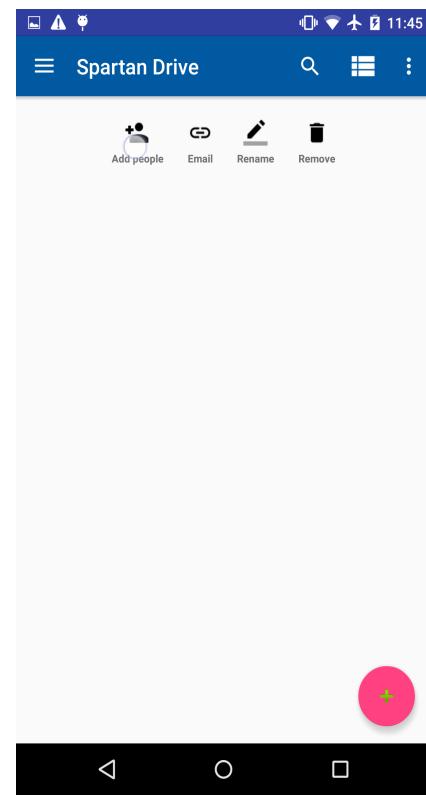


SpartanDrive

Share file and folder:

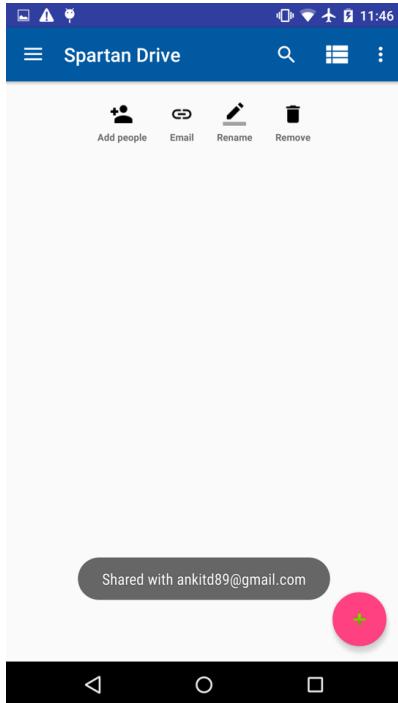


Click on Add people:

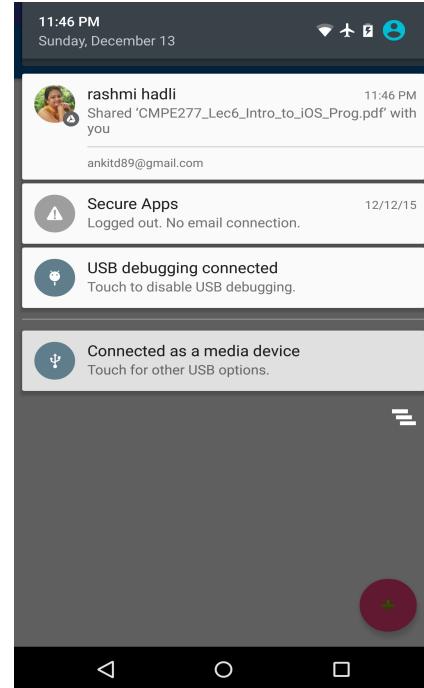


SpartanDrive

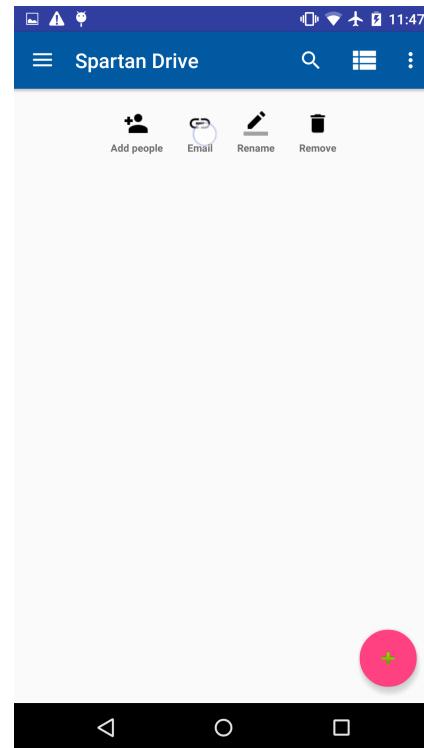
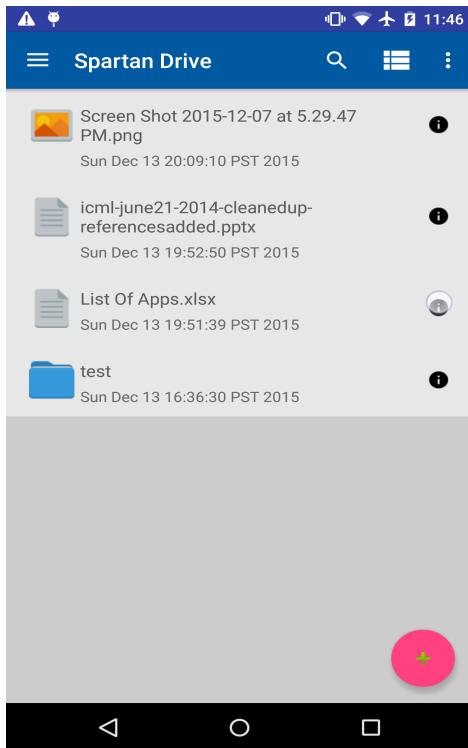
Sharing Confirmation:



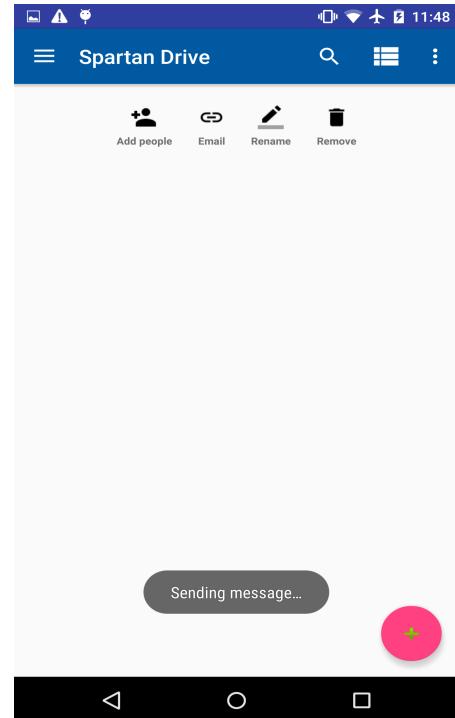
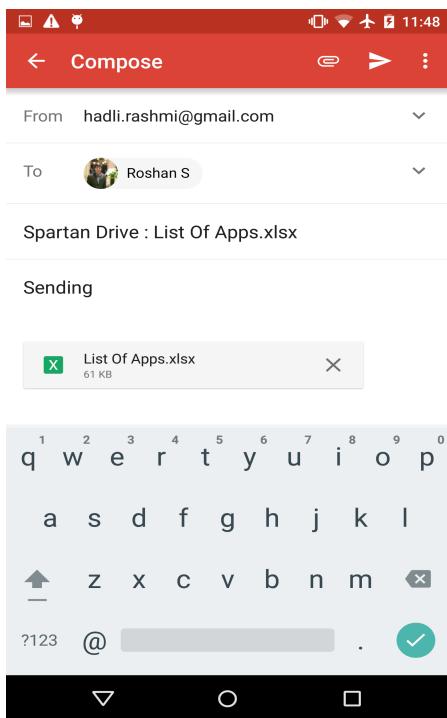
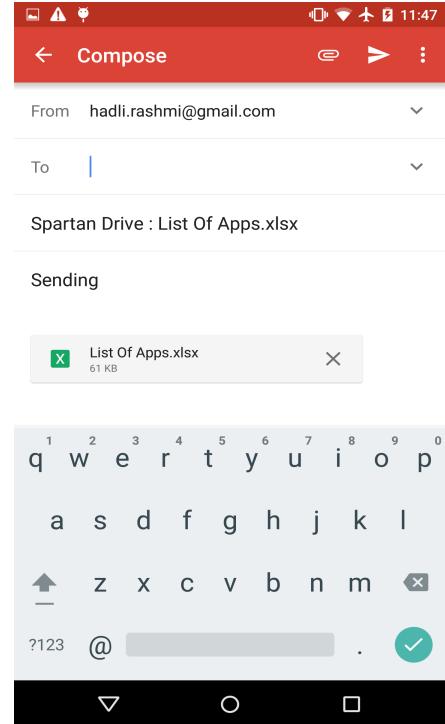
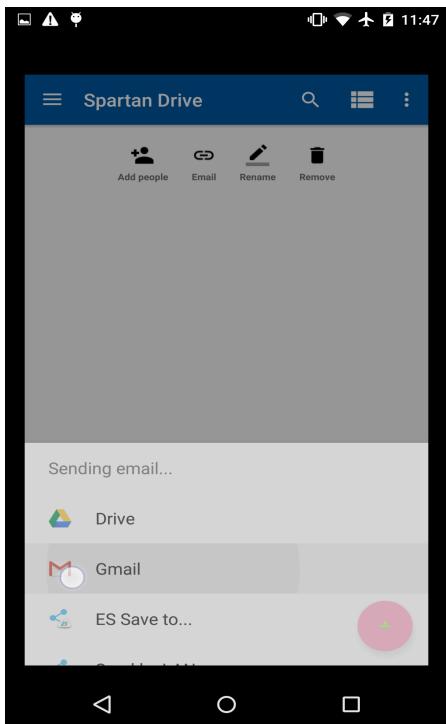
Notification:



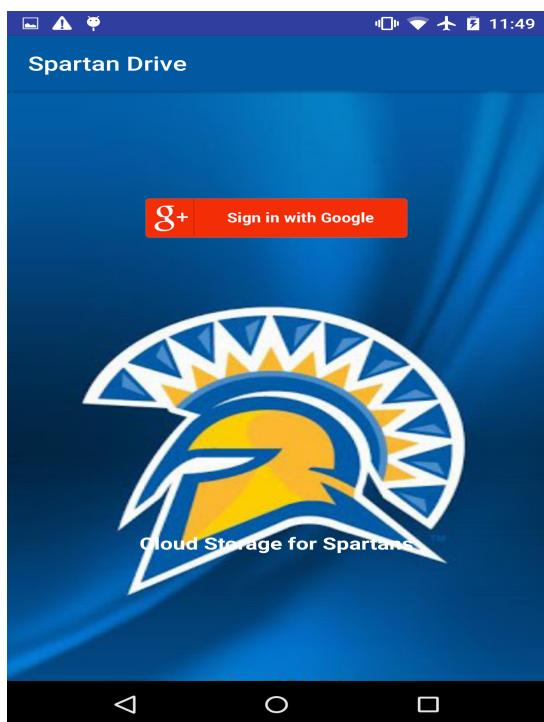
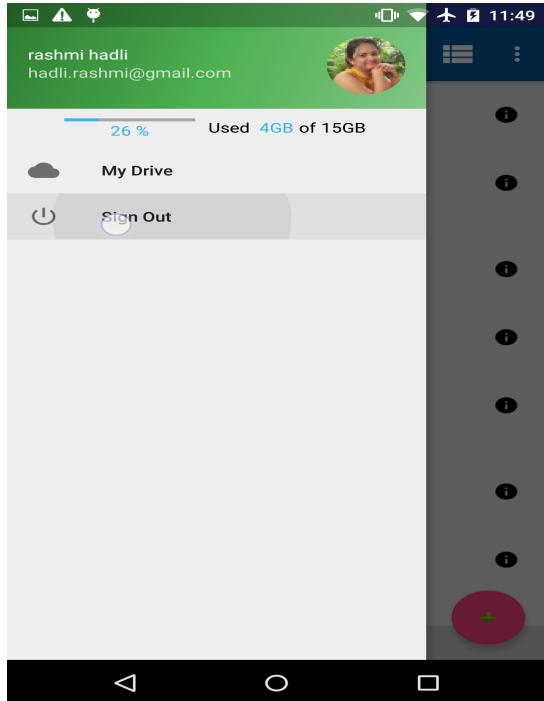
Email a file:



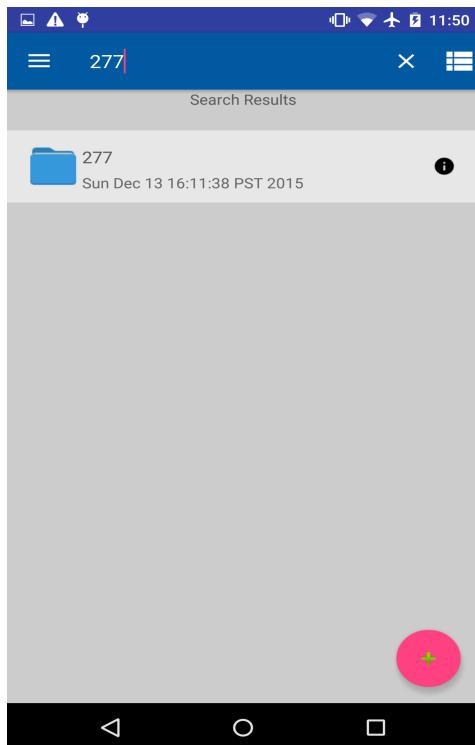
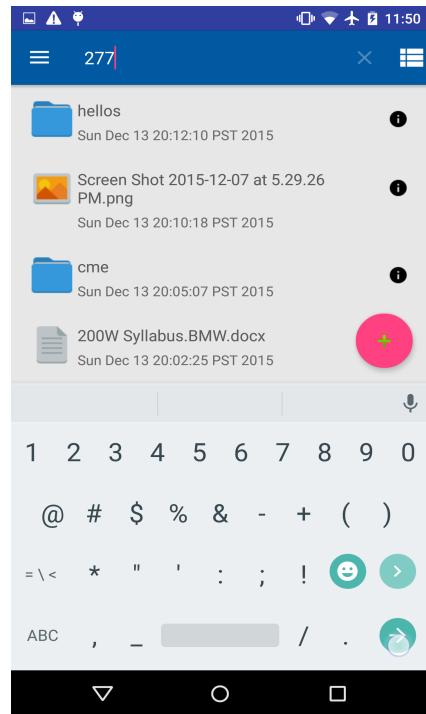
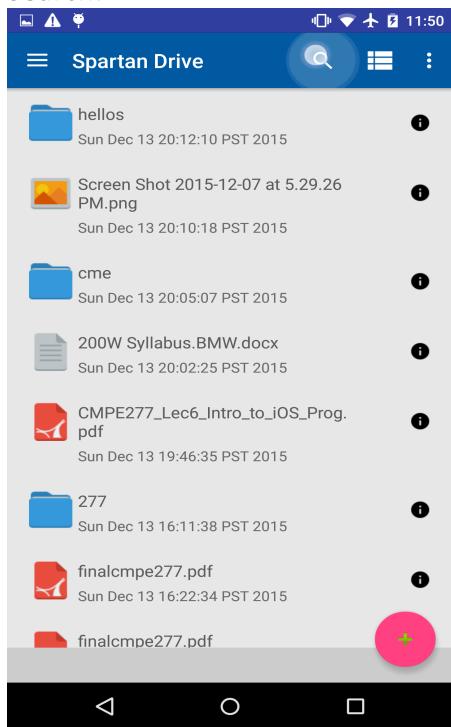
Select the Email Client



SignOut:



Search:



6. TESTING PLAN

Testing plan executed and results

Sr No	Test Name	Expected Result	Result got	Status
1.	Login to SpartanDrive	User can login to application by selecting his desired google account	User logged in to the application	Success
2.	Root screen display	This screen should display the list of all the files/folders in the drive	User is able to view list of all the files/folders in the Drive	Success
3.	Grid view of Drive contents	This screen should display the grid view of all the files/folders in the drive	User is able to view grid layout of all the files/folders in the Drive	Success
4.	Create a Folder	User should be able to create a folder by specifying name after selecting Folder option from the floating button	User can create a folder by specifying its name as required.	Success
5.	Upload a Folder/File	User should be able to upload a folder by from the Device	User can upload a folder from the Device.	Success
6.	Delete a Folder/File	User should be able to Delete a folder/file from the Drive	User can Delete a Folder/File from the Device	Success
7.	Share a Folder/File	User should be able share a folder using email id of the receiver	User can Delete a Folder/File by specifying email-id of recipient	Success
8.	Search a File/Folder	User should be able to search for Files/Folders	User can Search for the file/folder using its name	Success
9.	File Information and icon	In the root screen the user should be able to view file name and basic information of the file	User can view the file name and basic file information	Success
10.	View each file	User should be able to open and view the file	User can open and view each file	Success
11.	Email file	User should be able to email the file selected	User can email the file selected	Success
12.	Usage Report	User should view the report of the amount of memory used by the app	User can view the usage details	Success
13.	Navigate through directory	User should be able to navigate through the directory structure of the folder in the drive	The User is able to navigate to through the	Success
14.	Push notification	User should be able send notification to User on sharing/unsharing	User can send notification on sharing and unsharing	Success

7. LESSONS LEARNT

We implemented our project using Android Studio and using the concepts of Android application development

PROJECT CONCEPTS

- **Activity and Broadcast Receiver**

Using this component the user can initiate any action. Every activity is integrated with the UI using a Window. When the application is first launched the main activity is loaded. After the main activity the series of other activities follows and a schedule is maintained one after the other.

We can issue broadcast announcements using a broadcast receiver. Generally all these are system based and the app can also issue its own broadcast receiver.

- **Parse.com Push notification**

Push notifications are implemented using Parse.com. We learnt how to use Parse.com to publish and subscribe to channels. Each subscriber of a channel will get a notification on the occurrence of event.

- **Google Drive API**

The API is used to handle a number of tasks for access and syncing. We can read and write files as if the Drive is a local file system.

Metadata class is used for handling all the details about a file or folder and includes all details about a file or folder including the title, the MIME type, and whether the file is editable, starred or trashed

Files can be accessed using the DriveApi.getFile method or by allowing the user to select a file using a file selector activity created with the Activity Builder.

- **Team Work**

Working with a team of 6 was a challenge for us. We learnt a lot while collaborating and merging each other's part. The end result however was very impressive.

8. FUTURE WORK

We wish to make our app more user friendly. We would like to improve the user experience by improving the UI and the performance of the application. We also wish to implement features like Enabling feature of editing .docx or .xls files, Able to sync the contacts from your gmail account so that you don't have to specify a contact every time and Issuing alert the user so that he can maintain a history of event being taken place in the drive.

9. CONCLUSIONS

In this way we have developed an application that is on the similar grounds as Google Drive that enables us to share manage folders/files. Thus using Google Drive API for the backend development and Studio for UI development.