МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

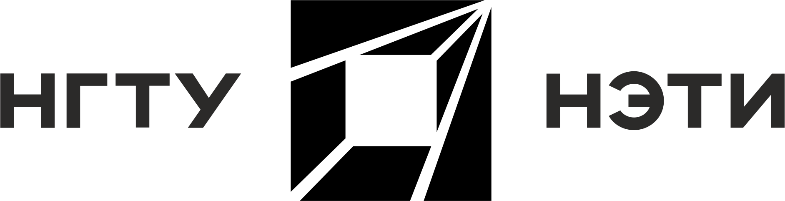
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики



**Техническое задание**

по дисциплине: Разработка мобильных приложений

Факультет: ФПМИ

Группа: ПММ-21

Студенты: Сухих А.С., Черненко Д.А., Бариев Р.Р.

Преподаватель: Самочернов Игорь Валентинович

Новосибирск 2022

**1. Название проекта**

Система оплаты покупок.

**2. Цель**

Сделать покупки товаров более простыми, удобными и быстрыми с помощью моментальной системы оплаты покупок путем сканирования телефоном товаров с NFC-метками или QR-кодами. Это позволит сократить очереди посетителей магазинов и внедрить, либо распространить программу лояльности для пользователей.

**3. Характеристики решения**

Система будет выполнена в виде фронтенда, представленного мобильным приложением для ОС Android, устанавливаемым пользователем через магазин приложений и бэкенда. Бэкенд будет состоять из базы данных, хранящей информацию о товарах, пользователях и их покупках, а также биллинговой системы, обрабатывающей платежи и покупки пользователей.

На каждый товар будет наклеен QR-код и NFC-метка, определяющие его в системе. Для оплаты покупатель будет использовать NFC-сканер своего телефона, либо камеру для считывания QR-кода. Поскольку для оплаты банковскими картами требуется подключение интернет-эквайринга, требующего согласование приложения с банком, оплата будет производиться с виртуального счета приложения.

Мобильное приложение будет разработано на языке Kotlin, API - на Python, а в качестве СУБД будет использован PostgreSQL.

**4. Структура проекта**

**4.1 Экраны**

Приложение содержит 4 основных экрана:

* авторизации;
* сканера товаров;
* корзина;
* личный кабинет

Окно авторизации будет открыто при запуске приложения. В нем пользователь может ввести свой логин и пароль для входа, либо заполнить форму регистрации.



Рисунок 1. Макет интерфейса авторизации

После входа в приложение пользователю открывается интерфейс сканирования товаров. Для записи товара в электронную корзину необходимо считать QR-код камерой или поднести телефон с включенным NFC к NFC-метке, расположенной на товаре.

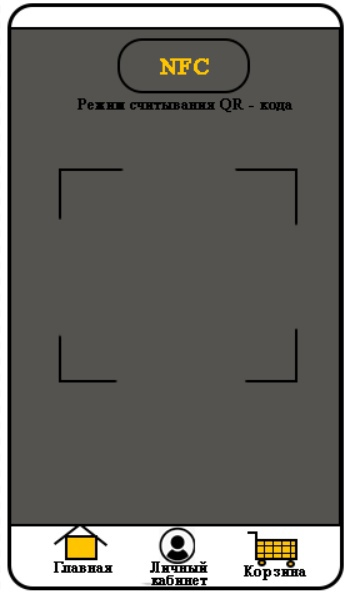


Рисунок 2. Макет интерфейса сканирования товаров

Для оплаты отсканированных товаров пользователь переходит в Корзину, где подтверждает свои покупки и нажимает кнопку “Оплатить”

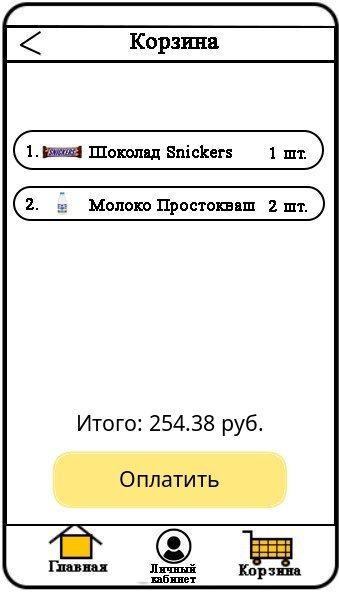


Рисунок 3. Макет интерфейса корзины

Пользователь может посетить свой личный кабинет для изменения каких-либо настроек программы и своего профиля, просмотра баланса и истории счета, бонусов и скидок.

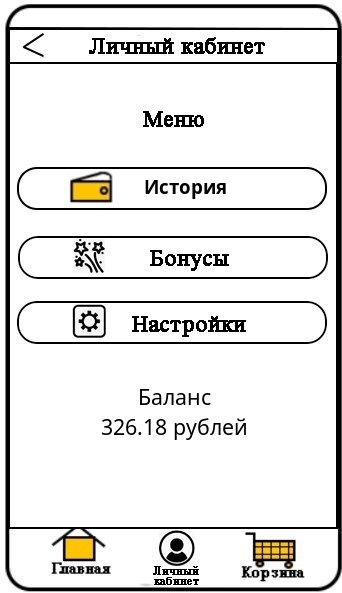


Рисунок 4. Макет интерфейса личного кабинета

**4.2 API**

С помощью API будет производиться обработка оплаты товаров, поиск необходимых товаров при сканировании, а также авторизация. Авторизация выполняется с помощью токена, получаемого от API при авторизации. Данный токен необходимо передавать при запросе к методам API. Программный интерфейс, реализующий взаимодействие мобильного приложения с базой данных будет выполнен в виде REST API со следующими ресурсами:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| URI | Метод | Описание | Входные п-ры | Выходные п-ры |
| /login | POST | Авторизация | телефон, пароль | токен |
| /products | GET | Получение списка товаров |  | список товаров (ID товара, наименование) |
| POST | Добавить новый товар | описание, стоимость, прочие параметры | ID товара |
| /products/<product-id> | GET | Информация о товаре | ID товара | описание, стоимость, прочие параметры |
| PATCH | Изменить | ID товара, параметры | ID товара |
| DELETE | Удалить | ID товара | ID товара |
| /users | POST | Создать нового пользователя (регистрация) | номер телефона, имя, почта, пароль | токен |
| /users/<user-id> | GET | Информация о пользователе | ID пользователя | ФИО, логин, телефон |
| PATCH | Изменить информацию | ID пользователя, параметры | ID пользователя |
| DELETE | Удалить пользователя | ID пользователя | ID пользователя |
| /users/<user-id>/  cart | GET | Товары в корзине | ID пользователя | список товаров |
| POST | Добавить товар в корзину | ID пользователя, ID товара | информация о добавленном товаре |
| DELETE | Очистить корзину | ID пользователя | список удаленных товаров |
| /users/<user-id>/  cart/<product-id> | DELETE | Удалить товар в корзине | ID пользователя, ID покупки | информация об удаленном товаре |
| /users/<user-id>/  purchases | POST | Оплата покупки | ID пользователя, ID купленных товаров | ID покупки |
| /users/<user-id>/  deposit | POST | Пополнить счет | ID пользователя, сумма пополнения | текущий баланс |
| /users/<user-id>/  history | GET | История операций | ID пользователя, период | список операций |

**4.3 База данных**

База данных необходима для хранения данных о товарах, покупателе и выполненными им покупках. Структура базы данных описывается следующей ER-диаграммой

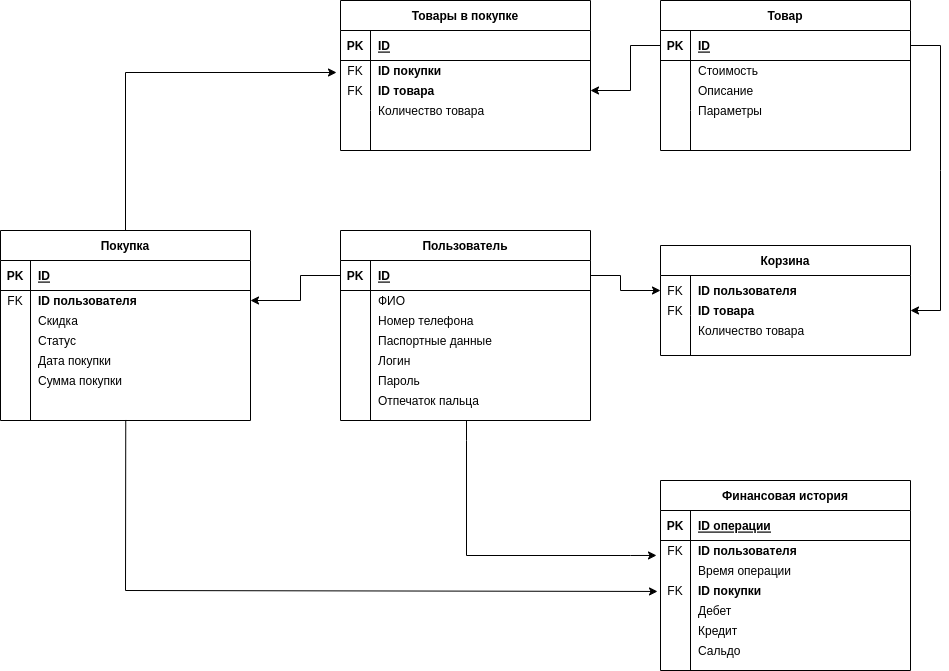


Рисунок 5. Диаграмма сущность-связь для базы данных мобильного приложения

В качестве СУБД была выбрана PostgreSQL как open-source решение, активно применяющееся для построения транзакционных баз данных.

**4.4 Взаимодействие между экранами и компонентами**

В процессе работы приложение выполняет запросы к различным методам API, который в свою очередь обращается к базе данных для получения и записи информации, требующей сохранения.

4.4.1 Регистрация и авторизация

UML-диаграмма регистрации и авторизации пользователя показана на рисунке 6.

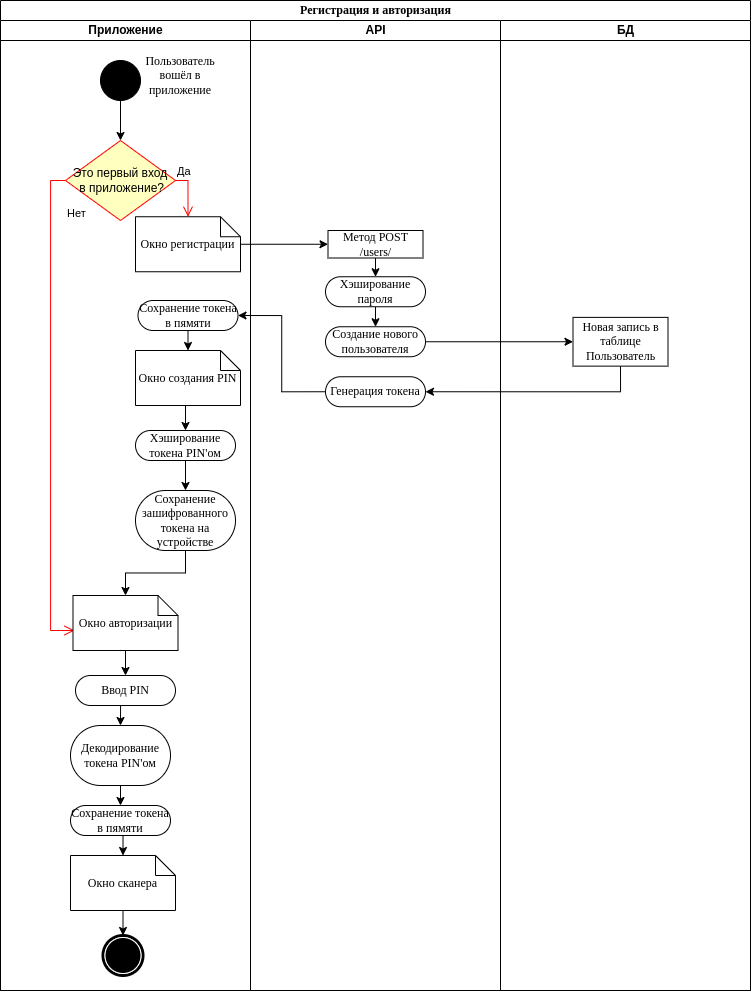


Рисунок 6. UML-диаграмма регистрации и авторизации пользователя

При первом входе в приложение на своем телефоне пользователю выводится экран авторизации (рис. 1), в котором он может ввести свой номер телефона и пароль, либо зарегистрироваться. Номер телефона и пароль передаётся API, в ответ приложение получает токен. Этот токен должен быть зашифрован с помощью PIN-кода, который должен задать пользотваель.

После пользователю открывается форма ввода только что созданного PIN-кода, который используется приложением как ключ к расшифровке токена для доступа к API.

4.4.2 Сканирование товаров

Диаграмма сканирования товаров представлена на рисунке 7.

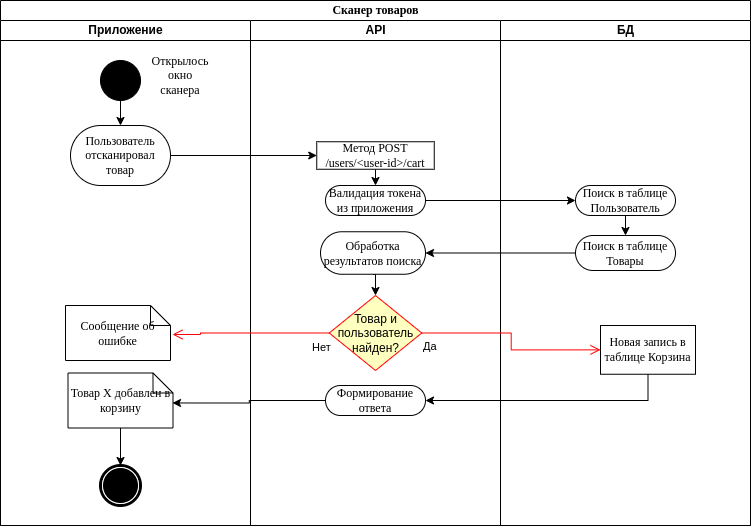


Рисунок 7. UML-диаграмма сканирования товаров

После считывания NFC-метки или QR-кода приложение определяет необходимый ID товара и делает соответствующий запрос к API на добавление товара в корзину. Каждый запрос сопровождается передачей токена, с помощью которого производится проверка доступа пользователя к запрошенному ресурсу. При успешной проверке товар добавляется в корзину в БД.

4.4.3 Оплата покупок

Чтобы оплатить покупки пользователь заходит в интерфейс корзины (рис. 3), в котором приложение делает запрос на получение данных о корзине. При нажатии кнопки “Оплатить” запускается процессы проверки средств на счете. Если средств достаточно, то содержимое корзины переносится в завершенные покупки, а сумма стоимостей товаров списывается со счета абонента.

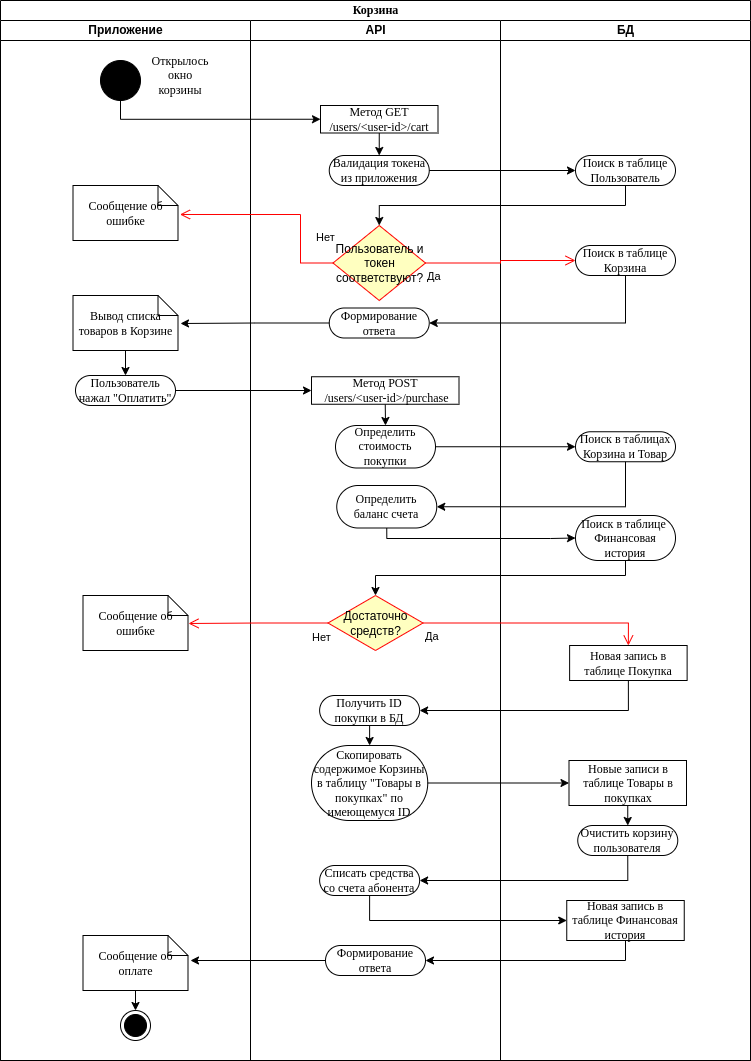


Рисунок 8. UML-диаграмма оплаты покупок в корзине

4.4.4 Личный кабинет

Пользователь может использовать интерфейс личного кабинета (рис. 4) для изменения, настроек, просмотра баланса и истории операций, пополнения счета. При открытии окна вызывается метод API получения данных о пользователе, которые отображаются в окне. Далее, например, при выборе в меню “Истории” выполняется запрос к API для получения финансовой истории покупателя.

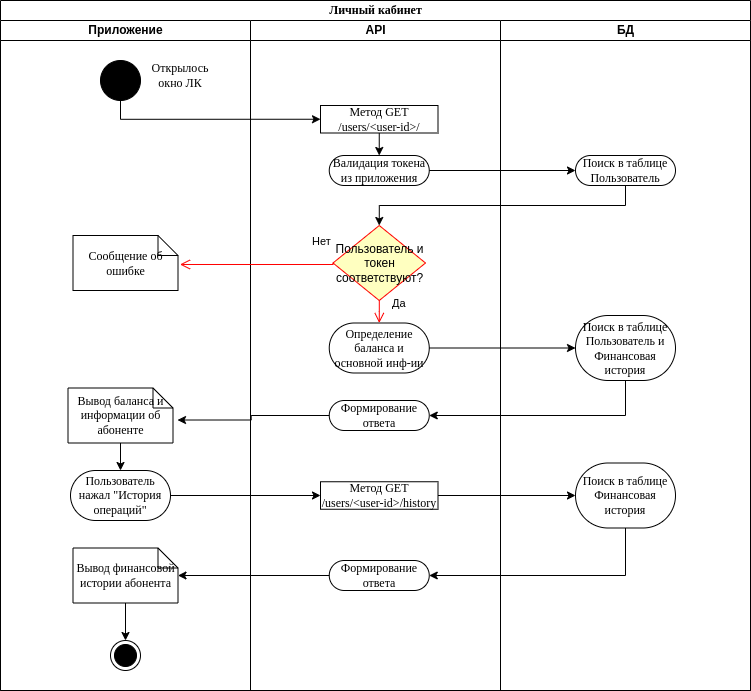


Рисунок 9. UML-диаграмма личного кабинета

**5. План проекта (диаграмма Ганта)**

Разработка проекта был поделена на 4 основных этапа:

* верстка интерфейса приложения;
* разработка функционала приложения;
* разработка API;
* проектирование API.

В разработке принимают участие 3 человека. На каждую часть назначен ответственный за выполнение поставленных сроков.

Диаграмма Ганта этапов разработки проекта представлена на рисунке 10, а также доступна в интерактивном виде в системе Notion по ссылке:

https://careful-vein-736.notion.site/99f3afb962ff4ea98a44d786727ebb27?v=8eaf34d494ee4987b95f622b3ed9f093

Рисунок 10. Диаграмма Ганта этапов разработки проекта

