



Мобильное приложение для работы пиццерии Joe Mama's Pizza

Город: Москва

Институт: РТУ МИРЭА

Студент: Середа Илья Андреевич

Группа: ИВБО-07-19

Руководитель: Степанов Павел Валерьевич

Введение

В наше время все больше людей предпочитают заказывать еду через мобильные приложения, так как это удобно и быстро. Сеть пиццерий “Joe Mama's Pizza” должна следовать этому тренду и предоставить клиентам такую же возможность.

Для повышения эффективности работы пиццерии необходимо разработать мобильное приложение со всеми возможностями очного посещения и даже больше: заказа пиццы дистанционно, которую получить можно самовывозом или доставкой, сохранение заказов в историю заказов и так далее.

Мобильное приложение также поможет увеличить продажи, так как многие люди предпочитают заказывать еду онлайн. Кроме того, мы сможем собирать данные о наших клиентах, что поможет нам лучше понимать их потребности и предоставлять более персонализированный сервис.



Целевая аудитория

Целевая аудитория пиццерии охватывает людей с разными вкусовыми предпочтениями:

- люди возрастом от 20 до 40 лет (70%);
- молодые люди до 20 лет (25%);
- остальные (5%).

Важным для всех них является скорость обслуживания и цена.



Существующие аналоги

У разрабатываемого приложения имеются прямые и косвенные аналоги-конкуренты.

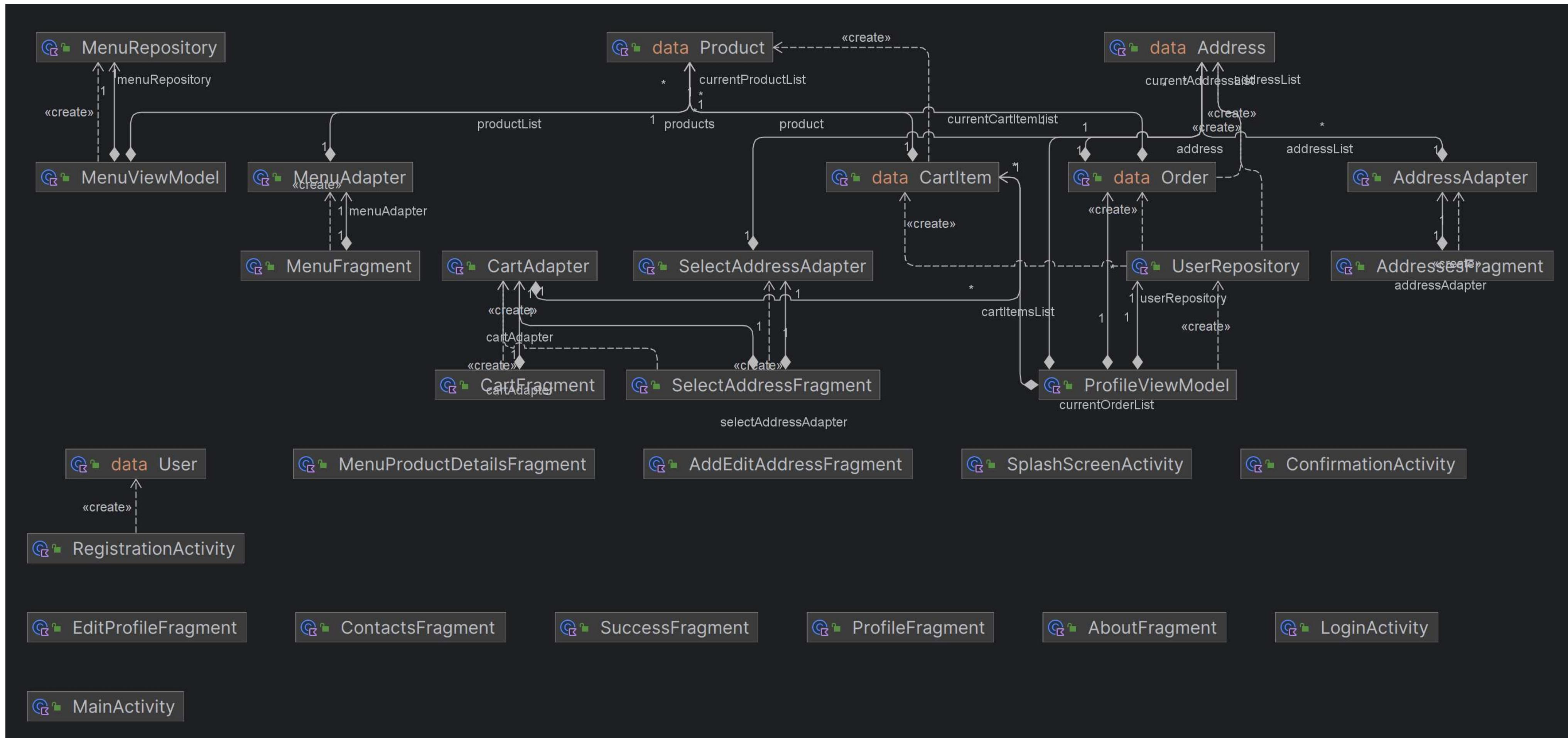
Прямыми аналогом является мобильное приложение «Алло! Пицца», реализующее сервис для заказа и доставки товаров из ближайшей по местоположению пиццерии.



К косвенным аналогам можно отнести мобильное приложение “FlowWow”, которое является площадкой для салонов цветов, которые размещают там свои товары, предназначенное для заказа и доставки, в основном, букетов цветов, а также различных других товаров.



Диаграмма классов



Используемые зависимости

```
dependencies {  
    implementation 'androidx.core:core-ktx:1.10.1'  
    implementation 'androidx.appcompat:appcompat:1.6.1'  
    implementation 'com.google.android.material:material:1.9.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
  
    // Lifecycle components  
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.6.1'  
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.6.1'  
  
    // Navigation components  
    implementation 'androidx.navigation:navigation-fragment-ktx:2.5.3'  
    implementation 'androidx.navigation:navigation-ui-ktx:2.5.3'  
  
    // Firebase components  
    implementation platform('com.google.firebaseio:firebase-bom:31.4.0')  
    implementation 'com.google.firebase:firebase-database-ktx:20.2.1'  
    implementation 'com.google.firebase:firebase-auth-ktx'  
  
    // Glide component  
    implementation 'com.github.bumptech.glide:glide:4.15.1'  
  
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'  
  
    // Test components  
    implementation 'androidx.navigation:navigation-testing:2.5.3'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'  
    androidTestImplementation 'androidx.test.espresso:espresso-contrib:3.5.1'  
    androidTestImplementation 'androidx.test:runner:1.5.2'  
    //noinspection GradleDependency  
    androidTestImplementation 'androidx.test:rules:1.2.0'  
}
```

- Lifecycle компоненты
- Nagivation компоненты
- Firebase Database
- Firebase Auth
- Glide
- JUnit 4

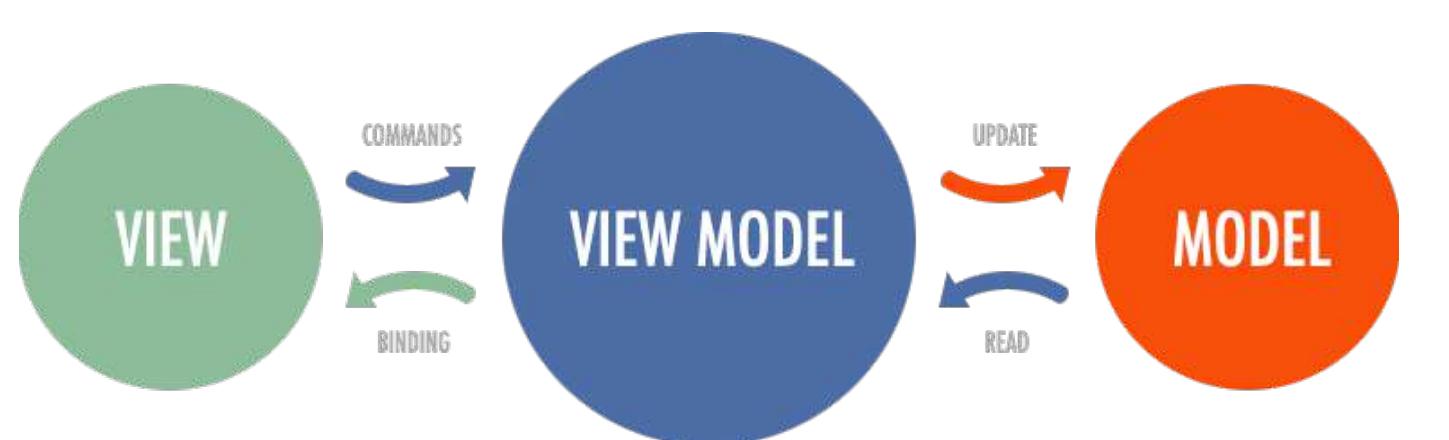


Архитектура проекта

В качестве шаблона проектирования архитектуры клиентского приложения был выбран MVVM (Model-View-ViewModel), так как разделение задач в MVVM позволяет легко тестировать каждый компонент отдельно. Особенно легко тестировать ViewModel, поскольку она не зависит от пользовательского интерфейса, что делает возможным тестирование в изоляции.

Также использование данного паттерна позволяет иметь возможность повторного использования: Модель ViewModel может использоваться совместно различными представлениями, что позволяет более эффективно повторно использовать код. Это может привести к значительному сокращению времени разработки, особенно при создании больших приложений.

В качестве альтернативы реализации серверного решения используется готовая облачная реализация вида Mobile Backend as a Service (MBaaS) от компании Google - Google Firebase.



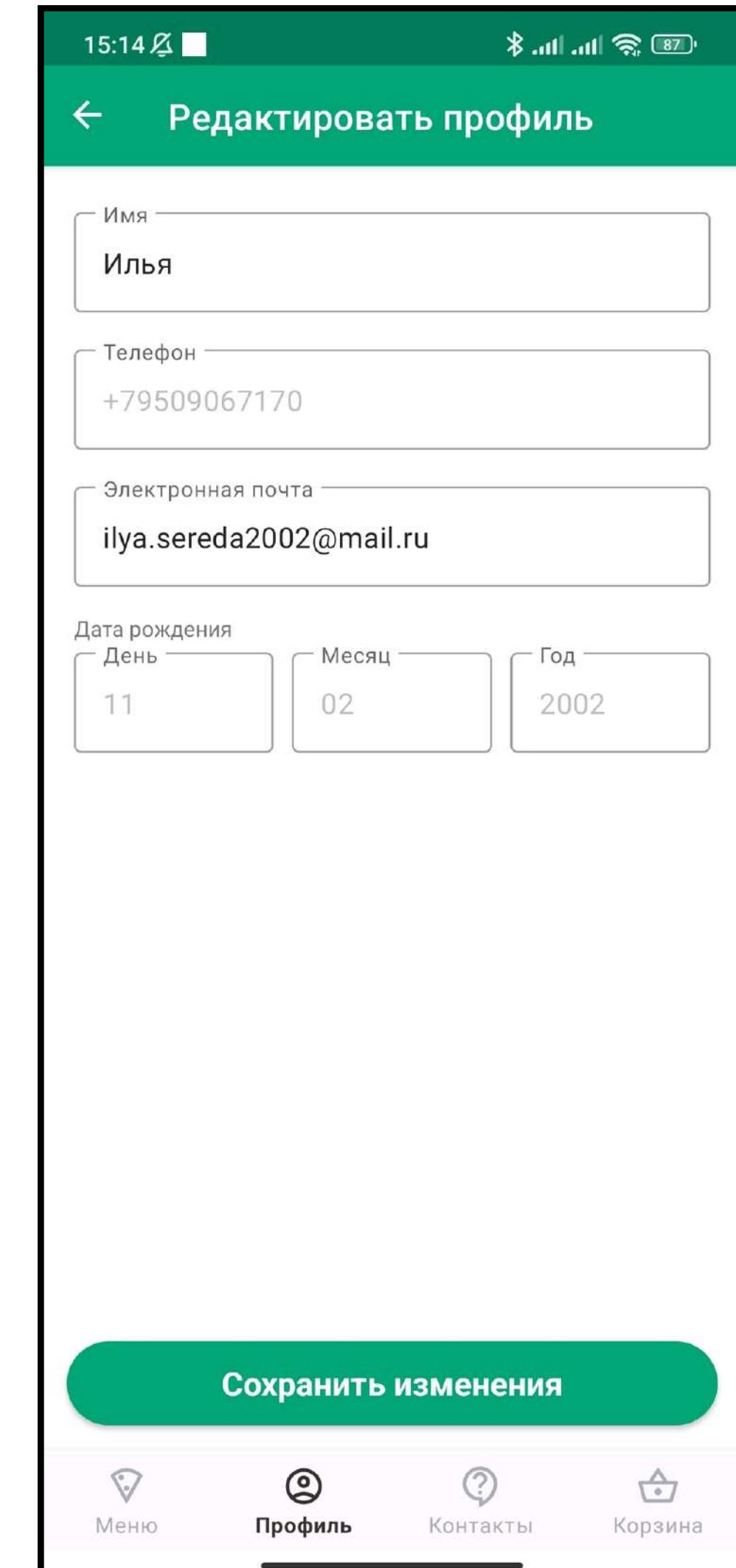
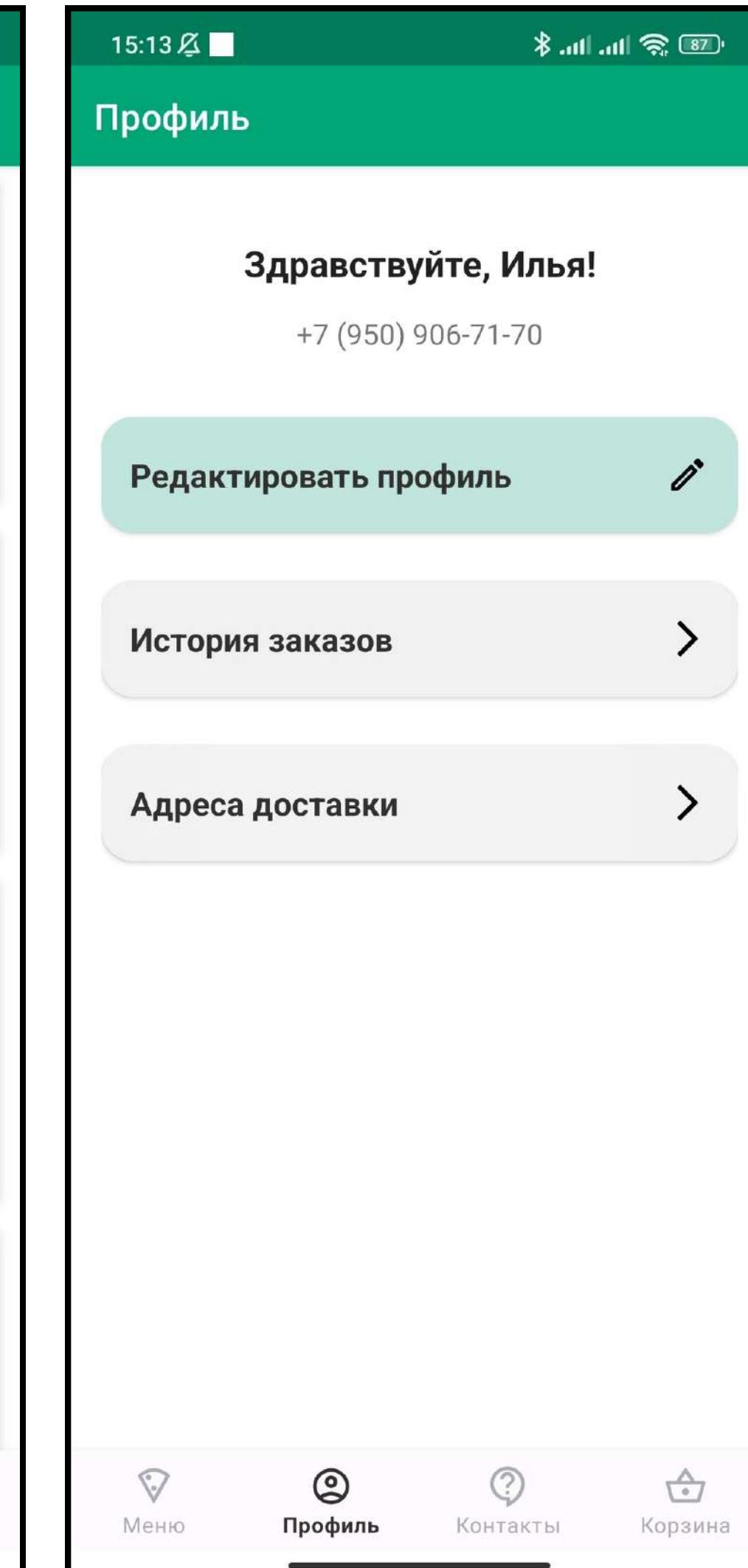
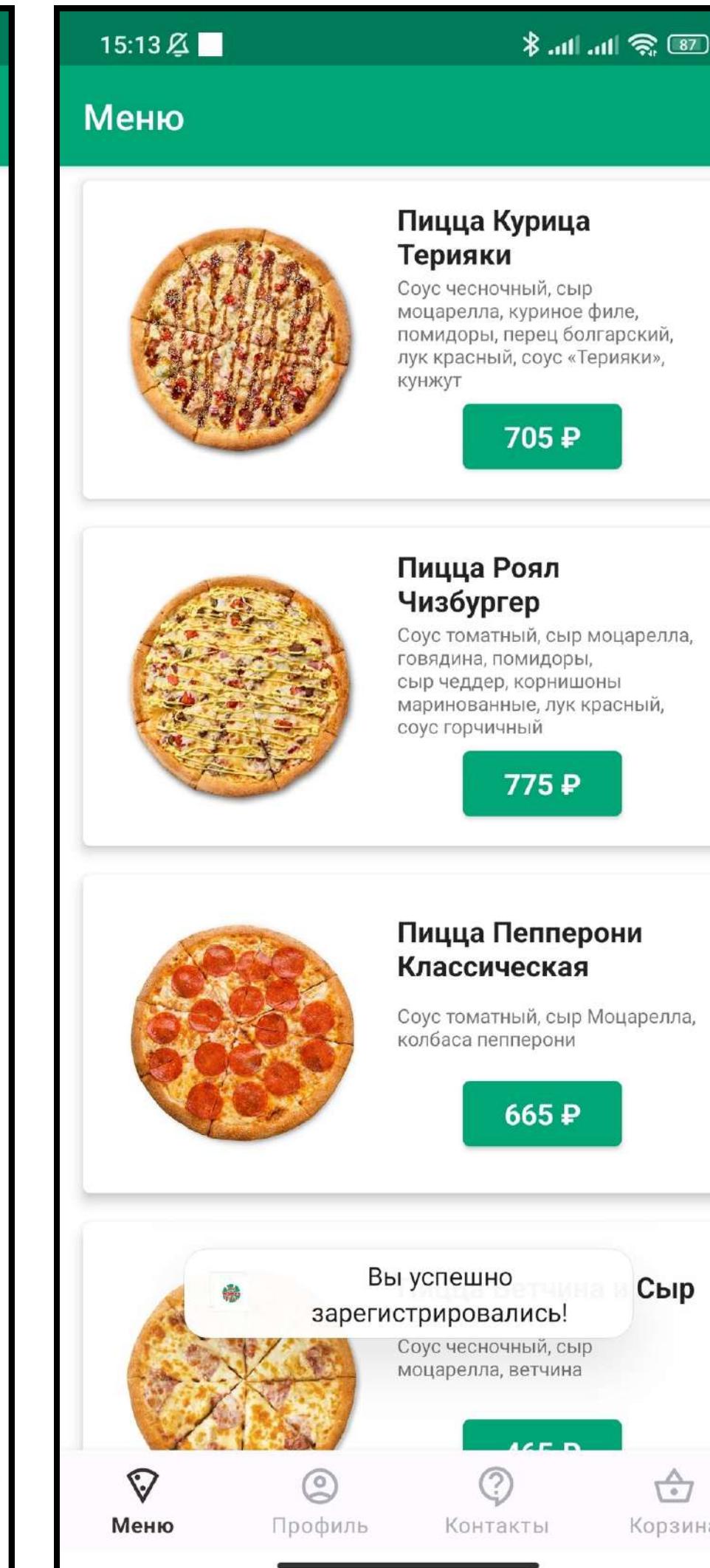
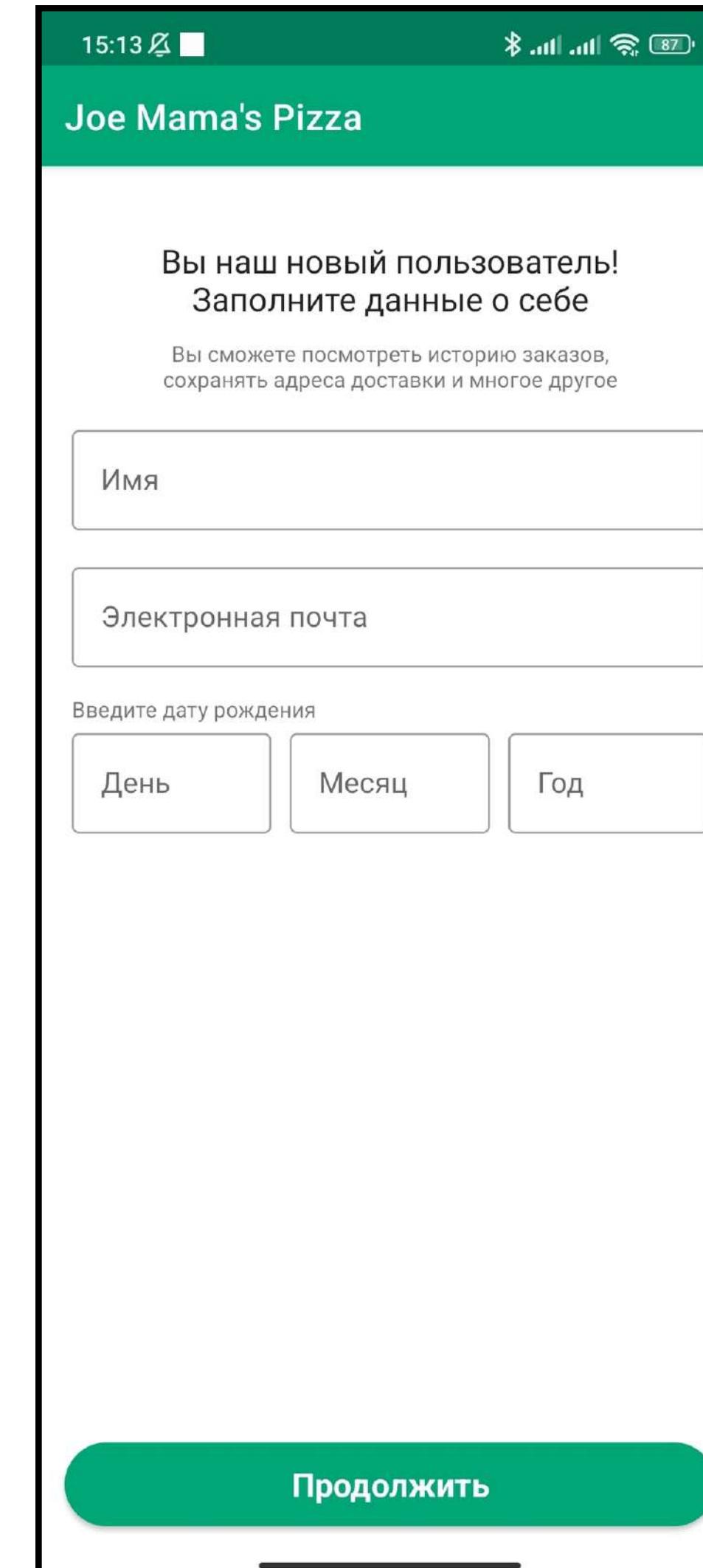
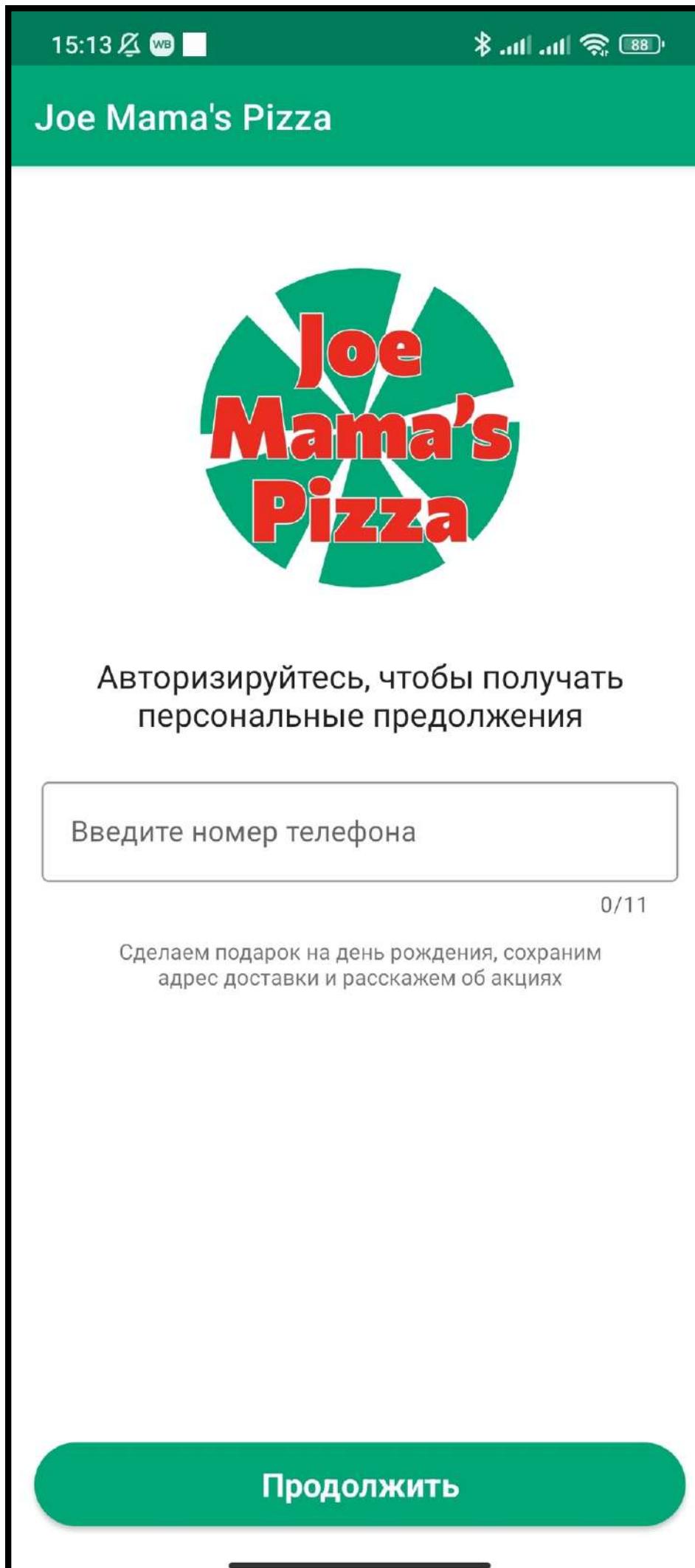
Уже реализовано

Реализован следующий функционал приложения:

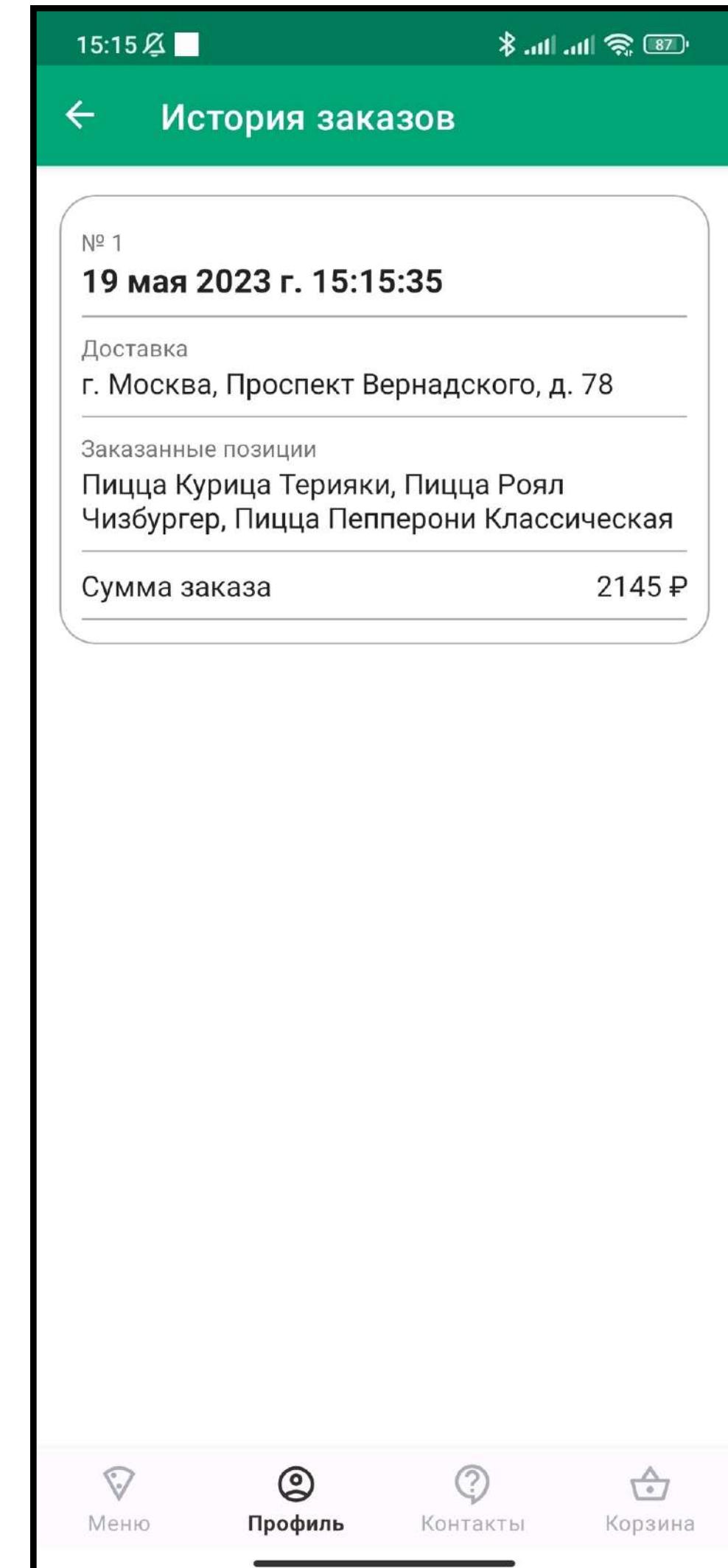
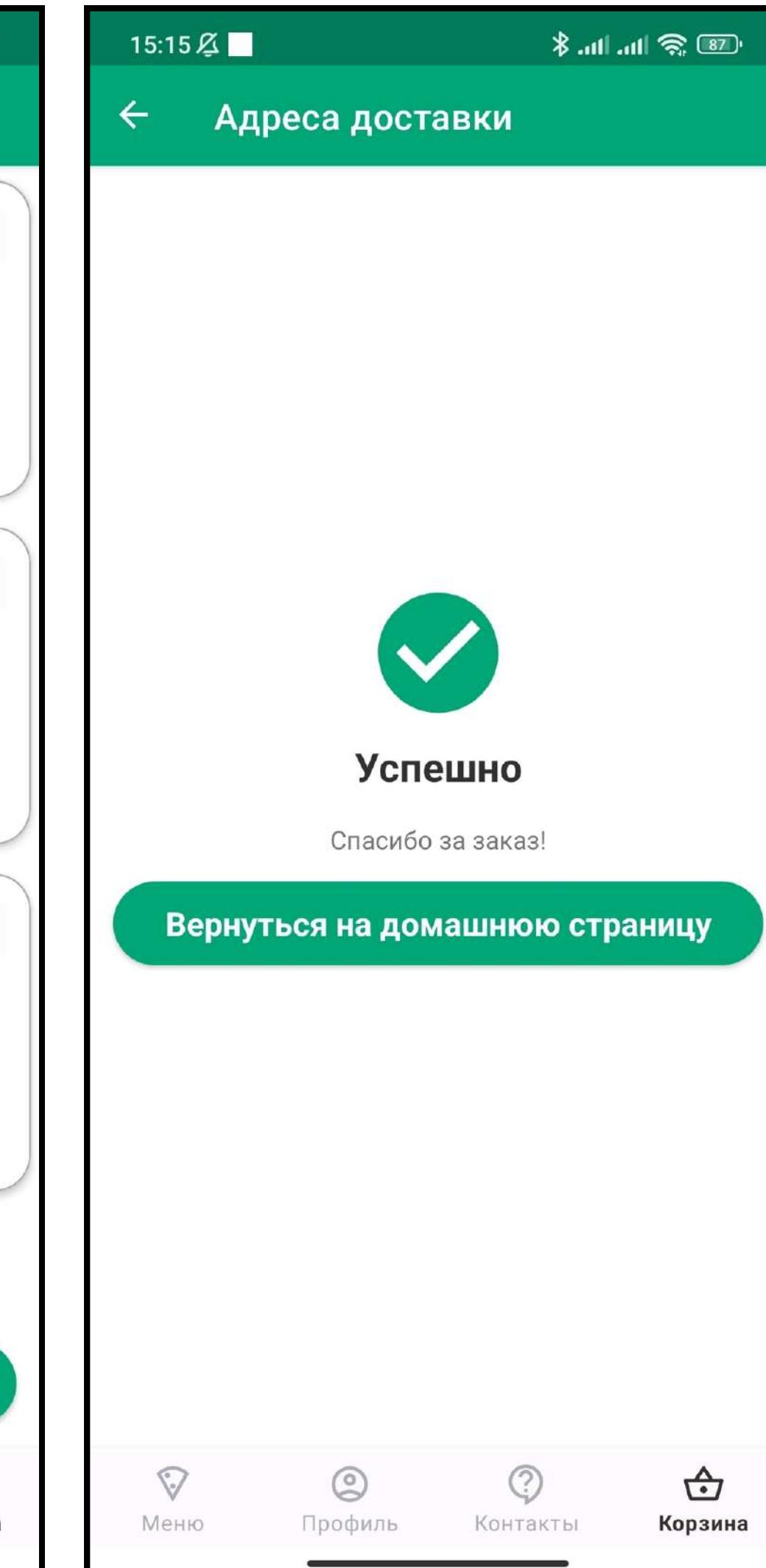
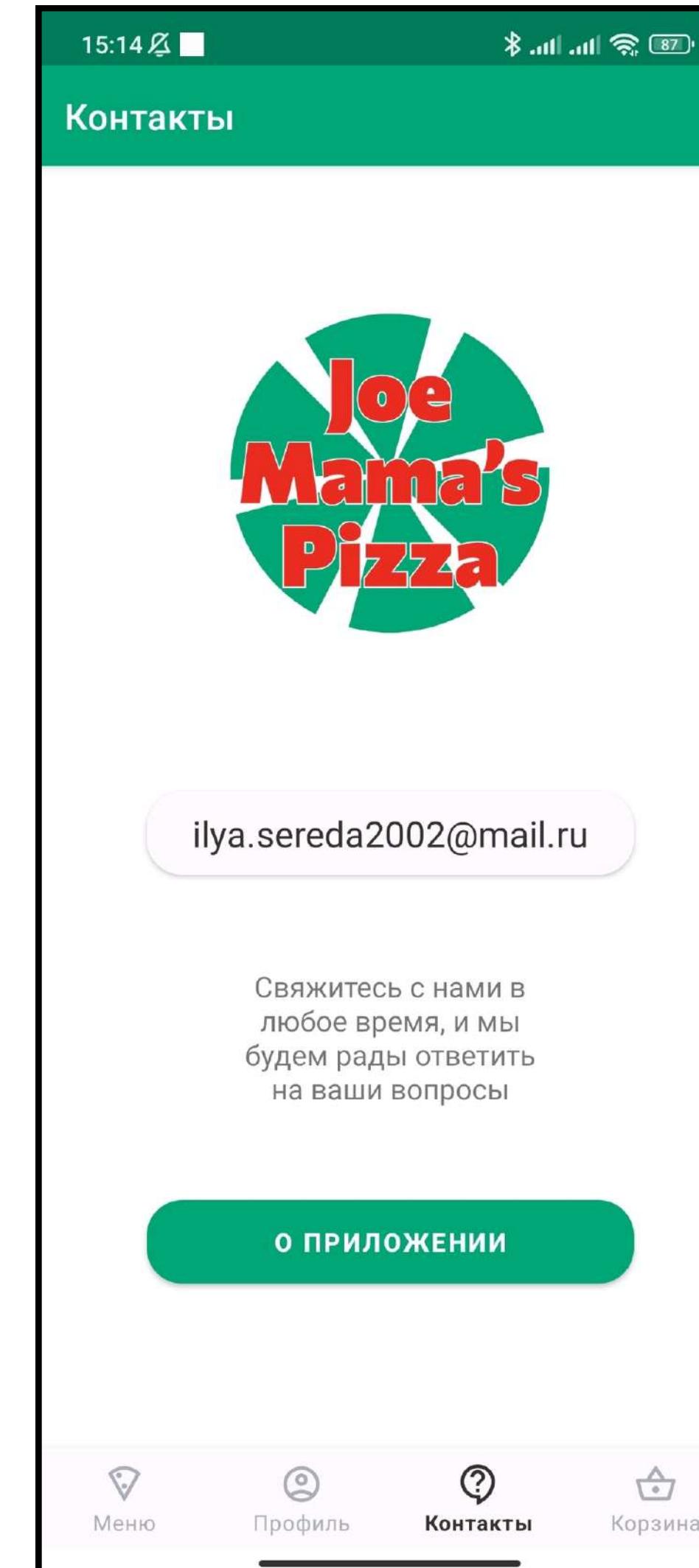
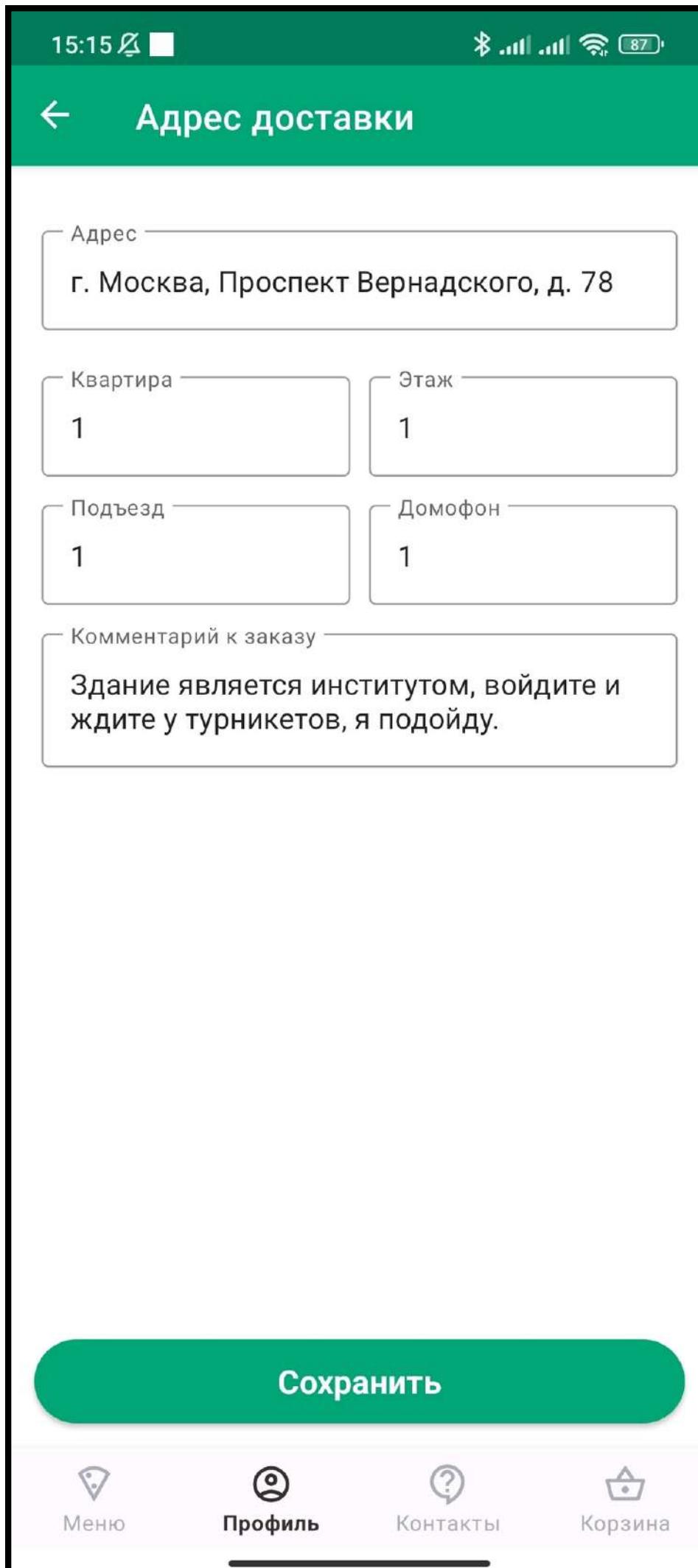
- отображение каталога меню;
- возможность выбора определенной позиции и добавления её в корзину;
- личный кабинет пользователя;
- добавление адресов доставки;
- регистрация и аутентификация;
- корзина выбранных продуктов;
- оформление заказа;
- отображение информации о приложении;
- отображение оформленных заказов в профиле;
- редактирование данных пользователя.



Скриншоты приложения



Скриншоты приложения



Планы по развитию

- Добавить карту и возможность определения местоположения, благодаря которым можно удобнее определять адрес доставки.
- Добавить разделение категорий для продуктов.
- Добавить отслеживание состояния заказа.
- Добавить возможность более подробного оформления заказа с выбором вида оплаты.
- Добавить приложение в Google Play.





Спасибо за внимание!

[Ссылка на видеоролик с демонстрацией работы приложения](#)

[Ссылка на облако с проектом](#)