

Pinpoint

Jake Moretz, David Lee, Alexander Wolff

App Description [10 points]

Goal: The main goal of Pinpoint is to track packages from around the globe

Functionality: Pinpoint allows users to create an account which then brings them to a screen with a list of options. These options include package tracking, shopping, cart, support and coupons. Package tracking allows users to track packages, shopping allows users to purchase items from different stores/websites,

Target audience: The target audience of pinpoint are teens and adults who shop online

App Features [15 points]

1. Register: Allows users to create an account
2. Login: Allows Users to login with the email and password they registered
3. Reset password: Allows users to change their password if they forgot
4. Welcome page: page shown after logging in where the user can then pick what they want to do
5. Track package: page that brings up a tracking website to track the user's package
6. Shop: Allows users to shop for items they want
7. Support: Page that gives description and mission of pinpoint and also allows the user to send a support ticket

UI Design [15 points]

1. Register

Pinpoint
All-in-one Delivery Tracking

Full Name

Age

Email

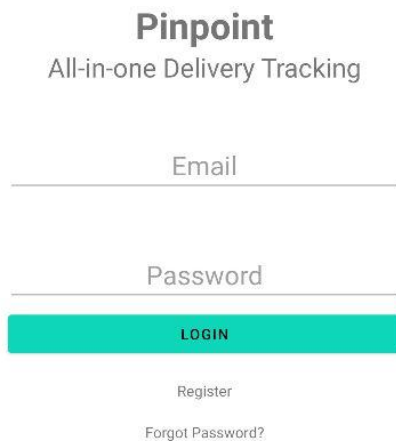
Password

REGISTER

A simple form that users enter their information to be used in conjunction with Firebase so that they may receive an authentication email to login to our app. All the fields check whether valid information has been entered. If the user does not enter information into one of the fields, an error will remind them to fill out all the fields.

Constraint Layout, two TextViews, four EditTexts, one Button

2. *Login*



The login form for the Pinpoint app. It features a header with the app name "Pinpoint" and the tagline "All-in-one Delivery Tracking". Below the header are two input fields: "Email" and "Password". A red "LOGIN" button is positioned below the password field. At the bottom of the form, there are two links: "Register" and "Forgot Password?".

After registering the users have to check their emails for an authentication email (usually found in the spam folder). After which they can login to the app. The app also tells the user if an invalid email or password has been entered. Every Layout page has a Pinpoint subheader and header, which the user can click to return to the previous page (Usually Login or Welcome page once logged in).

Constraint Layout, four TextViews, two EditTexts, one Button

3. *Forgot Password*

Pinpoint

All-in-one Delivery Tracking

Email

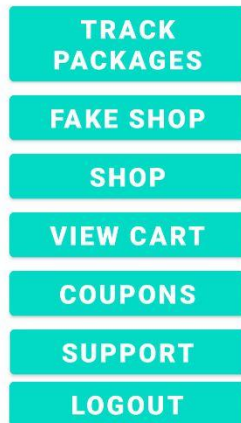
RESET PASSWORD

The user can request a password reset by clicking on the “Forgot Password?” text view in the main activity. Again, only valid emails will receive a password reset email.

Constraint Layout, two TextViews, one EditText, one Button

4. Welcome Page

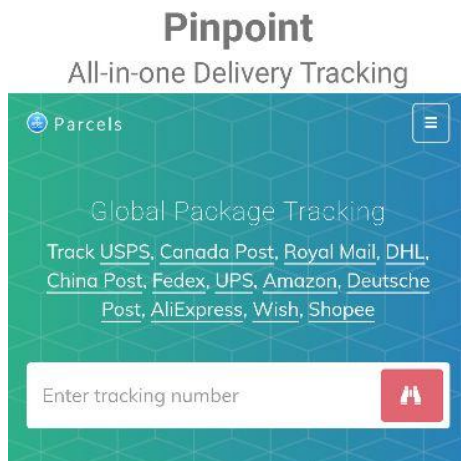
Pinpoint
All-in-one Delivery Tracking
Welcome, Alexander Wolff!



After a successful login, users should be greeted to the welcome page where they can navigate to our other features or simply log out whenever they are done.

Constraint Layout, three TextViews, Seven Buttons

5. *Track Packages*



Remember to hit Pinpoint to go back a page / go back to the welcome page. This page lets users input their shipping information.

Constraint Layout, two Textviews, one Webview

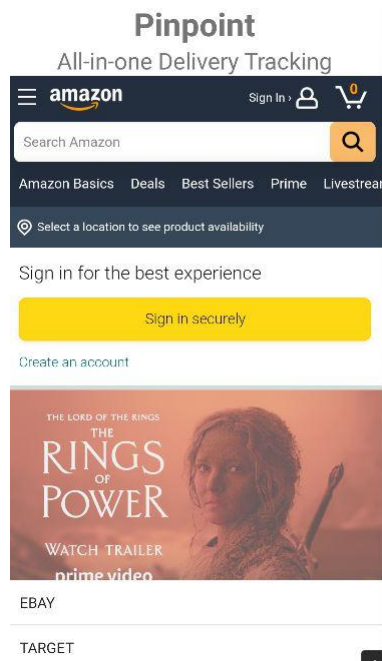
6. Fake Shop

Pinpoint	
All-in-one Delivery Tracking	
iPhone	\$800.0
MacBook	\$1400.0
test	\$1400.0

We added a fake shopping tab to show what we have learned this semester when it comes to adding items to a recyclerview and interacting with them. We hardcoded some of the items in this shop, which the user can click on to add them to a shopping cart. Unlike the webviews from the coupon, track package, and shopping tab, the products here are just items in an arraylist. We implemented this feature to stay on theme with the tracking package and shopping design of our app.

Constraint Layout, two Textviews, one Recyclerview

7. Shop



Remember to hit Pinpoint to go back a page / go back to the welcome page. This page lets users browse shopping websites and scroll through the listview to see different webviews.

Constraint Layout, two Textviews, two Fragments -> two WebViews

Pinpoint

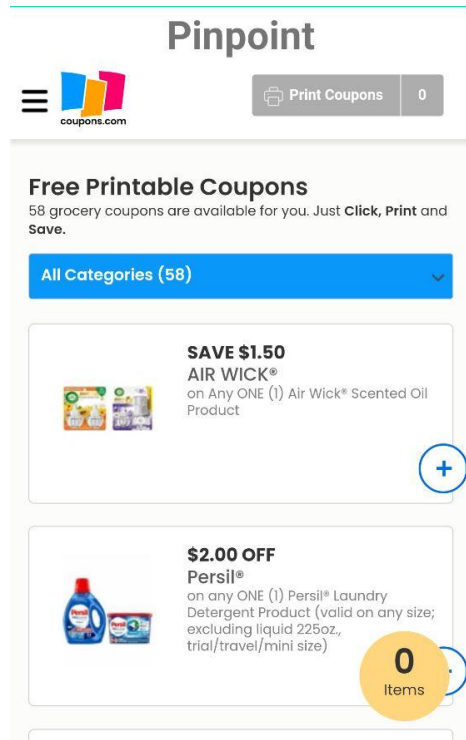
All-in-one Delivery Tracking

iPhone	\$800.0
test	\$1400.0
MacBook	\$1400.0

CALCULATE PRICE

Overall Price:

Constraint Layout, three Textviews, one Recyclerview, one Button



In the coupons tab, users can check out several available coupons from the biggest coupon website.

Constraint Layout, two Textviews, one Webview

Code Design [10 points]

A. MainActivity

- a. onCreate()
- b. onClick()
- c. userLogin()
- d. onComplete()

First screen presented to the user that lets them register or login.

B. RegisterUser

- a. onCreate()
- b. onClick()
- c. registerUser()
- d. onComplete()

Register page that lets users input their information into the firebase server to login.

C. ResetPassword

- a. onCreate()
- b. resetPassword()
- c. onClick()
- d. onComplete()

Lets users update the firebase information to login successfully again.

D. Support

- a. onCreate()
- b. onClick()

E. Welcome

- a. onCreate()
- b. onClick()
- c. onDataChange()
- d. onCancelled()

General splash page that lets users navigate to different pages of the app. When hitting the top “Pinpoint” on any app it redirects them to the welcome page.

F. User

- a. User()

G. TrackPackages

- a. onCreate()
- b. onClick()

Shows a webview of a website that lets users input their tracking information to see their arrival dates.

H. InfoWebFragment

- a. onCreate()
- b. onCreateView()

Used in conjunction with ListFragment.

I. ListFragment

- a. onCreate()
- b. onCreateView()
- c. onItemClick()

Used in conjunction with InfoWebFragment.

J. Shop

- a. onCreate()
- b. onClick()

Presents a webview and a listview of available websites. Webview lets users see the webpage clicked.

K. Coupons

- a. onCreate()

Presents a webview of a popular coupon website.

L. FakeShop

- a. onCreate()
- b. onClick()
- c. createItem()
- d. buildRecyclerView()

Shows a recyclerview with items stored in an Arraylist. The items are clickable and get send to a database.

M. ViewCart

- a. onCreate()
- b. onClick()
- c. onDataChange()
- d. buildRecyclerView()

Shows the items that were bought from the fake shop. A recyclerview grabs the items from a database. By using a snapshot, the price gets calculated and shown at the bottom when a button is clicked.

APIs [15 points]

We used Firebase to handle the login and registering of users into our app.

Challenges [10 points]

Unlike the assignments, the project used a lot more interfaces and java classes. There were a lot more variables to keep track of as well as individual ids. When encountering issues, it was sometimes very time intensive to find a solution. The debugger was a big help in that regard, but we still needed to improve our Java skills to fix a lot of the bugs that we encountered.

Deviation from proposal/Incomplete Features [5 points]

When development began for the app we noticed some limitations on what we knew how to do and the scope of our initial intentions for the app. We haven't implemented a social media aspect of the app where users can share pics or updates on their products. However, we did implement a register/login page and that can be used in future iterations of the app for when we do implement a social media aspect. We have also included an all around shipment website tab where users can enter their tracking information to check on their packages.

Lessons Learned [10 points]

This section can include anything from coding and APIs, to dividing the work, to working on a team, to scheduling, to allocating time for different features.

How About This [10 points]

a) Topics that you wish had been covered.

David: Maybe making something with animations in the app? I can't really think of much since I think we covered quite a bit in the time that we had for a summer semester.

Jake: I don't know

Alex: I agree with David that animations would have been a really cool way to show what you can do with Android Studio. The Picasso feature and adding images from an URL to an imageview was fun, but I would have liked to go a bit more in depth with it and discussed how to make the app look better. Perhaps explaining more about the cartview feature would have been great.

b) Topics that you wish had been covered earlier or in more depth

David: I never quite got assignment 2 to work properly with the SMS messages updating the ticker list. I think if we went a little more in depth with how to handle those BroadcastReceivers then it would have been a little easier.

Jake: I wasn't able to understand assignment 2 until towards the end of the semester and even then I don't know it too well, so if the topics for assignment 2 went more in depth I think it would have gone better.

Alex: The beginning of the semester felt a bit rushed. At the end, I really got the hang of it because assignment 4 and 5 were super intuitive and made it necessary for me to understand how the different components of the java classes and the layouts interacted. Also, the modules for the later assignments gave a lot more insight on how to get everything to work. With the first and second assignment, I had to look up a lot of information from the Internet instead of having it presented in the lecture slides.