# Project ideas

## First Idea:

* We want to design a very straightforward file sending and receiving app.
* We want two key features
  + Send – send the files to a phone
  + Receive – receive the files from a phone
* Use Wi-Fi direct to sends files
* Or use parse.com as a server

## Second Idea:

* We want to design a clean interface for users to mange their files.
* Design many buttons to enhance user experience
  + Delete
  + Undo
  + Search
* May or may not implement the sending and receiving feature

# User Analysis

The users for our application and their characteristics are:

**College Students**

Characteristic:

* Attend classes on a daily basis
* Have many digital files
* Are experience smart phone users
* Are experience social media app users

Users goals:

* Share digital files with classmates, friends, and professors
* Want to perform the sharing tasks under one minute
* Want to access the files in a nice interface

Users knowledge:

* Smart phone experience
* Familiar with modern app interface

**Business People**

Characteristic:

* Spend most of their time on the plane
* Attend many meeting
* Read many digital documents
* Use smartphone for contacting and emailing

User goals:

* Share digit files with secretary, business partners, and other employees
* Want to perform the sharing tasks under one minute
* Want to access the files in a nice interface
* Want to sort the files easily
* Want to retrieve a specific file under 30 seconds

User knowledge:

* Basic smart phone experience
* Comfortable with modern app interface

**Photographers**

Characteristic:

* Take many digital photo
* Have a large library of photo
* Use smartphone to take casual pictures and view pictures
* Travel to different places

User goals:

* Directly share digit files from smart phone to tablet
* Want to perform the sharing tasks under one minute
* Want to have a consistent interface for both smart phone and tablet

User knowledge:

* Basic smart phone experience
* Comfortable with modern app interface
* Expert in using camera application

# Task Analysis

Deleting a file

1. Click on an item in the list
2. Click on an item in the sub-list
3. Select the items to be deleted
4. Deselect items to not send (Optional)
5. Press the trash icon that appears on the action bar
6. Press on the reverse arrow to undo (Optional)

Sending a file

1. Click on an item in the list
2. Click on an item in the sub-list
3. Select the items to be sent
4. Deselect items to not send (Optional)
5. Press the up arrow symbol on the action bar
6. Select a device
7. Deselect a device (Optional)
8. Press the send button on the sending dialog

Receiving a file

1. Press on the down arrow symbol in the action bar

Searching a file

1. Click on the symbol with the magnifying glass in the action bar
2. Type in the text field to start to search
3. Press the plus icon to add the item to the dialog
4. Press the send button to send the files added to the dialog

# Problem Scenarios

**A Computer Science Study Group Working on Assignments Together**

After almost three years at UCSB, students have formed study groups and gained a social network of friends. Around the time before an assignment is due, the students realize that they are unsure whether their answers are truly correct. The students confirm with each other by sending documents or pictures of the assignments to each other. Students quickly browse through each other’s work to check both their friends’ work and their own work.

**A Computer Science Student Trying to Organize Files Downloaded**After almost three years at UCSB, students have most likely downloaded a bunch of pdf’s and documents pertaining to a project description or assignment description that have been outdated. However, students also store important personal information in their phones. Some of these data may be important, and some may not. The student wishes to organize all the files inside his or her phone efficiently. Furthermore, the student may wish to preview the document before deciding whether to delete it or not.

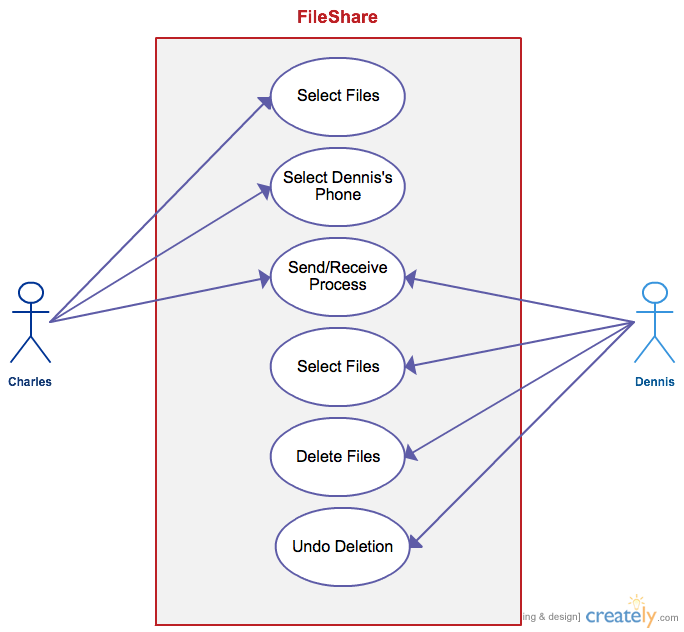
**A Businessman Who is in a Rush**

A Businessman who often travels a lot and rarely has time to sit down and open his laptop needs to send documents and pictures to his or her secretary. The user must efficiently send files within a timeframe or else he or she will miss the plane flying to LA.

**A Photographer Who Forgot his Camera**

A Photographer wants to go the Africa to take pictures of the indigenous tribes and giraffes. He did not bring his professional camera to take a picture of a beautiful scene on a particular moment of the day, so he decides to pull out his phone to take a picture. Now he wants to send the picture to a android tablet without taking that it out.

# Use Case



In this use case, Charles is initiating the FileShare application. He first selects which files he wishes to send. Then a new window appears to select an Android device to send to. He selects “Dennis’ Phone” as the receiver. The file is now in the “Send/Receive Process” where the file is being sent from Charles’ phone to Dennis’. Dennis receives the files and is able to select that file in his list view. He decides that he does not want what Charles has given him so he selects the file and deletes it. Later, he realizes he has clicked the wrong file to delete, so he is able to click the undo button to revert back to the original files.

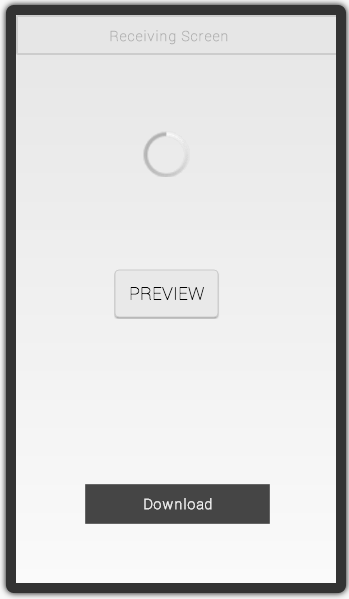
# Prototypes (Initial Idea)

The following prototype is our first design:

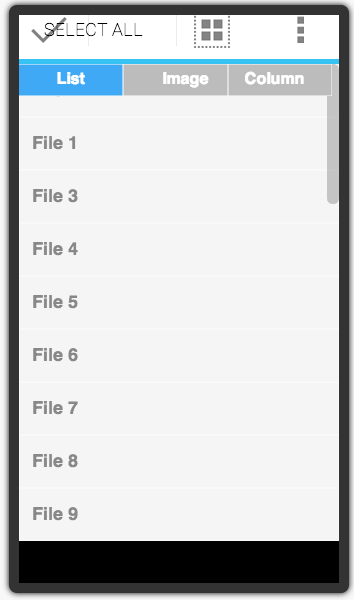
## Send

****

## Receive

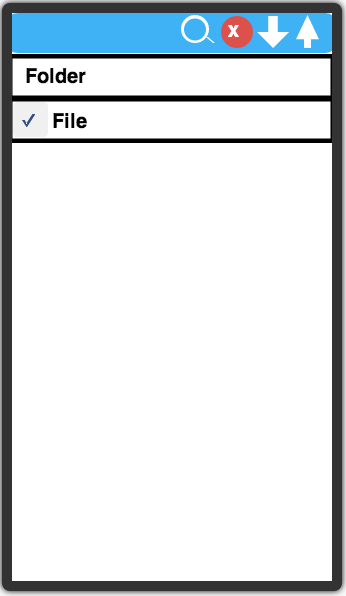


## File View

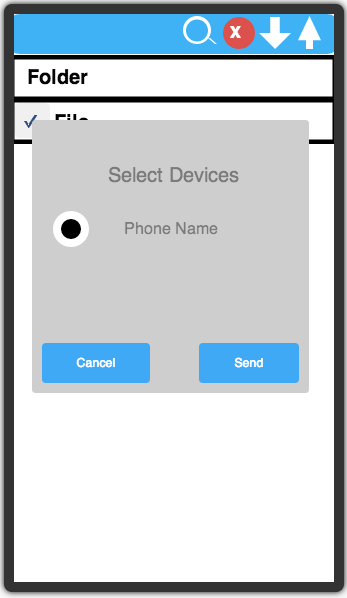


# Prototypes (Final Idea)

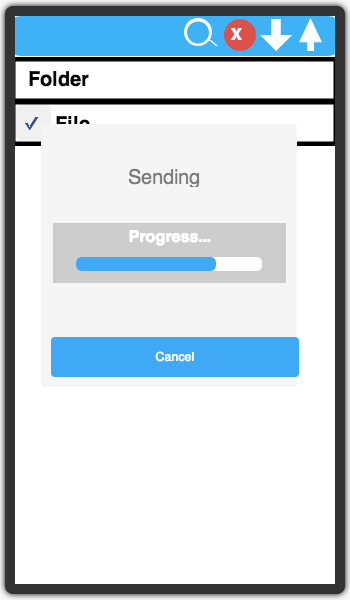
## Main File Editing Screen

****

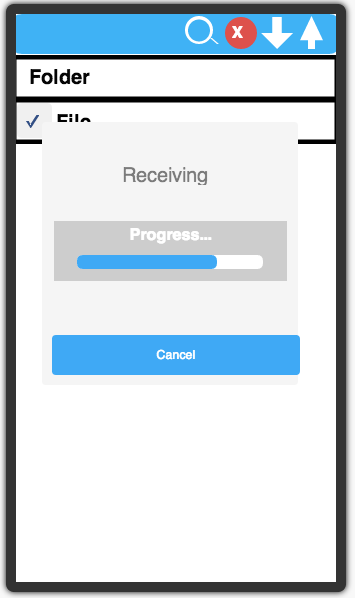
## Device Selection Screen

****

## Sending Date Screen



## Receiving Data Screen



# Qualitative Evaluation

Our team designs a set of questions and interview four people before implementing the features of FileShare. This evaluation took place at a very early stage of our design process. The goal of this evaluation is to understand the potential needs of the users. The volunteers include three college students and a CPA (certified public accountant) examiner. The interviews were conducted on campus in UCSB and through Skype. Before they answer the questions, we first briefly explain to them the idea of FileShare. These are the questions and results:

**Volunteer:** Aura Roy

**What do you think about the idea of FileShare?**

I think it is a very good idea because my phone doesn’t have NFC. I think it would be very convenient to browse through my files and send them to other people.

**What features do you want to see in FileShare beside sending and receiving?**

I think it will be good if you guys can include a file navigation system. I found it very inconvenient to not able to manage my files.

**Volunteer:** Julio Lemus

**What do you think about the idea of FileShare?**

It is a good idea because sometime it is very frustrated for me to send my picture and music files to other people. I would like to use an app like file share.

**What features do you want to see in FileShare besides sending and receiving?**

I think it is a good idea to include an organize view of my files and able to open the files directly from the app.

**Volunteer:** Cameron McNair

**What do you think about the idea of FileShare?**

I would like to use an app like FileShare because it is always a pain for me to send files from devices to devices. I use Dropbox a lot but it is as direct as FileShare

**What features do you want to see in FileShare besides sending and receiving?**

In terms of the features, I think sending and receiving are sufficient. I would hope that you guys would implement this for different platform since I have an iPod touch, windows phone, and a windows laptop.

**Volunteer:** Shu-fen Weng

**What do you think about the idea of FileShare?**

I think it is a great idea for Android devices, but I feel like there may be better applications for this exact feature.

**What features do you want to see in FileShare beside sending and receiving?**

Being able to see a detailed view for my files and some better sort features.

The result of the survey shows that the users like file managing interface. Therefore, we decided to focus on file managing and enhance a better user experience.

# Quantitative Evaluation

This KLM Analysis took place at a later stage when we have prototypes and some implementation. The goal of this evaluation is to get the estimate time of performing varies tasks such as sending, receiving, and deleting files when using FileShare.

This KLM is based on the Use Case we developed earlier:

* Assume there are 2 files
* Assume there are 3 devices
* Assume there are 2 users (one sender/ one receiver)

**This is the KLM for Sender:**

The Sequence Actions consist of:

1. Select a folder tab
2. Select 2 files
3. Click on the send Icon
4. Select device(s)
5. Click on the send Button

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KLM Operator | M inserted | M deleted | Time (s) | Comment |
| P | Yes | No | 2.45 | This P is for selecting the folder tab. Mental preparation is required. |
| P | Yes | No | 2.45 | This P is for selecting files under the folder tab. Mental preparation is require. |
| P | Yes | Yes, rule 2 | 1.1 | This P is for selecting the second files under the tab. M can be remove because it belongs to the cognitive unit |
| P | Yes | No | 2.45 | This P is for selecting the send icon on the action bar. Mental preparation is required. |
| P | Yes | Yes, rule 2 | 3.55 | This P is for selecting the initial device you want to send to. Mental preparation is needed for the first selection |
| P | Yes | Yes, rule 4 | 2.45 | The P is a terminating command for the cognitive unit of selecting a device. M is deleted |
| Total |  |  | 14.45 |  |

**This is the KLM for Receiver:**

The Sequence of Actions consist of:

1. Click on the receive icon
2. Select a folder tab
3. Select 2 files
4. Click on the delete icon
5. Click on the OK button inside the confirm dialog
6. Click on the undo icon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KLM Operator | M inserted | M deleted | Time | Comment |
| P | Yes | No | 2.45 | This P is for selecting the receiving button. Mental preparation is required. |
| P | Yes | No | 2.45 | This P is for selecting the folder tab. Mental preparation is required. |
| P | Yes | No | 2.45 | This P is for selecting files under the folder tab. Mental preparation is required. |
| P | Yes | Yes, rule 2 | 1.1 | This P is for selecting the second files under the tab. M can be remove because it belongs to the cognitive unit |
| P | Yes | No | 2.45 | This P is for clicking on the delete icon on the action bar. Mental preparation is required. |
| P | Yes | Yes, rule 4 | 1.1 | This P is for click on the OK button for the confirmation dialog after deletion. The confirmation dialog is a terminator |
| P | Yes | No | 2.45 | This P is for clicking on the UNDO button for undoing the deletion. Mental preparation is required |
| Total |  |  | 14.45 |  |

The result of the KLM analysis exceeds our expectation. Our goal is to implement a straightforward app that is able to perform sending and receiving files under one minute.

# Summary of Design Decisions

Our initial goal was to create an application similar to Apple’s airdrop capabilities for the Android. We wanted to be able to send documents, pictures, and any other files from Android devices all through Wi-Fi. This would not only allow fast file transfers for users, but also give us an opportunity to really dive into Android’s capabilities and a taste for UI design.

Initially, our first prototype was a simple three-tab interface with a listview, an icon view, and a column view (similar to the Apple’s finder file management). This design would allow the user to view the same files in three different ways and click on the files they wish to send. After a few group discussions, we moved on from this implementation and decided that it was unnecessary for users to view the same files three different ways. As a result, a simple list view with checkboxes is what we came up with. After selecting the files, clicking send would bring a new activity for a list of nearby Android devices to send the files to.

With this design in mind, the actual implementation process began. As we were implementing the design, having a “send” button at the end of the listview would potentially block a file at the end of the listview, so we decided to incorporate the action bar for the send button. The send button would only appear when items are selected in the listview. Once this was created, we expanded the application to include a delete button, an undo button (for after a file deletion), a search feature (to search through files, but due to time constraints was not fully implemented), and the actual send and receive feature (not implemented because we wanted to focus more on the UI design, since it is the focus of the class).

Finally with the core of the application design finished, we added a splash screen and created an icon for our application. Themes and colors were added to better suit the UI design.

# To be implemented

* Send and Receive feature
* Detailed view of files
* Date modified and size of files
* Search feature for the listview
* Search for nearby Android devices
* Multiple undo deletions

These features were not implemented due to time constraints and were not a priority for the UI design process. We wanted to focus more on the UI design rather than getting bogged down by the implementation process. These features are what we wished to implement to make our application better.

# Sources

<http://stackoverflow.com/questions/8855791/how-to-create-standard-borderless-buttons-like-in-the-design-guidline-mentioned>

<http://www.javacodegeeks.com/2013/06/android-action-bar-with-tab.html>

<http://www.androidhive.info/2013/07/android-expandable-list-view-tutorial/>

<http://stackoverflow.com/questions/8855791/how-to-create-standard-borderless-buttons-like-in-the-design-guidline-mentioned>

<http://stackoverflow.com/questions/18525793/how-to-set-the-text-color-of-titlebar>

<http://stackoverflow.com/questions/5049852/android-drawing-separator-divider-line-in-layout>

<http://developer.android.com/reference/android/widget/CheckedTextView.html>

<http://developer.android.com/reference/java/util/ArrayList.html>

<http://developer.android.com/reference/java/util/HashMap.html>

<http://stackoverflow.com/questions/4461134/android-listview-adapter-onclicklistener-issue>

<http://stackoverflow.com/questions/3988337/how-to-keep-expandablelistview-in-expanded-status>