

University of Alabama in Huntsville

**LOUIS**

---

Honors Capstone Projects and Theses

Honors College

---

7-12-2019

## Listify-Organize your Life

Venika Gaur

Follow this and additional works at: <https://louis.uah.edu/honors-capstones>

---

### Recommended Citation

Gaur, Venika, "Listify-Organize your Life" (2019). *Honors Capstone Projects and Theses*. 353.  
<https://louis.uah.edu/honors-capstones/353>

This Thesis is brought to you for free and open access by the Honors College at LOUIS. It has been accepted for inclusion in Honors Capstone Projects and Theses by an authorized administrator of LOUIS.

# Listify-Organise your life!

by

**Venika Gaur**

**An Honors Capstone**

**submitted in partial fulfillment of the requirements**

**for the Honors Diploma**

**to**

**The Honors College**

**of**

**The University of Alabama in Huntsville**

**07/12/2019**

**Honors Capstone Director: Dr. Tathagata Mukherjee**

**Assistant Professor, Computer Science Department**

---

Student (signature)                      Date

---

Director (signature)                      Date

---

Department Chair (signature)                      Date

---

Honors College Dean (signature)                      Date



Honors College  
Frank Franz Hall  
+1 (256) 824-6450 (voice)  
+1 (256) 824-7339 (fax)  
honors@uah.edu

### Honors Thesis Copyright Permission

This form must be signed by the student and submitted as a bound part of the thesis.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Venika Gaur

Student Name (printed)

*Venika Gaur*

Student Signature

07/12/2020

Date

# Table of Contents

|                              |    |
|------------------------------|----|
| 1. Dedication                | 4  |
| 2. Abstract                  | 5  |
| 3. Introduction              | 6  |
| 4. Design                    | 7  |
| 5. Setting up Android Studio | 9  |
| 6. Creating the app          | 10 |
| 7. Project experience        | 13 |
| 8. Conclusion                | 15 |

# Dedication

I would like to dedicate this project to my parents Pradeep & Meenakshi Gaur, my brother Satvik Gaur, and my boss Elisabeth Johnson. Their constant love and support motivates me to work hard and achieve my goals.

# Abstract

“Listify” is a simple to-do list app that allows users to add a note, edit it, delete it, and save the data when user exits the app. It was prototyped using Adobe XD and created using Android Studio. Through this project, the author aims to understand the process of designing, developing, and deploying an android application. This paper attempts to describe the process of designing and creating an Android application. The author will also describe the challenges she encountered and the things she learnt during the duration of this project.

# Introduction

Android app development is getting increasingly popular in the tech world. To meet the requirements of Honors college diploma, all students need to submit a capstone thesis. However, instead of using my senior design class project as my capstone, I decided to learn something that was not taught in my computer science classes- Android app development. I wanted to use this opportunity to gain extra knowledge and also create something that I felt I really needed in my life- a simple to-do list app. I like making to-do lists on paper, but I almost always end up losing them before finishing them up. Almost all the apps I have tried on Appstore have added functionality that makes app a lot better but takes away its simplicity. The additional features are great for others but for me, they are a distraction because I start to focus too much on using pictures and voice notes to write the task or connecting social media accounts for added incentive or setting calendar reminders for the tasks rather than doing them. I wanted an app that acted just like a pen and paper to-do list with no extra frills and features to distract me from focusing on getting the tasks done.

I have also been trying to get into Android app development for a while but have been putting it off because of classes and work. This project seemed to be a perfect opportunity to finally start learning how to make applications and finally get into the field of app development. Creating an application that I was in dire need of was just the added bonus of this endeavor.

# Design

Before getting started with android app development, I researched all the tools that I would need to start planning the design process. Since I was learning the app development from scratch, I was able to try different software and tools that were available and then pick the ones that were best suited for the job. I used the following tools to aid me during this whole process-

## 1. Adobe XD

Adobe XD is Adobe prototyping tool for creating wireframes, prototypes and screen designs for websites and mobile applications. I used Adobe XD for designing the initial template for the app. While the Android Studio itself has a way to design the app, Adobe XD's interface allowed me to test different UI for the app and then choose the one that best suited the app design I had in mind. It was very useful in the initial app designing stage.

## 2. GitHub and Git Command line

I used GitHub as my online repository and used Git command-line tool to commit to this repository.

## 3. Trello

I used Trello to keep track of my project progress and manage all the resources that I was using to learn Android App Development. I also used it too keep track of

how far along I am in the application building process and what other features I still need to work on.

#### 4. Android studio

Android Studio is the official integrated development environment for Google's Android OS and is designed specifically for Android development. It is based on the JetBrains IntelliJ IDEA editor. For this project, I used Android Studio as my IDE.

After choosing my IDE, I had to pick between Kotlin and Java as my programming language. Kotlin is a newer programming language and has been chosen by Google as the official programming language for android app development. However, Java is still a widely use language for Android app development and has been around for a while. Ultimately, I decided to go with Java because I had prior experiences with the language and decided to learn Kotlin once I am proficient in using Java to write android apps.

## Setting up Android Studio

After installing Android Studio, I began to set it up. I used the “Standard” installation because that was more than enough for me to get started. To begin building the app, I needed to be able to try and test the app I am building while I am writing the code. Now this can be done either by plugging in an Android device and testing the app on the device or by using an Emulator. Since using an emulator was the easier and quicker option, I decided to go with that during the testing period. Luckily, the Android Studio now comes with an emulator so all I needed to do was set it up. It is recommended to test the app on a physical device before releasing it because the emulator just stimulates a device and is not an actual copy of the physical device. Since I have an android device available, I will do the final testing using that device.

After setting up a new project using Java, I used Git command-line tool to commit my code to a GitHub repository. Using Git allowed me to back up my code and keep track of all the changes I make from this point on. It came in handy in later part of the project when a change that was made to the code broke it and I had to go back a version. Using GitHub to host my code online had another benefit. It allowed me to work on my projects from different machines which helped a lot since my personal laptop was ill-equipped in terms of memory and resources that were needed for the proper functioning of the Android Studio.

# Creating the App

To begin the process of creating the app, I started with prototyping my idea on Adobe XD by creating the different user screens and connecting them in a way that I wanted the app to flow. Since I wanted a minimalistic app that just does what a pen and paper list does, designing a basic prototype was not hard. The application had 3 screens- a main screen which displays the tasks in the list and also where we can add the items to the list, an edit items screen and finally a delete items screen. Here is what the prototype looked like in Adobe XD-

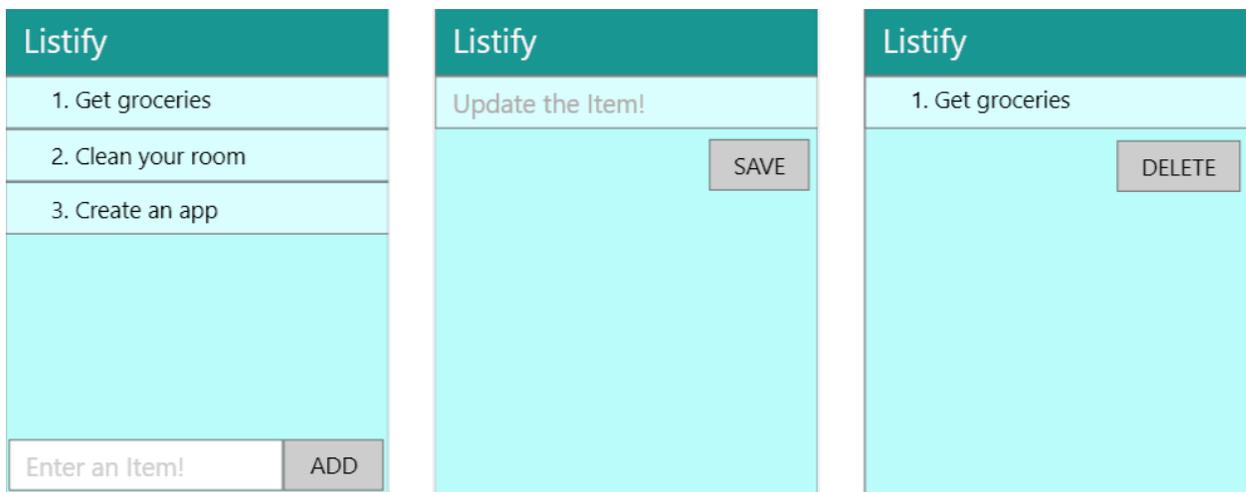


Fig. 6.1 Main screen, Update Item screen and Delete Item screen prototypes in Adobe XD

After designing the prototypes in Adobe XD, I began working with Android studio. To start creating my app, first I had to pick a View Layout. View Layouts are the base classes upon which we build the interactive UI components. View Layouts act as a container for various view components like buttons, text fields, etc. The 3 layouts in View Layout are linear layout, relative layout, and constraint layout.

Linear layout, as the name suggests, aligns all the children views in a single direction, horizontally or vertically. Relative layout displays children views relative to the chosen view and constraint view is a mix between the two. It is also the default view in Android studio. However, for this project, I decided to go with Linear view since there is so much symmetry between the views. After choosing a view layout, I started to create views for the app, i.e., choosing a background, adding clickable buttons to perform user actions and editable text fields for user inputs, etc. Next, I worked on adding event listeners and writing methods connected to each view that performed the necessary actions when the user interacted with them. To display my task strings as a vertical scrolling list, I used something called a RecyclerView. Before that, I was trying to use a ListView adapter to make my tasks list work. RecyclerView is a successor to the ListView but it still was by far one of the most difficult part of this entire project. It took me a while to understand the basics of it and how to use it in my project.

During the app creation process, I realized that my initial prototype idea of having a separate delete screen was unnecessary. Since I was trying to make this app as simple as possible, I made the decision to simplify the deletion of the task by adding an event listener so that if the user long presses an item, that deletes it. I also added little toasts that pop up when an item is added or deleted.

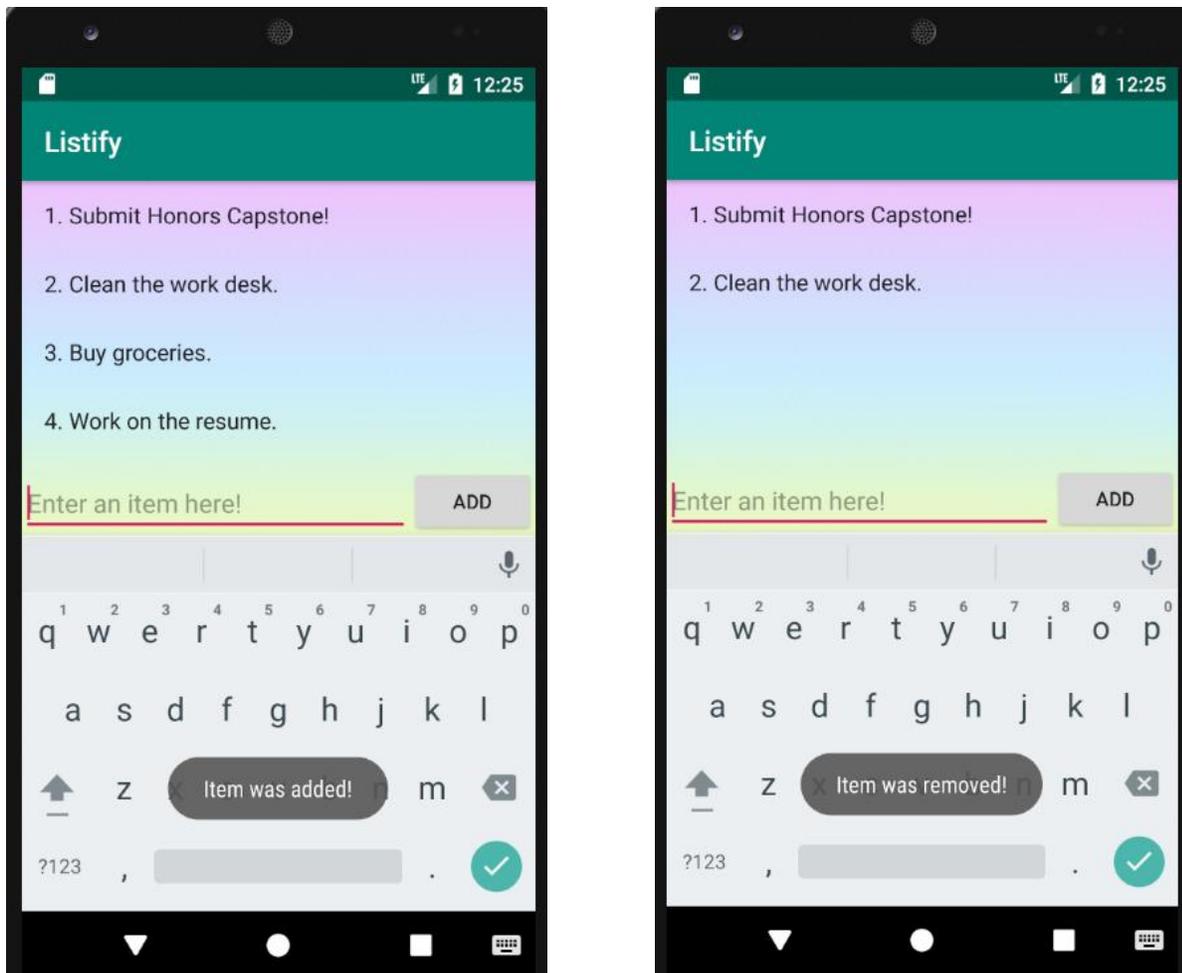


Fig. 6.2 App screen with little toasts on task creation and deletion

The rest of the app creation process was fairly simple. A big challenge I ran into was when after adding toasts to the code, the app just stopped running. I tried various ways to fix the issue but for some reason, I was missing a long list of utilities even though the app was running just fine a day ago. Luckily, I had all the versions on my code on GitHub and I used it to compare the most recent version with the previous one. Turns out, while adding toasts, I accidentally removed a single line of code that was used to include the import all the utils. A single line of code was all it took to break the entire application. This taught me two valuable lessons, one- to always leave spaces between the code blocks and two- to always back up my code using Git to allow version control.

# Project Experience

This entire experience was as challenging as it was rewarding. Initially, I struggled a lot because Android OS was a new territory for me and to create apps, I needed to understand the Android directory structure and the resources. Luckily, I was taking a network security class with my project mentor, Dr. Tathagata Mukherjee that talked about Android operating system and had us go deep into the Android file structure and source to understand the vulnerabilities of the system. While navigating the android file structure and adding or modifying code to add or fix vulnerabilities, I gained a much deeper understanding and appreciation for the Android OS. That course gave me a foundation in Android upon which I built my knowledge of Android app dev.

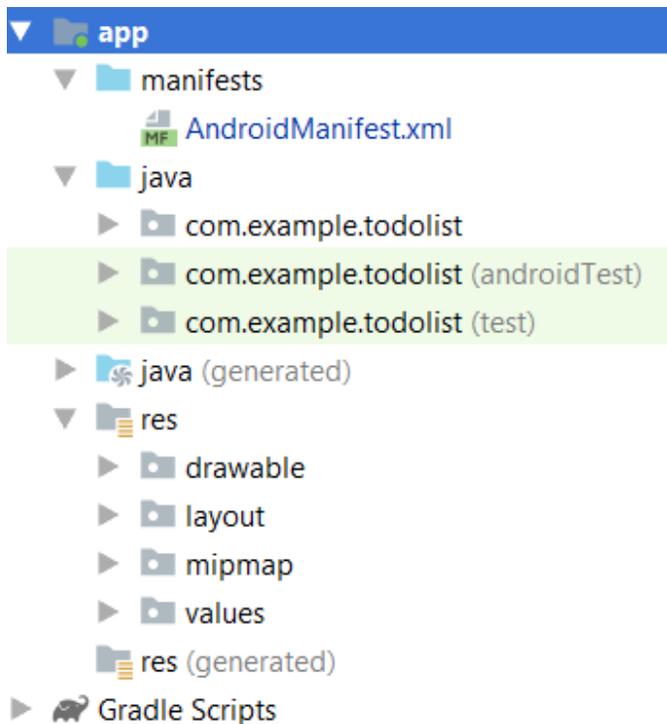


Fig. 7.1 Android app's directory structure

The challenges I ran into during the course of this project had to do a lot with the android studio itself. During this process, I came to realize that just knowing the Java language was not enough to create android applications. I needed to understand how to make my code do what it is supposed to do across multiple screens. So, initially, I struggled a bit with sending data across different screens. For example, if a user deletes a note, I had to send that info on to the main screen/activity without needing to load the app again. Or if the user exits the app, I needed to figure out how to save all the notes so that when the app is opened again, they do not just disappear. After a few trials, I was able to figure it out and things got a little easier after that.

After finishing up my app, I decided to work on improving its user interface and user experience. And it was not easy as I thought it would be. I did not realize how much time and effort it takes to come up with unique designs and backgrounds for the app. Before this project, I was just a programmer with enough skills to create a simple app background and that is it. All my buttons were little rectangles filled with basic colors. The app was not ugly by any means, but it was definitely missing that polished touch. So, to improve my skills, I went ahead and did a quick course on UI/UX design. It added a week more to my total project time, but it definitely was worth it.

## Conclusion

After finishing up this project, I can honestly say that android app development was not as daunting as it initially seemed. After figuring out the bare bones of the application, it was just sort of rinse and repeat till the app does everything it is supposed to do. When I started this project, I expected to come out as a better programmer, but I am happy to report that it has also made me a better designer.

Overall, this capstone project was a very positive project experience. I believe I have grown as a programmer and as an independent learner. Initially, I tried to replicate the exact app that I envisioned in Adobe XD. But during the actual coding process, I allowed myself to deviate from the original plan to make tweaks that made the application better. The result was a to-do list app that was just as simple as using pen and paper. Since then, I have been trying to tweak the app more. The aim was to build an app that is perfect for people that get distracted too quickly and I believe that I was successful in doing so.

After this project, I decided to continue developing apps and have so far created a Twitter bot and an Instagram bot by using official APIs. Working with APIs was also challenging at first but once I got over the initial hurdle, the apps just started to come together. I am currently working on a messaging bot and trying to incorporate online money transfer into the application. It is a little challenging, but I cannot wait to see how far I go in this journey!

I am extremely glad that I decided to choose Android app development as my Honors Capstone Project. This project allowed me to grow as a programmer outside the normal classroom curriculum and has opened an entire new field for me in terms of career opportunities after school.