GitHub Username: Ajrulovski

Ready!

Description

Problem:
Let's consider several every-day situations that most of us are facing:

1. We leave our clothes for dry-cleaning, and they tell us to come back tomorrow after 17:00. When we go there, the clothes are not done, they ask us to wait for an hour and then come back.
2. We leave our car at the car wash. We go to the nearby mall to finish up on other stuff, we hurry back, and yet the car is not ready.
3. We order a pizza (or e.g. coffee) in a half-service restaurant. We have to wait for the pizza to be ready, and if the place has no buzzer based system to inform you when the pizza is done, you will probably have to stand around and wait for your name to be called. Maybe you want to go out and get something else done meanwhile, but hey, you will have to wait.
4. You just love those burgers from a burger joint 3 blocks north of your place, but they don't have delivery. You need to go there, order food, and wait for it to be done and ready to go.

Actually, all of these situations happened to me. And not only in my hometown, but all around the world, especially in cities that are more inclined toward half-service businesses (e.g. Stockholm, Sweden).

Solutions:
Here comes in Ready!.
Ready! is designed as an Android app with two modes of work: Business mode, and client mode. Also Ready! will have two types of users, let's call them "business user" and "client user".

With business mode, the Micro and SME (or why not, even franchise) businesses would be able to open their business user account, and manage a menu of their services. With client mode, client users would be able to open a client account and place orders for the services in the menus built by the business accounts (e.g I could order a coffee from my favourite coffee-to-go shop). After each placed order, the business user would get a notification that there is a new incoming order (my coffee shop would see a notification from @Ajrulovski) and the order would appear in their orders cue. The business user accepts (or declines) the order and estimates the time for which it would be ready (so that the client user has an approximate idea of how long should he wait). After processing the order, the business user would mark the order as ready, which would trigger a notification towards the client user (the one that ordered the service)

informing him that his order is ready to be picked up (e.g. I would get a notification saying that my latte macchiato is ready). So, the client user goes to the place of business, picks up whatever he ordered, and marks the transaction as done. Prior to marking the transaction as done, the client gets a chance to rate the business and the business gets a chance to rate the client.

Intended User

Ready! is intended for all categories of users that are in need of making their order-to-pickup process more efficient. Of course, early adopters are more likely to replace their typical way of ordering services than other.

Features

App's main features are:

- Order food/services/goods even before reaching the place of business
- Get a notification when your goods are ready to be picked up
- Browse businesses and their menus of goods/services
- Keep a minimal info dataset about your business and broadcast it to the users
- Manage a list of your services (if you are a business user)
- Get notified when you get a new order from a client
- Review orders as Active and Historical orders

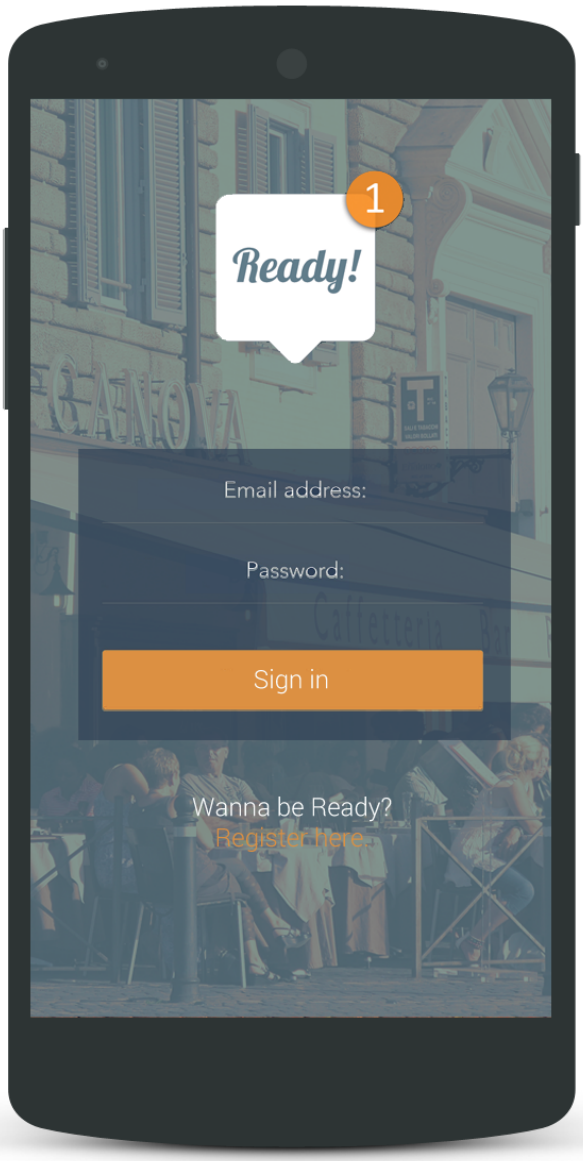
One of the main benefits of the application is its enormous potential to save client's time on one hand and increase business's exposure by giving them a new channel for marketing their services toward the clients.

User Interface Mocks

Screen 1

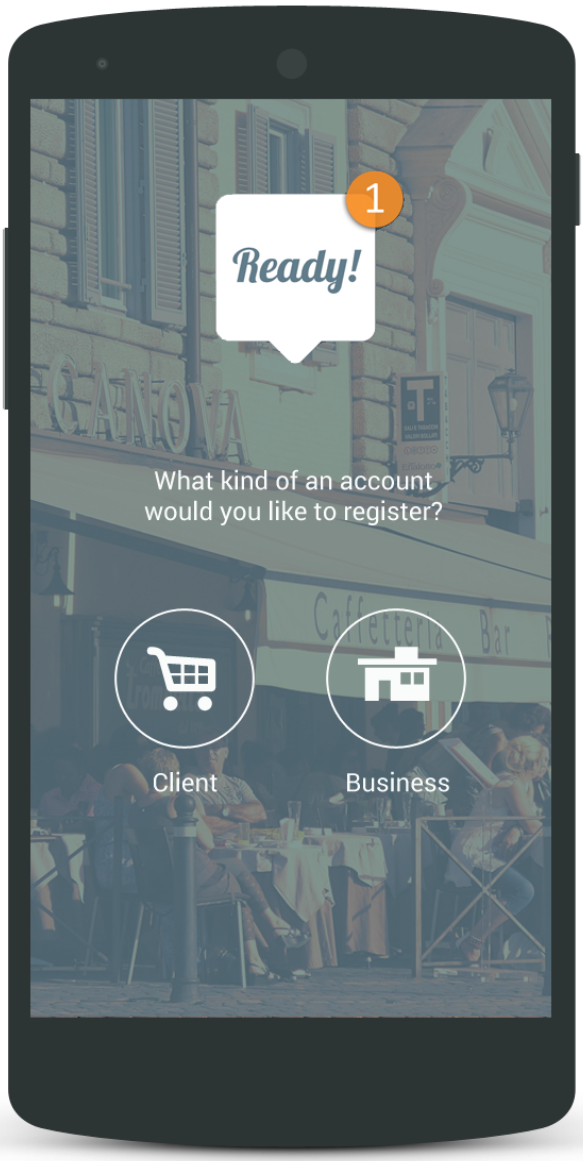
First splash screen, showing the app's logo.

Screen 2



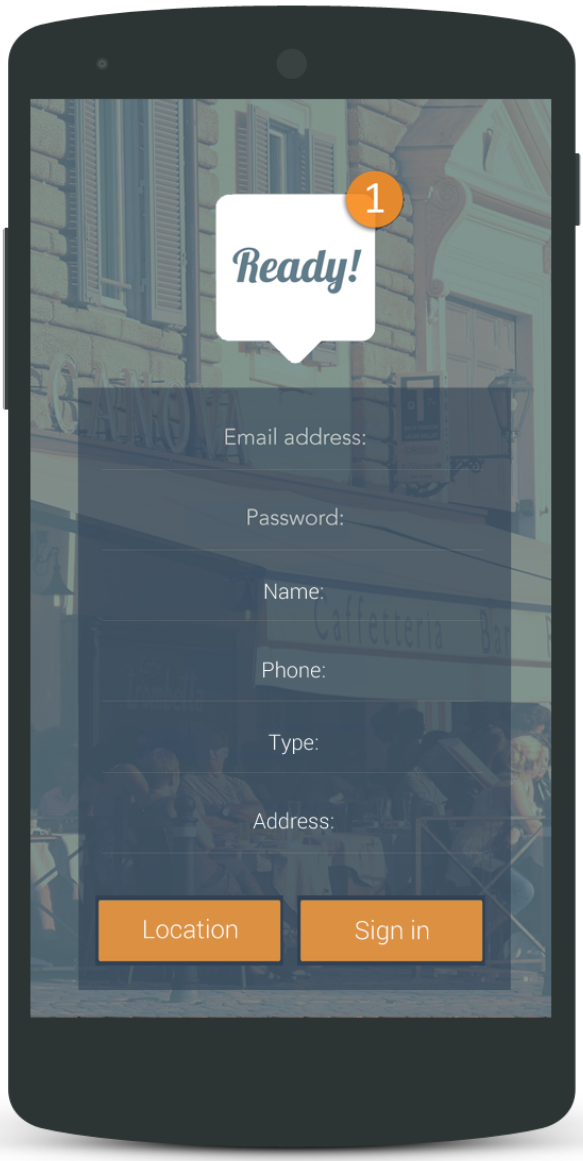
Login screen for both client and business users.

Screen 3



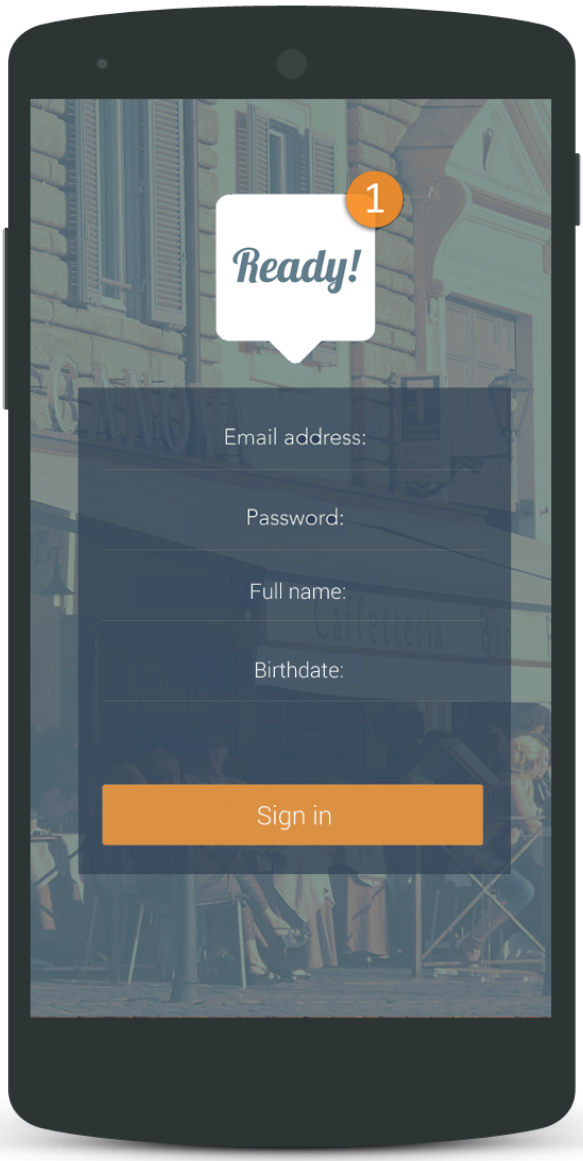
If the user has no account, he/she can open a new one, where first they need to choose the account type.

Screen 4



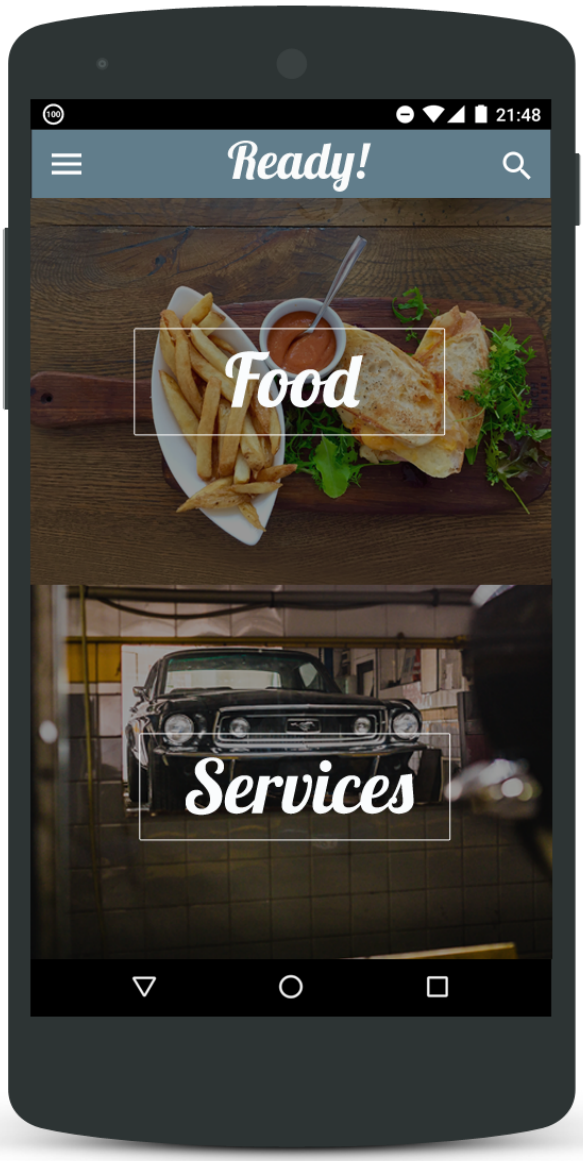
Register activity for businesses. Filled once when creating an account, information like email, name, type and location are obligatory.

Screen 5



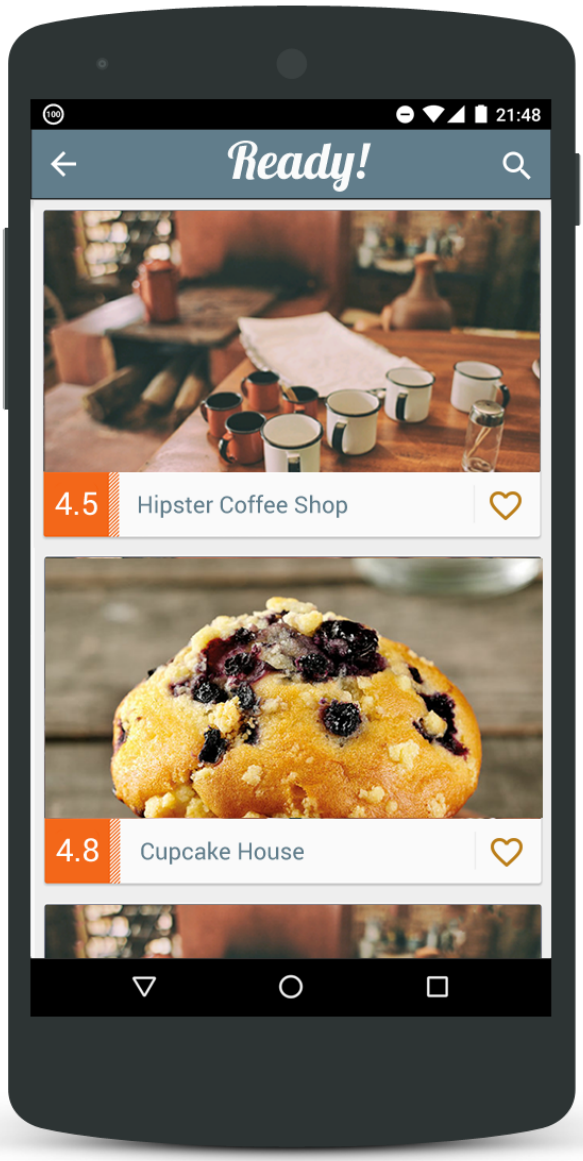
Register activity for clients. All fields are obligatory.

Screen 6



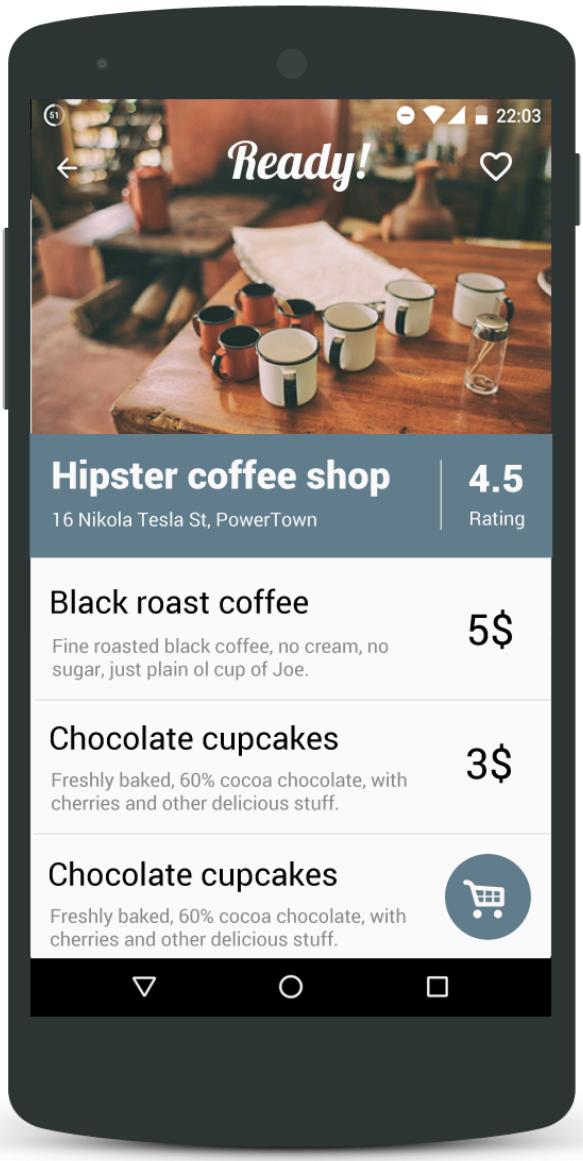
After a succesfull sign in/registration, a user of a type client can choose to browse businesses via their type (food, services, shopping etc), or to search for a client by typing several letters in the search box (activity opened with the icon in the upper right corner).

Screen 7



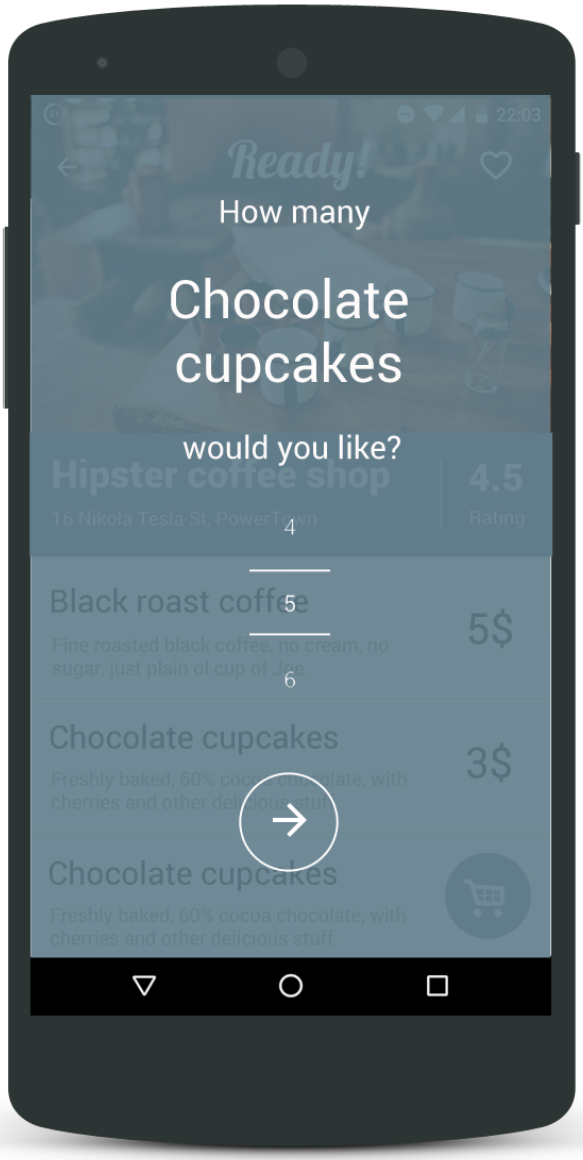
This is a mockup of an activity that would be started after you choose a business type, or search for a business by it's name.

Screen 8



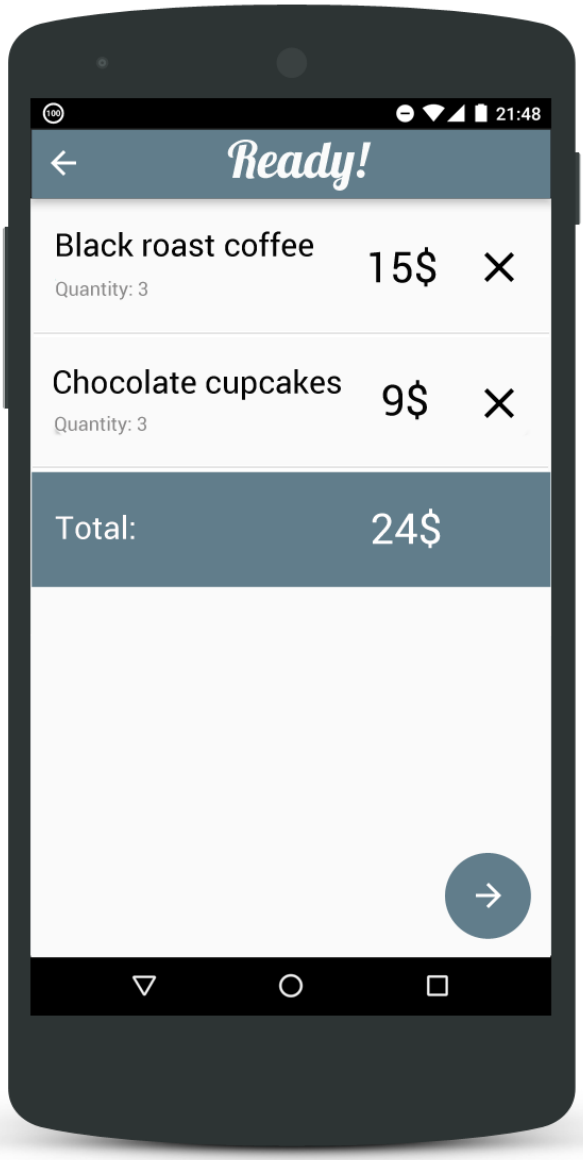
Then, after the client taps the business of their choosing, an activity appears that looks like this. Here, the client can review some basic info about the business, mark the business as favourite and view the complete service menu of the business. By tapping the menu item, an activity appears (shown on the following screen) that asks for the quantity to be entered, so that it can be added to the order list. When the user is done with adding items to the order list, he/she taps the shopping cart FAB to place the final order/checkout.

Screen 9



The client chooses the quantity of the order item, and by tapping the arrow button the item is added to the order list.

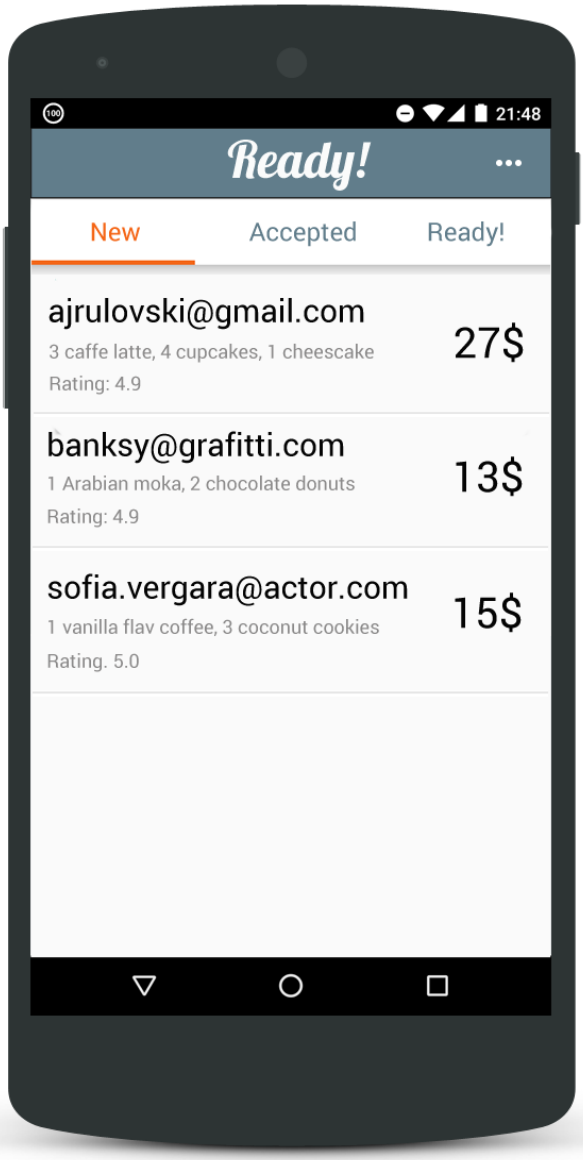
Screen 10



When the client taps on the place order/checkout FAB button (as seen in screen 8), they are redirected to the checkout activity. Here they can review all of the items in their order list, where they can remove an item from the list or go back to the menu to add more items. When they are done, by pressing the arrow FAB button they are sending out the order. At this point a

notification is sent to the business that received the order, and the client is waiting for a Ready! Notification for pick up.

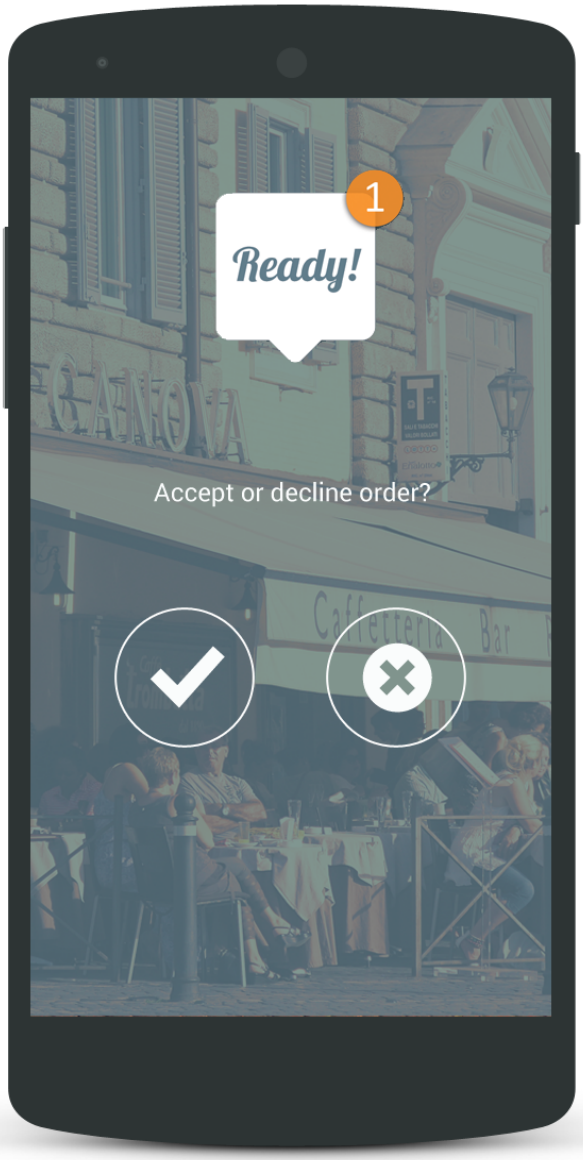
Screen 11



When the business user is logged in, his main activity is the activity with the orders they are receiving. The orders are first coming into the New tab. These orders can be accepted or

declined by the business (e.g. they do not have the item in stock or they simply don't want to serve the client because of his bad rating).

Screen 12

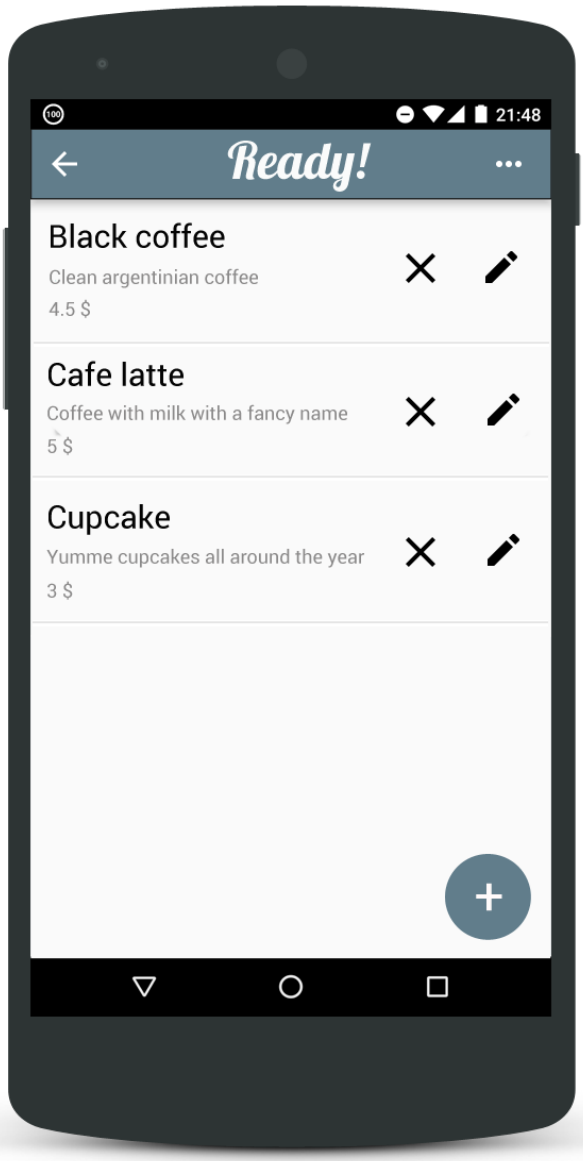


If the business accepts the order, it is moved to the Accepted tab (Screen 11). If they decline the order, the client is notified that their order has been declined, and the business is required to leave a reason (entered in a simple textview) for the refusal of the order.

Screen 13

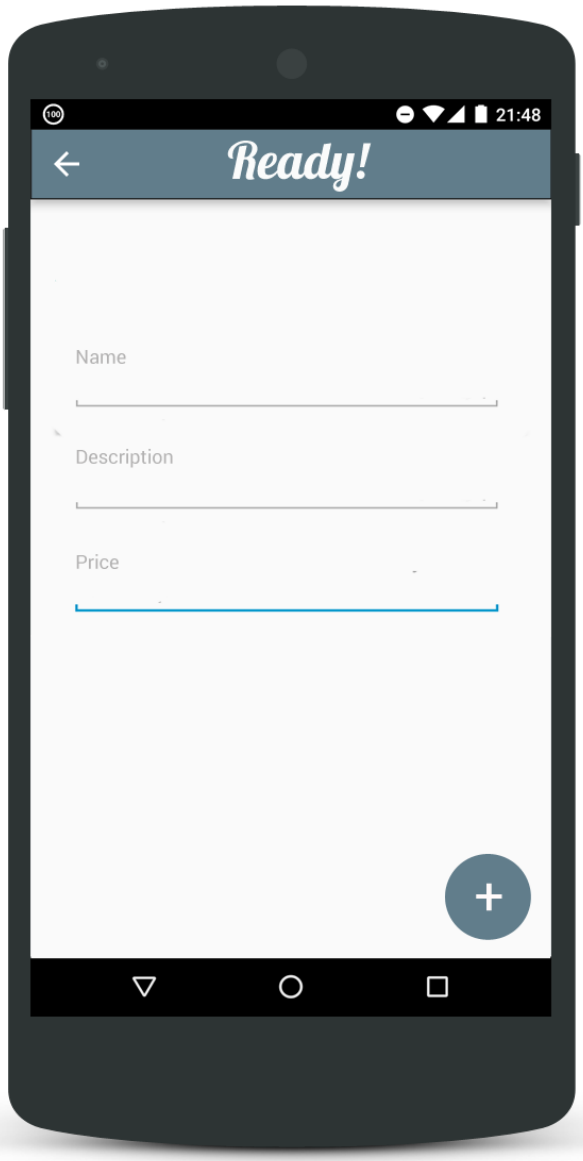
When an item in the Accepted tab is tapped, this is the activity that appears. The business decides if the order is ready to be picked up or not. If it is, by tapping the checkmark button, the order moves from the Accepted to the Ready! Tab (screen 11), and a notification is sent to the client that his order is Ready!

Screen 14



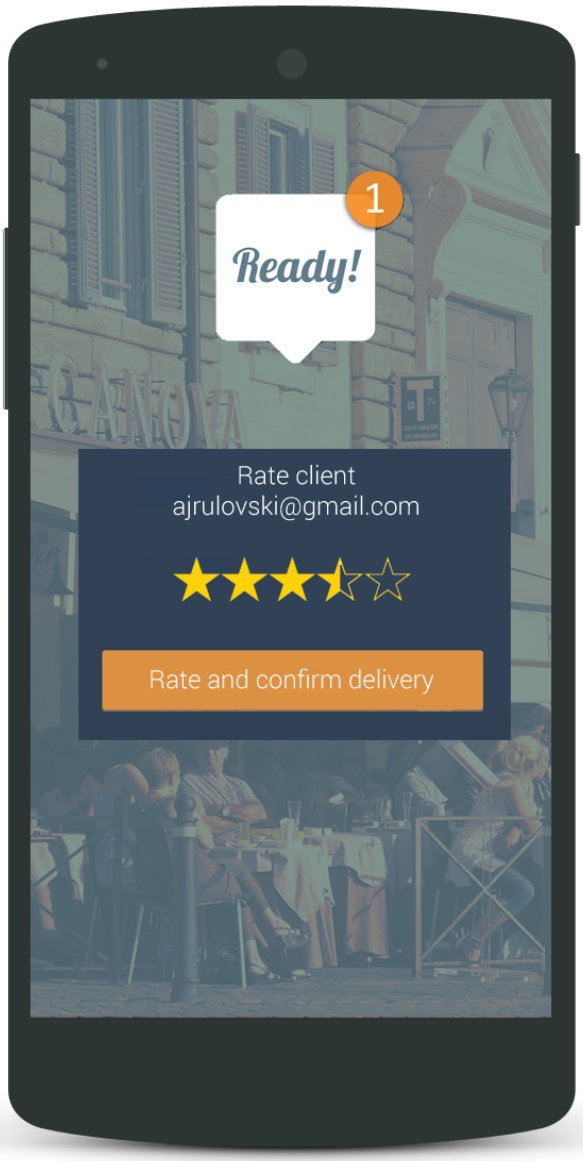
Each business will be able to maintain their menu of services/products via the menu editing activity. This activity is invoked via the menuitems reached from the button in the appbar's upper right corner in the Main order list (Screen 11).

Screen 15



New menu items can be added via this activity, invoked by the FAB button in the Screen 14.

Screen 16



Last but not least, one of the more important features of the system is the rating feature. Clients and businesses will be able to rate each other, so that the future users are aware of how good/bad their service is, or how good/bad the client is.

Key Considerations

How will your app handle data persistence?

All the data would be kept in a NoSQL structure hosted with Firebase. Uploaded files (e.g. photos) will be kept in Firebase file storage.

Describe any corner cases in the UX.

All the activities are going to be setup in a hierarchical order, and each parent screen will be accessible via the back buttons in the app bar. Rating of clients/businesses is merged with the delivery/pickup confirmation so that the users are subtly asked to rate their experience as the rating is an important part of the concept.

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

A number of libraries will be used in the implementation of Ready!, but mostly the usual suspects:

- Picasso for loading images
- Retrofit for async and service calls
- Google Maps API (for registering new business user account)
- Firebase libraries for utilisation of Firebase database, file storage and FCM (Firebase cloud messaging)

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

Initial phase of the implementation will require some basic setup actions:

- Create a new project in Visual Studio
- Set it up on Github
- Generate a new project on Firebase

- Include planned libraries (of course some of them I may include as the development goes)
- Make a data model that will be used/followed when keeping data in Firebase database
- Configure notification engine (via FCM)

Task 2: Implement Login activity

The layout and the respective functionality for the Login activity needs to be built

- Build UI/layout for LoginActivity
- Implement a needed data structure in Firebase database
- Implement calls for reading and checking the login data

Task 3: Register account for client activity

Users need to be able to open a client account for themselves.

- Create layout
- Implement a data structure to keep the client user data
- Implement the classes for creating new users
- Implement format control for the parameters (e.g. mail address format, allowed characters, password strength etc)
- Implement required fields check

Task 4: Register account for business activity

Users need to be able to open a business account for themselves.

- Create layout
- Implement a data structure to keep the business user data
- Implement the classes for creating new users
- Implement format control for the parameters (e.g. mail address format, allowed characters, password strength etc)
- Implement required fields check
- Implement the choose location on map functionality

Task 5: Choose business by type activity

The first activity that the client user will see will be a RecyclerView with a CardView item element.
With this activity, the user can get a quick list of businesses of a certain type: Food and drinks (Pizza places, Burger spots, Coffee shops, half-service restaurants and coffee bars), Shopping (local grocery stores and supermarkets), Services (carwashes, mobile phone service points, dry cleaning...).

- Create layout (Screen 6)
- Implement a data structure to keep the business types
- Implement the calls to read the types found in the database

Task 6: Search business by name

The user will be able to search for businesses by inputting the business's name. Search box should appear in the appbar after a tap on the search icon (Screen 6). The results will appear in the List of businesses activity described in Task 7.

- Create layout
- Implement the calls to search the businesses in the database

Task 7: List of businesses activity

After searching for a certain business, or choosing a business type, the result will appear in the List of businesses activity (Screen 7). The activity will have a RecyclerView with a complex item layout made out of an ImageView and several textviews showing important info for each shown business. Tapping the RecyclerView item will open the menu activity for the tapped business (Screen 8)

- Create layout
- Implement the calls for filling the necessary data adapters and loading/reloading the RecycleView
- Implement the Add to favourites functionallity

Task 8: Product/Service Menu activity

Each business will have a menu of products/services offered to the client (Screen 8). The client will be able to compile an order list by tapping the individual menu items and choosing the quantity for the order (Screen 9). The layout will use the business image as a header and will show a RecyclerView with a simple item layout to show the menu items as seen in Screen 8. When tapping a menu item, a quantity dialog will appear and after choosing the quantity the item will be added to the order list, kept in the SharedPreferences of the device. When the client decides that his order list is OK, he will tap the FAB button in the lower right corner to proceed to finalizing the order list.

- Create layout for the menu activity
- Implement the needed data structures in the Firebase database to keep the menus related to the business entities
- Implement the calls for filling the necessary data adapters and loading/reloading the RecycleView
- Implement the Add to favourites functionality
- Implement the order list mechanism (keeping the data in SharedPreferences)

Task 9: OrderList/Checkout activity

When the client decides that his order list is ready, he can check it out for the final time, before placing an order (Screen 10). The client can remove an item from the order list or go back to the menu and add a new item. When ready, the client can place the order.

- Create layout for the order list activity
- Implement the needed data structures in the Firebase database to keep the orders related to the business entities and client entities as well
- Implement the functions for filling the necessary data adapters from SharedPreferences and loading/reloading the RecycleView
- Implement the calls for placing an order
- Implement notification mechanism to inform the business that they have a new order in queue

For the time being, the client's activities end here. He is now waiting for a notification from the business informing him that his order is Ready!

Task 10: Live orders activity

On the business side, the manager of the business account will start with the Live orders activity where they can see the incoming orders in three tabs: New, Accepted and Ready! (Screen 11). The orders in the New tab are all orders that came from the client but the business has not responded to them yet. To respond to them, they should simply tap the order and a screen will appear that asks them if they want to accept the order or decline it (Screen 12). If they decide to decline it (e.g. item is not on stock) they should provide a reason for declining an offer and the client will get a notification saying that the offer is declined and showing the reason. If they decide to accept the offer, the order is removed from the New tab and will be transferred to the Accepted tab. For the business,this will mean that they are in the process of preparing the product/doing the service before it is ready for pick up. When they are finished, by tapping the item in the Accepted tab, they will see a choice (Screen 13) prompting them to mark the order as ready. If they tap on the checkmark button, the order will be moved from Accepted to the Ready! Tab, and a notification will be sent to the client that his order is ready for pick up.

- Create layout for the order list activity
- Create layout for the accept/decline/mark as ready order
- Implement the functions for filling the necessary data adapters for all tabs and loading/reloading the RecycleViews
- Implement the needed notification mechanisms

Task 11: Rate and confirm delivery/pickup activity

By tapping an item in the Ready! Tab, the rating activity (Screen 16) appears and the business can rate the client and their experience with him. This same activity will appear for the client as well when they tap on the notification saying that their order is ready, or by tapping an list item in the Client order history activity. At the same time, the user confirms the pickup/delivery of the goods, as this is the point when the transaction between the involved parties is done.

- Create layout for the rating activity
- Implement the needed data structures in the Firebase database to keep the ratings related to the business entities
- Add calls for recording the rating in the database
- Add calls for marking the pickup/delivery as confirmed

Task 12: Client order history activity

The client order history activity will hold all the orders that the client made. When tapping on the item they will be able to confirm the order pickup and rate the business, via the activity described in Task 11.

- Create layout for the Client order history
- Add calls for loading the placed orders in a RecyclerView

Task 13: Menu management activity

The business user will be able to manage and maintain their service/products menu via this activity (Screen 14). It will be a simple RecyclerView with a more or less complex item showing the product name, description and price for single item. New items can be added by tapping the FAB button in the lower right corner.

- Create layout for the Menu management activity
- Model the database entities for representing the menu items
- Add calls for loading the menu in a RecyclerView

Task 14: Add new menu item activity

With this activity (Screen 15) the business user will be able to add a new item to their menu.

- Create layout for the new menu item activity
- Add calls for inserting the items into the database

Task 15: Favourites activity

This will be a simple activity showing a RecyclerView of favourite places chosen by the client user. The list will look similar as the mockup in Screen 7.

- Create layout for the Favourites activity
- Add calls for loading the RecyclerView

Project plan:

As there is a rather large number of tasks and the app deadline is 11th of July, the task schedule is rather tight but doable. Here is a rough project plan for task implementation.