

GitHub Username: heleneshaikh

James

Description

Easily order & pay your meals at a restaurant.

Intended User

For everyone who likes to eat at restaurants, i.e. couples, families, friends, ...

Features

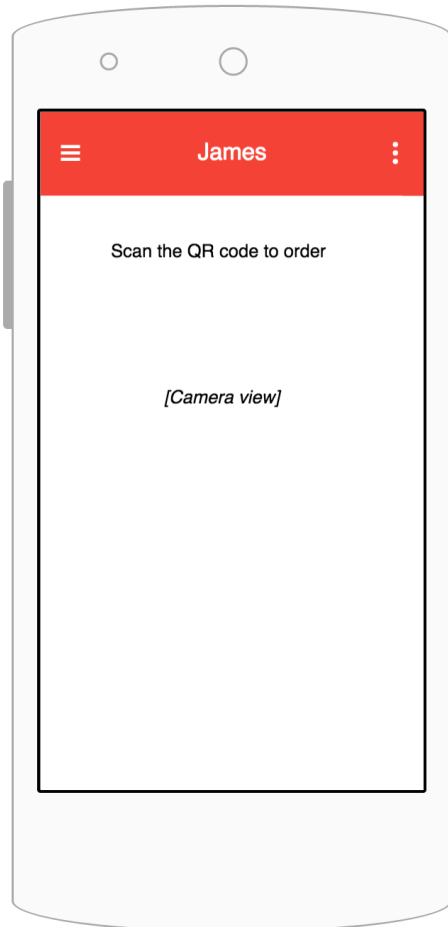
List the main features of your app. For example:

- Displays the menu of the restaurant.
- Order drinks/food.
- Lets you review your order and pay your receipt (cash or via the app) .

User Interface Mocks

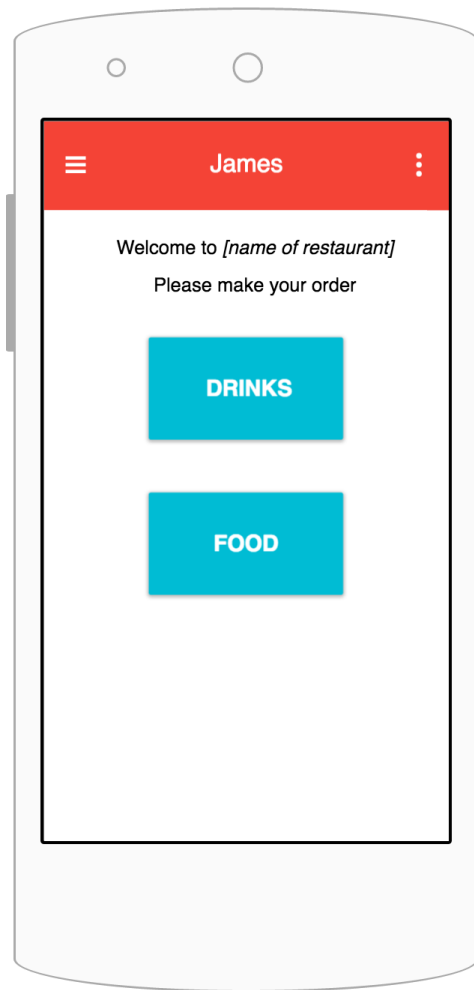
[Live preview](#) (if you download the proto.io app on your mobile, you can follow along as if it were a real app. To start, click on “Scan the QR code to order”)

Screen 1 - MainActivity



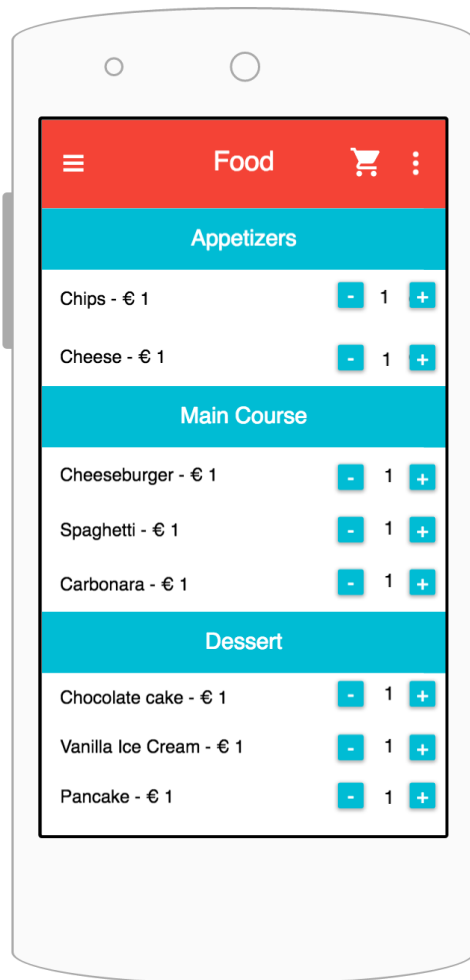
- Title: Scan the restaurant menu QR-code
- QR-scan camera view

Screen 2 - MenuActivity



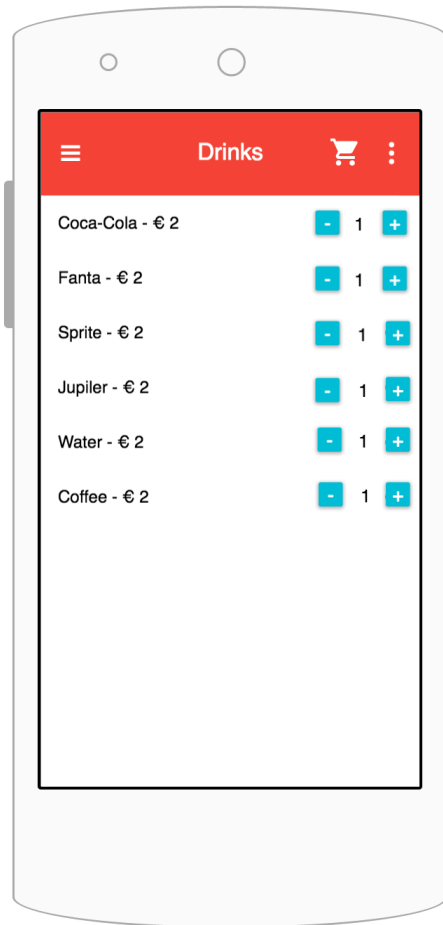
- Title: Welcome to [name of restaurant]. Please make your order
- “Food”-button. OnClick opens DrinkActivity
- “Drinks”-button. OnClick opens FoodActivity
- Back button to return to MainActivity

Screen 3 - FoodActivity



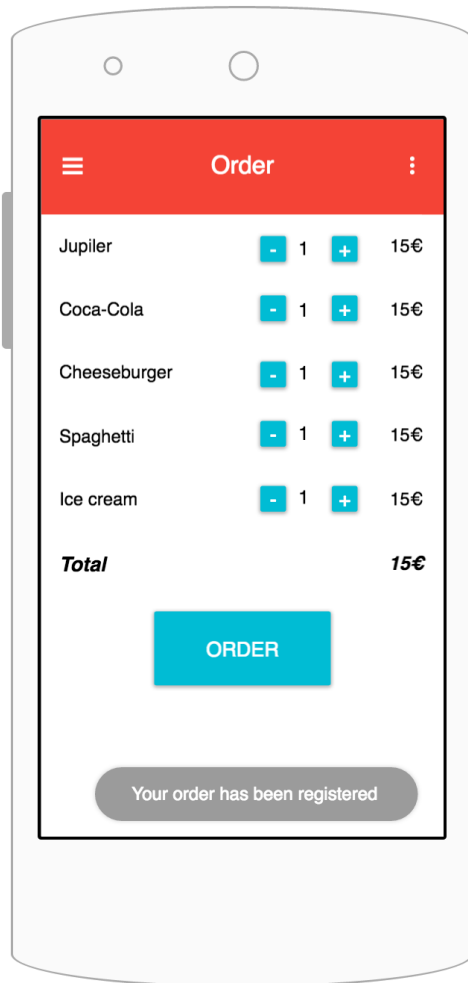
- Optional: click on title and see the dropdown menu
- Include shopping cart icon in toolbar. Every time the user clicks on + or -, animate the icon with a number indicating how many items the user has ordered so far. This way, the user knows his order has been registered every time he orders something.
- Title: Appetizers
- List of appetizers with + - number picker to choose amount and price
- Title: Main Course
- List of foods with + - number picker to choose amount and price
- Title: Dessert
- List with all desserts with + - number picker to choose amount and price
- Back button to return to MenuActivity
- Double click on back button to return to MainActivity

Screen 4 - DrinkActivity



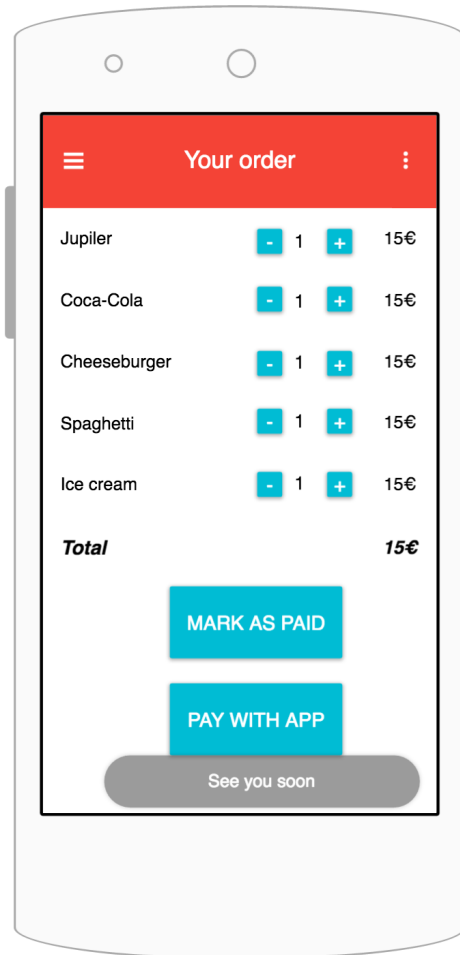
- List of all available drinks with + - number picker to choose amount and price
- Back button to return to MenuActivity
- Double click on back button to return to MainActivity
- Include shopping cart icon in toolbar. Every time the user clicks on + or -, animate the icon with a number indicating how many items the user has ordered so far. This way, the user knows his order has been registered every time he orders something.

Screen 5 - ShoppingCartActivity



- Overview of order
- Button to send the order to the waiter

Screen 6 - OrderOverviewActivity



- Title: Drinks
- List with chosen drinks, amount and price
- Title: Main Course
- List with chosen main course, amount and price
- Title: Desserts
- List with chosen desserts, amount and price.
- Total price
- Button "mark as paid". Toast "see you soon". Redirect to MainActivity.
- Button "pay with app". Via Android Pay, user goes through purchase flow

DrawerLayout

- List items: Drinks, Food, Order Overview, Pay
- Extra: Log out

Widget

Widget will display the user's current bill so that he can view it while eating/drinking. To keep track of his expenses.

How will your app handle data persistence?

Although it's brand new to me, I'm going to use the Firebase Realtime Database to hold and retrieve the data of the user's order. I can share this with the bartender's system as well.

Data is stored as JSON and synchronized in realtime to every connected client. Link:

<https://firebase.google.com/docs/database/>

Describe any corner cases in the UX.

Can't think of any for the moment.

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

Butterknife, easy for data binding. Saves you a lot of work.

Retrofit to request my data from the API.

EventBus to pass events to other classes.

(I'll probably use other libraries as I'm progressing in my app).

Describe how you will implement Google Play Services.

Android Pay will be integrated in the OrderReviewActivity so that the user can pay their bill at the end of their meal.

Mobile Vision will be integrated in the MainActivity so that the user can scan the restaurant's QR code and access the menu.

(extra: Google Analytics, Google Mobile Ads)

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.

- Read Firebase documentation
- Make an API with dummy data for the restaurant's menu using <http://myjson.com/>
- Read Mobile Vision documentation on how to implement a QR-code reader that redirects you to the restaurant's menu
- Figure out how to animate shopping cart SVG icon

Task 2: Implement MainActivity

- Create layout
- Incorporate a QR-reader using Mobile Vision

Task 3: Implement MenuActivity

- Create layout and add buttons "food" and "drink". Redirect user to either FoodActivity or DrinkActivity.
- On tablet: FoodActivity or DrinkActivity would appear on the right pane.

Task 4: Implement FoodActivity

- Create layout
- Request Appetizers, Main Course, Dessert data from myjson.com using Retrofit
- Write the user's choice to Firebase Realtime Database
- Include shopping cart icon in toolbar. Every time the user clicks on + or -, animate the icon with a number indicating how many items the user has ordered so far.

Task 5: Implement DrinkActivity

- Create layout
- Request list of drinks from myjson.com using Retrofit
- Write the user's choice to Firebase Realtime Database
- Include shopping cart icon in toolbar. Every time the user clicks on + or -, animate the icon with a number indicating how many items the user has ordered so far.

Task 6: Implement ShoppingCartActivity

- Create layout
- Display order overview
- Include button that will send the user's order to the waiter via Firebase.

Task 7: Implement OrderOverviewActivity

- Create layout
- Retrieve data from Firebase Realtime Database
- Display the user's order and total amount due
- Include buttons to allow users to pay cash or via the app
- If user clicks on "pay cash", display a toast saying "see you soon" (the bartender receives an alert saying that the user is ready to pay and comes at the user's table)
- If user clicks on "pay with app", user is redirected to payment via Android Pay where user has to complete the usual payment steps. When payment is done, display toast.

Task 8: Create a widget

- Portray the user's current bill, retrieve data from Firebase Realtime Database.
-