

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[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.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** OniiChanStudio

# LaunchPadify

## Description

The LaunchPad Simulator lets users create, record and share music with their friends and community or any other social network. User can select a music preset of selected sounds from the menu, or they can just create their own. Login with any of your social network and save your creations to the drive directly.

## Intended User

Music enthusiasts and/or people trying to make music with their mobile phone. Play music on the go.

## Features

Main features of the LaunchPad Simulator include

- Social Network Login
- Create Music with the simplest UI
- Record Music
- Share Music on most of the Social Apps
- Save your creations to the drive directly
- Choose from a lot of presets of music
- Or create your own preset with a real Recording feature

## User Interface Mocks

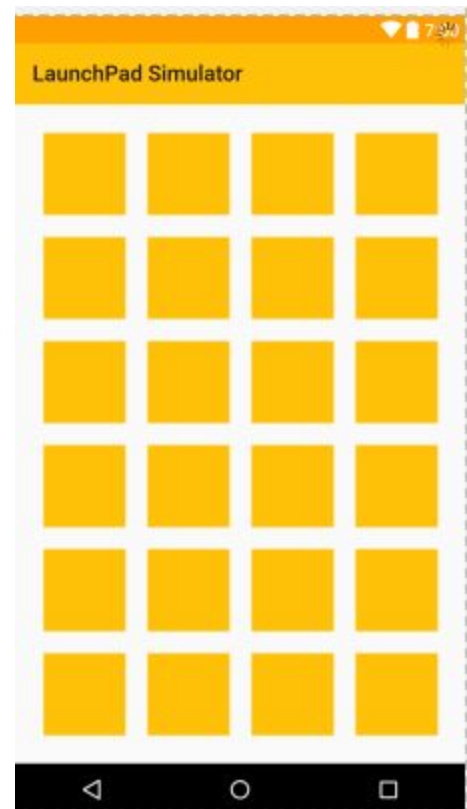
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

### Screen 1

This screen of the app shows a set of keys  
Pressing the keys will produce a sound.

You will also be able to define the number of keys you want  
In the app.

The best part will be that with this you can define any sounds  
to your keys, directly from your microphone or you can use a  
file present on your SD card.



## Key Considerations

How will your app handle data persistence?

The app will be connected to a TBA service, which is an an existing content provider.

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

Any back button presses twice on an Activity other than the Launchpad main activity will return them to the main activity. Pressing back button twice on the main screen will exit the app.

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

TBA

**Describe how you will implement Google Play Services.**

TBA

## 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: 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
- Watch Dayum tuts for what I can't do right now and everything else
- Planning and Sketching out all the activities
- Earlier there was stuff like, Recording, Sharing, Saving to Drive, Changing the layout itself. If possible, and adding sounds as they please (This feature will be epic if I could pull it off)
- Think about each of the above
- Lets think now
- Watch tutorials for Recording in-app sounds and audio
- Implement that in the app
- Watch tutorials for Sharing with almost all social networks
- Again Implement that into the app

- Watch a piano making tutorial and figure out what I'm gonna do for the actual main layout
- Design the main layout and code the app with the default clap sounds or anything i can find.
- Now record and sharing Test with it
- Actually this should have been done first, Oh well
- Now the recording, where to save it.
- Save that to drive
- Ah, almost forgot, Login with social networks is also needed. But can be added later,
- Or it'll slow down the testing time.
- I know I can always replace the launcher activity
- NEXT
- Changing the layout itself
- Yeah, dream feature, if the user wants to add 15 buttons or 3 buttons, its up to them
- If I can just generate buttons procedurally according to an input number
- Then go and ask to assign audio to each one of them. It'll be real awesome.
- Fukin buy a play store dev a/c
- Publish the app as a paid one without ads at all. (Everybody does it, don't they ?)
- Also publish a free alternative with ads
- I can also put a few in app purchases like yeah a few different presets of music.
- Hey I am gonna need tutorials for the app itself, like how to start playing and have a demo play also. Now i am trying to think big, I will do this when the app itself is complete.
- Fukin save presets also.
- Fukin make the buttons glow as long as they are playin or being touched !
- Let users download the fukin presets with sounds and also apply in app purchases to a few of awesome presets. Who cares anyway. What presets will anybody care for ? Synthesizers maybe. For fun, lets add LostPause preset and other anime friends presets too.
- Ah just thought of it, lets put that Singup activity or login whatever on hold. I mean users can skip em, But they'll still should be able to download presets. Thats noice.

Divesh or Dipak are not prime helpers though any help is appreciated.. I'll keep on adding more dicks if I can take em all.

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

Sub tasks will be mainly fukin everything other than the naming

- Build UI for MainActivity
- Build UI for the RecyclerView: das gonna have a a list view
- Lets define the Recycler a little
  - It'll have a top part with the user info (obvious) less make it smaller

- Recent presets list (Actually I'll have two of the recent presets there.
- Which means a new headache.
- I'll have to think up how to search about to save the last two or three used presets and always update them on the recent list
- I just thought of making the recent part a little down arrow ^ kind of thing, that'll save up space and can have a list of recents into that. No idea how to do that as of now.
- Now design a few layouts for the actual presets menu. That'll be only one to look out for, will be easy I guess. To design and will be a dick in the ass while implementing. #NOHOMO
- Okay, jokes aside, now I'll have to think of how to display shit, where I am going to put In app purchases.
- Basically making UI will be second nature to the fuking dick list above.
- Even then, The new preset menu, which will select how many keys and how many fukin add sounds to it. Layout for all of them, If i can figure out how that shit works.

### Task 3: Your Next Task

- Make the UI responsive for every device you can think of .
- OK not really every device, but generic tabs and mobile phones.

### Task 4: Your Next Task

- TBA

### Task 5: Your Next Task

- Publish the App and fukin Earn Dollars

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