GitHub Username: mbeevor

Soiree

Description

Plan your next dinner party - search thousands of recipes online, and store them for your starter, main and pudding. Make notes against each recipe and add a list of guests. You'll never make the same meal for the same group of guests again!

Intended User

For anyone who hosts dinner parties, and is looking for new recipe ideas, or a simple place to store recipes.

Features

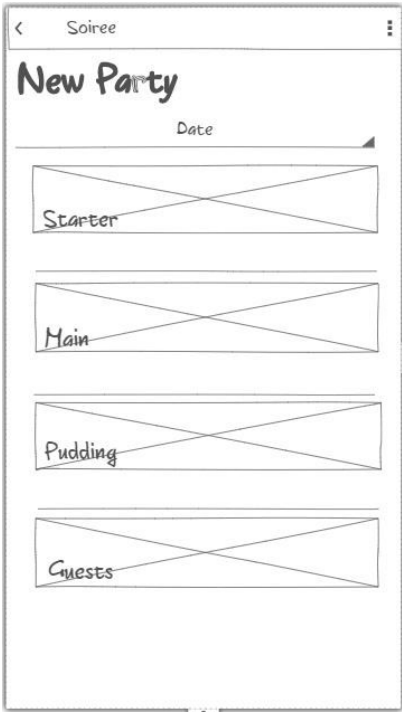
- Search for thousands of recipes using our in-built API
- Store your chosen recipe for up to three courses: starter, main and pudding
- Make your own notes for each recipe to remember for next time
- Store details about the party, like the date and the guests who attended

User Interface Mocks

Screen 1

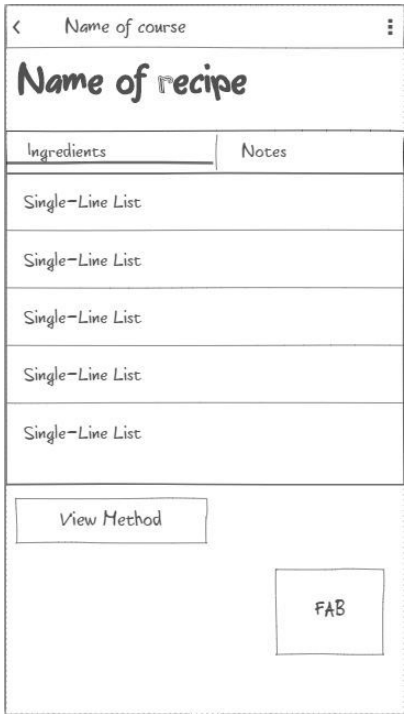
MainActivity screen - showing list of upcoming parties and a FAB to create a new dinner party. Prominent image at top of screen.

Screen 2 - DetailActivity



DetailActivity, used when creating new party, or pre-populated if editing an existing party. Cards to select each course, and the list of guests.

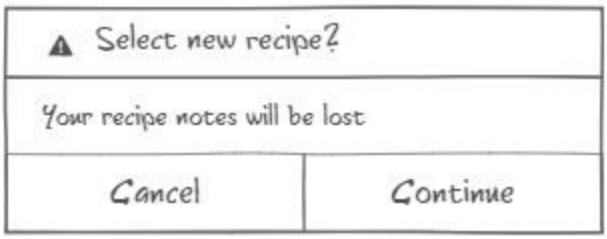
Screen 3 - RecipeFragment



Each course will have two tabs:

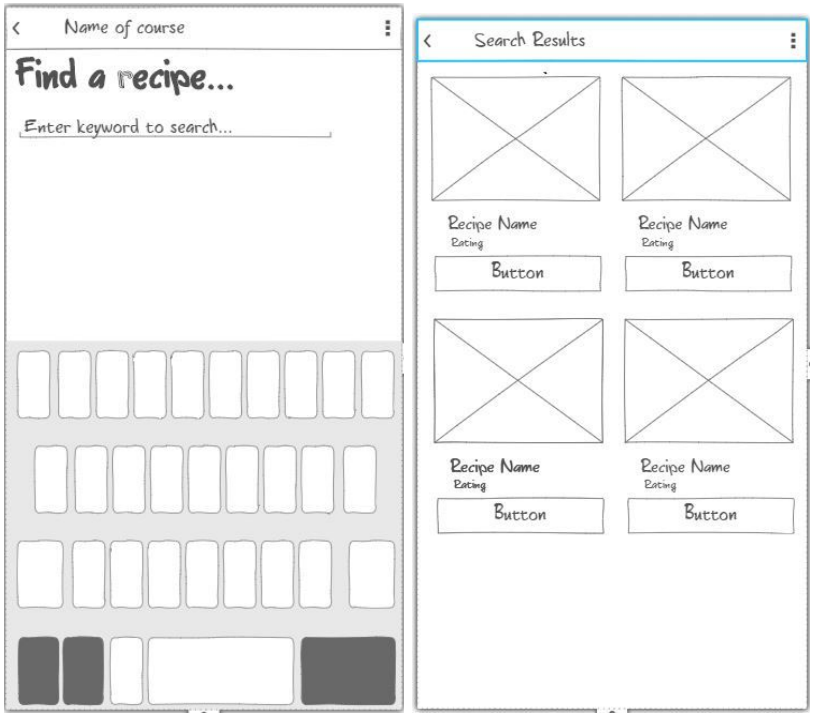
- ingredients(showing list of ingredients) and a button which links to the recipe method online with an intent to open the recipe in a browser [the API doesn't include recipes, only links]
- notes (where the user can add their own notes and comments).

Each tab will display default text if no recipe has been selected.
The FAB is used to search for a new recipe. The user is warned if an existing recipe is already selected, and asked to confirm before they can proceed.



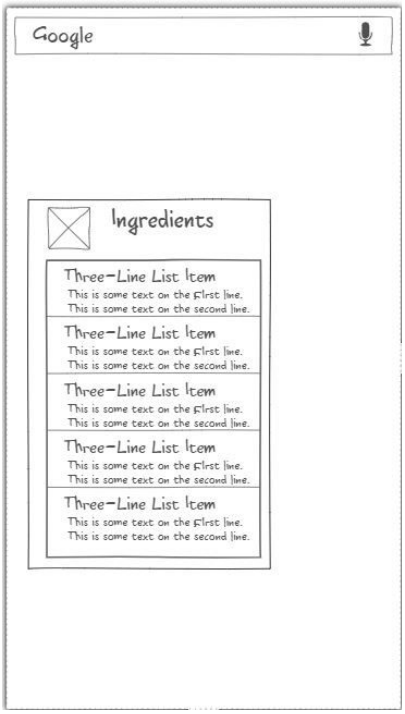
The RecipeFragment will appear in a new activity, or alongside the DetailActivity depending on space available.

Screen 4



The search field will ask the user to enter an ingredient to search and then show a grid view list of results with the recipe name, the average rating and an image. A button will be used for the user to add the recipe to their dinner party menu.

Widget



The app widget will show the recipe ingredients for the selected recipe.

Key Considerations

How will your app handle data persistence?

Data will be handled using a content provider.
Recipes will be fetched from the Food2Fork API and then stored in a local database when saved to a dinner party. The same database will also store a list of dinner party guests and any notes made by the user.

Describe any edge or corner cases in the UX.

The recipe activity is designed to show details about the chosen recipe, but this will be blank if no recipe has been selected. Pre-populated text will be displayed here instead.

Similarly, the user may not choose three recipes, opting for only one or two. Again, pre-populated text will be shown to ensure the app doesn't crash.

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

Picasso will be used to handle the loading and caching of images. Picasso is a well-known and maintained library that I have experience of using.

The app will also use Butter Knife to bind views without having to use 'findViewById'. Again, this is a well-known, maintained and recognised library.

The app will be written entirely in Java and will be built using Android Studio, powered by Gradle. The table below shows the versions of libraries and apps the project will use:

Tool / Library	Version number
Android Studio	3.1.3
Gradle	3.0.1
Picasso [com.squareup.picasso]	2.5.2
Butter Knife [com.jakewharton:butterknife]	8.8.1
AppCompatLibrary [com.android.support:appcompat]	27.1.1
Android Support Design Library [com.android.support:design]	27.1.1
Android Support Annotations [com.android.support:support-annotations]	27.1.1
Gson [com.google.com.code.gson:gson]	2.8.2
RecyclerView [com.android.support.recyclerview]	27.1.1
CardView [com.android.support:cardview]	27.1.1

The app will keep all strings in a strings.xml file and will enable RTL layout switching on all layouts.

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

The app will use AdMob to display ads, and Firebase Analytics to gain insight on app usage and user engagement.

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

- Create a new blank Android Studio project which implements AppCompatActivity and latest versions of all repositories
- Configure libraries for Picasso and BindView
- Ensure you have an API key for the Food2Fork API (https://food2fork.com/about/api).

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
    - Create layout activity file with RecyclerView and FAB
    - Set up OnClickListener for RecyclerView
    - Set up OnClickListener for FAB

- Build UI for DetailActivity and DetailFragment
    - Create layout file using cards, and use Intents from MainActivity
    - Set up OnClickListeners for each part of UI

- Build UI for RecipeActivity
    - Create layout using tabs and cards

- Build UI for SearchActivity
    - Create RecylcerView and Adapter for showing results based on search query

Ensure each ImageView in each layout file includes content description for accessibility purposes.

Task 3: Set up Google Firebase

- Create new Google Firebase project, to allow integration of AdMob and Firebase Analytics
- Update project to create ‘free’ and ‘paid’ variants. Include AdMob dependencies and Ad layouts in free version.
- Implement Firebase Analytics

Task 4: Set up Database and create Models

- Create model Menu, using Parcelable, consisting of starter, main and pudding
- Create models for starter, main and pudding
- Create new content provider, contract and database handler
- Create new Loader and Adapter to populate MainActivity RecyclerView
- Create ProgressBar to display when cursor data loading in MainActivity and static content for showing in RecyclerView when loader returns null (i.e. there are no saved dinner parties)

Task 5: Set up API

- Create Adapter and AsyncTask for querying the API

Task 6: Set up widget

- Create widget for displaying ingredients; include tools for user to select which recipe ingredients to show