**How to Use this Template**

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

---



**GitHub Username**: kdrosado

# Trendy

## Description

(Write a brief summary of what your app does. What problem does your app solve?
Not sure how to write a good description? Search 5-star apps on the Play Store for inspiration.)

Trendy is an Android application that lets you share the latest trends on Etsy with your Google+ followers.  So if you have always wanted to share items from Etsy with your friends and family this is the app for you.

# Intended User

This Android application is for the following users:

- Google+ users
- Etsy users
- Android users
- Family, friends

# Features

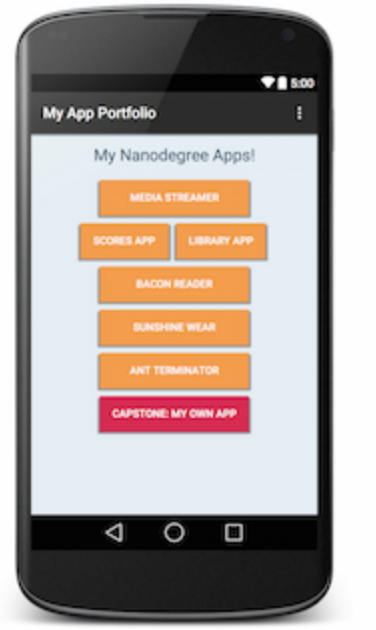
List the main features of your app. For example:

- Uses Etsy API to store a list of latest trends
- Uses Google+ API to allow users to share list from Etsy with Google+ followers
- Using Admob for ads

# User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

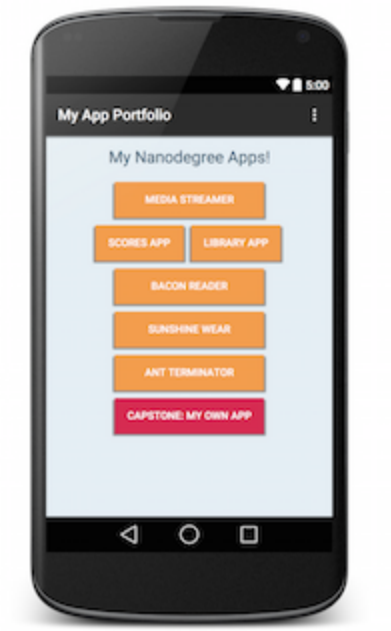
## Screen 1




Replace the above image with your own mock [ click on the above image, then navigate to Insert → Image… ]

Provide descriptive text for each screen

### Screen 2




Replace the above image with your own mock [ click on the above image, then navigate to Insert → Image… ]
Provide descriptive text for each screen

Add as many screens as you need to portray your app's UI flow.

## Key Considerations

### How will your app handle data persistence?

Describe how your app with handle data. (For example, will you build a Content Provider or use Firebase Realtime Database?)

### Describe any edge or corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

**Describe any libraries you'll be using and share your reasoning for including them.**

*(For example, Picasso or Glide to handle the loading and caching of images.)*

I will be using the following two libraries:

- **Picasso:** To load images from the internet without worrying about advanced topics like caching, multi-threading, or memory constraints
- **Retrofit:** To create readable Java interfaces and classes that match the API we are accessing, as well as allowing us to interact with simple Java objects instead of having to parse network responses manually

**Describe how you will implement Google Play Services or other external services.**

*(Describe which Services you will use and how.)*

- **Etsy API:** To get a list of trendy items on Etsy
- **Google+ API:** To share the list of trendy items from Etsy with Google+ followers
- **Admob:** To display ads

# Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

## Task 1: Project Setup

*(Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.*

*You may want to list the subtasks. For example:*

- *Configure libraries*
- *Something else*

*If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.)*

- **Creating a new Android project**
    - Open Android Studio
    - Start a new Android Studio project
    - Name the project “Trendy”
    - Accept all defaults
    - Run project;  If it was built properly you should a Hello World Message. Otherwise repeat the previous steps.

- **Setting Up the API’s:**  When interacting with API’s,  you need to register your application with the API provider to show that you are accessing the data in a secure way.

    - **Setting Up the Etsy API:**
        - You will need to get an API key from Etsy by going to the following site: https://www.etsy.com/developers/
        - If you do not have an Etsy account, you will need to create one by selecting “Register as a developer” on the left column.  Otherwise, you can log in with your account.
        - Once you are logged in, you will be asked to create a new app.  Fill out the following fields:
            - Application Name
            - Description
            - Application Website
        - Under “What type of application are you building?”, select mobile
        - Under “Who will be the users of this application?” select just myself
        - Under “Is your application commercial?”, select no
        - Leave the following unchecked:
            - Upload or edit listings
            - Read sales data
            - Send email
        - Agree to terms to create your app
        - This will take you to your API key
        - Go back to your project on Android Studio and create a class and store your api key inside a string

    - **Setting Up Google Play Services & the Google+ API:**
        - You will need to get an API key from Google Play Services for the Google+ API by going to the following site: https://console.developers.google.com/cloud-resource-manager
        - If you do not have an account you will need to create one.
        - Once logged in, select “Create Project”
        - Name your project.  Note the project id is auto generated.
        - For location, accept default which should be “No organization”

- Press create
- Once created go to top menu and select API's & Services
- Select View All, then select Google+ API
- Click Enable API
- Press button "Create Credentials"
- Select "OAuth Consent Screen" from top menu
- For Email address list your email address
- For Product name use the name of your app
- You can skip the rest of the items as they are all optional. Click save.
- Now select Credentials from top menu, then select OAuth Client ID
- Under application type, select Android
- For Name, accept the default
- For the Signing Certificate Fingerprint, go to Application Type and select link "Learn More" next to Android
    - Open a terminal on your computer and type the following:
        - keytool -exportcert -alias androiddebugkey -keystore ~/.android/debug.keystore -list -v
        - Password: android
        - This will generate a fingerprint
        - Copy contents of SHA1 and paste in the form under Signing Certificate Fingerprint
- For package name, go to your project in Android Studio and select the package name from the manifest
- Press Create, you're now done and can move to the next step

- **Adding Code Library & Permissions:**
    - In Android Studio, go to the Android SDK Manager
    - Scroll down to the Extras section and make sure the following are installed:
        - Google Repository
        - Google Play Services
        - Android Support Library
        - Android Support Repository
    - Make they are all up to date, otherwise, download the updates
    - In your AndroidManifest.xml file add the following meta-data tag:

```
<meta-data
  android:name="com.google.android.gms.version"
  android:value="@integer/google_play_services_version" />
```

    - Inside your build.gradle file add the following dependency:
        - implementation "com.google.android.gms:play-services-plus:15.0.1"

- **Add 3rd Party Libraries:**

    - To add Picasso & Retrofit, add the following dependencies in your build.gradle file:
        - implementation 'com.squareup.picasso:picasso:2.71828'
        - implementation 'com.squareup.retrofit:retrofit:1.9.0'

- **Create the Data Model Classes:**
    - Read the Etsy API documentation @ https://www.etsy.com/developers/documentation
    - Select API Reference, then select Listing
    - Scroll down to Searching Listings, select method findAllListingActive to view guidelines on how to use this method
    - Now go to GettingStarted, select API Basics, and scroll down to Accessing the API
    - You will be making the call to the API at the following: https://openapi.etsy.com/v2/listings/active?includes=Shop,Imager&api_key=MY_KEY (insert your API KEY)
    - Insert above link into browser to view json data you will be working with
    - Create a class called ActiveListings to parse the data from the Etsy API
    - Create a class called Listing for the data objects
    - Create a class called Api, type should be Interface
    - Import the ActiveListings class you just created and the following retrofit classes
        - import retrofit.Callback;
        - import retrofit.http.GET;
        - import retrofit.http.Query;
    - Create the call & callback inside your Api class
    - Inside your Etsy class add a method that retrieves an instance of the Api
    - Inside theEtsy class set-up the API KEY

## Task 2: Implement UI for Each Activity and Fragment

*(List the subtasks. For example:*

- *Build UI for MainActivity*
- *Build UI for something else )*

**Using Recyclerview to create the UI**

- Go to your res folder and create a new class layout for your listing
- Change layout to LinearLayout
- Add an ImageView to display listing image and TextViews to display title/shop/price
- Add the following to your dependencies in the build.gradle file
    - implementation 'com.android.support:recyclerview-v7:27.1.1'

## Task 3: Your Next Task

Describe the next task. For example, “Implement Google Play Services,” or “Handle Error Cases,” or “Create Build Variant.”

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

## Task 4: Your Next Task

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

## Task 5: Your Next Task

Describe the next task. List the subtasks. For example:

- Create layout
- Something else

Add as many tasks as you need to complete your app.

---

**Submission Instructions**

- After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
    - Make sure the PDF is named “**Capstone_Stage1.pdf**”
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”

- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"