GitHub Username: curveo

Software Engineer to do's

Description

Software Engineers have a lot of tasks to track and are required to multitask a multitude of priorities and while there are many ways to digitally track ones to do's there isn't one that is centric to the job of a software engineer. This App gives you the ability to create a to do task as you would think but also gives you the ability to add snippets of code that are contextual to the given task. Your efficiency and organization will no doubt increase when you have this app installed on your Android Device!

Intended User

This app is intended for use by Software Engineers of all kind that are looking for a quick and easy way to track one's priorities.

Features

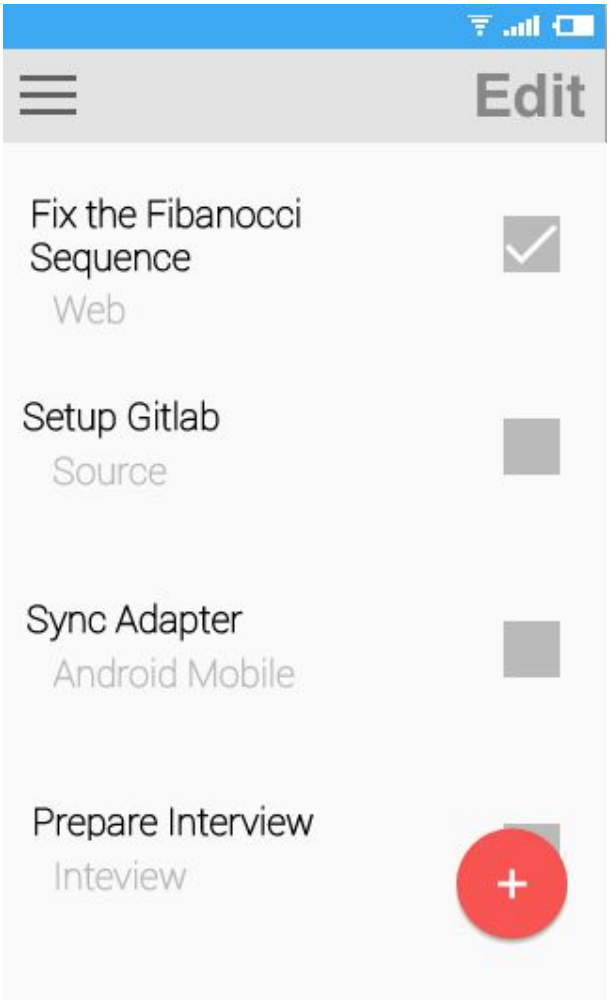
Primary features include:

- Create Simple Tasks
- Checkbox to mark task complete
- Synchronized with the server
- Add code or other snippets of information

User Interface Mocks

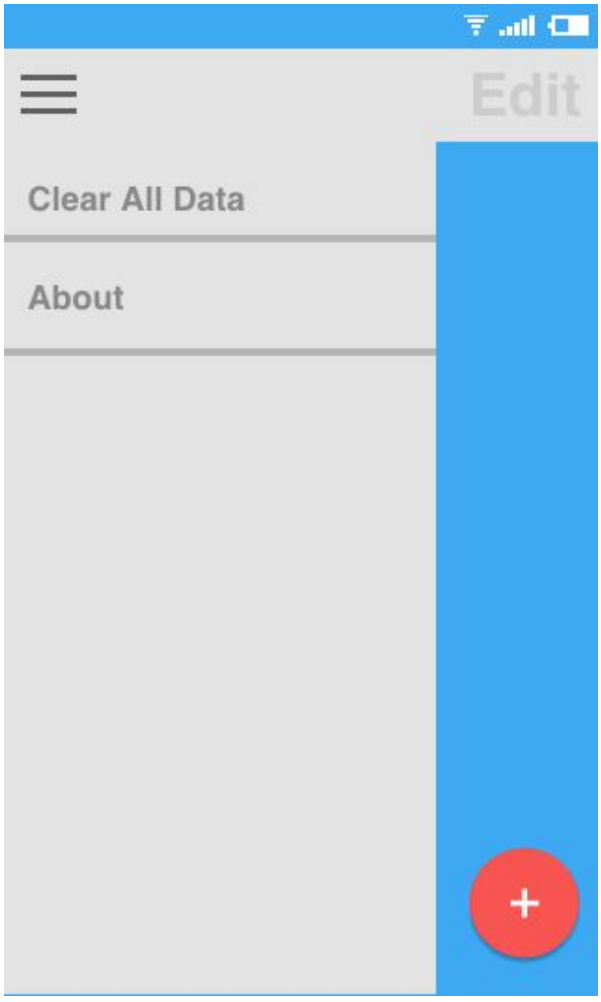
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1 - Start Screen

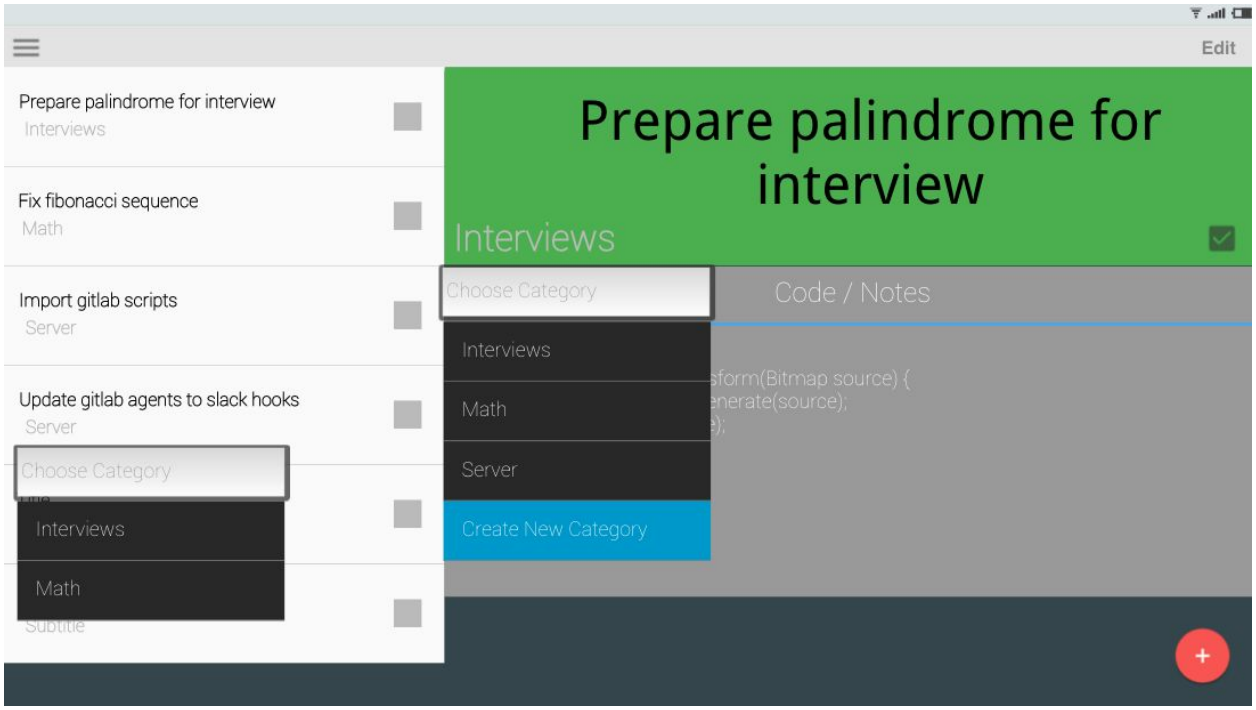


User is presented with the current to do's on their list and ability to add, mark done, categorize, or view the detail of.

Features include a menu to access the Settings and Floating Action button to add a new to do.

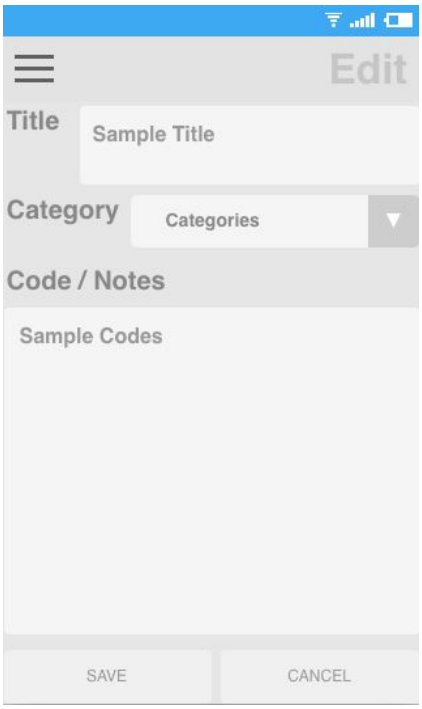


The left drawer will be used to clear all data or view the about page that contains the version information.



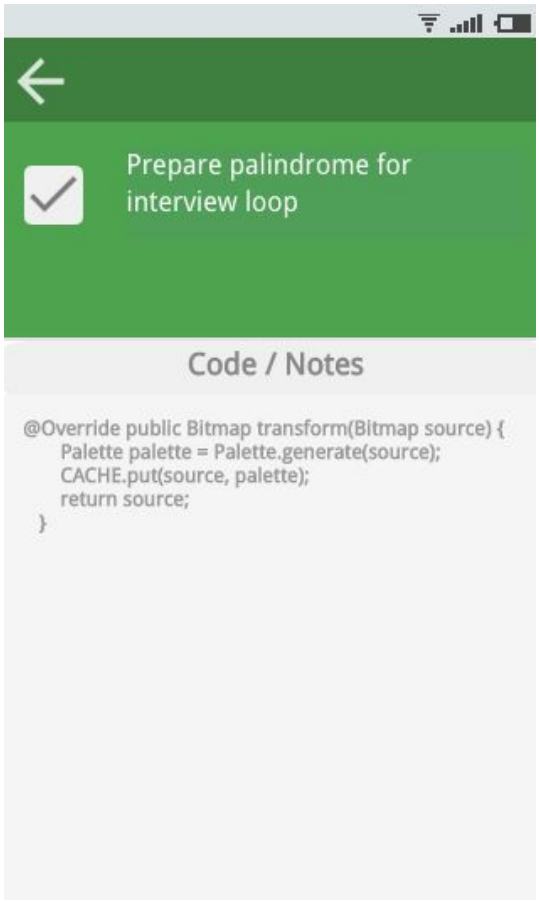
Tablet variant of the Start Screen. In this view the user is able to see the list of current todo's and take advantage of the multicolumn layout to view the details of a to do simultaneously.

Screen 2 - Add To Do Screen



This screen is for creating a new to do. The code / notes section is optional.

Screen 3 - Details View



This screen allows the user to view the full details of the to do including code / notes added when the to do was created.

Key Considerations

How will your app handle data persistence?

This app will create a custom content provider to cache the data. It will also sync with a simple web service that uses the UUID of the endpoint to backup the data into the cloud.

Describe any corner cases in the UX.

To do status corner case:

1. Toggling the status of a particular to do: User will be able to access completed tasks by scrolling to the very bottom of the to do's list.
2. User will be able to edit the text in the detail view for Title, Category, and notes
3. User will be warned that clearing data will also erase all stored cloud data before clearing
4. User will not lose any data entered into fields on orientation changes.
5. Back button will work as expected with the backstack and will return to Launcher screen if at the list activity.

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

This app uses the MaterialEditText library for enhanced textview look and feel for all input fields. It also uses Google AdMob and Google Analytics to analyse how the app is used. The following gradle libraries will be needed:

- `compile 'com.rengwuxian.materialedittext:library:2.1.4'`
- `compile 'com.google.android.gms:play-services-ads:8.4.0'`
- `compile 'com.google.android.gms:play-services-analytics:8.4.0'`

Next Steps: Required Tasks

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

Task 1: Define Data Schema and Backend

Setup Lamp (Linux, Apache, Mysql, PHP) service

- Amazon S3 instance or Local hosting over http

Define the data model scheme

- Define the Data Objects in UML (I like to use LucidChart Diagrams)
- Create script to generate database structure for backend mysql
    - Database name: todos

Configure Apache for host todos.curveo.net

Setup mysql on server and execute script created above to create empty database

Create php project

- Initialise empty git repo gitlab.curveo.net
- Build out API entry points for GET, PUT, and DELETE methods
- Implement API call functions to mysql
- Build unit test for coverage of each API

Task 2: Setup Android Project

Create an empty git repo for the directory where the project will be stored
Create new Android Project using Studio new project wizard
Setup the gradle file to include the following compiles:

- Google Admob
- Google Analytics
- MaterialEditText

Task 3: Build data models and service connections

This task is to integrate with the server and build out the data models.

- Define how to generate UUID for device (in lieu of login accounts)
- Integrate data model schemas (refer to UML)
- Create service connection class that maintains sync with the server
- Define sync strategy

Task 4: Create Activity classes for UI

Build out the following classes that xml:

- Create To Do's List activity
- Create new To Do activity
- Create Detail View Activity

Task 5: Add Google Services

Build out the following google services:

- Admob
- Google analytics

Task 6: Create tablet UI

Create a custom 2 column layout for tablets that includes the list of to do's and individual to do detail view.