

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Main Screen](#)

[Slide Out Menu](#)

[Upload Menu](#)

[My Images](#)

[Viewing an image](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for main gallery](#)

[Task 3: Settings](#)

[Task 4: Image Uploading](#)

[Task 5: Implement 'My Images'](#)

[Task 6: Image Viewing](#)

[Task 7: Tablet View](#)

**GitHub Username:** `sc0tt`

# Upscoot

## Description

Upscoot is an image uploading and gallery application that allows users to upload images on their android device and share them easily among their friends.

## Intended User

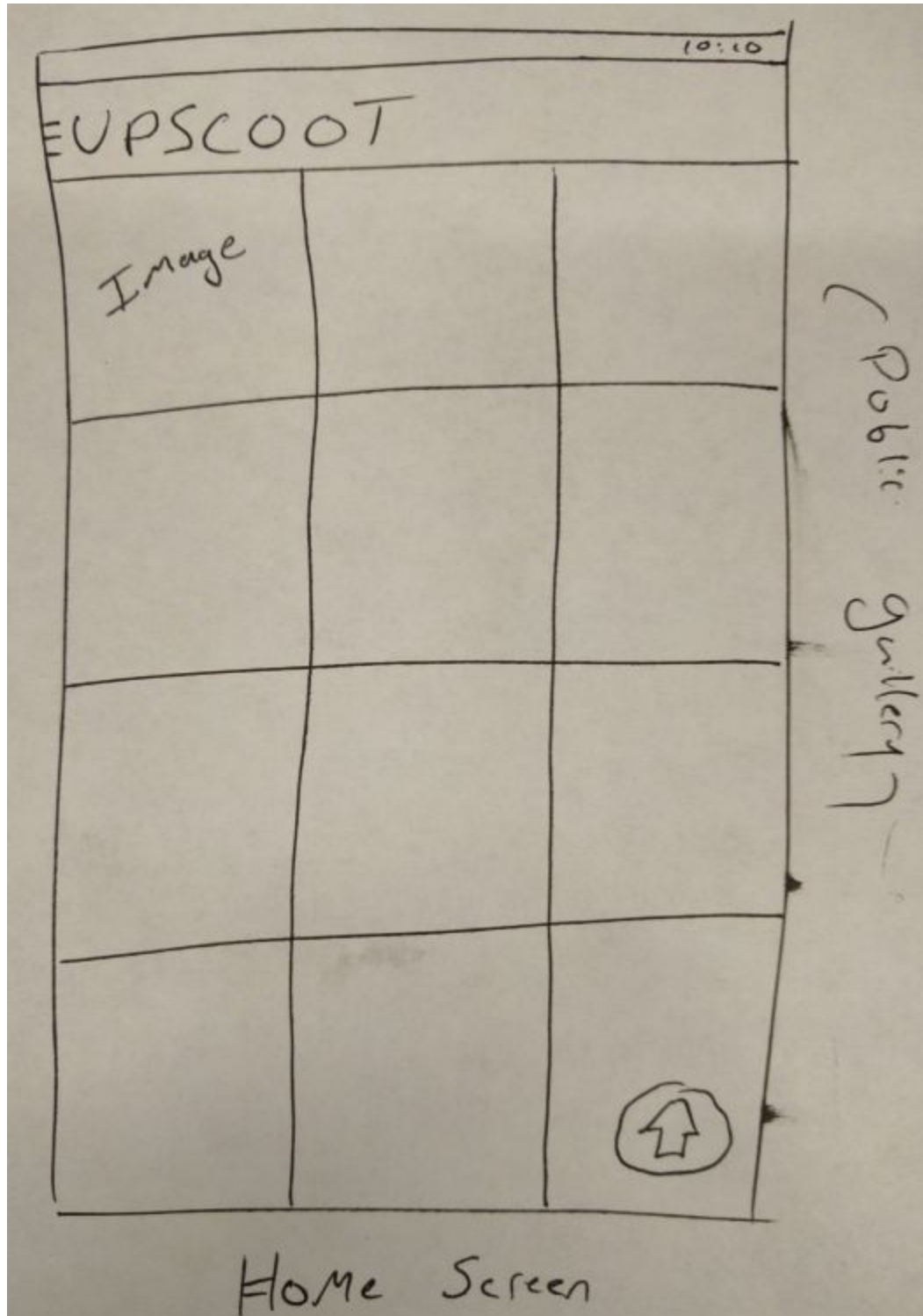
Upscoot is intended for groups of friends who find themselves sharing images they have taken or saved. Upscoot allows the user to publicly or privately upload images and provides them with the direct image URL to share with.

## Features

- Upload images directly to the Upscoot servers
- Upload privately, allowing only the uploader to know the image url
- Upload publicly, allowing anyone to view the image.
- Gallery containing all publicly uploaded images
- Upload via the share menu from any application or gallery

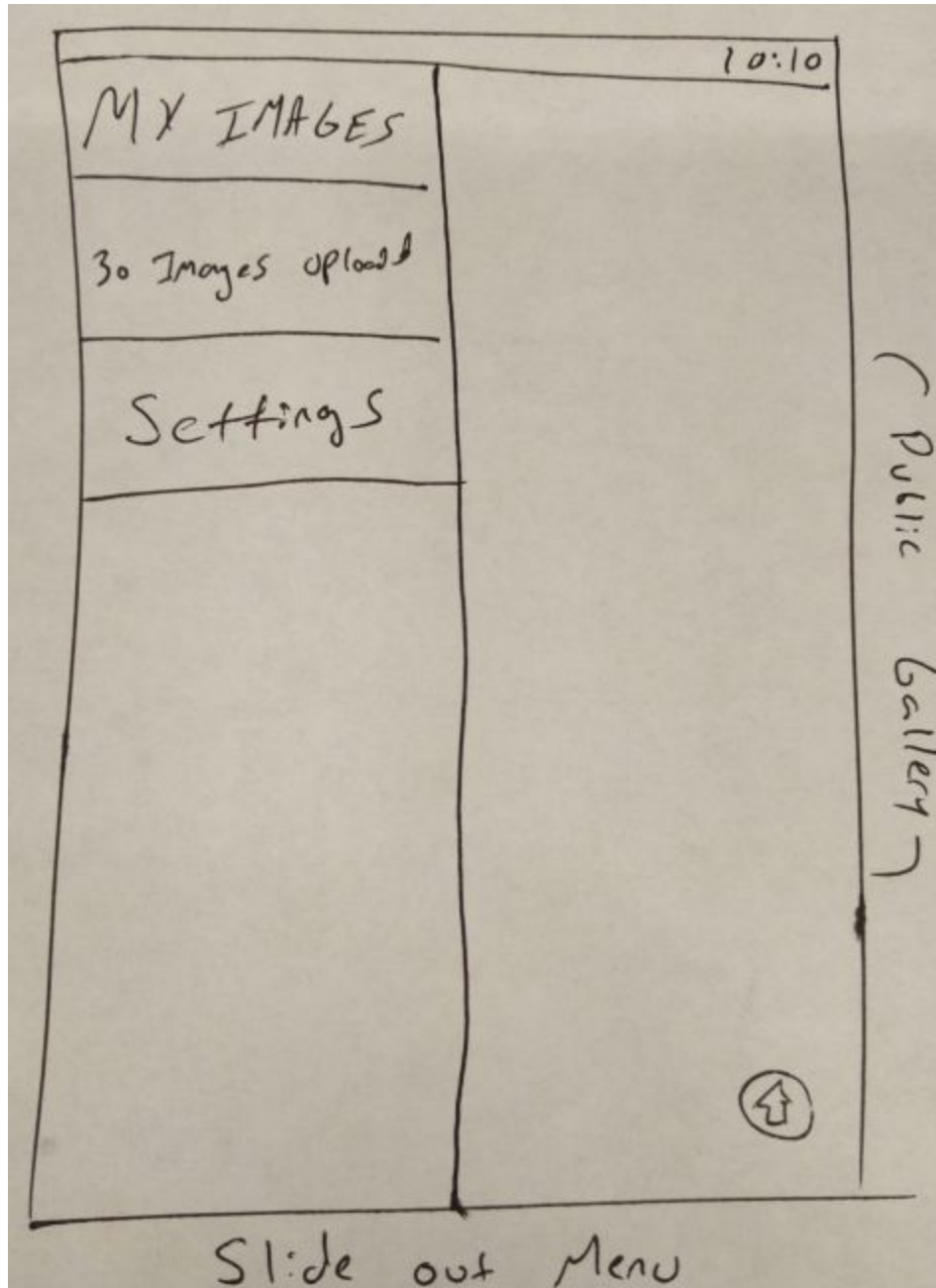
## User Interface Mocks

### Main Screen



This screen will display a public gallery of all images uploaded to the service that are public. Pressing an image will display that image in full view. A FAB on the bottom right will display the users gallery, allowing them to upload any image they choose. A slide out menu will also be available to display certain options for the user.

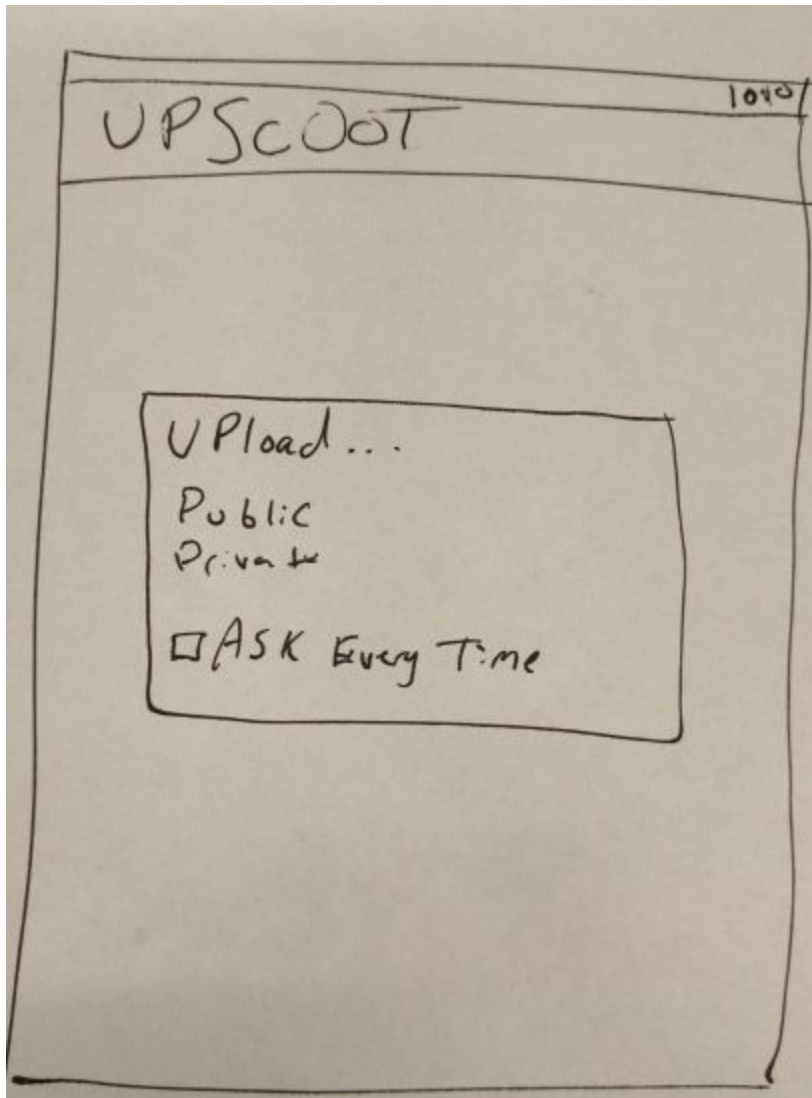
### Slide Out Menu



This menu will appear on top of the main gallery content, just as one would expect with a slide out menu. The menu will allow the user to see how many images they uploaded and adjust the

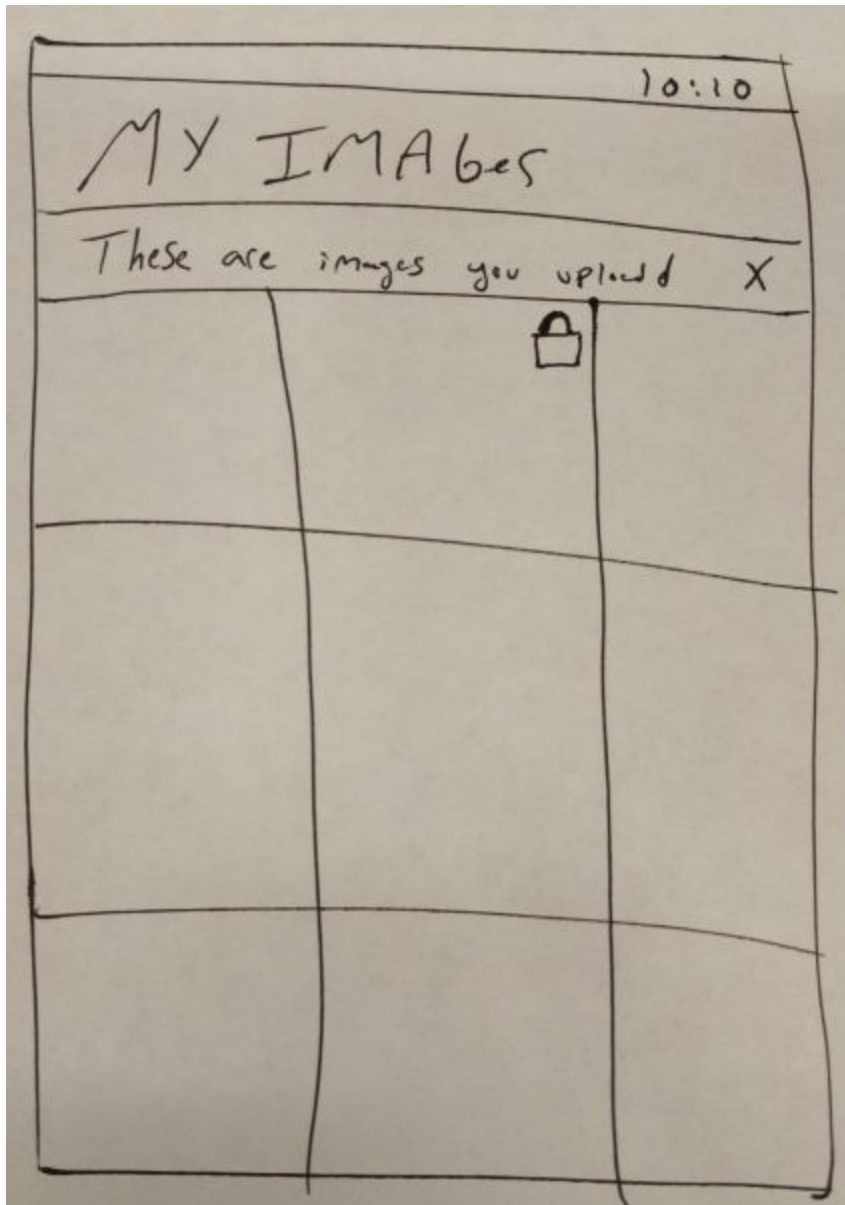
app settings. Clicking on 'n Images Uploaded' will bring the user to the screen to display all their uploaded images.

## Upload Menu



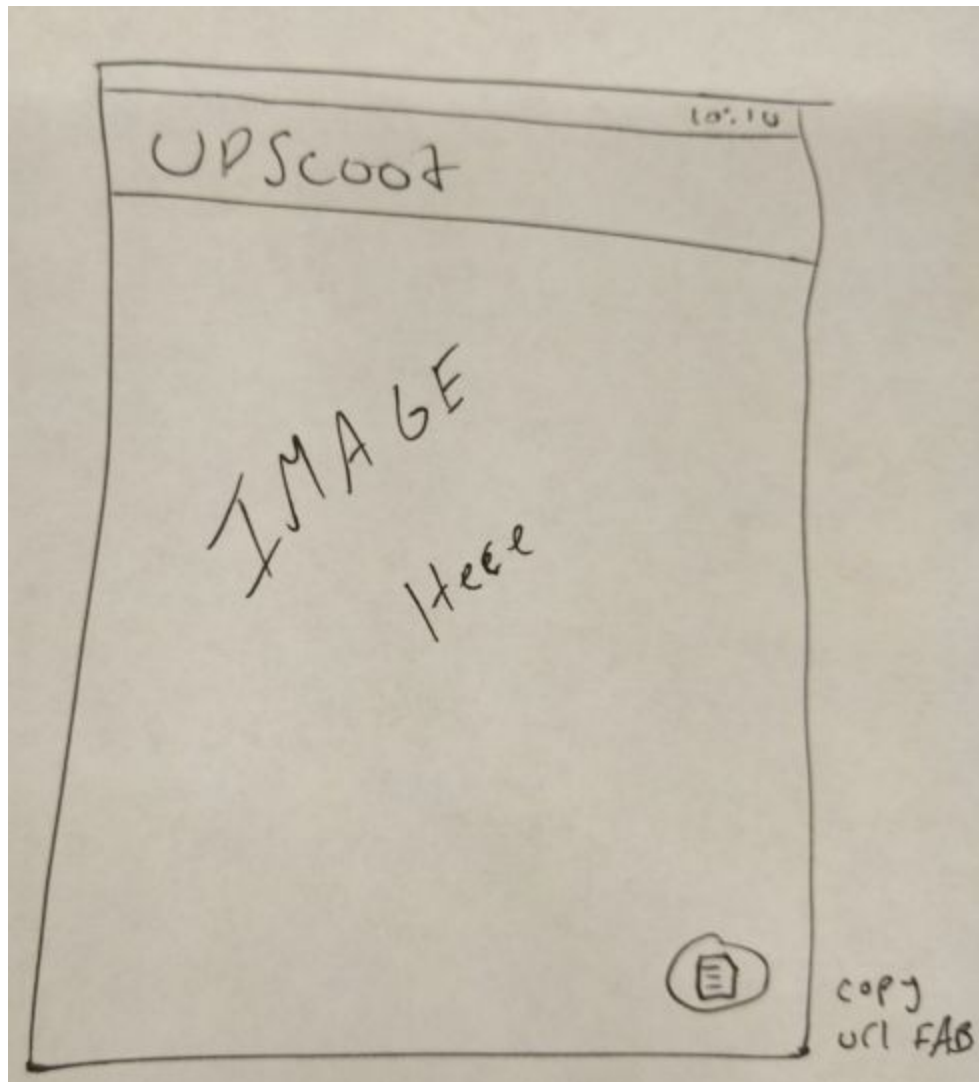
A user can choose to upload from within the app by clicking the FAB on the main screen or from the share menu of other apps within android. Both will present the user with a dialog box asking them if the image should be public or private, with a checkbox to ask every time. If the user unchecks 'Ask Every Time', it will use whichever option they selected for future uploads without a prompt. This will be configurable in the settings screen.

## My Images



This screen will display images that the user uploaded through the app. Clicking an image will bring it full screen. A lock icon will appear above certain images if they are considered private.

## Viewing an image



When a user selects an image to view in the app, it will appear full screen with a FAB displayed to allow them to copy the URL to the clipboard.

## Key Considerations

### How will your app handle data persistence?

Public gallery images will be stored server-side and the list of gallery images will be obtained from the app during launch. Images uploaded to the server from the user will be stored in a database to be kept track of. This will allow the section "My Images" to exist, as the server is intentionally programmed to keep the image uploads anonymous.

### **Describe any corner cases in the UX.**

If an image is unable to be loaded the app should not error out, but rather use a placeholder image that lets the user know there was a problem. Clicking on the errored image will attempt to re-download it.

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

Ion or Picasso will be used for the retrieval of images to simplify the process of displaying them.

## **Next Steps: Required Tasks**

### **Task 1: Project Setup**

- Configure android studio and git repository
- Initialize web server to accept image uploads.

### **Task 2: Implement UI for main gallery**

- Build UI for MainActivity
- Implement retrieval of server-side images to be displayed in the main activity

### **Task 3: Settings**

- Build a settings activity to store user preferences, such as upload type preference and auto-copying of uploaded image urls to clipboard

### **Task 4: Image Uploading**

- Implement android share intents so that the application will accept images to be shared to it
- Implement gallery permissions and use the new android permission model to request for permission if needed.
- Implement Dialog to handle the prompt for public or private image uploading
- Implement usage of stored settings when deciding to display the prompt or not
- Implement usage of stored settings when copying the url to the clipboard.



### **Task 5: Implement 'My Images'**

- Have a reference of uploaded images stored in a database to identify them.
- Display the uploaded images in a grid user interface similar to the main gallery

### **Task 6: Image Viewing**

- Build a user interface to display images full screen
- Implement tap-to-zoom which allows the user to zoom and pan the image
- Implement floating action button to store the images url to the clipboard
- 

### **Task 7: Tablet View**

- Implement a master-details view for the main gallery to better support tablet usage.