Table of contents

GitHub Username: elshadsm

Bio Organic

Description

The Bio Organic App aims to present a natural alternative to non-healthy, industrialized chemical foods that have become one of today's most pressing problems. Users online can easily get the stuff they need with this app.

The products are divided into different categories so that they can be targeted and find easily. Each food contains important information such as detailed information about it, rules of usage, place of manufacture, duration of use and so on. Buyers have the right to comment on it after receiving the food.

Intended User

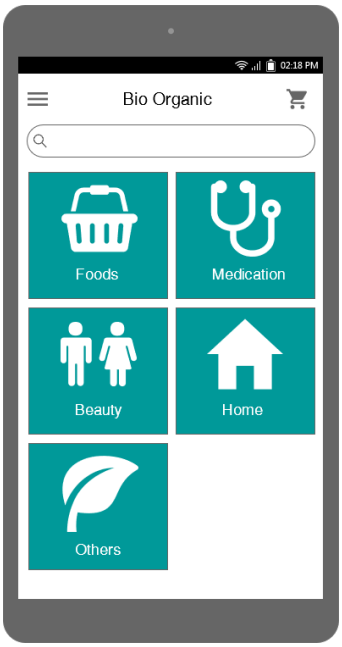
This program is especially designed for adolescents and older people. Some of the foods may be dangerous for health if they are not properly used. For this reason, it is not recommended for younger people to order some products by using this app.

Features

- Displays products categories.
- Provides product search functionality.
- Enables registration for a user.
- Displays product lists.
- Displays detailed information about a product.
- Enables to add a product to the shopping cart.
- Enables to add a product to the favorites list.
- Provides ordering functionality of an item.

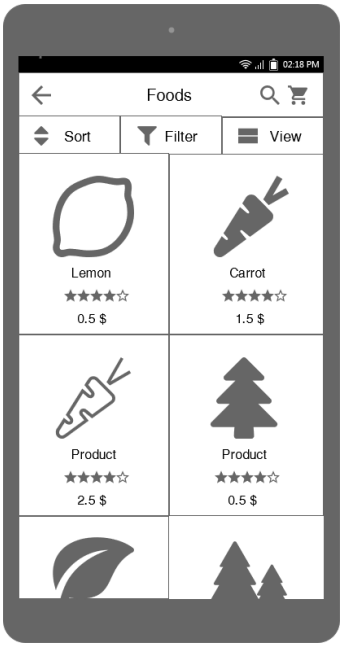
User Interface Mocks

Main Screen



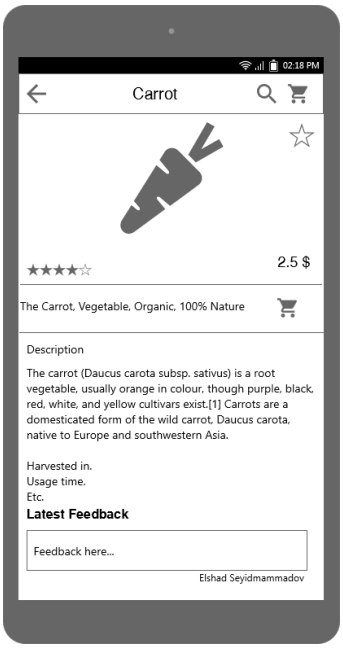
The main screen indicates product categories, search features, navigation menus, and shopping cart action. The categories presented on this page are intended to help the user find the product more convenient.

Product List Screen



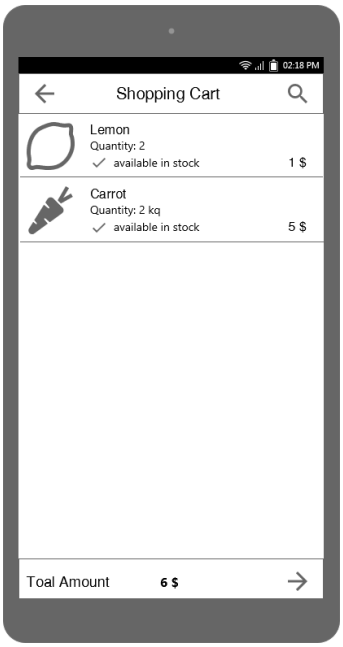
This screen represents a list of products to users. Users can perform sorting, filtering, and image distribution operations on products data.

Product Details Screen



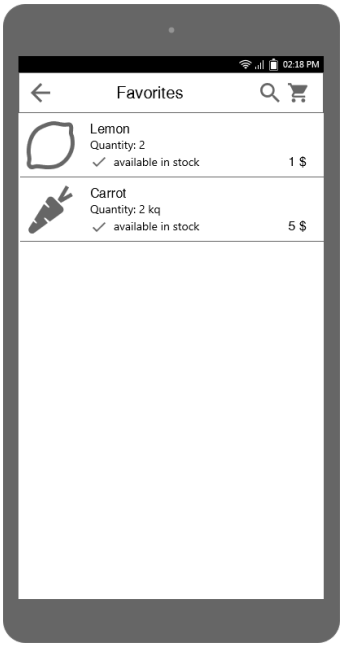
Users can get detailed product information on this screen. Additionally, they can add this product to their favorite list or add the product to their shopping cart.

Shopping Cart Screen



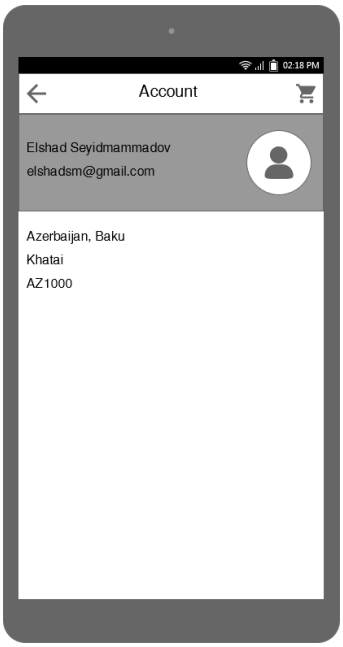
All products which were added to the shopping cart and the final price are displayed on this screen. The next process is to buy the products.

Favorite List Screen



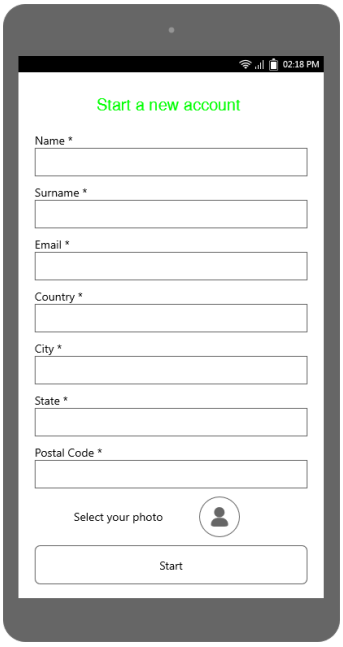
All products which added to the favorites list is displayed on this screen.

Account Screen



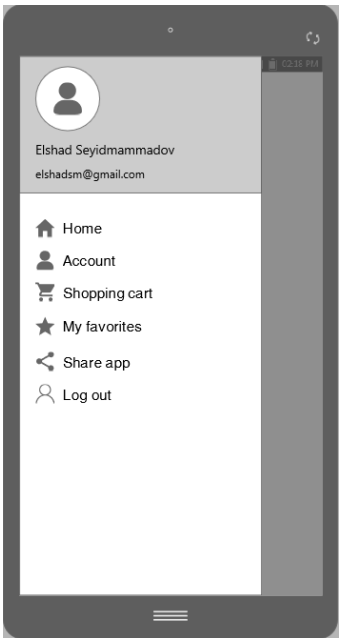
Displays account information.

Account Registration Screen



The screen is for registering a new account.

Menu



This menu is for performing different operations.

Widget

The simple app widget includes app logo, search input, and favorite icon. When a user clicks on the search input, the app opens with the focused search input. In addition, when the user clicks the star button, the favorites list screen opens.

Key Considerations

How will your app handle data persistence?

I am going to use Firebase Realtime Database to fetch data from the server. I don't have experience on Firebase Realtime so if I am not able to use it successfully, I will use mock files with the static data in it.

Additionally, I am going to use Content Provider to cache some data.

Describe any edge or corner cases in the UX.

In case of using the app without internet connection, the data will be retrieved from local storage.

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

I plan to use the libraries below:

1. Picasso to handle the loading and caching of images.
2. Butterknife automatically cast the corresponding view.
3. Retrofit to use as a type-safe HTTP client.

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

Firebase analytics to provide the app analytics.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new project.

- Setup project structure.
- Configure libraries.
- Create util/config classes.
- Handle resource files.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity.
- Build UI for ProductActvity.
- Build UI for ProductListFragment.
- Build UI for ProductDetailsFragment.
- Build UI for ShoppingCartActivity.
- Build UI for FavoritesListActivity.
- Build UI for AccountRegistrationActivity.
- Build UI for AccountActvity.
- Build UI for navigation menu.

Task 3: Implement Backend Support

- Design/Create the schema for the app on the Firebase Realtime Database.
- Fill mock data to the database.
- Synchronize the app with this database.

Task 4: Build Persistence Support

- Design database for caching.
- Define entity models.
- Define database layer.
- Define content providers.

Task 5: Implement Google Services Integration

- Implement analytics support for crash and analytics.