How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

GitHub Username: kunotatemaki

Cooking with Cookeo

Description

Categorize your Moulinex Cookeo recipe collection. Search through all recipes by types (main courses, starters, desserts… even vegetarian food!). You will be able to create your own recipes and share them with all Cookeo lovers.
Bon Appetite!

Intended User

All people who love cooking and own the Moulinex Cookeo cooking robot.

Features

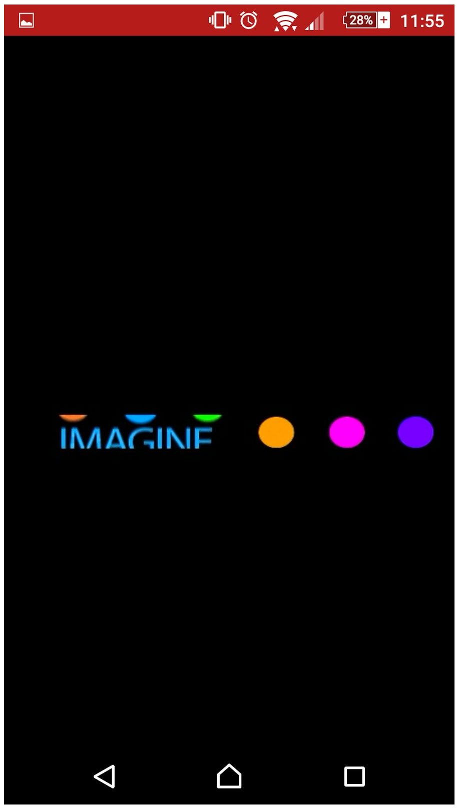
List the main features of your app. For example:

- Saves information
- Takes and crops pictures
- Receives Push Notifications
- Downloads content from server
- Manage internal and external memory
- Uses Loaders and content providers
- Plays Video as an introduction
- Applies Material design guidelines
- Uses latest SDK tools, as RecyclerViews, CardViews, Floating Action Buttons, swipe refresh layout…
- Uses SQlite database
- Uses shared preferences
- Uses broadcast receivers locally

User Interface Mocks

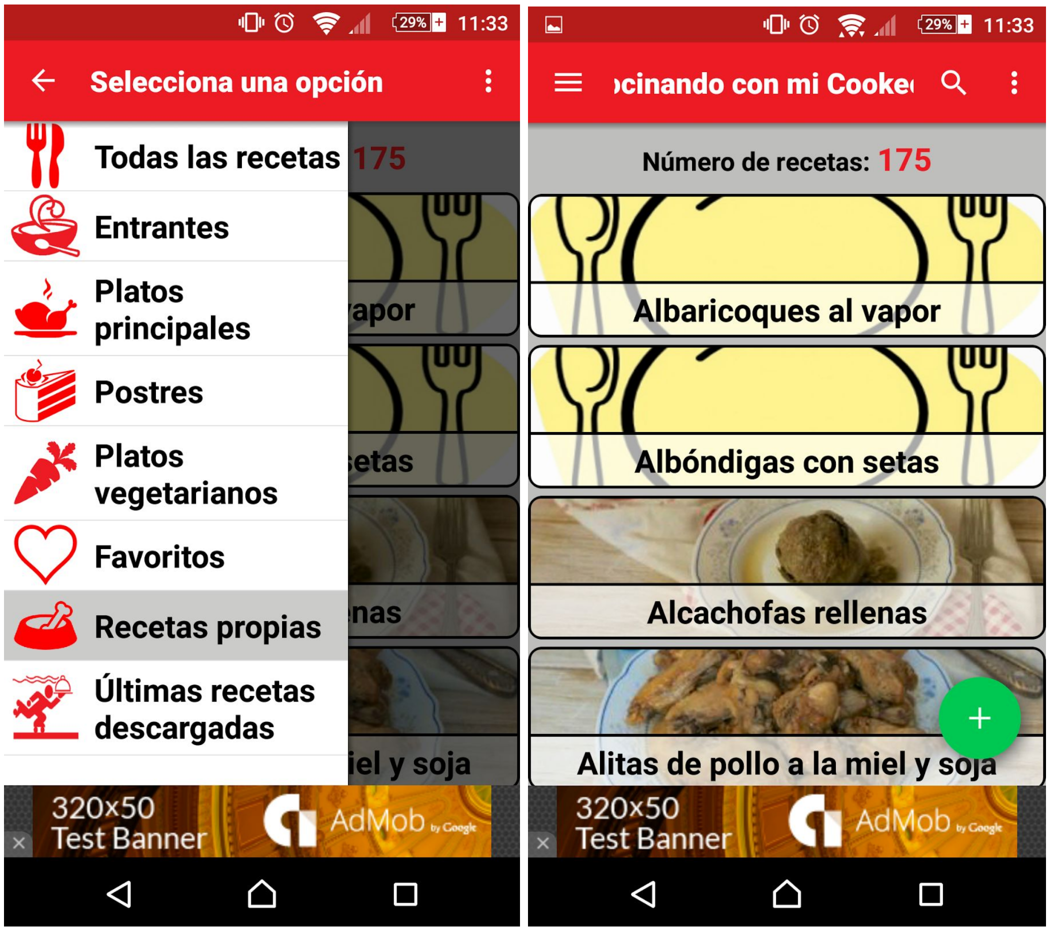
The app is based in a previous one I developed before taking this Nanodegree to teach myself about Android Development. The mocks are taken from that app. They will be redesigned according to the Material design guidelines, with material styling and new required components (FAB, loaders, content providers, recyclerViews, Cardviews…). I show them here because the concept, number and distribution of views are going to be pretty similar. Although the screenshots are in spanish, the app will be available in both spanish and english.

Screen 1



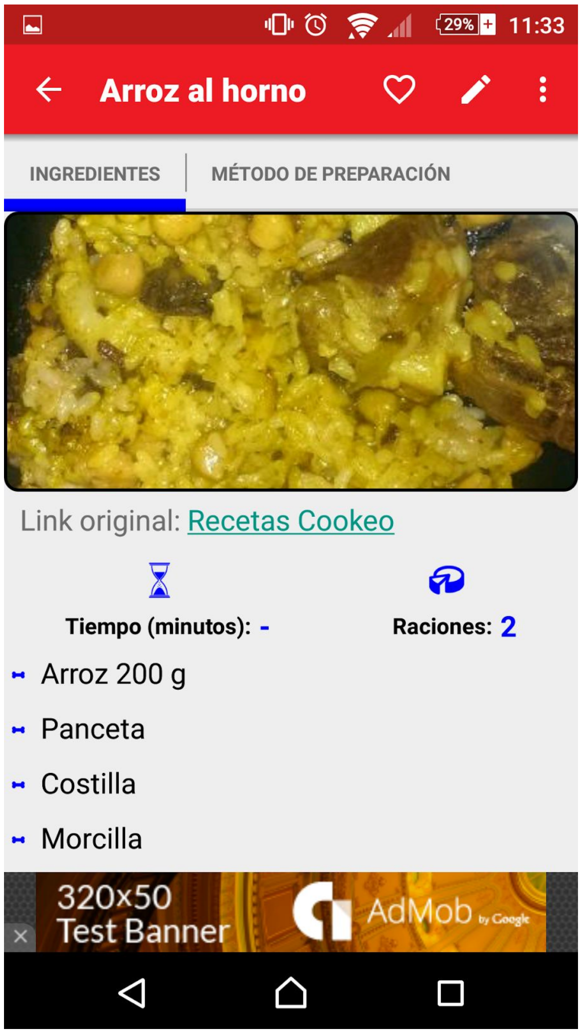
Intro activity. It will be shown only once, when application is loaded (not in memory). Is a videoplayer which plays an intro video of 5 seconds. This full screen activity won't allow back button.

Screen 2



Main activity: It will show all recipes. A search icon will allow you to search a specific recipe by name, and the navigation drawer will allow you to show recipes by type (main, starters, desserts…). The floating action button will start a new activity for creating your own recipe.

Screen 3

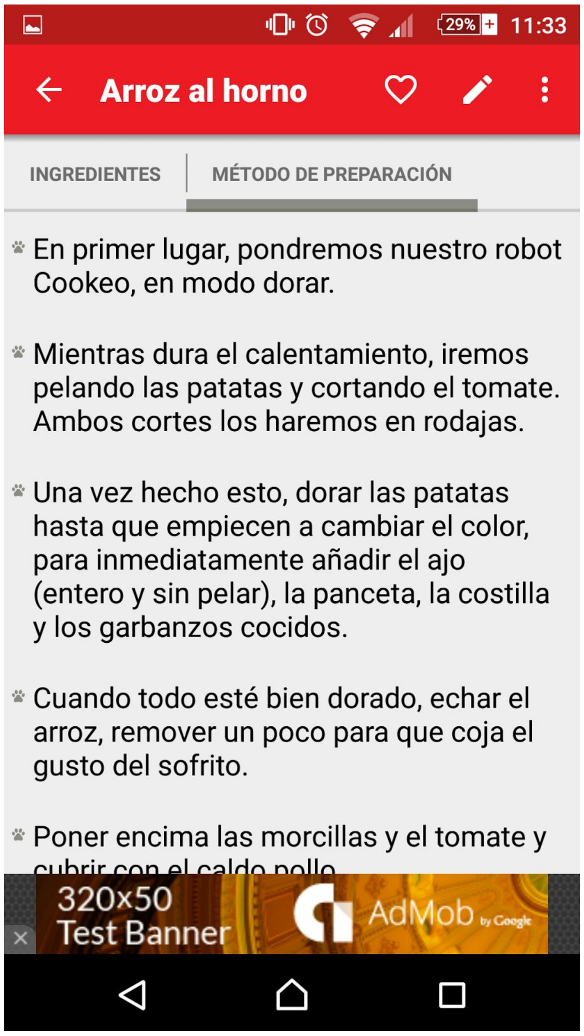


Link original: Recetas Cookeo

Tiempo (minutos): -

Raciones: 2

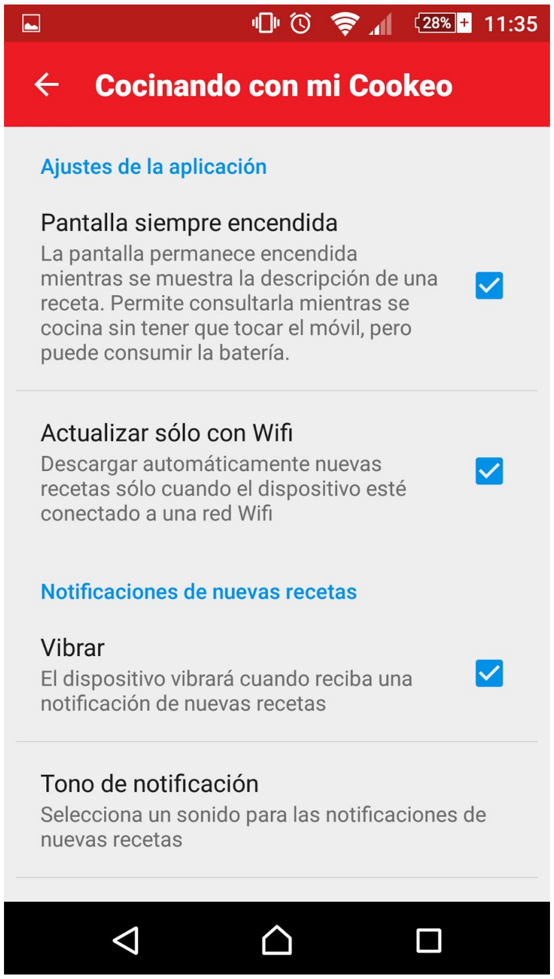
- Arroz 200 g
- Panceta
- Costilla
- Morcilla



- En primer lugar, pondremos nuestro robot Cookeo, en modo dorar.
- Mientras dura el calentamiento, iremos pelando las patatas y cortando el tomate. Ambos cortes los haremos en rodajas.
- Una vez hecho esto, dorar las patatas hasta que empiecen a cambiar el color, para inmediatamente añadir el ajo (entero y sin pelar), la panceta, la costilla y los garbanzos cocidos.
- Cuando todo esté bien dorado, echar el arroz, remover un poco para que coja el gusto del sofrito.
- Poner encima las morcillas y el tomate y cubrir con el caldo pollo

Recipe details activity: it will show details from the recipe. In these mocks, the information is shown using tabs (one for pic and ingredients and the other for the procedure). But I will show all information without tabs, using a coordinator layout with material design effects, like parallax scrolling, cardviews, a FAB for the "heart icon" that will be animated...

Screen 4



Cocinando con mi Cookeo

Ajustes de la aplicación

Pantalla siempre encendida [✓]
La pantalla permanece encendida mientras se muestra la descripción de una receta. Permite consultarla mientras se cocina sin tener que tocar el móvil, pero puede consumir la batería.

Actualizar sólo con Wifi [✓]
Descargar automáticamente nuevas recetas sólo cuando el dispositivo esté conectado a una red Wifi

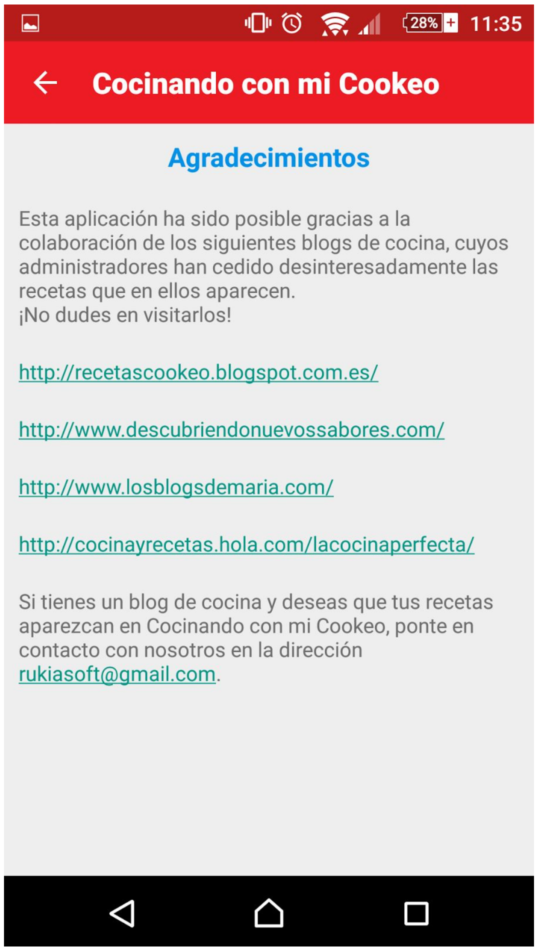
Notificaciones de nuevas recetas

Vibrar [✓]
El dispositivo vibrará cuando reciba una notificación de nuevas recetas

Tono de notificación
Selecciona un sonido para las notificaciones de nuevas recetas

Settings activity: configuration of the app. It allows you to choose options like keep screen on when showing a recipe, download new recipes only with wifi connection or choose sound for notifications.

Screen 5



11:35

Cocinando con mi Cookeo

Agradecimientos

Esta aplicación ha sido posible gracias a la colaboración de los siguientes blogs de cocina, cuyos administradores han cedido desinteresadamente las recetas que en ellos aparecen.
¡No dudes en visitarlos!

http://recetascookeo.blogspot.com.es/

http://www.descubriendonuevossabores.com/

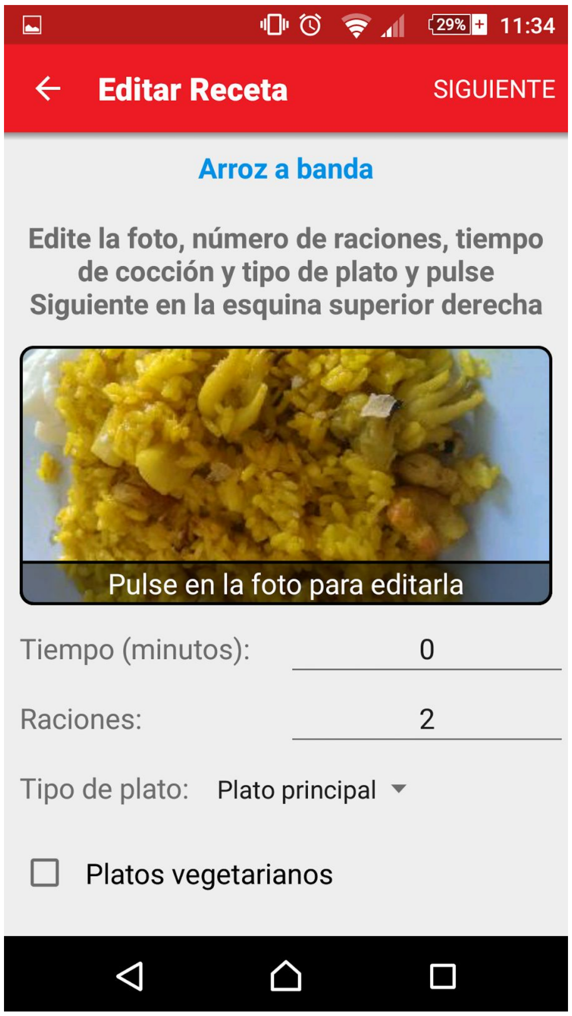
http://www.losblogsdemaria.com/

http://cocinayrecetas.hola.com/lacocinaperfecta/

Si tienes un blog de cocina y deseas que tus recetas aparezcan en Cocinando con mi Cookeo, ponte en contacto con nosotros en la dirección rukiasoft@gmail.com.

Thanks Activity: all recipes are taken from blogs (every blog is linked in the recipe's description), so in this activity I will mention and thank all blogs that allowed me to take recipes from their sites.

Screen 6



Arroz a banda

Edite la foto, número de raciones, tiempo de cocción y tipo de plato y pulse Siguiente en la esquina superior derecha

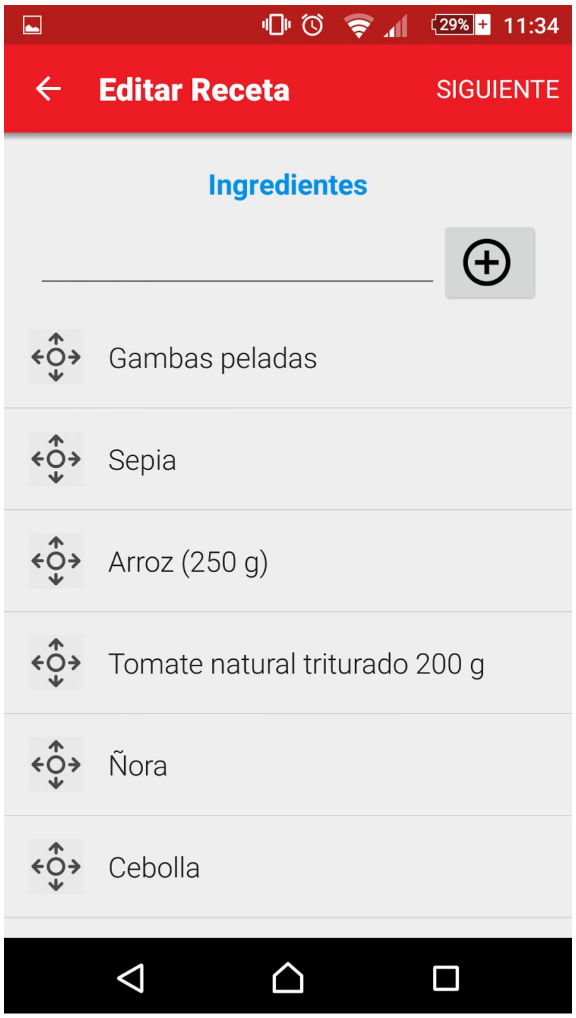
Pulse en la foto para editarla

Tiempo (minutos): 0

Raciones: 2

Tipo de plato: Plato principal

Platos vegetarianos



Ingredientes

- Gambas peladas
- Sepia
- Arroz (250 g)
- Tomate natural triturado 200 g
- Ñora
- Cebolla

Edit/Create Activity: You will be able to create a recipe from the scratch (using the FAB in main activity) or modify an existing one (using the pencil icon in activity details). This activity holds 3 fragments
1- set name, author, time of preparation, number of persons and select the picture (taken by the camera or picked from the gallery, and then cropped)
2- set the ingredients
3- set the steps (procedure) and any piece of advise.

TABLETS

Tablets will make better use of the space, using similar techniques as in Project 5: Make your app Material. In main activity it will show recipes in more than one or two columns, edit activity will show all 3 fragments at the same time…

App Engine Backend with Google cloud messaging

A small module will be developed in order to test push notifications. it will send a push notification to the app. This notification will include a link to download a new set of recipes. Those recipes would be stored in a server in a production app, but in this case they will be stored in a public folder of a Dropbox account. The app will download the zip, decompress it, store the recipes in the device memory, delete the zip and stores the link in a database in order to avoid downloading it again.

Key Considerations

How will your app handle data persistence?

Recipes will be stored as xml files in internal or external memory (depending of the recipe type). The application will have a number of default recipes, which will be provided as assets. Recipes downloaded from the server will be stored in the internal memory (in order to delete them when app is uninstalled). Created and edited recipes will be stored in external memory, in order to keep them saved even if the app is uninstalled. The possibility of a cloud backup will be studied.
A loader will load all recipes to populate the recyclerview.
During the app life-time it will use “onSaveInstanceState” and “fragment setRetainInstance(true)” to keep data when orientation changes, navigate from one activity to another, or any other issue.

Describe any corner cases in the UX.

The work-flow of the app is as described below:
App starts with Main Activity. In onCreate checks an “intro” flag. if false, starts intro activity. Then set the flag to true in order to avoid showing the intro activity again.
At the beginning it will connect to the server to ask if there are new recipes available. If true it will download them (if allowed)
When pressing any option in the navigation drawer, recipes will be filtered. The search icon will filter recipes as well, according to the typed text.
If a recipe is clicked, activity details will be launched, showing details from this activity.

Clicking back button from details activity will show Main Activity.
Clicking back button from main activity will close the app (a dialog to confirm will be shown). If Main activity is showing only a group of filtered recipes, clicking back will show all recipes, and clicking back again will close the app.
Clicking back button in create/edit activity will close the activity in tablets and show previous fragment in phones. If first fragment is being showing, click back will close the activity as well. App will always show a confirm dialog.

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

Glide to handle the loading and caching of images.
Android Design Support Library to add floating action buttons and apply material design style guidelines
Android support library, to allow old devices compatibility (the app will allow api level 14 and higher for sure, but the idea is to support API level 9 - Gingerbread, if not too much extra work is required)
Google Play Services, to allow push notifications, Analytics, Admob…
Nineoldsandroid
simple-xml to read the recipes (each recipe will be a xml file and a jpeg file)
butterKnife
GSON or codehouse jackson-mapper and jackson-core, to handle JSON data
okhttp to connect to the server
alertdialogpro-theme-material to show alert dialogs with material style in pre-llolipop devices
¿acra?: I have used it in a previous project to register unhandled exceptions. In this case I will try Analytics to do the same (as I have learned in the Google play services course), but in case it does not satisfy my expectations, I will get back to Acra again.

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

- Configure libraries
- Add activities, services and permissions in Manifest.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for DetailsActivity
- Build UI for CreateRecipe
- Build UI for ThanksRecipe
- Create xml files for settingsActivity and menus
- Create UI for IntroActivity
- Select/create icons and drawables

Task 3: Play intro Video

- start intro activity from mainactivity
- play video
- return to main activity and check video is not played again

Task 4: Create and cofigure backend

- Create backend
- Create service that connect app with backend
- Check if app gets response
- Show notification with response

Task 5: Load and Store Recipes

- Create a loader to load recipes when app starts
- Create functions to safely write and erase files from internal and external memory
- Create functions to load images into Imageviews

Task 6: Create activities.java

- Create java classes for any activity and fragment
- Create recyclerview adapters
- Create work flow between activities
- Create some beauty transitions (animations, shared elements…)

Task 7: Download data

- Create functions for downloading zips form Dropbox, unzip, store content and erase container.
- Create database to store used links

Task 8: Edit/Create recipe

- Create classes for taking pictures from camera or gallery and crop them
- Create recycler views that allow drag and drop, and swipe to dismiss, in order to edit elements (ingredients, procedures…)

Task 9: Error Handling

- Use Analytics to send unhandled exceptions or...
- Use Acra library

Task 10: Ads

- Use admob to show ads and interstitial ads

Task 11: Sharing

- Allow to share your own recipe. For this non commercial app, it will send the recipe as an email using the gmail account (or any other) to the developer address. In a production app, it would send it to a server, share them app to app, or upload to a social network (facebook, g+, etc…).

Task 12: Last Steps

- configure gradle to run tests
- configure the application sign
- configure proguard rules to obfuscate the code
- configure gradle to avoid including non used resources

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”