ResX

Final Design Report

Genton Mo

1. Introduction

1.1. Project Overview

ResX, at its core, is a mobile app for Android that is able to transfer files from one phone to another using NFC and Bluetooth technologies. The initial purpose of this app was to both reduce paper consumption at career fairs and other related events and save recruiters time by filtering exceptional résumés from the rest. Instead of bringing dozens of physical copies of their résumé, a user would upload their résumé to their Android device and would transfer their résumé to a company representative via NFC. The representative would then be able to open the résumé and add notes and other markings, as they would with a physical copy. Once the representative is finished marking up the résumé, they would be able to mark whether if it was exceptional or not. The edited copy of the résumé will then be saved to the device’s local storage and/or uploaded to a cloud, such as Google Drive or Dropbox.

1.2. Stakeholders

The main stakeholders in this project are company representatives that appear at school career fairs or events of the like. At these events, representatives receive hundreds of résumés per day. ResX would save the representatives many hours of time sifting through the thick stack of résumés and searching for the exceptional ones. Along with saving paper and time, ResX also acts as a quick method of file transfer between two recruiters when conversing. Without ResX, only one recruiter would have a person's physical résumé, but with ResX, both recruiters can have digital copies with their own personal markups.

1.3. Goals and Objectives

1.3.1. Project Goals

- To create a more eco-friendly way to share résumés at career fairs
- To increase the shareability of résumés

1.3.2. Project Objectives

- To be able to understand and use NFC on NFC capable devices
- To be able to access and read existing files and create new files and directories on an Android device
- To be able to access and read existing files and create new files and directories using Google Drive and Dropbox APIs

1.4. Outcomes and Deliverables

The end goal of the project is to have a working prototype with no crashes occurring. The app should be able to access and read the sender's local files, Google Drive, Dropbox, and any other app that may store pdf or jpg files. The app should also be able to send the read file over the the receiving device and write it to that device's local storage or cloud storage. Upon completion, the existing project files will be submitted, as well as an Android application package (APK) of the app.

2. Background

For every career fair I have attended, I have noticed the absurd amounts of paper that are given out to recruiters. I was asking a few friends how many résumés they printed out per career fair and most of them responded around the 20 to 25 range. Giving a rough estimate of 1,500 students attending the career fair here at Cal Poly, that's about 33,750 sheets of paper printed per career fair. According to howstuffworks.com, one 60-foot tall pine tree with a one foot diameter produces around 80,000 sheets of paper. So for every career fair that happens at Cal Poly, half a tree's worth of paper is used.

The main goal for ResX is to reduce the amount of paper waste that occurs at career fairs and other related events. The original idea for the app came from two separate inspirations put together. The first inspiration was from a friend who wanted to create a business card exchange app using either NFC or QR codes. The second came from noticing Cisco Systems taking pictures of résumés at a career fair and not keeping the physical copies. I put both of the ideas together and that's how ResX came about.

3. Project Requirements

At the minimum, ResX is expected to transfer files from one Android device to another via NFC and Bluetooth. The file types that shall be able to be transferred are pdfs and jpgs.

The user on the sending device shall be able to search through their device's file system to select a file if there is a pre-existing app on the phone that allows them to do so. Otherwise, only files in the Downloads folder will be available to select from. If available, the sending user may also select files from cloud storages, given there is an internet source and an account logged in to said cloud.

The user on the receiving device shall be able to select the location of where they want the received file to be placed. The default location for file placement is the Beam folder, which is automatically created when ResX is used for the first time. If available, the receiving user may also select a location inside a cloud storage, given there is an internet source and an account logged into said cloud.

The minimum hardware requirements for ResX are shared between the sending device and the receiving device. However, the device requirements in terms of permissions depend on whether the device is sending the data or receiving the data. In addition to having these minimum requirements, there are also optional requirements, allowing the user to fully utilize ResX.

Minimum Requirements

- Device must be running at least version Android 4.1 (Jelly Bean)
- Device must have both an NFC chip and Bluetooth chip embedded inside
- File transfer time will take at most 2 seconds to complete

Optional Requirements

- Device may have either a WiFi chip embedded or a GSM chip inside
- User may have a file explorer app on their device to search for a file on any directory they have access to
- User may have a Google Drive, Dropbox, or any other cloud storage account logged in on the device

Sending Files

- User must give ResX read permissions to access internal and external storage files

Receiving Files

- User must give ResX write permissions to access internal and external storage files

4. Design

4.1. Method of File Transfer

Since ResX was supposed to be a file transfer app, the biggest question of ResX was how files were going to be transferred from one device to another. A few of the available options were using NFC, Bluetooth, WiFi, and QR codes. My first thought was to definitely not implement Bluetooth because it would be a mess trying to pair two devices with hundreds of phones with Bluetooth enabled. I also decided to eliminate WiFi due to due to WiFi not being always available. This left only NFC and QR codes on the table. After putting in some thought and doing some research, I realized in order to do file transfers using QR codes, I would need to set up a web server of some sort to host all the files. NFC ended up being the only and, in fact, most feasible solution for file transferring.

What NFC essentially was used for in ResX was establish a Bluetooth connection between two devices. In order for the two phones to detect each other, the NFC chips on each phone needed to be at most 4 centimeters apart from each other. This solved the problem of taking too long to pair devices. As soon as the connection is established, the file gets sent via Bluetooth to the receiving device.

The speed of the file transfer is relatively slow compared to other existing technologies, however. With Bluetooth 3.0 and 4.0, the speeds are capped around 25 mbps. With smaller files, such as résumés, this speed is ok, as it would take at most a second or two to complete the transfer. The scalability for larger files, however, is terrible and would require the implementation of a different technology, such as WiFi Direct.

4.2. File Selection

By default, the sending user has the option to select files from only their Downloads folder and Recents folder. If the user has a file explorer app, such as ES File Explorer or File Manager, the user may traverse through their phone's directories and select any supported file. If the user has an account that is linked to a cloud storage app and has an internet connection, the user may also search through those drives and select a file from there. Allowing these options to the user provides a huge convenience to the user, which I felt was paramount in terms of user experience.

4.3. File Storage and Organization

After speaking with some recruiters at career fairs, many of them agreed that the biggest feature they were looking for was to be able to organize groups of résumés easily. This was easily solved by allowing the user to select which directory they want files to be put into. This way, organizing résumés "easily" is left up to the discretion of the user instead of the developer.

4.4. Costs and Concerns

The financial cost of creating ResX is zero dollars. All development was done on Android Studio and all research and help found was done through Google and StackOverflow. There is no direct financial cost in using this app as of now. However, people may spend money for more space on cloud drives or expandable storage such as microSD cards.

Some concerns that may arise are listed below

- Eye strain may occur after staring at a device's screen for prolonged periods of time
- Devices can be stolen and many users' personal information can be leaked
- Devices may run out of battery or malfunction throughout the day

5. Testing

Test Number	Test Specification	Test Description	Result
1	NFC available	Checks if the phone is capable of NFC	Success
2	Internet available	Checks if there is a stable internet connection	Success
3	Bluetooth available	Checks if the phone is capable of Bluetooth	Success

Table 1: Tests to check hardware specifications

Test Number	Test Specification	Test Description	Result
1	Select local pdf file	Tests to see if pdf MIME types are able to be selected. Also tests file uris work	Success
2	Select local png file	Tests to see if other file types fail when selected	Success
3	Select remote pdf file from Dropbox	Tests to see if pdf MIME types are able to be selected. Also tests if content uris work	Success
4	Select remote pdf file from Google Drive	Tests to see if pdf MIME types are able to be selected. Also tests file uris work	Success

Table 2: Tests to check if files can be selected correctly

Test Number	Test Specification	Test Description	Result
1	Open received file	Tests to see if the file was received correctly	Success
2	Check location of received file	Tests to see if file was placed in the default location	Success
3	Check location of file copy	Tests to see if file was copied to desired location correctly	Success

Table 3: Tests to check if files were received correctly

6. Conclusion

I think this working on this project has been a really good experience for me. It was really cool having an open ended project and having to decide which design choices would be best to implement. This project was also a good test of Google-fu whenever I got stuck. It was really different trying to research so many topics at once and trying to understand how exactly each thing worked.

ResX is, unfortunately, not fully completed due to time constraints. Had I had another few weeks to work on it, I would have liked to make it functionally complete and to have pushed it onto the Google Play Store. Although there is still some work that can be done to improve the app, the parts that do work function well. I think if ResX were to be continued and refined, it really could change how career fairs worked in terms of saving paper.