

Приложение,

REStaura

которое помогает пользователю подобрать рестораны на основе выбранных кухонь.

Идея

RESTaurant — это iOS-приложение для эффективного подбора ресторанов.

Теперь не будет споров и долгих раздумий на выбором места, где провести хорошо время и вкусно поесть,

Зачем это нужно

- анализирует предпочтения
- персонализирует подборки
- экономит время

Целевая аудитория

- 1 Кто: Жители и гости Москвы (25–45 лет), ценящие гастрономический опыт.
- 2 Интересы: Кулинария, новые места, удобство.
- 3 Привычки: Планируют ужин заранее, но открыты к спонтанным решениям.

ДИЗАЙН

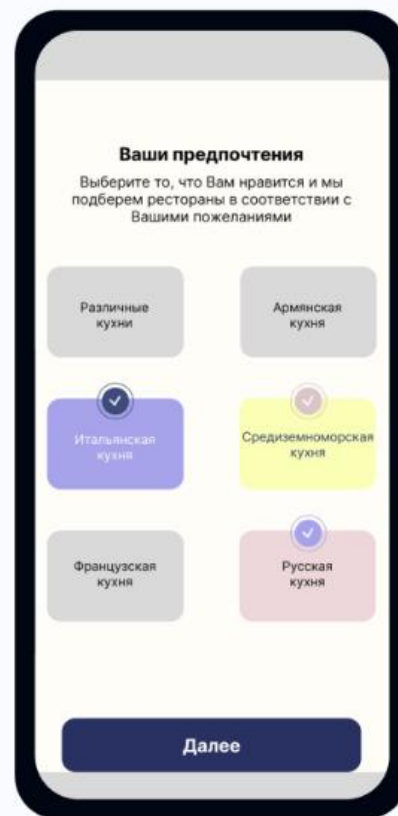
Home Screen



Screen Page



Choose Page



ДИЗАЙН

Sign Up

Регистрация

email

имя

имя пользователя

телефон

пароль

Далее

Есть аккаунт? Войти

The Sign Up screen features a light yellow background with a gray header and footer. It contains a registration form with five input fields: email, name, username, phone number, and password. A dark blue button labeled 'Далее' (Next) is at the bottom of the form. A link 'Есть аккаунт? Войти' (Have an account? Log in) is located at the very bottom.

Sign In

Крутая картинка

имя пользователя

пароль

Войти

Нет аккаунта? Зарегистрироваться

The Sign In screen has a light yellow background with a gray header and footer. It features a large gray placeholder for a 'Крутая картинка' (Cool picture). Below it are input fields for 'имя пользователя' (username) and 'пароль' (password), followed by a dark blue 'Войти' (Log in) button. A link 'Нет аккаунта? Зарегистрироваться' (No account? Register) is at the bottom.

Account

Имя пользователя

Избранное История Рекомендации

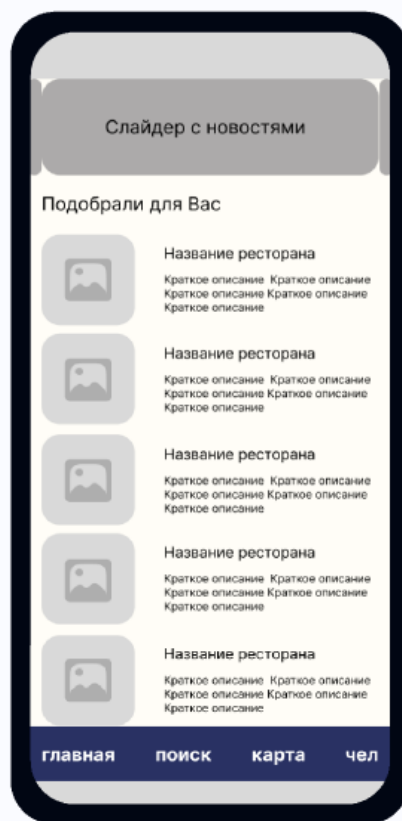
Название ресторана

главная поиск карта чел

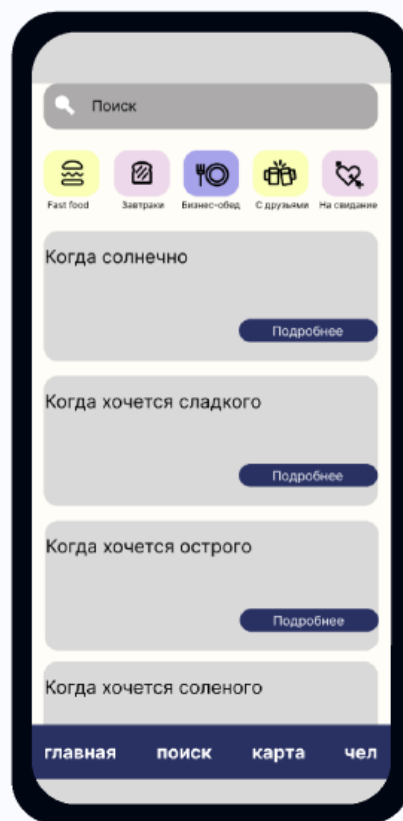
The Account screen has a light yellow background with a gray header and footer. It displays the user's 'Имя пользователя' (username) and a settings gear icon. Below are three tabs: 'Избранное' (Favorites), 'История' (History), and 'Рекомендации' (Recommendations). A large gray placeholder for a restaurant image is shown, with the label 'Название ресторана' (Restaurant name) underneath. At the bottom is a dark blue navigation bar with four items: 'главная' (Home), 'поиск' (Search), 'карта' (Map), and 'чел' (Profile).

ДИЗАЙН

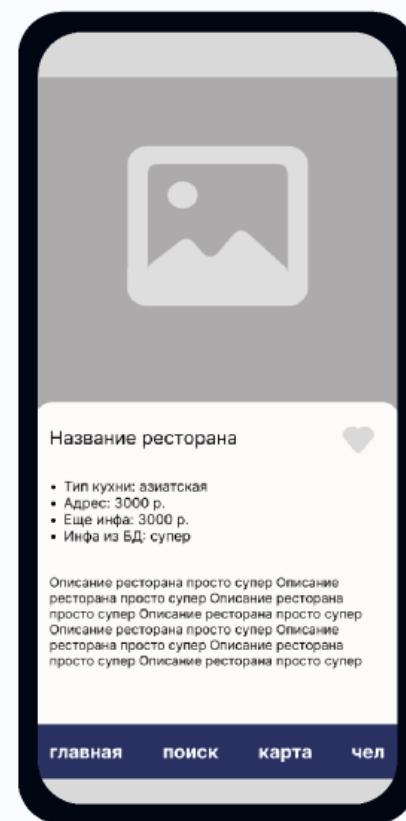
Restaurants



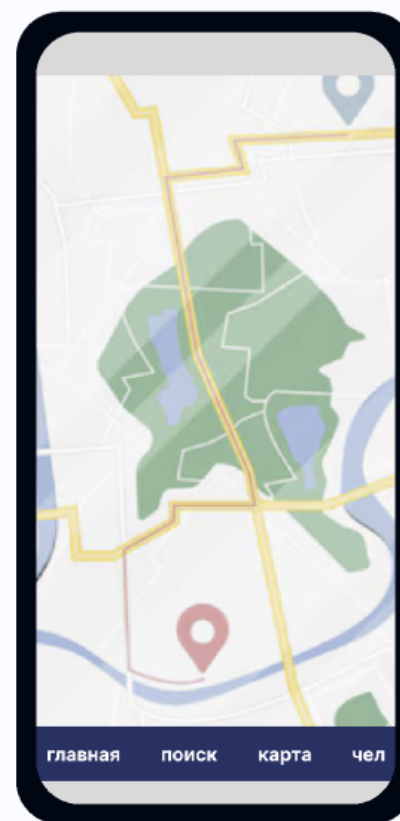
Product Page



Place Page

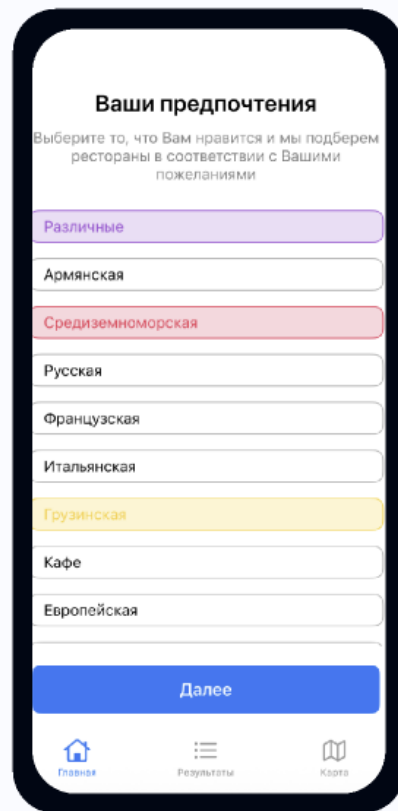


Map

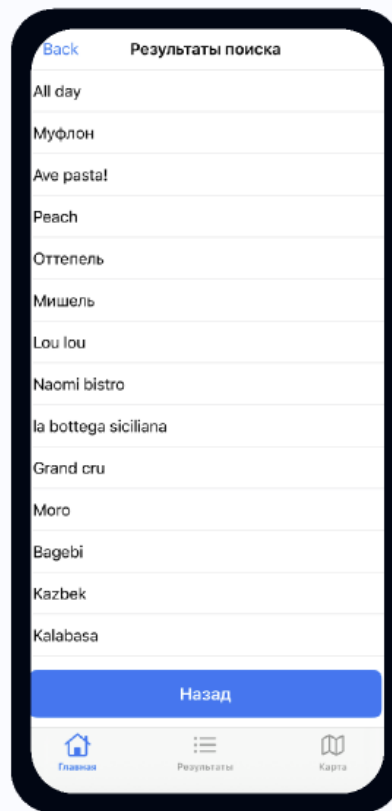


РЕАЛИЗАЦИЯ

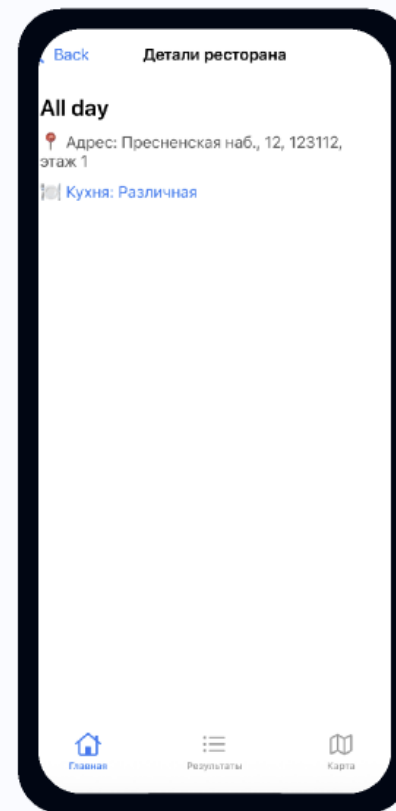
Choose



Recomendations

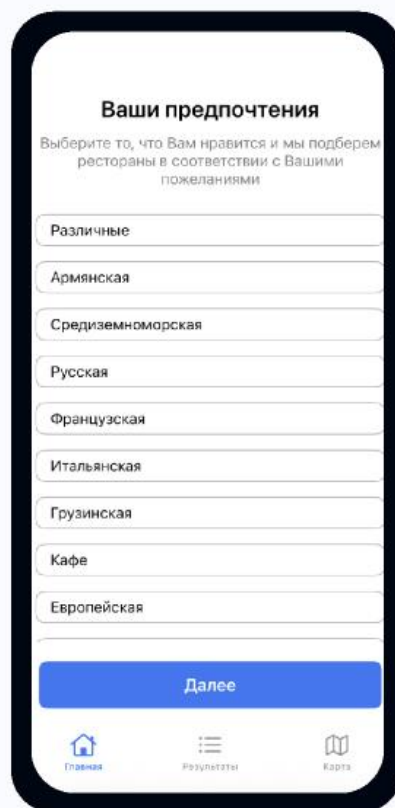


Place Page

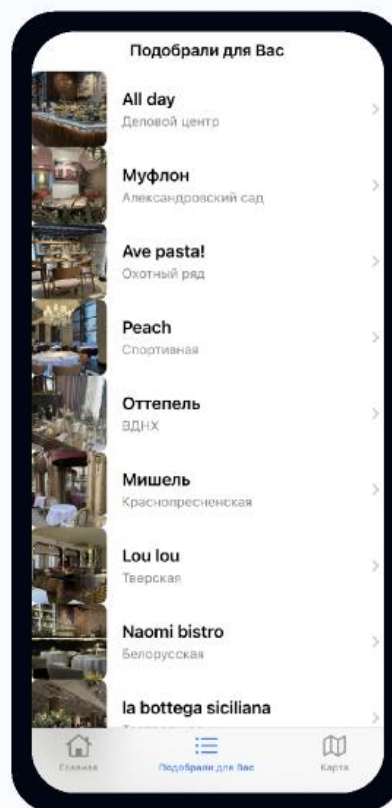


РЕАЛИЗАЦИЯ

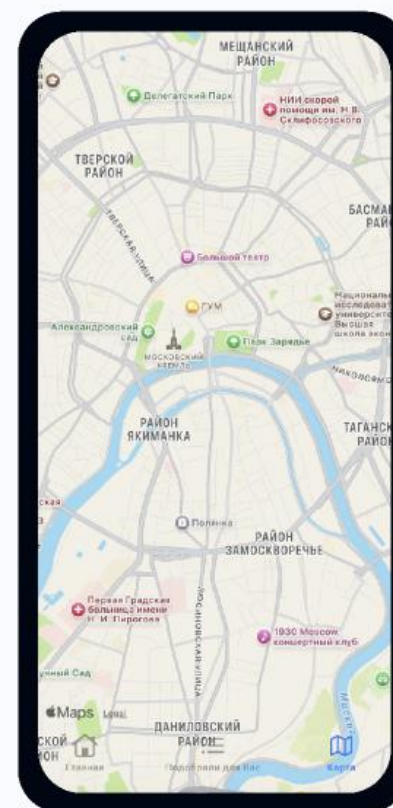
Choose



Restaurants



Map



ОБЗОР

Ваши предпочтения

Выберите то, что Вам нравится и мы подберем рестораны в соответствии с Вашими пожеланиями

Различные

Армянская

Средиземноморская

Русская

Французская


Итальянская


Грузинская


Кафе

Европейская

Далее

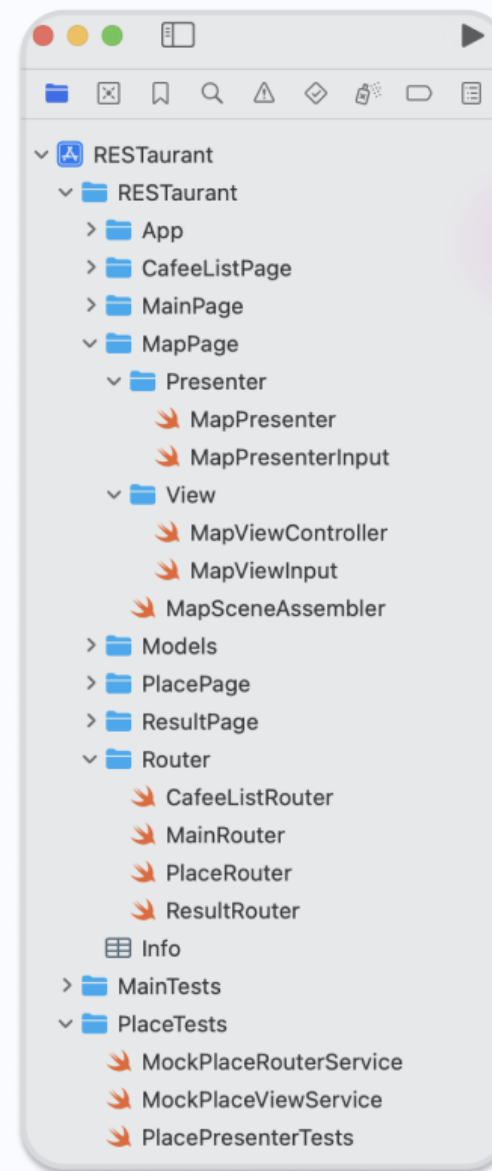

Главная


Подобрали для Вас


Карта

Архитектура

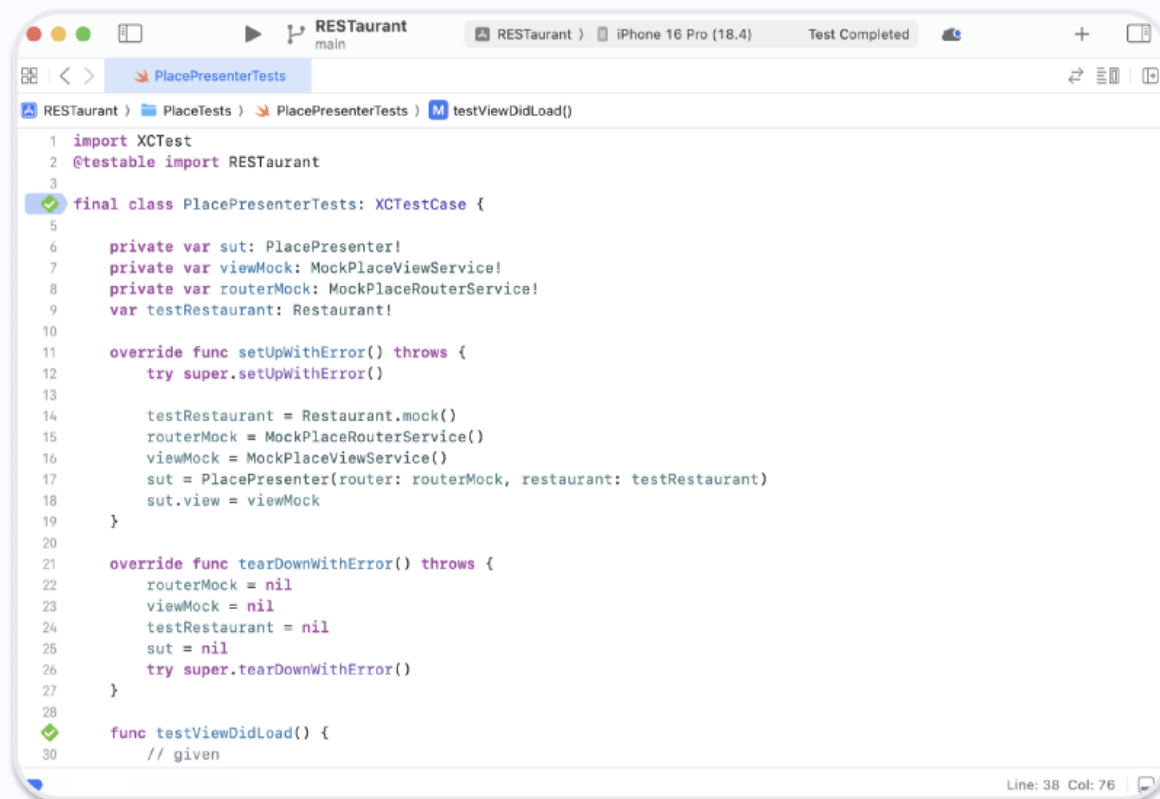
- Все папки с окончанием Page содержат подпапки Presenter, View и файл SceneAssembler.swift.
- Логика навигации вынесена в папку Router.
- Модели данных находятся в папке Models.



ТЕСТЫ

В проекте реализованы тесты:

- MainTest — тестирование главной страницы и бизнес-логики.
- PlaceTest — тестирование функционала страницы ресторана.



The screenshot shows an Xcode editor window with the following details:

- Toolbar:** Includes standard macOS window controls (red, yellow, green buttons), a play button, and a location indicator showing 'RESTaurant' and 'main'.
- Navigation Bar:** Displays the current file path: 'RESTaurant' > 'PlaceTests' > 'PlacePresenterTests' > 'testViewDidLoad()'. It also shows a 'Test Completed' status and a zoom level of 'iPhone 16 Pro (18.4)'.
- Code Editor:** Contains Swift code for the 'PlacePresenterTests' class, which inherits from ' XCTestCase '. The code includes:
 - Imports for 'XCTest' and 'RESTaurant'.
 - A 'final class' definition for 'PlacePresenterTests'.
 - Private variables for 'sut' (PlacePresenter!), 'viewMock' (MockPlaceViewService!), 'routerMock' (MockPlaceRouterService!), and 'testRestaurant' (Restaurant!).
 - An 'override func setUpWithError()' method that initializes the test environment with mocks and the 'sut' object.
 - An 'override func tearDownWithError()' method that cleans up the test environment by setting variables to 'nil'.
 - A 'func testViewDidLoad()' method with a comment '// given'.
- Status Bar:** At the bottom right, it indicates 'Line: 38 Col: 76'.