

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»

**СОГЛАСОВАНО**  
Научный руководитель, доцент  
департамента «Программной инженерии»,  
кандидат технических наук

**УТВЕРЖДАЮ**  
Академический руководитель  
образовательной программы  
«Программная инженерия», кандидат  
технических наук

\_\_\_\_\_  
Е.Ю. Песоцкая  
«\_\_» \_\_\_\_\_ 2024 г.

\_\_\_\_\_  
В.В. Шилов  
«\_\_» \_\_\_\_\_ 2024 г.

**Приложение для обмена товарами**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.01-01 01-1-ЛУ**

Исполнитель:  
студент группы БПИ227

\_\_\_\_\_  
/ Судакова Д. Е. /  
«\_\_» \_\_\_\_\_ 2024 г.

**Москва 2024**

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Приложение для обмена товарами

Пояснительная записка

RU.17701729.05.01-01 01-1-ЛУ

Листов 25

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**СОДЕРЖАНИЕ**

<b>СОДЕРЖАНИЕ</b> .....	<b>3</b>
<b>АННОТАЦИЯ</b> .....	<b>4</b>
<b>1. ВВЕДЕНИЕ</b> .....	<b>5</b>
1.1. Наименование программы .....	5
1.2. Документ, на основании которого ведется разработка .....	5
<b>2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ</b> .....	<b>6</b>
2.1. Назначение программы.....	6
2.1.1. Функциональное назначение .....	6
2.1.2. Эксплуатационное назначение .....	6
2.2. Краткая характеристика области применения .....	6
<b>3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ</b> .....	<b>7</b>
3.1. Постановка задачи на разработку программы .....	7
3.2. Описание и обоснование архитектуры программы.....	7
3.2.1. Описание архитектуры программы .....	8
3.2.3. Обоснование архитектуры программы.....	11
3.5. Описание алгоритма работы программы.....	11
3.6. Описание выбора метода организации входных и выходных данных.....	12
3.7. Описание и обоснование выбора состава технических и программных средств .....	12
3.5.1. Состав технических и программных средств.....	12
3.5.2. Обоснование выбора технических и программных средств .....	13
<b>4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ</b> .....	<b>14</b>
4.1. Ориентировочная экономическая эффективность .....	14
4.2. Предполагаемая потребность.....	14
4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.....	14
<b>1. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ</b> .....	<b>15</b>
<b>ПРИЛОЖЕНИЕ 1</b> .....	<b>17</b>
<b>ПРИЛОЖЕНИЕ 2</b> .....	<b>18</b>
<b>ПРИЛОЖЕНИЕ 3</b> .....	<b>19</b>
<b>ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ</b> .....	<b>25</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## АННОТАЦИЯ

Настоящий программный документ представляет собой пояснительную записку к программному проекту «Приложение для обмена товарами «Droplet».

Раздел «Введение» включает в себя наименование программы и документ, на основании которого ведётся разработка, с указанием организации, утвердившей данный документ.

В разделе «Назначение и область применения» содержатся функциональное и эксплуатационное назначение программы и краткая характеристика области её применения.

В разделе «Технические характеристики» находятся функциональные требования и задачи для разработки проекта, архитектура приложения, взаимодействие клиент-серверной части и схема базы данных.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с аналогами.

Программный документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## ВВЕДЕНИЕ

### 1.1. Наименование программы

Наименование программы: «Приложение для обмена товарами «Droplet».

Наименование программы на английском языке: «Droplet» mobile app for exchanging goods»

Краткое наименование программы: «Droplet».

### 1.2. Документ, на основании которого ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта.

Документ утвержден организацией: Национальный исследовательский университет «Высшая школа экономики», Факультет компьютерных наук, образовательная программа «Программная инженерия».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1. Назначение программы

#### 2.1.1. Функциональное назначение

Приложение «Droplet» позволяет пользователям обмениваться товарами на основе внутренней валюты «айтем». Пользователи могут публиковать свои объявления или просматривать чужие, просматривать объявления по категориям и добавлять их в «Избранное», использовать вещи совместно в разделе «Совместное использование», а также связываться друг с другом по номеру телефона, указанного в личном кабинете. Айтемы можно тратить, используя один айтем для покупки одного товара, или получать, когда другие пользователи покупают ваши товары. Таким образом обеспечивается не прямой бартер и бесплатный обмен.

#### 2.1.2. Эксплуатационное назначение

Приложение предназначено для широкой аудитории пользователей, которые поддерживают идею повторного использования вещей вместо покупки новых или хотят экологично избавиться от ненужных вещей, отдав их другим людям и взяв взамен другие вещи. «Droplet» позволяет сделать поиск вещей и контакт с их обладателями удобным и быстрым, а также дает возможность совместно использовать вещи.

### 2.2. Краткая характеристика области применения

«Droplet» - приложение для бартера, которое предоставляет все необходимые функции для удобного обмена вещами и даже услугами, а также поддерживающее отдельную функцию совместного использования. Такое приложение будет полезным любому человеку, готовому избавиться от старых вещей или обменять их на новые.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

#### 3.1. Постановка задачи на разработку программы

«Приложение для обмена товарами «Droplet» должно предоставлять пользователям следующие функции:

1. Регистрация в приложении.
2. Авторизация в приложении для зарегистрированных пользователей.
3. Добавление данных об аккаунте: имя, фамилию, номер телефона, адрес электронной почты, адрес, аватар.
4. Изменение данных аккаунта, в том числе пароля от аккаунта.
5. Создание объявления с указанием его названия, описания, категории и добавления фото объявления.
6. Редактирование объявления, в том числе его название, описание, категорию и фото.
7. Архивацию объявления.
8. Просмотр всех объявлений из базы данных приложения.
9. Просмотр объявлений по категориям с помощью выбора категории из всех возможных.
10. Просмотр недавно опубликованных объявлений.
11. Добавление объявления в избранное.
12. Удаление объявления из избранного.
13. Просмотр истории покупок.
14. Просмотр всех объявлений, созданных пользователем, в том числе архивных.
15. Возможность отклика на объявление и отправки запроса на покупку пользователю-автору объявления.
16. Просмотр всех текущих запросов на объявления пользователя и возможность отклонить или одобрить их.
17. Возможность связи с пользователями по номеру телефона, указанному в профиле.

#### 3.2. Описание и обоснование архитектуры программы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.2.1. Описание архитектуры программы

Для «Droplet» была выбрана стандартная клиент-серверная архитектура мобильного приложения. Серверная часть была разработана на языке программирования Java с использованием фреймворка Spring для облегчения обеспечения функционирования сервера. Серверное приложение состоит из моделей, представляющих собой программные сущности таблиц базы данных, DTO (Data Access Object), используемых при передаче данных серверу, репозиториях, предназначенных для не прямой работы с SQL запросами, сервисов, каждый из которых работает со своим репозиторием, и контроллеров, которые определяют интерфейс API для подключения клиентской части и принимают HTTP-запросы. В качестве базы данных была выбрана PostgreSQL.

Клиентская часть разработана на языке Kotlin в среде разработки Android Studio. Для отправки запросов серверу используется библиотека Retrofit, позволяющая минимизировать количество кода для создания API. Созданные API соответствуют сущностям в базе данных и моделям серверной части. Каждое API имеет соответствующий интерфейс Repository и класс RuntimeRepository, реализующий этот интерфейс, которые конструируют необходимые DTO на основе пользовательских данных и перенаправляют запросы классам API. Все API инициализируются в классе Dependencies, общим для всей программы, и доступны из любой точки клиентской части программы. Клиентская часть также имеет модели и DTO, повторяющие сущности, находящиеся в программе.

Структура фронтенда клиентской части программы следующая: программа имеет 4 активности для вкладок приложения, определяющимся нижней навигацией, активность регистрации, активность авторизации, а также несколько фрагментов, определяющих поведение программы, таких как фрагмент страницы объявления, фрагмент добавления объявления и т.д. Все фрагменты имеют соответствующий класс viewModel, перенаправляющий пользовательские данные API а затем серверу, и использующий механизм корутин в Kotlin и suspend-функций. Для списков на экране пользователя, таких как список объявлений, список запросов на покупку, список категорий и т.д. используются интерфейсный элемент recyclerView и адаптер, соответствующий каждой необходимой сущности.

Рассмотрим последовательный механизм отправки запроса серверу. Сначала приложение собирает данные, введенные пользователем, с помощью интерфейсных элементов, и передает их в фрагмент или активность, которые асинхронно запускают методы классов RuntimeRepository для соответствующего API-класса. API-классы, в свою очередь, по указанному URL, совершают

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

запрос на сервер, передавая все необходимые параметры. Далее запросы с серверной стороны направляются в контроллеры по указанным URL-адресам, которые вызывают методы классов-сервисов. В них происходит валидация данных и обработка исключительных ситуаций. Классы-сервисы взаимодействуют с базой данных с помощью классов-репозитория, которые отправляют данные PostgreSQL.

### 3.2.2. Архитектура базы данных

Рассмотрим архитектуру базы данных PostgreSQL.

База данных имеет следующие таблицы и поля:

1. `_user` – таблица для хранения данных о пользователях.
  - 1.1. `id` – уникальный идентификатор.
  - 1.2. `address` – адрес пользователя.
  - 1.3. `email` – электронная почта пользователя.
  - 1.4. `first_name` – имя пользователя.
  - 1.5. `items` – количество айтемов на счету пользователя.
  - 1.6. `last_name` – фамилия пользователя.
  - 1.7. `password` – пароль.
  - 1.8. `phone_number` – номер телефона пользователя
  - 1.9. `photo` – аватар пользователя.
2. `advertisement` – таблица для хранения данных об объявлениях.
  - 2.1. `id` – уникальный идентификатор.
  - 2.2. `category` – категория объявления, *поле связано с полем `id` в таблице `category`.*
  - 2.3. `description` – описание объявления.
  - 2.4. `name` – название объявления.
  - 2.5. `photo` – фото объявления.
  - 2.6. `status_active` – статус активности объявления (активно/в архиве)
  - 2.7. `owner_id` – идентификатор владельца объявления, *поле связано с полем `id` в таблице `_user`.*
3. `category` – таблица для хранения данных о категориях.
  - 3.1. `id` – уникальный идентификатор.
  - 3.2. `name` – название категории.
4. `favourites_list` – таблица списков избранных объявлений пользователей.
  - 4.1. `id` – уникальный идентификатор.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- 4.2. user\_id - идентификатор владельца списка, *поле связано с полем id в таблице \_user.*
5. favs\_ads – таблица связи списков избранных и объявлений.
- 5.1. fav\_list\_id – идентификатор списка объявления, *поле связано с полем id в таблице favourites\_list.*
- 5.2. ads\_id – идентификатор объявления, *поле связано с полем id в таблице advertisement.*
6. purchases\_list – таблица списков покупок пользователей.
- 6.1. id – уникальный идентификатор.
- 6.2. user\_id – идентификатор владельца списка, *поле связано с полем id в таблице \_user.*
7. purchases\_ads – таблица связи списков покупок и объявлений.
- 7.1. purchases\_list\_id – идентификатор списка объявления, *поле связано с полем id в таблице purchases\_list.*
- 7.2. ads\_id – идентификатор объявления, *поле связано с полем id в таблице advertisement.*
8. shared\_usage\_list – таблица списков совместного использования пользователей.
- 8.1. id – уникальный идентификатор.
- 8.2. user\_id – идентификатор владельца списка, *поле связано с полем id в таблице \_user.*
9. sharedusage\_ads – таблица связи списков совместного использования и объявлений.
- 9.1. shared\_usage\_list\_id – идентификатор списка объявления, *поле связано с полем id в таблице shared\_usage\_list.*
- 9.2. ads\_id – идентификатор объявления, *поле связано с полем id в таблице advertisement.*
10. query – таблица для хранения данных о запросах на покупку объявлений.
- 10.1. id – уникальный идентификатор запроса.
- 10.2. status – статус запроса (ждет ответа/отклонен/принят)
- 10.3. ads\_id – идентификатор объявления, *поле связано с полем id в таблице advertisement.*
- 10.4. user\_id – идентификатор автора запроса, *поле связано с полем id в таблице \_user.*

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

### 3.2.3. Обоснование архитектуры программы

Объясним несколько архитектурных решений, принятых при разработке приложения «Droplet»:

1. В качестве среды разработки была выбрана Android Studio, поскольку это наиболее современный и удобный инструмент для конструирования Android-приложений. В ней удобно создавать приложения с графическим интерфейсом, также есть встроенный эмулятор, на котором можно тестировать приложение, кроме того, Android Studio, предоставляет возможности для автоматического создания xml-файлов разметки, а также активностей и фрагментов.
2. Была выбрана клиент-серверная архитектура как стандартная архитектура для Android-приложений. Такая структура программы позволяет изолировать работу с запросами и базой данных от интерфейсного приложения и удобно разделить передачу и валидацию данных на слои.
3. В качестве языка программирования для клиентской части был выбран Kotlin, поскольку это ведущий на данный момент язык разработки мобильных приложений. Он позволяет избежать исключений типа NullPointerException на этапе компиляции, и, кроме того, предоставляет удобный инструмент работы с потоками – корутины. Для «Droplet» это решение показалось самым оптимальным. Также была выбрана библиотека Retrofit, поскольку она позволяет очень быстро и с минимальным количеством кода обращаться к API и передавать запросы.
4. Серверная часть была написана на Java с использованием Spring, поскольку это удобный и быстрый способ организовать архитектуру с использованием слоев модель-сервис-контроллер. Кроме того, Java поддерживается Android Studio, а значит серверную часть также можно легко и удобно запускать и тестировать из этой среды разработки.

### 3.5. Описание алгоритма работы программы

Рассмотрим последовательно алгоритм работы «Приложения для обмена товарами «Droplet». Сначала необходимо пройти регистрацию или авторизацию, если пользователь уже

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

зарегистрирован в системе. Далее нет четкой последовательности действий, пользователь может просматривать объявления, создавать свои объявления, редактировать данные аккаунта, и пользоваться всеми функциями, описанными в разделе «Постановка задачи на разработку программы». Однако рассмотрим подробно процесс совершения сделки по объявлению:

Для того чтобы откликнуться на объявление и забрать вещь себе, необходимо иметь не менее одного айтема на балансе.

1. Пользователю-владельцу объявления отправляется запрос, который он может принять или отклонить.
2. При принятии запроса пользователю-владельцу начисляется 1 айтем, у пользователя-покупателя списывается 1 айтем, объявление архивируется;
3. При отклонении запроса транзакции не происходит.
4. После успешного совершения покупки объявление помещается в список покупок текущего пользователя, для пользователя-владельца объявление становится архивным. В случае, если объявление находится в категории «Совместное использование» объявление не архивируется, а остается активным.

### **3.6. Описание выбора метода организации входных и выходных данных**

Серверная часть приложения не имеет входных данных. Все данные хранятся в базе данных, описанной в пункте «Архитектура базы данных». Клиентская часть приложения оперирует входными данными, получаемыми через взаимодействие с экраном мобильного устройства от пользователя.

### **3.7. Описание и обоснование выбора состава технических и программных средств**

#### **3.5.1. Состав технических и программных средств**

Для нормальной работы серверного приложения необходимо:

1. Не менее 1 Гб ОЗУ
2. Не менее 1 Гб свободного пространства на диске
3. Доступ в локальную (общую с сервером) или глобальную сеть

Для нормальной работы мобильного приложения необходимо:

1. Не менее 100 Мб свободного пространства на диске
2. Не менее 512 Мб оперативной памяти

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. Доступ в локальную (общую с сервером) или глобальную сеть

Программные требования для нормальной работы программы:

1. ОС Android версии 10 и выше
2. ОС Windows 7 и выше или ОС Linux
3. Java версии 11 или выше с библиотекой Retrofit и фреймворком Spring

### 3.5.2. Обоснование выбора технических и программных средств

В требованиях указано минимум 512 Мб ОЗУ, так как запросы на сервер могут занимать время и ресурсы.

Работу серверного приложения может обеспечивать стандартный компьютер, однако при росте количества пользователей и размера базы данных должны расти и мощности компьютера, поскольку при большом количестве записей запросы и загрузка страниц будет занимать все больше времени.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

#### 4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

##### 4.1. Ориентировочная экономическая эффективность

В данном курсовом проекте не предусмотрено экономической эффективности.

##### 4.2. Предполагаемая потребность

Ожидается, что приложение будет полезно независимо от времени года и будет наиболее популярно у людей, поддерживающих идею апсайклинга и повторного использования вещей.

##### 4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

В таблице 1 приведена сравнительная характеристика с приложениями, наиболее близкими к теме разрабатываемого проекта.

Таблица 1 – Аналоги

	«Karmitt»	«Sharing Map»	«SwopShop»	«Swop.it»	Droplet
Регистрация	+	+	+	+	+
Категория «Совместное использование»	-	+	-	-	+
Обмен по принципу внутренней валюты	+	+	+	+	+
Категория «Услуги»	-	-	-	-	+
Обмен внутренней валюты на товары от партнеров	+	-	-	-	-
Доступно в России	+	+	+	-	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

- [1] ГОСТ 19.103–77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [2] ГОСТ 19.104–78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [3] ГОСТ 19.401-78 Текст программы. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [4] ГОСТ 19.105–78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [5] ГОСТ 19.404-79 Пояснительная записка. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [6] ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [7] ГОСТ 19.106–78 Требования к программным документам, выполненным печатным способом.  
//Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [8] ГОСТ 19.201–78 Техническое задание. Требования к содержанию и оформлению.  
//Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [9] ГОСТ 19.301–79 Программа и методика испытаний. Требования к содержанию и оформлению.  
//Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [10] ГОСТ 19.603–78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [11] ГОСТ 19.604–78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- [12] PostgreSQL 15.0 Documentation. // The PostgreSQL Global Development Group. 2023.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

URL: <https://www.postgresql.org/files/documentation/pdf/15/postgresql-15-A4.pdf> (дата обращения: 29.01.2023).

[13] Java Documentation. // Oracle Help Center. 2023. URL: <https://docs.oracle.com/en/java/> (дата обращения: 29.01.2023).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 1**

**ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ**

1. **База данных** – структура хранения данных, которая задает определенные правила для способа хранения информации, а также методы взаимодействия данных между собой.
2. **Мобильное приложение** – программное обеспечение для ОС Android.
3. **Транзакция** – в контексте данного приложения это процесс закрытия объявления и начисления 1 айтема пользователю-владельцу объявления и списание 1 айтема у пользователя-покупателя.
4. **Тост** – всплывающее сообщение внизу экрана мобильного устройства, характерное для ОС Android.
5. **Spring** – фреймворк языка Java, используемый для облегчения проектирования и создания приложений.
6. **Retrofit** – это библиотека для работы с REST API в Android, упрощающая взаимодействие с сервером и выполнение HTTP-запросов
7. **Репозиторий** – Локация, где хранятся и обслуживаются определенные данные. Обычно данные в репозитории представлены в виде файлов, доступных для распространения через сеть.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КОМПОНЕНТОВ  
ИНФРАСТРУКТУРЫ СИСТЕМЫ**

Приложение для обмена товарами «Droplet» состоит из двух частей – серверной и клиентской. Модуль клиентской части называется app и находится в соответствующей папке в репозитории на Github по ссылке: <https://github.com/daryasan/DropletBarterApp.git>.

Модуль серверной части называется backend и находится в одноименной папке в репозитории Github по той же ссылке.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 3**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ОСНОВНЫХ  
КЛАССОВ МОБИЛЬНОГО ПРИЛОЖЕНИЯ**

Ниже приведена таблица основных классов приложения, без учета вспомогательных классов и DTO (Data Transfer Object).

Класс	Поля	Методы	Назначение класса
<b>Активности</b>			
AdvertisementActivity	binding, adapter, viewModel	onCreate, onBackPressed, disableAdnHideElements, enableAndShowElements, setButtonsChange, checkVisibility, startAdsFragment, startAddAdsFragment, startSearch	Страница «Объявления»
MainActivity	binding, adapter, viewModel	onCreate, onBackPressed, authorize, disableAdnHideElements, enableAndShowElements, setButtonsChange, startAdvertisementFragment	Страница «Для вас»
ProfileActivity	job, binding, userDataDto, fragmentManager	onCreate, onDestroy, onResume, onBackPressed, disableAdnHideElements, enableAndShowElements, getUserData, setuserDataToScreen, onResult, logout, fetchUser, startEditDataFragment, startChangeLoginsFragment, startQueriesFragment	Страница «Профиль»
SearchActivity	binding, adapter	onCreate, onBackPressed, startSearchFragment, disableAndHideElements, enableAndShowElements	Страница «Категории»

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

20  
**RU.17701729.05.01-01 01-1**

LoginActivity	validator, toaster, binding, job, coroutineContext	onDestroy, onCreate, onResult, signIn	Страница входа в аккаунт
RegisterActivity	validator, toaster, binding, job, coroutineContext	onDestroy, onCreate, onResult, signUp	Страница регистрации
Фрагменты			
ChangeLoginsFragment	viewModel, binding, toaster, userData	onViewCreated, onCreateView, fetchUserData,	Страница изменения пароля, логина и телефона
EditDataFragment	viewModel, binding, toaster, photo, imageLoader	onViewCreated, onCreateView, setData	Страница изменения личной информации пользователя
QueriesFragment	viewModel, binding	onViewCreated, onCreateView,	Страница запросов на покупку
AdvertisementFragment	viewModel, binding, adapter, advertisement, ownerData, layoutResource	onViewCreated, onCreateView, setSimilar, setData, startAdsFragment, startEditAdsFragment, startProfileFragment	Страница объявления
AnotherProfileFragment	viewModel, binding, adapter, userDataDto, adsForUser, layoutResource	onViewCreated, onCreateView, setData, setAdvertisements, startAdsFragment	Страница профиля другого пользователя
EditAdvertisementFragment	viewModel, binding, advertisement, imageLoader, toaster, category	onViewCreated, onCreateView, setData, onItemSelectet, onNothingSelected, finishFragment	Страница редактирования объявления
SearchFragment	viewModel, binding, adapter, query, categoryPos,	onViewCreated, onCreateView, onResume, setAdvertisements, startAdsFragment, onItemSelected,	Страница просмотра объявления

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**21**  
**RU.17701729.05.01-01 01-1**

		onNothingSelected	
AddAdvertisementFragment	viewModel, binding, toaster, category, photo, name, description, imageLoader	onViewCreated, onCreateView, onItemSelected, onNothingSelected, finishFragment	Страница добавления объявления
viewModels			
ChangeLoginsViewModel	coroutineContext	getUserData, editEmail, editPhone, editPassword	Управление соответствующим фрагментом
EditDataViewModel	coroutineContext	saveEditedData, getUserData	Управление соответствующим фрагментом
QueriesViewModel	-	getQueries, removeClosedQueries	Управление соответствующим фрагментом
AdvertisementViewModel	-	findAdvertisements, findOwner, sendPendingRequest, hideAdvertisement, findSuggestions, findAllAdvertisement, getUserId,	Управление соответствующим фрагментом
AnotherProfileViewModel	-	findUser, findAdvertisementForUser, removeArchived,	Управление соответствующим фрагментом
EditAdvertisementViewModel	-	editAdvertisement, findAdvertisement	Управление соответствующим фрагментом
SearchViewModel	-	findAdvertisements, findAllAdvertisements, findCategoryAdvertisement, removeArchived	Управление соответствующим фрагментом
AddAdvertisementViewModel	-	addAdvertisement, getUserId	Управление соответствующим фрагментом
AdvertisementActivityViewMod el	-	findMyAdvertisement, getFavourites, getSharedUsage, getPurchases, getUserId	Управление соответствующим фрагментом
MainViewModel	-	findAllAdvertisements, removeArchived, findSharedUsage,	Управление соответствующей активностью

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

22  
**RU.17701729.05.01-01 01-1**

		findRecent	
API			
AuthApi	-	signInByEmail signUpByEmail logout refreshTokens	Класс для запросов по URL для входа и регистрации пользователя
AuthRepository	-	signInByEmail signUpByEmail logout refreshTokens	Интерфейс для работы с соответствующим API и конструирования DTO
RuntimeAuthRepository	-	signInByEmail signUpByEmail logout refreshTokens	Имплементация соответствующего интерфейса
AdsApi	-	addAdvertisement, findAdvertisements, editAdvertisement, findallAdvertisement, findAdvertisementsByCategory, findAdvertisementsOwnedByUser	Класс для запросов по URL для действий с объявлениями
AdsRepository	-	addAdvertisement, findAdvertisements, editAdvertisement, findallAdvertisement, findAdvertisementsByCategory, findAdvertisementsOwnedByUser	Интерфейс для работы с соответствующим API и конструирования DTO
RuntimeAdsRepository	-	addAdvertisement, findAdvertisements, editAdvertisement, findallAdvertisement, findAdvertisementsByCategory, findAdvertisementsOwnedByUser	Имплементация соответствующего интерфейса
FavouritesApi	-	addToFavourites deleteFromFavourites, findForUser	Класс для запросов по URL для списка избранных объявлений
FavouritesRepository	-	addToFavourites deleteFromFavourites, findForUser	Интерфейс для работы с соответствующим API и конструирования

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

23  
**RU.17701729.05.01-01 01-1**

			DTO
RuntimeFavouritesRepository	-	addToFavourites deleteFromFavourites, findForUser	Имплементация соответствующего интерфейса
PurchasesApi	-	addToPurchases, findForUser	Класс для запросов по URL для списка покупок
PurchasesRepository	-	addToPurchases, findForUser	Интерфейс для работы с соответствующим API и конструирования DTO
RuntimePurchasesRepository	-	addToPurchases, findForUser	Имплементация соответствующего интерфейса
SharedUsageApi	-	addToSharedUsage, findForUser	Класс для запросов по URL для списка совместного использования
SharedUsageRepository	-	addToSharedUsage, findForUser	Интерфейс для работы с соответствующим API и конструирования DTO
RuntimeSharedUsageRepository	-	addToSharedUsage, findForUser	Имплементация соответствующего интерфейса
QueryApi	-	addQuery, findQuery, editQuery, findQueryForUser	Класс для запросов по URL для запросов на покупку
QueryRepository	-	addQuery, findQuery, editQuery, findQueryForUser	Интерфейс для работы с соответствующим API и конструирования DTO
RuntimeQueryRepository	-	addQuery, findQuery, editQuery, findQueryForUser	Имплементация соответствующего интерфейса
UserApi	-	findUserById, editUser, editEmail,	Класс для запросов по URL для действий с

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## RU.17701729.05.01-01 01-1

		editPhone, changePassword	пользователями
UserRepository	-	findUserById, editUser, editEmail, editPhone, changePassword	Интерфейс для работы с соответствующим API и конструирования DTO
RuntimeUserRepository	-	findUserById, editUser, editEmail, editPhone, changePassword	Имплементация соответствующег о интерфейса

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц в докум.)	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм.	Измененных	Замененных	Новых	Аннулированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.15-01 81				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата