Taifun food delivery project

Оглавление

[Taifun Food delivery 2](#_Toc519201878)

[Description 2](#_Toc519201879)

[Intended User 2](#_Toc519201880)

[Features 2](#_Toc519201881)

[User Interface Mocks 3](#_Toc519201882)

[Main screen 3](#_Toc519201883)

[Left drawer menu 4](#_Toc519201884)

[Orders screen (no orders) 5](#_Toc519201885)

[Orders screen (with actual orders) 6](#_Toc519201886)

[Orders archive 7](#_Toc519201887)

[My info 8](#_Toc519201888)

[Food info 10](#_Toc519201889)

[Order details 11](#_Toc519201890)

[Order confirmation 12](#_Toc519201891)

[Change address for delivery 13](#_Toc519201892)

[Promo 14](#_Toc519201893)

[Widget 15](#_Toc519201894)

[Key Considerations 16](#_Toc519201895)

[**How will your app handle data persistence?** 16](#_Toc519201896)

[**Describe any edge or corner cases in the UX.** 16](#_Toc519201897)

[**Describe any libraries you’ll be using and share your reasoning for including them.** 16](#_Toc519201898)

[**Describe how you will implement Google Play Services or other external services.** 16](#_Toc519201899)

[Next Steps: Required Tasks 16](#_Toc519201900)

[Task 1: Project Setup 16](#_Toc519201901)

[Task 2: Setup data structures 16](#_Toc519201902)

[Task 3: Create service to load food info using existing API 16](#_Toc519201903)

[Task 4: Implement UI for Each Activity and Fragment 16](#_Toc519201904)

[Task 5: Create IntentService to send order to server 17](#_Toc519201905)

[Task 6: Create Foreground service with notification about order status 17](#_Toc519201906)

[Task 7: Handle FCM messages 17](#_Toc519201907)

[Task 8: Add login 17](#_Toc519201908)

[Task 9: Add share button for food details 17](#_Toc519201909)

[Task 10: Add widget 17](#_Toc519201910)

**GitHub Username**: Yahhi

# Taifun Food delivery

Description

Food delivery for Kazan, Naberejnie Chelny and Elabuga. Quickly get your lunch at work, or plan a tasty dinner with family. We have a lot of menu options! Find exclusive offers and deals to make your lovely food even cheaper.

You can focus on the things you love to do and have a tasty and healthy food at the same time. No need to choose only one option!

EASY ORDERING

Food is just a few taps away. Get your favorite meals faster with quick reordering.

SCHEDULE DELIVERIES  
Advance ordering allows you to get your food when it’s most convenient for you.  
  
REAL-TIME TRACKING  
See what’s happening with your order. No need to wait without any information.

Intended User

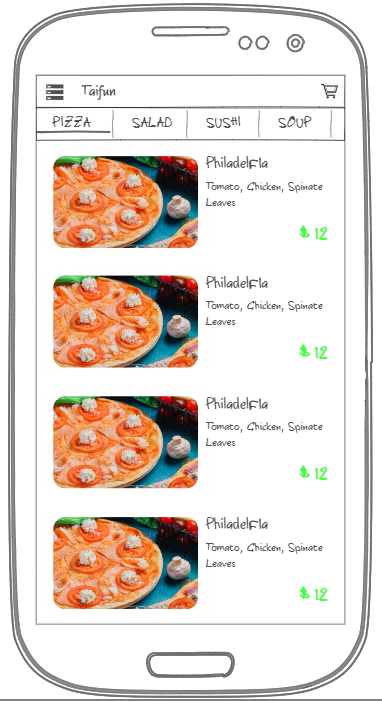
Intended user is firstly a habitant of the selected cities, where Taifun restaurant exists (Kazan, Elabuga, Naberezjnie Chelny). They are adults, who want to order food delivery in a fast and easy way.

Features

* Gives actual menu information (syncing with online catalog)
* Can form an order with multiple dishes
* Can exclude ingredients from the dish (for people with allergies or special taste)
* Saves contact info and addresses for future orders (for home, office, etc.)
* Counts total price for order (can apply discounts and delivery price)
* Can schedule delivery
* Can track order status
* Saves order history

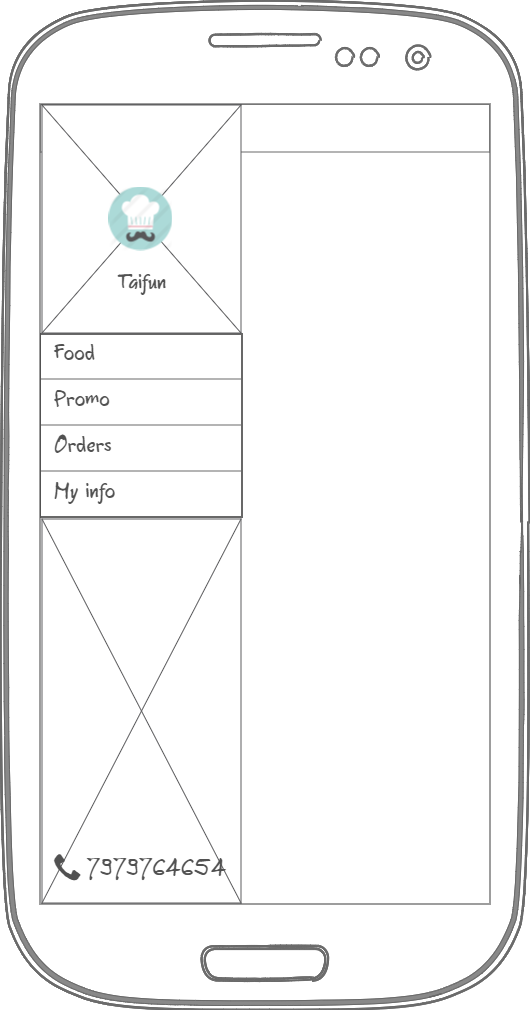
# User Interface Mocks

## Main screen



Shows categorized list of food

## Left drawer menu

****

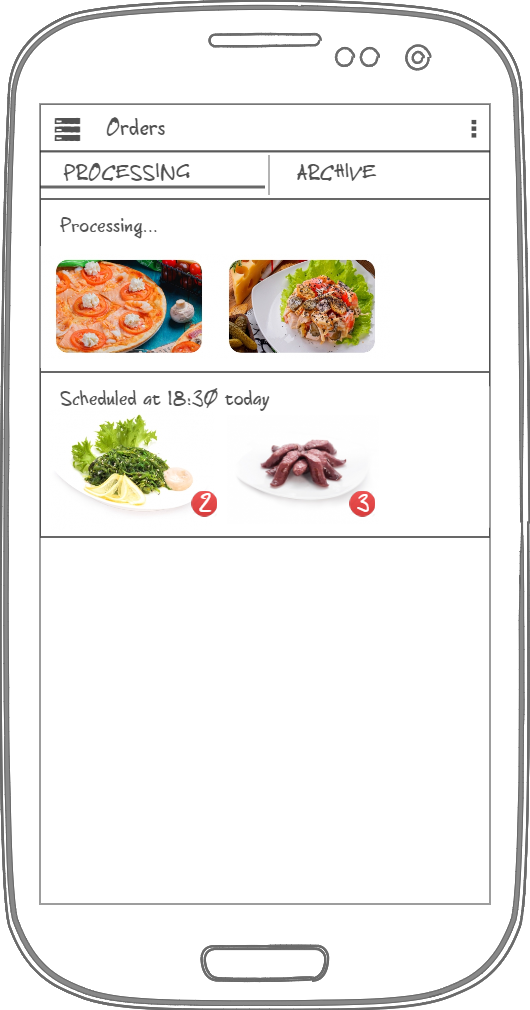
Logo of the restaurant and Menu options

## Orders screen (no orders)



Orders screen with 2 tabs: active orders and archive

## Orders screen (with actual orders)



Orders will show status (processing/scheduled/preparing/ready to pickup/delivering) and food in order (horizontal scrolling list)

## Orders archive

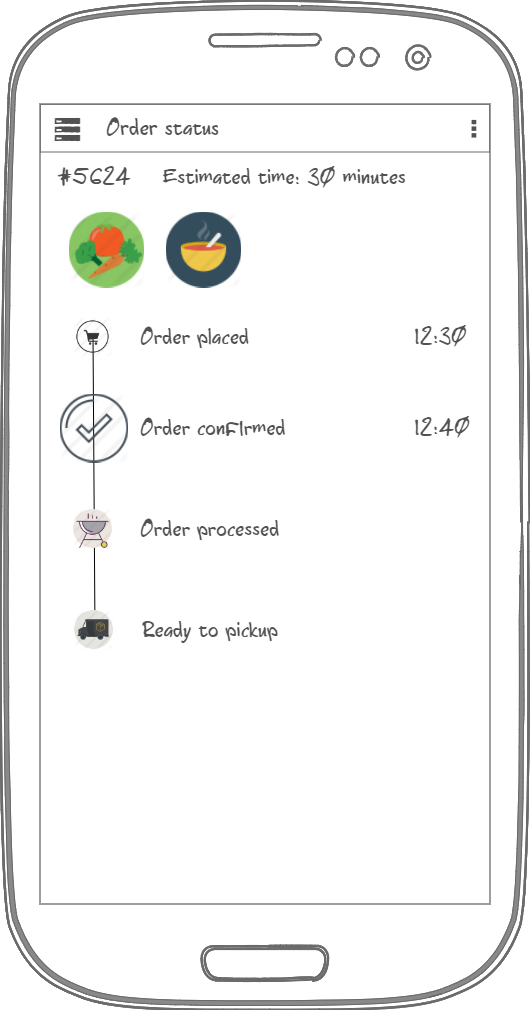


Every order in this list can be reordered

## My info

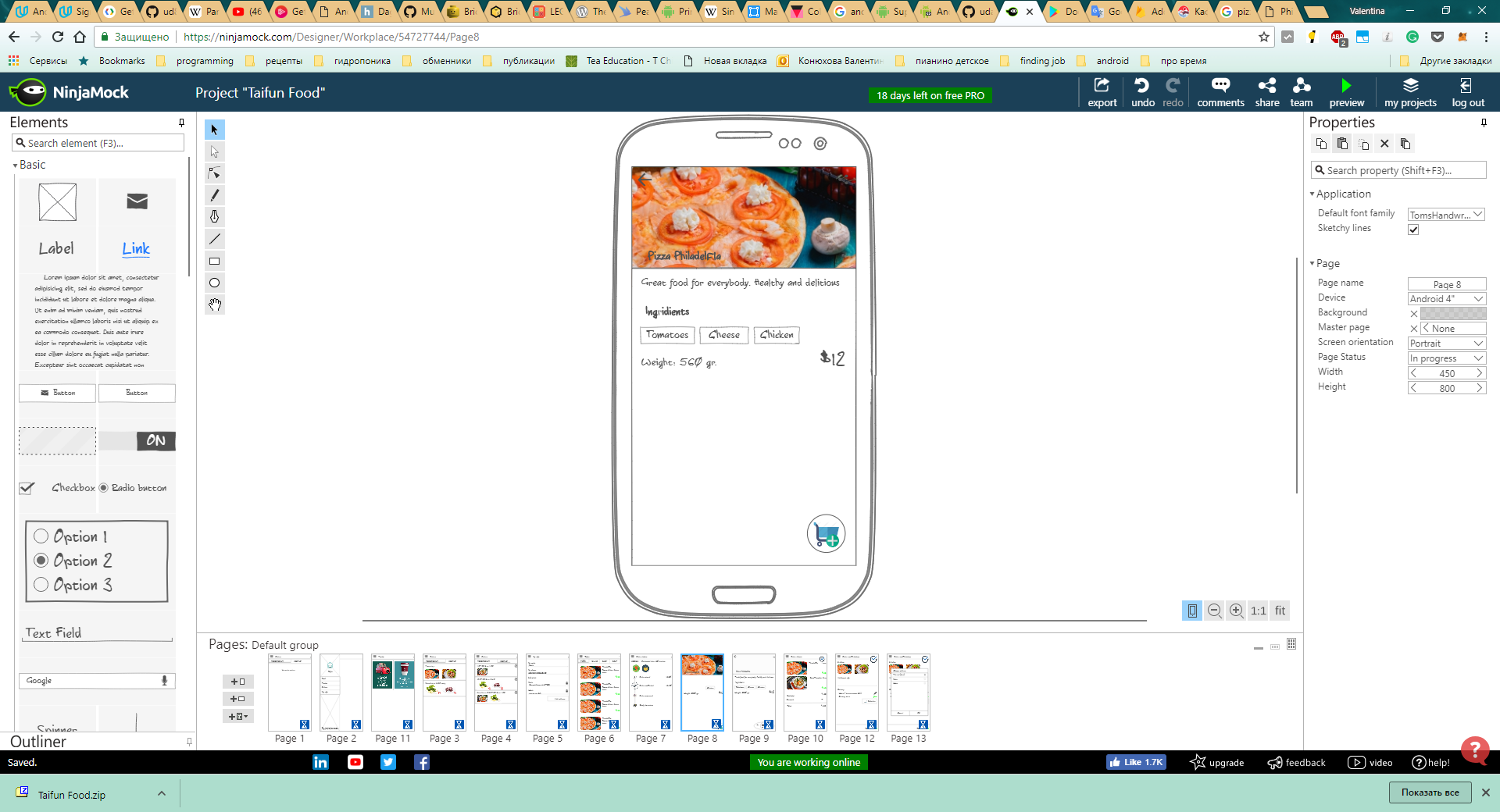


This info is used to make orders. User can add addresses for delivery



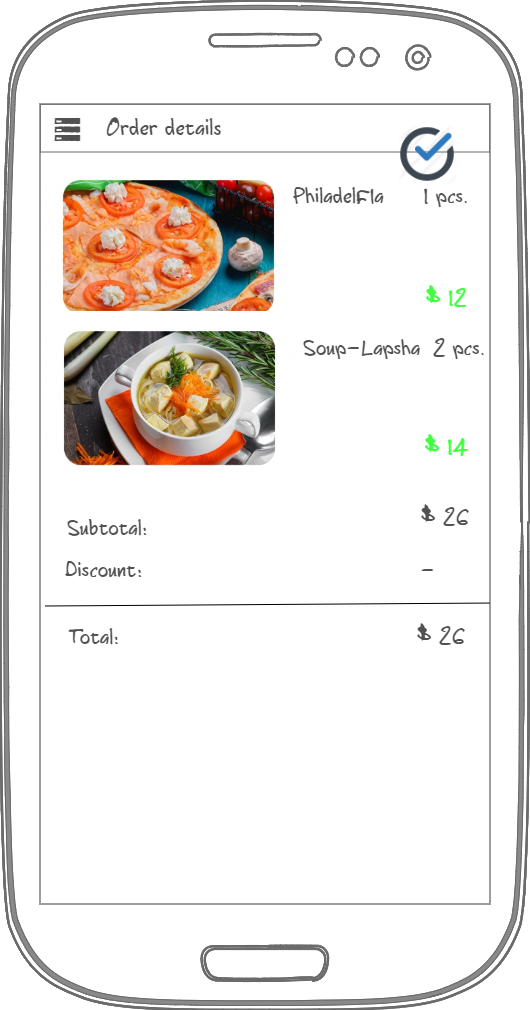
This screen is shown clicking on order in active orders list. There are food list scrolling horizontally If required and steps to prepare order with time if this step is completed

## Food info



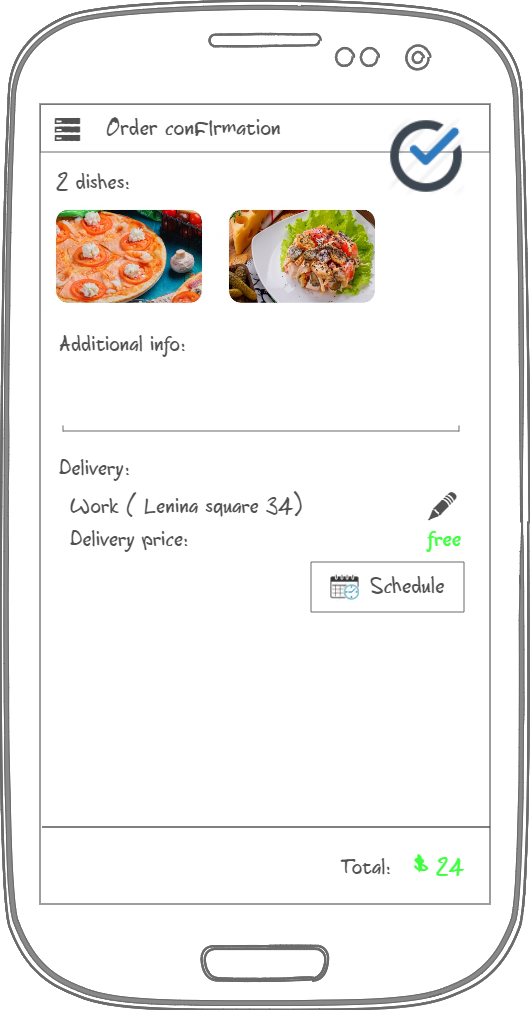
There are food description (optional), ingredients list, weight and price. User can remove ingredients (for example in case of allergy)

## Order details



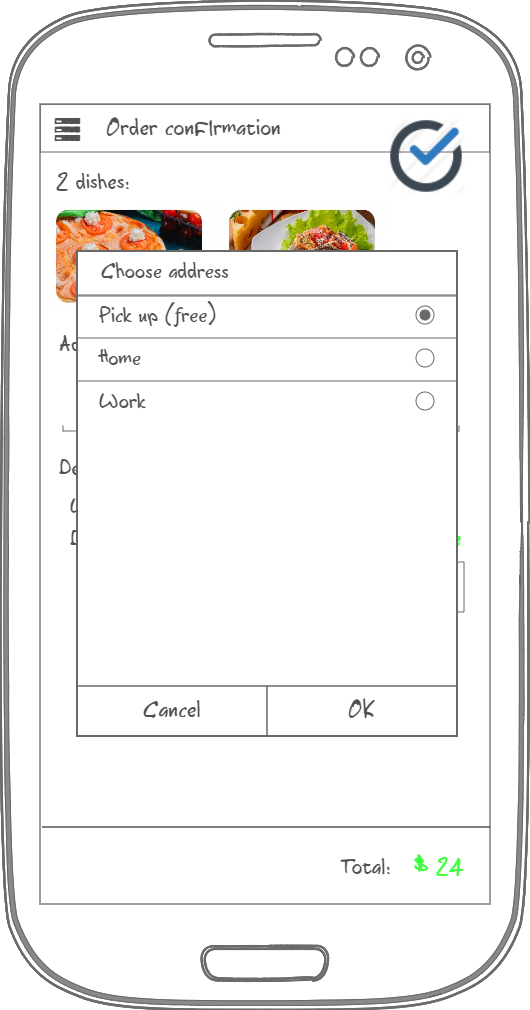
After adding something to order user can see order details with total price. FAB in top right corner is used to proceed to finishing order and setting delivery

## Order confirmation



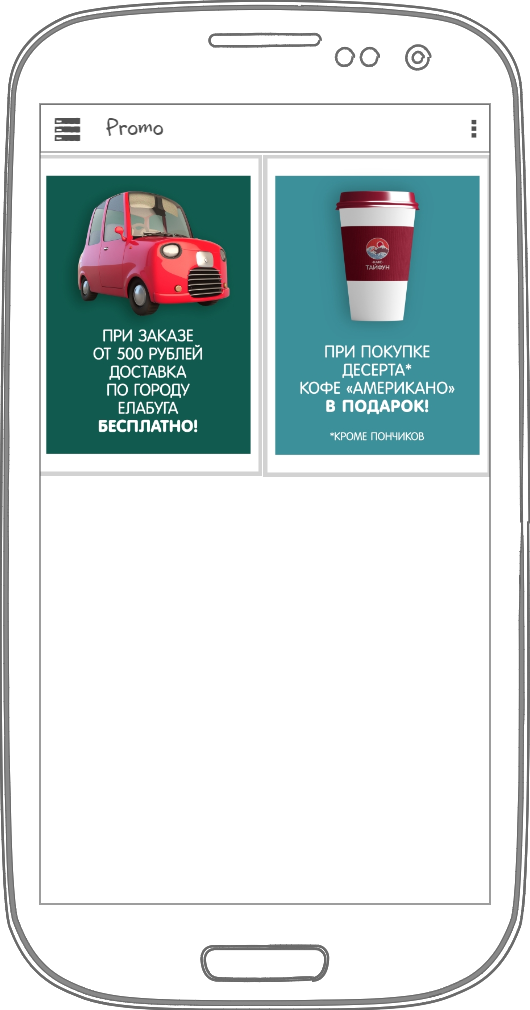
User can add comment to order, select delivery address and schedule delivery to most appropriate time.

## Change address for delivery



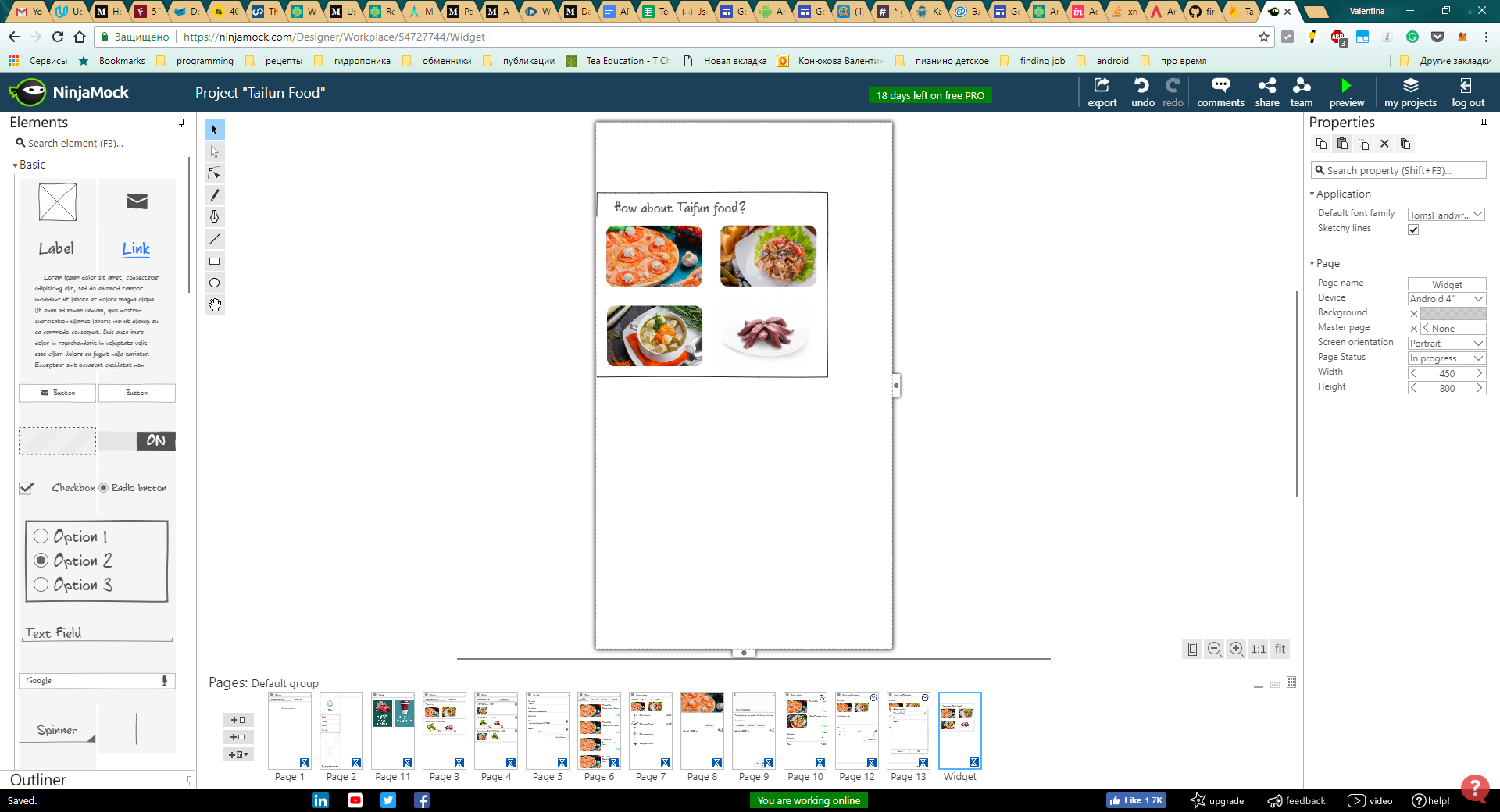
User can select address from existing list.

## Promo



There is a list with all deals and discounts available

## Widget



Widget will show a list of food suggestions to order. There can be new food (that was recently added) or food sorted by user views

# Key Considerations

App will be written solely in the Java Programming language.

App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

Accessibility support: app will use content description on all clickable images and apply directional navigation to all clickable elements.

**How will your app handle data persistence?**

I will use Firebase Database to store order history and user info in the cloud.

I will use Room for saving food info and orders locally.

**Describe any edge or corner cases in the UX.**

* When user clicks “add to order” fab it should disappear and show bottom sheet with +1 -1 buttons, button to see order details and total price.
* When user press back in order details, it should return to last seen food
* When order is in progress, foreground service should be started to show notification about order status and update it from time to time via existing API.

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

Glide 4.7.1 to handle images loading and caching

Retrofit 2.4.0 and GSON 2.8.5 to send order and get up-to-date menu via existing API

Android LiveData and ViewModel (v.1.1.1) to make lifecycle events handling easier

Room 1.1.1 to deal with local database

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

**Firebase Realtime Database (v.16.0.1)** to save user info and orders

**Firebase Auth** **(v.16.0.2)** to make user experience personal

**Firebase crash reporting** **(v.16.0.1)** to be aware of any errors appeared

**Firebase cloud messaging** **(v.17.1.0)** to send some notifications about deals or promos.

Next Steps: Required Tasks

### Task 1: Project Setup

* Setup firebase project in console
* Add dependencies and google-services.json to project

### Task 2: Setup data structures

* Set security rules on Firebase Database
* Create data structures and DAO for local database

### Task 3: Create service to load food info using existing API

Create service that will sync food in local database with food listed on server

### Task 4: Implement UI for Each Activity and Fragment

* Create food list item layout
* Create food details fragment layout
* Create order details fragment layout
* Create layout and menu for left drawer
* Create order confirmation fragment layout
* Create order status fragment layout
* Create user info with addresses layout
* Create promo fragment layout
* Create main activity with left drawer
* Create Food list fragment
* Create Orders fragment
* Create User info fragment
* Create Promo fragment
* Create order details activity
* Create order status activity
* Create order confirmation screen

### Task 5: Create IntentService to send order to server

Task 6: Create Foreground service with notification about order status  
  
It will ask server from time to time about order status and update progress

### Task 7: Handle FCM messages

Show notification when receiving FCM about promos.

### Task 8: Add login

* Add login using google button and functionality to info screen
* Let user save and load his personal data (addresses, phone, orders history) in the cloud so this data will not be lost after moving to another device.

### Task 9: Add share button for food details

### Task 10: Add widget