

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen Main](#)

[Screen Main-> Baking Time \(overview\)](#)

[Screen Baking Time \(overview\) -> Baking Time \(master\)](#)

[Screen Baking Time \(master\) -> Baking Time \(detail\)](#)

[Widget](#)

[Screen Main-> Popular Movies \(master\)](#)

[Screen Popular Movies \(master\) -> Popular Movies \(detail\)](#)

[Screen Main-> Sandwich Club \(master\)](#)

[Screen Sandwich Club \(master\)->Sandwich Club \(detail\)](#)

[Key Considerations](#)

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

[Describe any edge or 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 or other external services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

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

[Task 3: Make layouts](#)

[Task 4: Polish](#)

[Task 5: Test](#)

**GitHub Username:** [pacheco](#)

## Pacheco's Portfolio

### Description

Pacheco's Portfolio is a compilation of some smaller projects to show off my skillz to the world. So if you need an Android Developer, check this out and then hire me!

## Intended User

People searching for an Android Developer like recruiters or private persons.

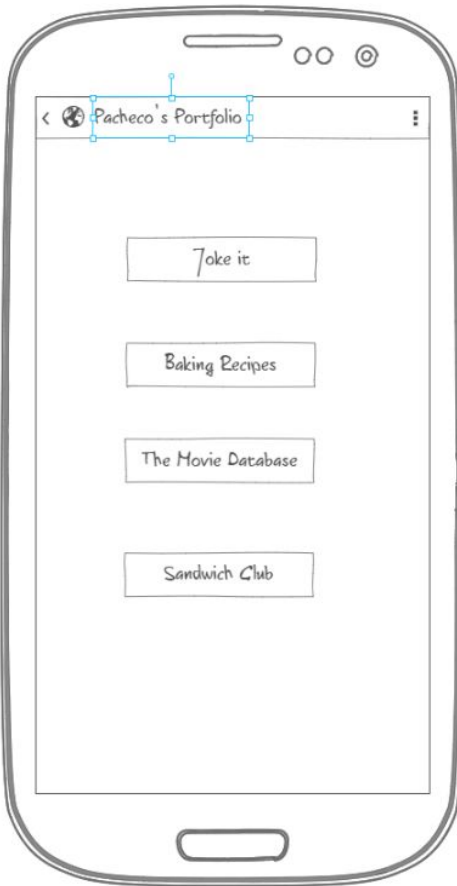
## Features

- Saves information of preferred movies
- Connects to internet for actual movies and recipes
- Uses a Java library
- Uses a firebase
- Tells jokes
- Shows actual movies

## 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](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

## Screen Main



This is the first screen, where you have an overview off the apps. The buttons will be in a RecyclerView, due to adding other apps. (Not required: The buttons may be cards with an image and a little description)

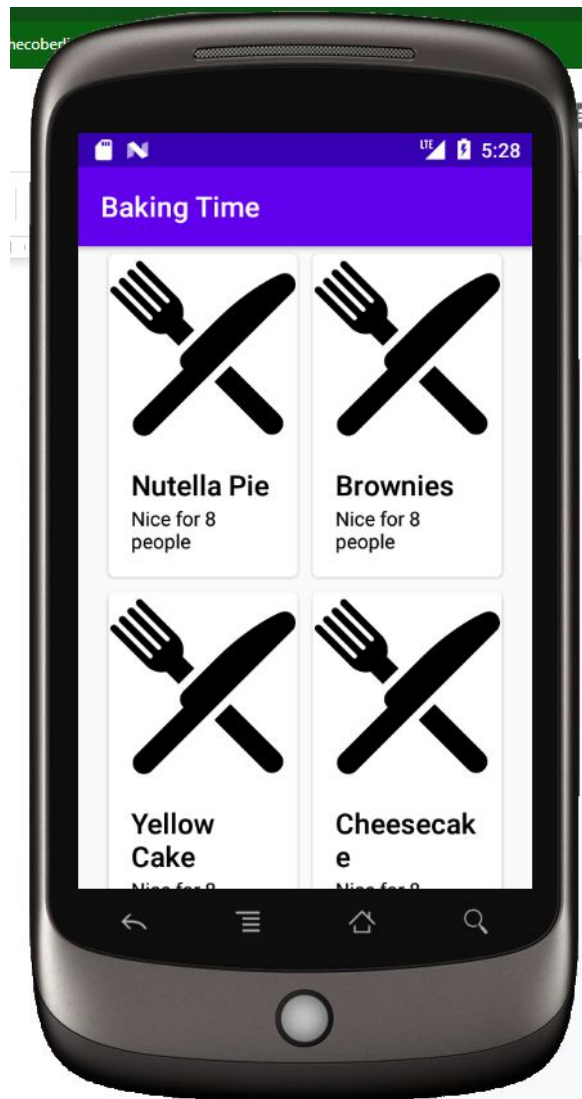
Button "Joke it" only makes a Toast.

Button "Baking Recipes" opens Screen Baking Time.

Button "The Movie Database" opens Screen Popular Movies (master)

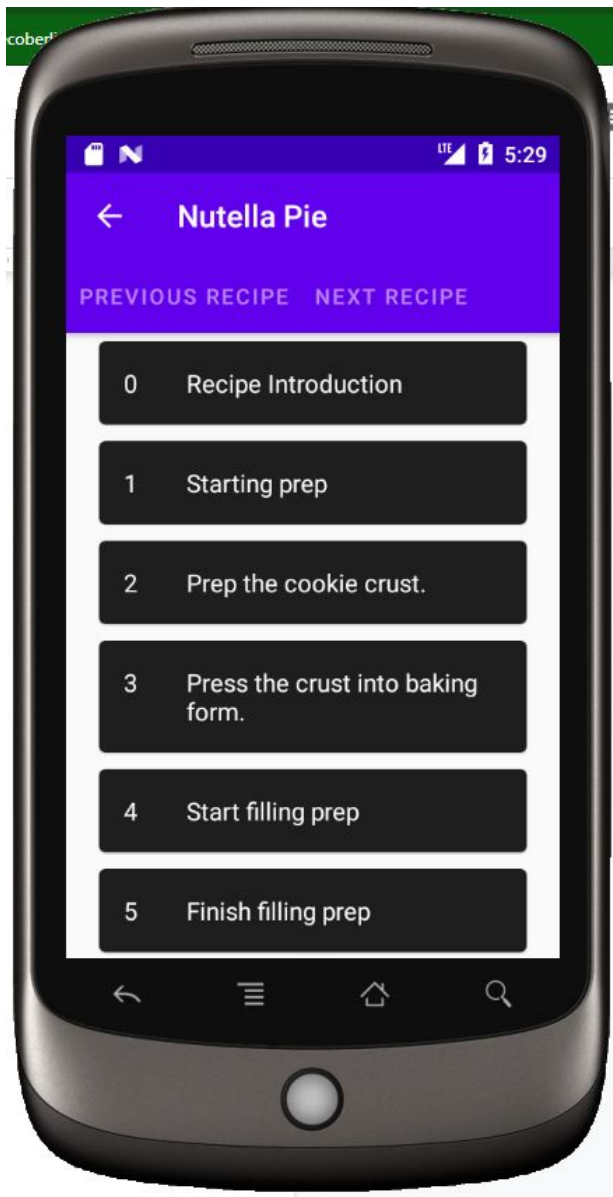
Button "Sandwich Club" opens screen Sandwich Club (master)

## Screen Main-> Baking Time (overview)



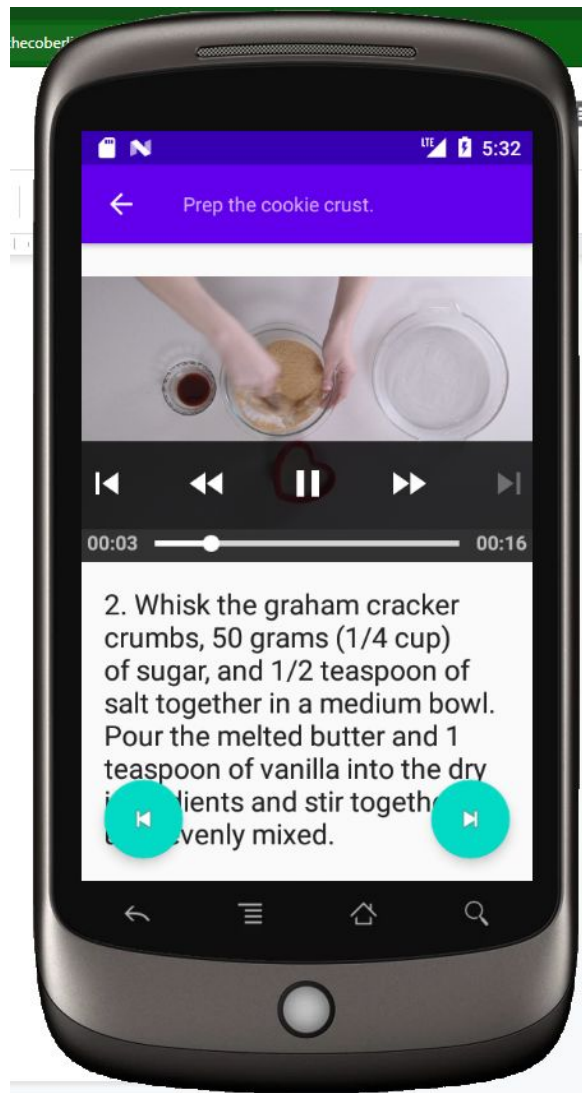
Overview of available recipes clicking on opens Screen Baking Time (master)

## Screen Baking Time (overview) -> Baking Time (master)



Clicking on next or previous recipe shows the steps of the according recipe.  
Clicking on a step opens Screen Baking Time (detail)

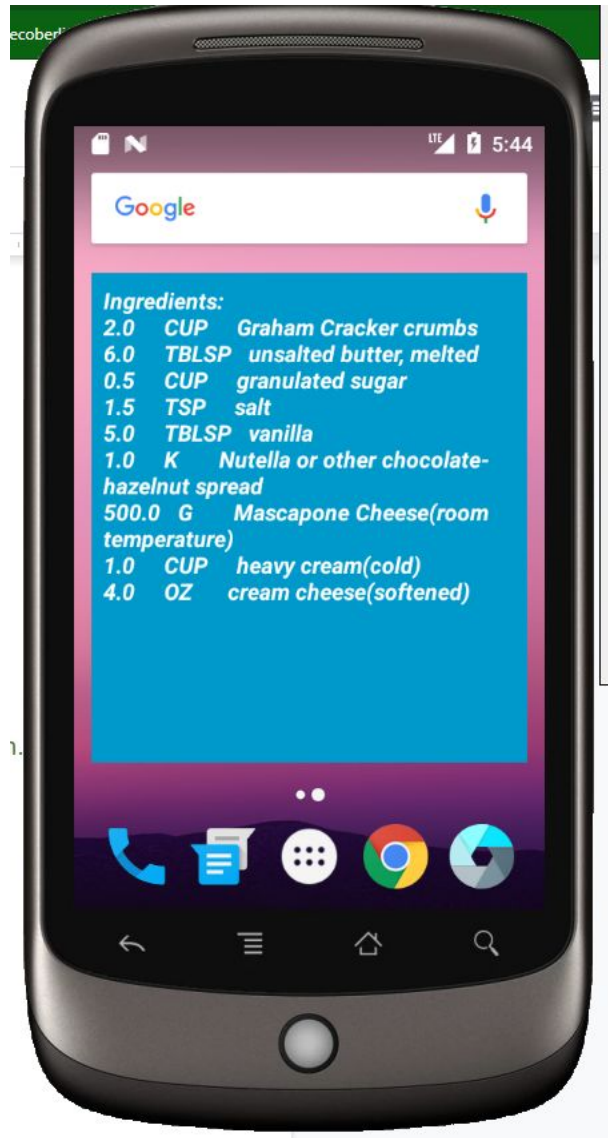
## Screen Baking Time (master) -> Baking Time (detail)



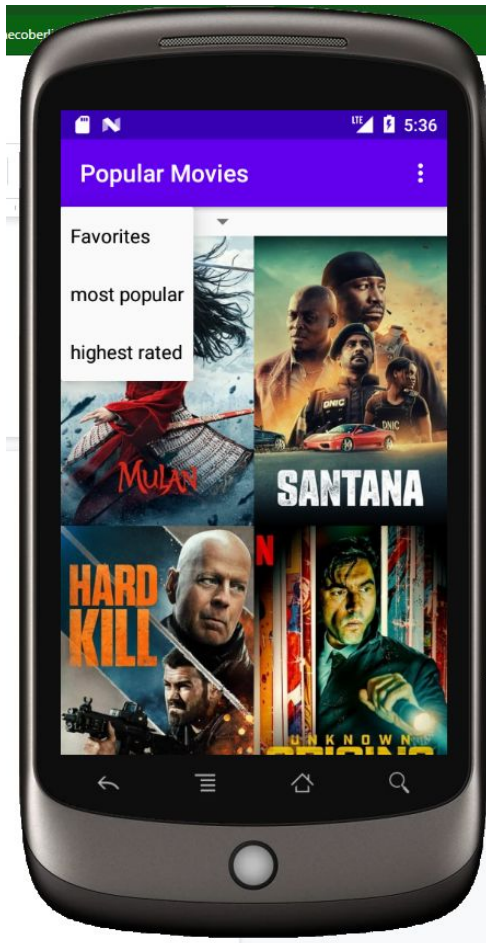
If a video is available its shown and have default controls. The FAB's goes through the available steps. In landscape video is full screen.

## Widget

Is not a screen and is only for Baking time, but there it is and shows the ingredients of a recipe chosen in the app:



## Screen Main-> Popular Movies (master)



Posters of the movies are shown and clickable. Three lists are available: favorites, most popular and highest rated. Clicking on a movie brings up the Screen Popular Movies (detail).

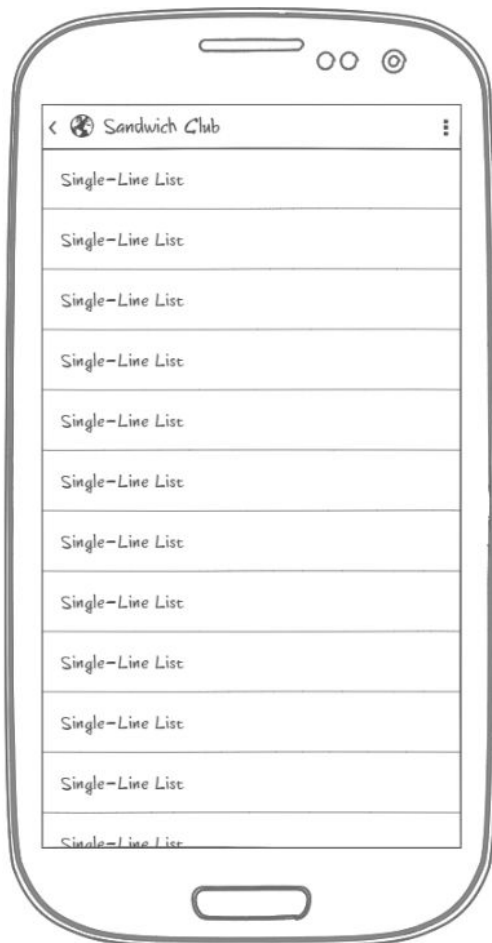


## Screen Popular Movies (master) -&gt; Popular Movies (detail)



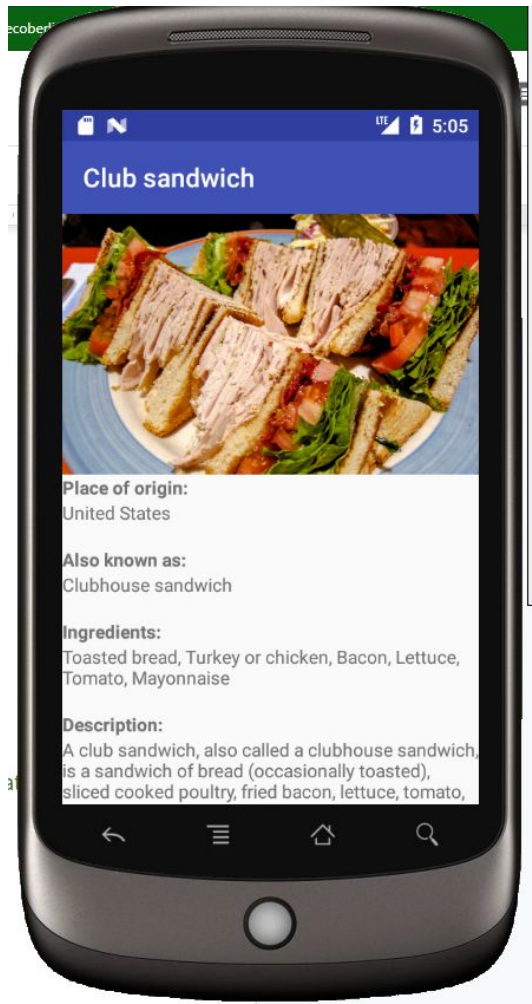
This details screen shows a little poster, some movie details, the movie can be marked as favorite, trailers are listed and youtube or the browser will start to show them and reviews are listed.

## Screen Main-> Sandwich Club (master)



A list of provided Sandwiches. Clicking on an item opens the Sandwich(Detail) screen.

## Screen Sandwich Club (master)->Sandwich Club (detail)



## Key Considerations

How will your app handle data persistence?

Room, Viewmodel, LiveData

Describe any edge or corner cases in the UX.

In Baking time(detail), when going back depending from which Step is returned, the list is scrolled to this position.

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

Picasso, handles the loading in background and has good placeholder and error options.  
Retrofit, handles REST calls in background and was several times suggested by reviewers (which hopefully know what they are doing 😊 ).

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

Ads are displayed on the main screen. Firebase, i do not know how it works, right now, but it will provide the jokes hopefully... because appengine endpoints is deprecated.

## 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

- Check how to setup firebase
- Set it up
- Think on how to bring all the single projects together, one multiproject or import them to one project only

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

- Build UI for MainActivity
- Make fragments for the master detail flows to support tablet in landscape mode
- Wire all up

### Task 3: Make layouts

Create Layouts for master detail flow to support tablets in landscape mode

### Task 4: Polish

Check material guidelines on:

- Spacing
- Elevation
- Colouring

And apply it.

## **Task 5: Test**

Make some (“connected”, does somebody say it like this, nowadays?) UI Tests with Espresso