Capstone Stage 1 Outline

GitHub Username: Ashitakalax

Scheduled Time Lapse

Description

Time lapse photography is a wonder way to see the world in a different perspective. This app has a specific purpose that other time lapse apps don't offer. It will take pictures at specific times, e.g. 8:00am everyday to see the sun rise. Or a picture every 10 mins of a sunflower slowly moving toward the sun.

Intended User

It is clear that this app isn't intended for users who use their phone every day, but for users who has an older existing phone that is just lying around collecting dust. This app will put those old phones to good use, creating long term time lapse photography. But this app isn't limited to just that. You can schedule photos from 10 seconds apart(possibly longer, need experience), to weeks apart.

Features

- Schedule starting time and the frequency of taking the picture
- Setup the app to awake from a sleep state to take the picture without user intervention(much like an alarm clock)
- Store the photos

User Interface Mocks

There are 2 ways to view this project, you can review the project from the snapshots below. You can also download the Pixate project(which also has transitions and key press interactions integrated into the ui mockups.

You can download a copy of the pixate project here.

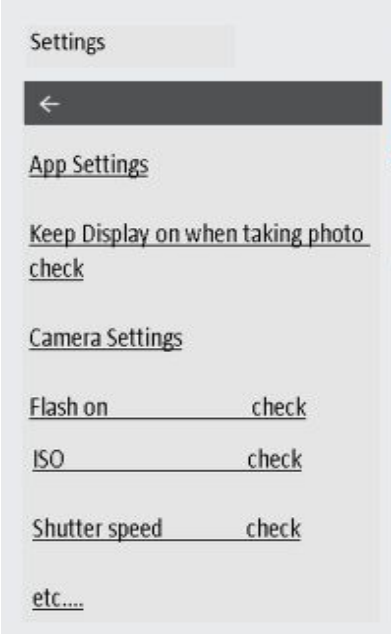
Screen 1

This is the home menu, it will show the last image taken by the project. Select the menu, or edit/view an existing project.

Screen 2

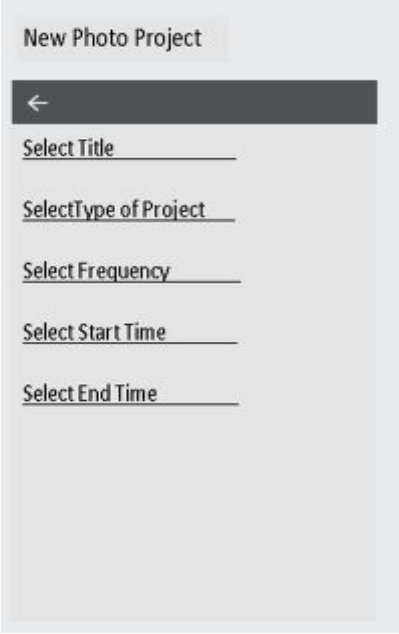
This screen is the side menu to allow users to quickly create a new photo project(which now looking at it This button may transition to a FAB). It will allow users to view system settings, and view the User manual for tips and tricks on how to use this app.

Screen 3



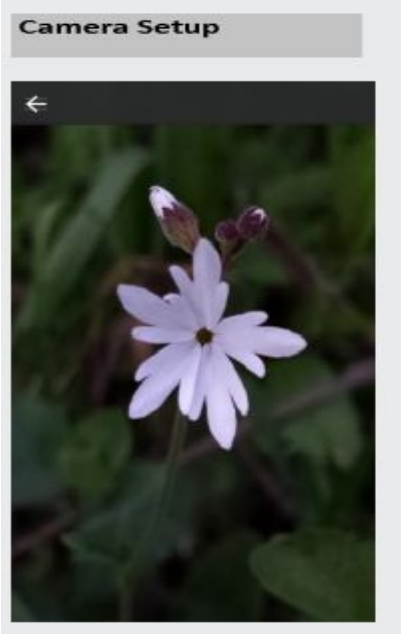
This allows users to set the camera settings.

Screen 4



This will allow users to control when, and how frequent, photos are taken.

Screen 5



This screen will allow users to lineup there camera with whatever they are taking a picture of.

Screen 6

Instructions on how to use this app, and also cool tips and tricks.

Key Considerations

How will your app handle data persistence?

The app will have to store multiple pairs of starting time and frequency(in seconds apart)
When the picture is done, it will store it just like a regular camera.

Describe any corner cases in the UX.

This app almost behaves as if it is two separate apps. One app for setup and configurations, and the second for when the app wakes up for taking the picture.

Part of the setup will have preview what the picture is going to take, allowing the user to see the frame of what will be taken in their absence.

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

I'll be using Picasso for handling the preview of the images(when image is selected it will start an intent to the default image viewer tool)

Next Steps: Required Tasks

Task 1: Project Setup

Setup the Android project to allow easier development for the both Tablets and phones. Setup the unit testing structure.

Task 2: SQLite data storage

Implement the Projects, and camera settings in the database.

Task 3: Implement UI for Each Activity and Fragment

Implement the activities, and fragments for each UI mock page. These pages will need to be polished later, this first draft is just to display the data that is needed for the project.

Task 4: Implement Scheduled Camera Triggering

The timing of the Camera is a key component to this app. It must be accurate and a top priority. If the camera is lined up with to take a picture of the moon just above a building. It needs to occur at that exact moment.

- Implement a Top Priority scheduler to execute the picture being taken
- Validate that the picture will be taken even if the phone is in different states.

    - Another app is open
    - Locked

Task 5: Camera Event Content Provider

To provide a way for other apps to work with this app, we will have a content provider for the objects that trigger the events. This way people would be able to make an addon app that would allow scheduling based on unique events(e.g. Moon Scheduling App that schedules based on moon timing).

Task 6: Implement Loader for Content Provider

To provide a way for other apps to work with this app, we will have a content provider for the objects that trigger the events. This way people would be able to make an addon app that would allow scheduling based on unique events(e.g. Moon Scheduling App that schedules based on moon timing).

Task 7: Implement Camera Screen

This screen will allow the user to line their camera up with what they are taking a picture of.

Task 8: Implement User Manual API

This will fetch the JSON data from an online source. It will need to do this in an AsyncTask in the background.

Task 9: Implement Upcoming Scheduled Events Widget

In order for users to be comforted knowing when the next picture will be taken, you view the widget on the home screen. It will show a list of upcoming events.

Task 10: Implement Google API's

The following will allow me fulfil the Google services requirement for the project.

- Implement Google Admob
- Implement Google Analytics to find which features users will be using

Task 11: Implement Transitions

Implement transitions between each fragment and activity.

Task 12: Polish UI

Touch up the UI to make sure everything fits well, and all the data is structured in the best way possible.