МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Разработка мобильного приложения по поиску товаров

Курсовой проект

09.03.02 Информационные системы и технологии

Программная инженерия в информационных системах

Допущены к защите

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Д. Махортов, к.ф.- м.н., доцент \_\_.\_\_.20\_\_

Обучающиеся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Папина, А.Д. Ильина 3 курс, д/о

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Х.А. Полищук, аспирант

Воронеж 2019

Содержание

[Введение 3](#__RefHeading___Toc876_1504883396)

[Аналоги 3](#__RefHeading___Toc878_1504883396)

[Методология 3](#__RefHeading___Toc880_1504883396)

[1. Постановка задачи 4](#__RefHeading___Toc882_1504883396)

[2. Анализ 4](#__RefHeading___Toc884_1504883396)

[2.1. Анализ предметной области 4](#__RefHeading___Toc886_1504883396)

[2.2. Анализ задач 5](#__RefHeading___Toc888_1504883396)

[2.2.1. Задача поиска товаров 5](#__RefHeading___Toc890_1504883396)

[2.2.2. Задача сохранения поискового запроса 6](#__RefHeading___Toc892_1504883396)

[2.2.3. Задача хранения понравившихся товаров 6](#__RefHeading___Toc894_1504883396)

[2.3. Анализ средств реализации 6](#__RefHeading___Toc900_1504883396)

[2.4. Диаграммы 7](#__RefHeading___Toc902_1504883396)

[2.4.1. Диаграмма прецедентов 7](#__RefHeading___Toc2047_1504883396)

[2.4.2. Диаграмма активности 8](#__RefHeading___Toc2049_1504883396)

[2.4.3. Диаграмма состояний 9](#__RefHeading___Toc2051_1504883396)

[2.4.4. Диаграмма последовательности 9](#__RefHeading___Toc2053_1504883396)

[2.4.5. Диаграмма взаимодействия 10](#__RefHeading___Toc2055_1504883396)

[2.4.7. Схема базы данных 11](#__RefHeading___Toc2057_1504883396)

[2.4.7. Диаграмма классов 12](#__RefHeading___Toc2059_1504883396)

[2.4.8. Диаграмма объектов 13](#__RefHeading___Toc2061_1504883396)

[2.4.9. Диаграмма развертывания 14](#__RefHeading___Toc2063_1504883396)

[3. Основная часть 15](#__RefHeading___Toc904_1504883396)

[Заключение 15](#__RefHeading___Toc906_1504883396)

[Список источников 15](#__RefHeading___Toc908_1504883396)

[Приложения 15](#__RefHeading___Toc910_1504883396)

Введение

В настоящее время в сети Интернет существует множество онлайн-магазинов. При покупке товаров через них, пользователь сталкивается с проблемой выбора и необходимостью просмотра всех доступных предложений на различных торговых площадках. Также, если пользователь желает приобрести наиболее новый товар, и он не использует никаких вспомогательных сервисов, ему приходится регулярно вручную перепроверять наличие новых предложений. Поэтому необходимо разработать автоматизированный сервис для решения этой задачи.

Аналоги

**Яндекс Маркет**

Удобный сайт для поиска и покупки товаров различных категорий. Есть такие функции как чаты, для общения с людьми и организациями, сравнение. Можно откладывать товары которые понравились, смотреть магазины на карте. Также представлено много параметров для поиска того или иного товара, что помогает покупателям подобрать для себя товар, соответствующий их требованиям. Сайт имеет простой и понятный дизайн. Из минусов можно отметить отсутствие уведомлений о новинках товаров.

**Wikimart.ru**

Большой, красивый и достаточно крупный агрегатор товаров. Объем трафика оценить тяжело, но всего в базе зарегистрировано 2 миллиона товаров. На каждой карточке товара есть стандартное описание, сравнение цен и отзывы. Из минусов: заказ оформляется на самом “Wikimart”, без перехода на сайт магазина.

Методология

При разработке приложения будет использоваться водопадная модель проектирования. Также во время написания программы будет использована парадигма объектно-ориентированного программирования.

1. Постановка задачи

Необходимо разработать сервис для поиска товаров. Система будет представлять собой Android-приложение со свободной регистрацией пользователей. Основными целями создаваемого приложения с точки зрения пользователя являются:

* Упрощение процедуры поиска товаров из различных Интернет-магазинов, с возможностью указания таких параметров, как наименование, категория, ценовой диапазона и рейтинг товара.
* Возможность перехода на сайт магазина-продавца товара.
* Предоставление функции сохранения поисковых запросов для более быстрого поиска по тем же критериям в будущем.
* Возможность сохранять понравившийся товар для повторного просмотра.

Для выподнения этих задач приложение должно предоставлять пользователям следующие возможности:

* искать товары по параметрам
* сохранять товары в закладки
* сохранять поисковые запросы, указав параметры поиска

Реализация этих функций также влечёт за собой ряд нефункциональных требований, а именно, приложение должно обращаться к внешнему API Интернет-магазинов для поиска товаров, и иметь централизованное хранилище для сохранения поисковых запросов и понравившихся товаров.

В следующем пункте проанализирована предметная область разрабатываемой системы, а также более подробно рассмотрена основные задачи приложения.

2. Анализ

2.1. Анализ предметной области

Современные Интернет - сервисы по поиску товаров предоставляют пользователям широкий выбор товаров различных категорий из множества Интернет - магазинов.

Основными критериями поиска товаров являются:

* название
* цена
* категория

Также при поиске товаров немаловажными являются их рейтинги и отзывы пользователей, которые заказывали те или иные товары.

Можно выделить некоторые основные возможности, которые сервисы по поиску товаров предоставляют пользователям:

* изучать подробные характеристики товаров
* откладывать понравившиеся товары в корзину
* просматривать магазины, где можно заказать выбранный товар

Для облегчения нахождения товаров, сервисы должны запоминать истории поиска пользователей и предлагать товары на их основании.

Магазины, предоставляющие возможность заказать выбранный товар, необходимо сортировать по стоимости этих товаров, так пользователь сможет быстро сделать заказ исходя из своего ценового диапазона.

2.2. Анализ задач

2.2.1. Задача поиска товаров

Рассмотрим задачу поиска товаров. Этапы, которые она в себя включает:

1. Проверка данных, которые ввел пользователь. Если данные введены неправильно, необходимо предложить ввести их повторно.
2. Формирование и отправление запроса с данными на сторонний ресурс (API сервер).
3. Обработка ответа, если таковой получен. Если ответ не получен или произошла какая-либо ошибка, то пользователю выводится сообщение об этом и предлагается ввести данные для поиска повторно.
4. Отображение итоговых результатов поиска пользователю.

2.2.2. Задача сохранения поискового запроса

Рассмотрим задачу хранения поисковых запросов. При этом должны храниться следующие данные:

* название товара
* категория
* цена
* рейтинг

Также должна храниться ссылка на пользователя, сохранившего запрос.

2.2.3. Задача хранения понравившихся товаров

Рассмотрим задачу хранения понравившихся товаров. Должна храниться следующая информация о конкретном товаре:

* + идентификатор товара
  + название товара
  + категория
  + цена
  + магазин, в котором можно заказать товар
  + рейтинг (если имеется)
  + ссылка на изображение товара (если имеется)

Также должна храниться ссылка на пользователя, сохранившего запрос.

2.3. Анализ средств реализации

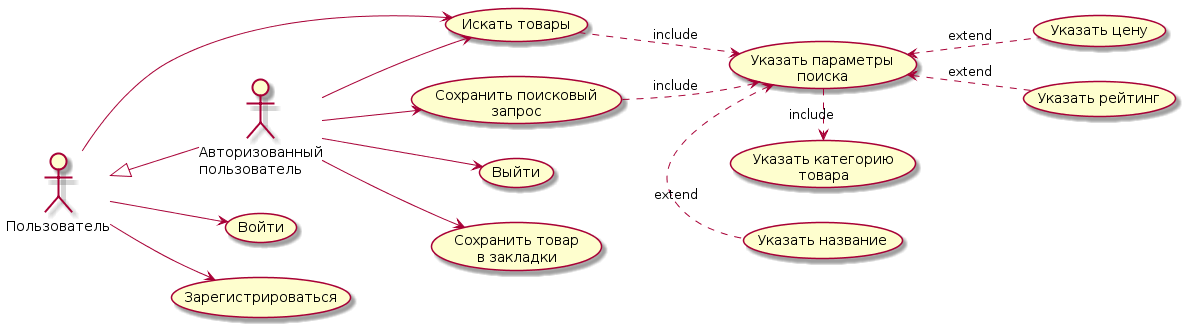
В качестве платформы, под которую разрабатывается приложение, была выбрана популярная операционная система Android. В качестве основного языка разрабатываемого приложения был выбран Java, поскольку на нем есть возможность написать как клиентскую Android-чать приложения, так и его серверную часть.

2.4. Диаграммы

Ниже приведены следующие диаграммы, описывающие работу системы:

* Диаграмма прецедентов
* Диаграмма активности
* Диаграмма состояний
* Диаграмма последовательности
* Диаграмма взаимодействия
* Схема базы данных
* Диаграмма классов
* Диаграмма объектов
* Диаграмма развертывания

2.4.1. Диаграмма прецедентов

Рисунок 1. Диаграмма вариантов использования

Данная диаграмма описывает различные вариантов использования системы. Ниже представлено краткое описание каждого из вариантов.

**Зарегистрироваться в системе**

В данном варианте использования пользователь регистрируется в системе, указываея свои личные данные (адрес электронной почты).

**Войти в систему**

В данном варианте использования пользователь авторизуется в системе, вводя адрес электронной почты и пароль.

**Выйти из системы**

В данном варианте использования авторизованный пользователь выходит из системы.

**Искать товары**

В данном варианте использования пользователь (как авторизованный, так и не авторизованный) может просматривать товары по категориям, а также осуществлять поиск товаров по наименованию, цене и рейтингу.

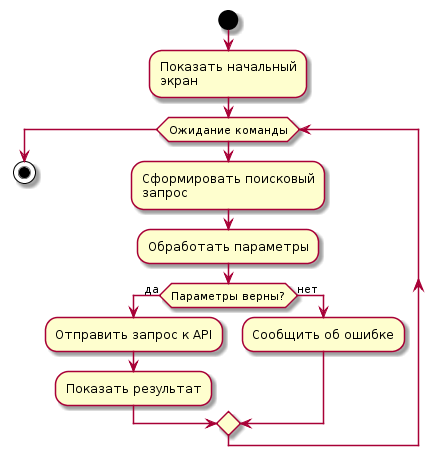
**Сохранить товар в закладки**

В данном варианте использования авторизованный пользователь может сохранить понравившийся ему товар в закладки для повторного просмотра в дальнейшем.

**Сохранить поисковый запрос**

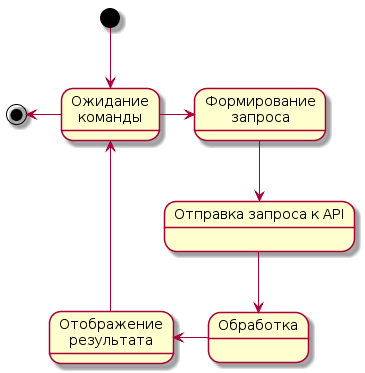
В данном варианте использования авторизованный пользователь может сохранять поисковый запрос, предварительно указав параметры поиска.

2.4.2. Диаграмма активности

Рисунок 2. Диаграмма активности

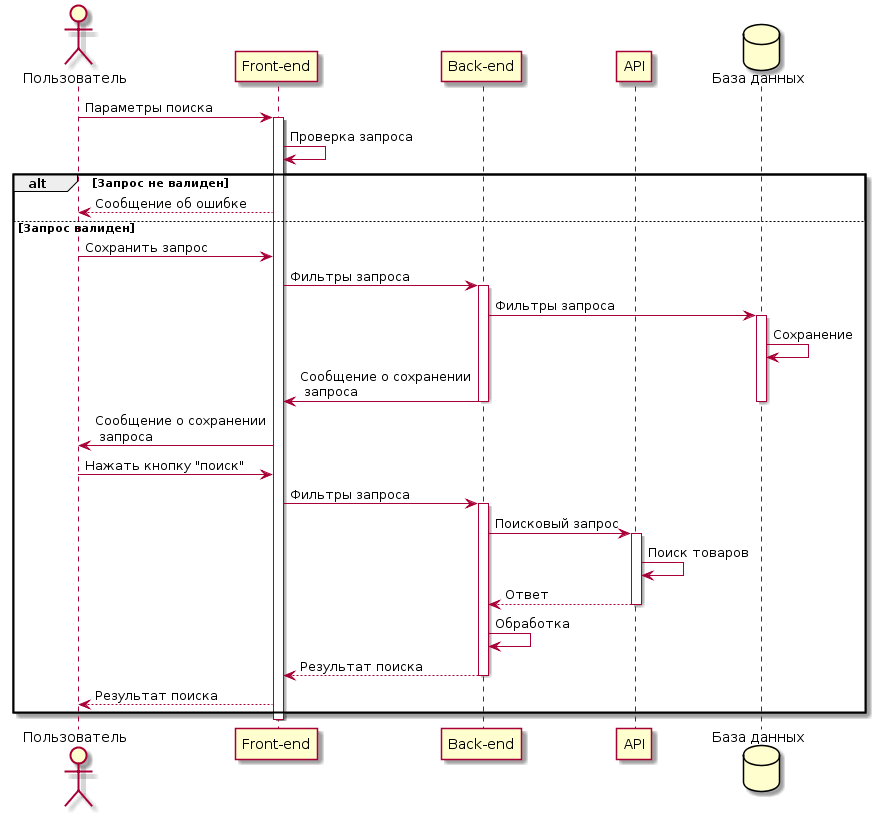
На диаграмме активности представлен бизнес-процесс поиска товаров. При старте приложение направляет пользователя на главный экран и ожидает его дальнейших действий. Если пользователь выходит из приложения, оно закрывается и бизнес-процесс завершается. Иначе, при вводе пользователем поискового запроса, тот обрабатывается в системе, и если введённый пользователем параметры верны, отправляется внешнему API Интернет-магазинов. В итоге пользователь видит результаты поиска, иначе сообщение об ошибке.

2.4.3. Диаграмма состояний

Рисунок 3. Диаграмма состояний

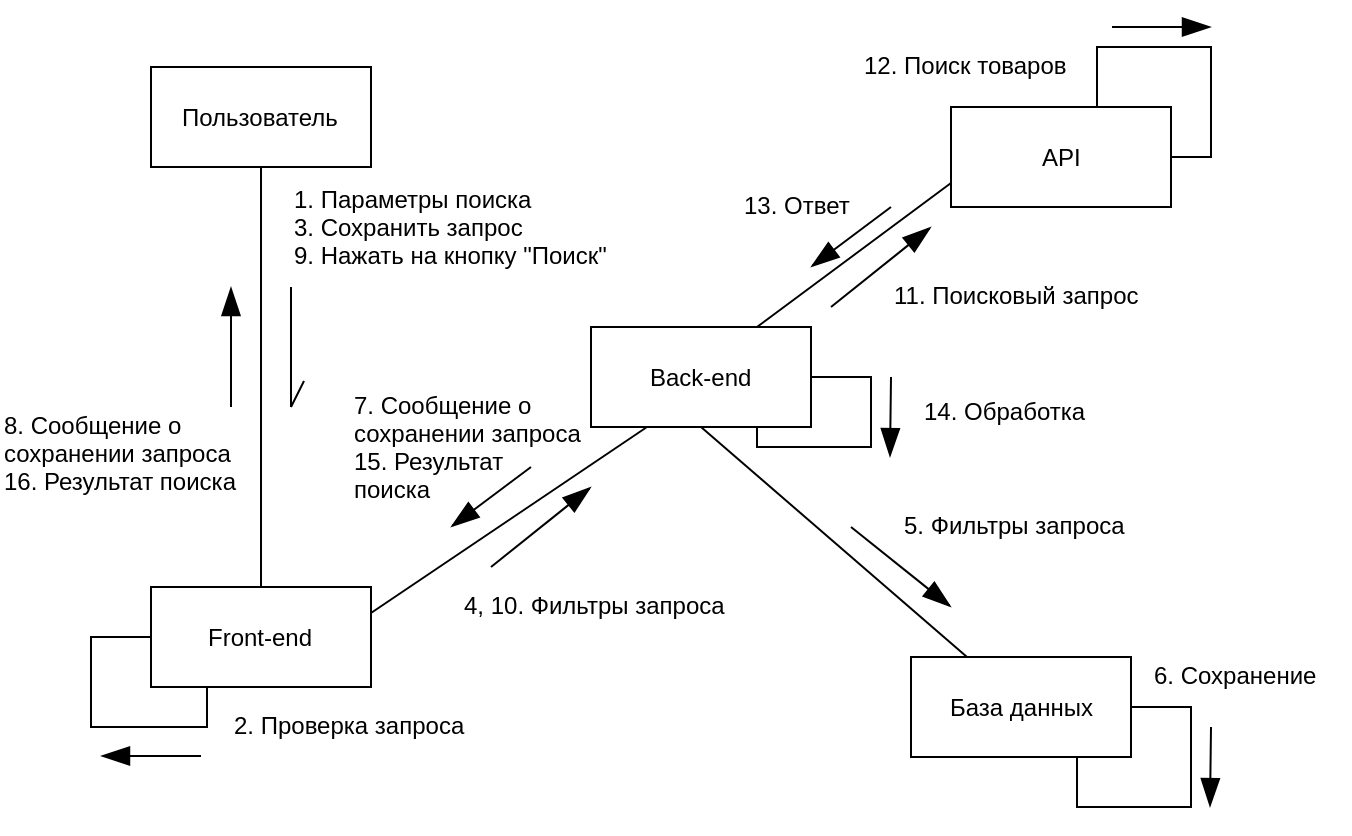
На этой диаграмме отображениы состояния системы при поиске товаров пользователем. После запуска приложение ожидает команды пользователя. Пользователь вводит параметры поискового запроса, после чего порисходит процесс формирования запроса к внешнему API. Затем осуществляется его отправка. Когда получет ответ от API, производится его обработка и отображение результата поиска.

2.4.4. Диаграмма последовательности

Рисунок 4. Диаграмма последовательности

На диаграмме последовательноти изображены процессы сохранения поискового запроса и его выполнения. Пользователь может сохранить поисковый запрос, как только введёт параметры поиска, и затем совершить поиск. При этом система осуществлят необходимую проверку и обработку данных и взаимодействует с базой данных и внешним API.

2.4.5. Диаграмма взаимодействия

Рисунок 5. Диаграмма взаимодействия

На диаграмме взаимодействия представлены те же процессы, что и на диаграмме последовательности.

2.4.7. Схема базы данных

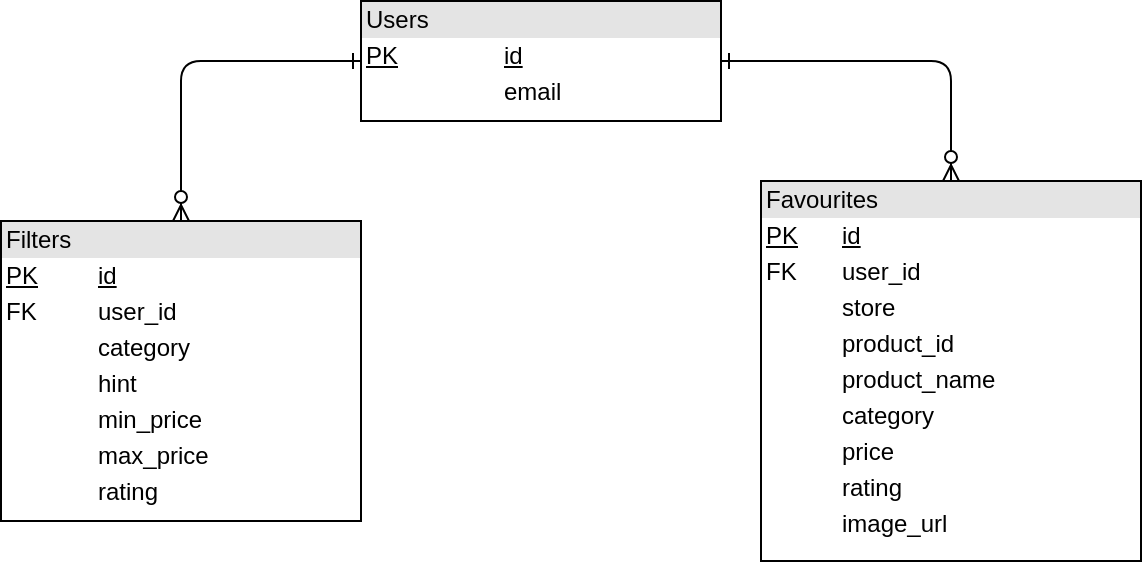


Рисунок 6. Схема базы данных

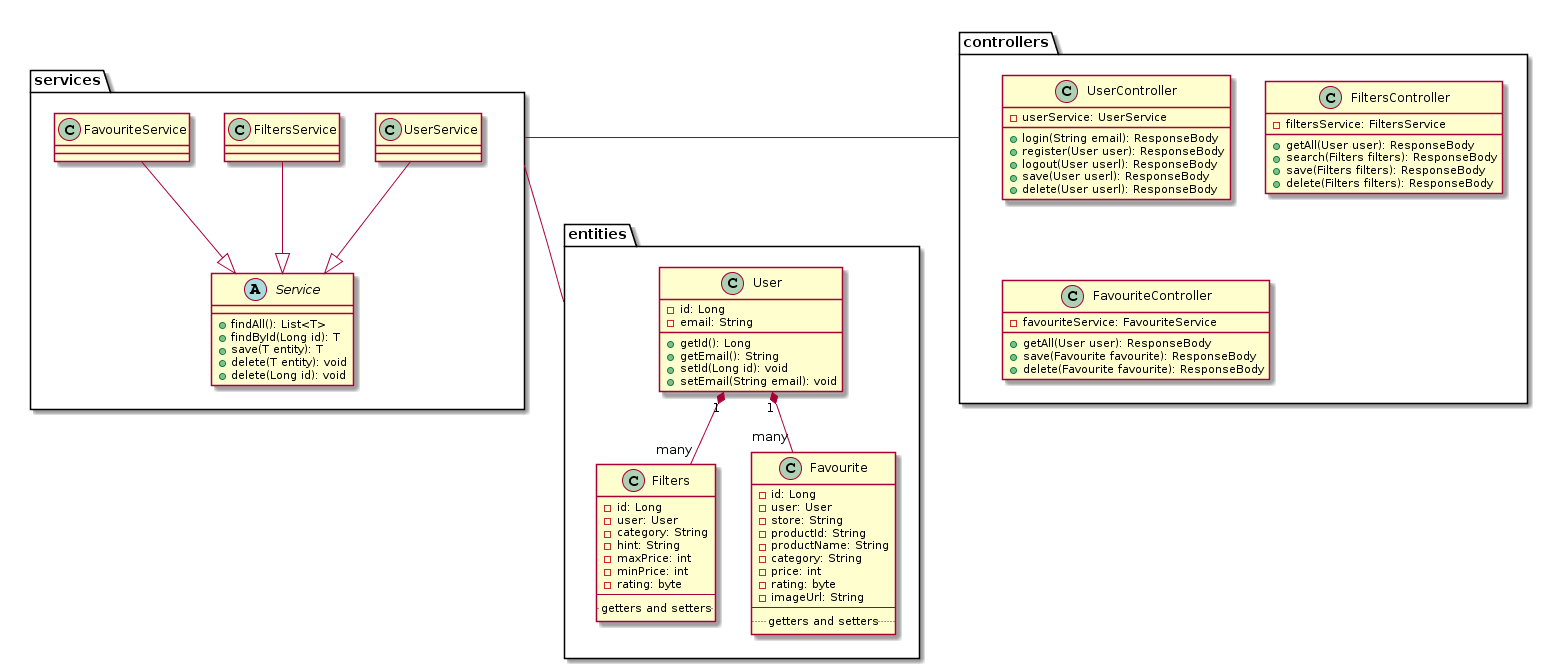
База данных приложения состоит из трёх таблиц:

* Users — хранит информацию о пользователях приложения
  + id — первичный ключ, идентификатор пользователя
  + email — адрес электронной почты пользователя
* Filters — хранит поисковые запросы пользователей
  + id — первичный ключ, идентификатор поискового запроса
  + user\_id — внешний ключ, ссылка на пользователя
  + category — категория (тип) товара
  + hint — название или часть названия товара
  + min\_price — минимальная цена товара
  + max\_price — максимальная цена товара
  + rating — рейтинг товара
* Favourites — хранит отложенные пользователями товары
  + id — первичный ключ, идентификатор отложенного товара
  + user\_id — внешний ключ, ссылка на пользователя
  + store — название магазина
  + product\_id — идентификатор товара в магазине
  + product\_name — полное наименование товара
  + category — категория (тип) товара
  + price — цена товара
  + rating - рейтинг
  + image\_url — ссылка на изображение товара

Отношения между таблицами:

* Users-Filters — один-ко-многим, пользователь может иметь произвольное количество сохранённых поисковых запросов, поисвый запрос обязательно ссылается на одного пользователя
* Users-Favourites — один-ко-многим, пользователь может иметь произвольное количество отложенных товаров, товар обязательно ссылается на одного пользователя

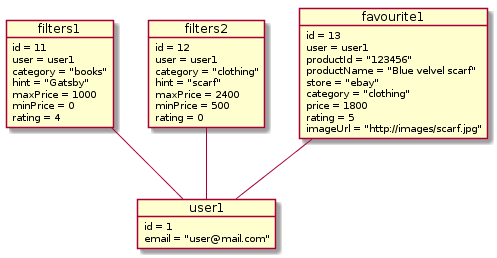
2.4.7. Диаграмма классов

Рисунок 7. Диаграмма классов

В соответсвии со схемой базы данных, back-end часть приложения имеет три класса сущностей (User, Filters и Favourite) с полями, соответствующим таблицам базы. Данные классы помещены в единый пакет entities. Помимо этого, есть также ещё 2 пакета: services, классы которого отвечают за взаимодействие с источниками данных (база данных и API), и controllers, классы которого отвечают за бизнес-логику приложения и обработку данных.

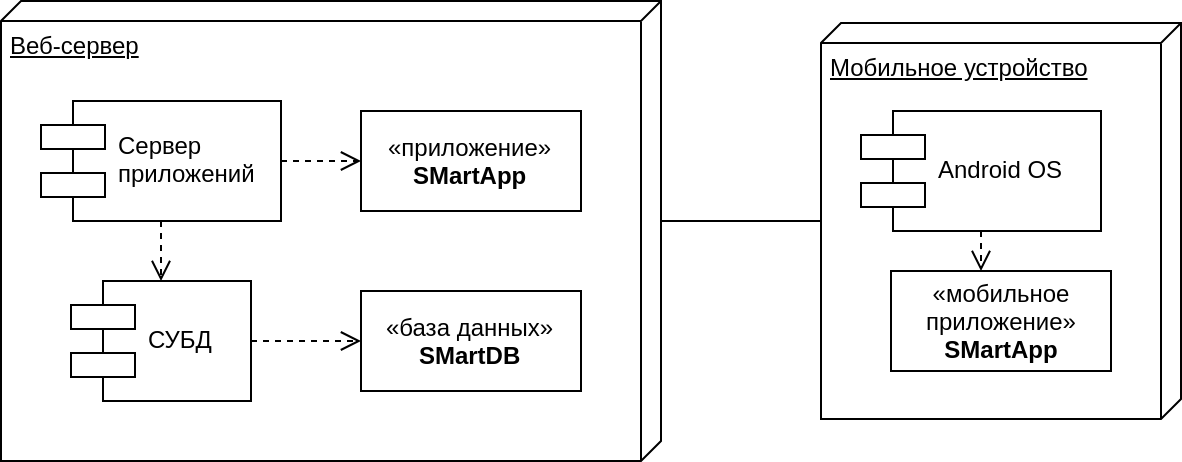
Все запросы от front-end части приложения поступают к классам-контроллерам, которые, в свою очередь, обращаются к сервисам.

2.4.8. Диаграмма объектов

Рисунок 8. Диаграмма объектов

На диаграмме объектов представлен пример экземпляров классов-сущностей. Поля поринициализированы в соответствии с объеткной моделью, изображённой на диаграмме классов. На экземпляр класса User user1 (конкретный пользователь с заданным адресом электронной почты) ссылаются экземпляры классов Filters filters1 и filters2 (некоторые сохранённые поисковые запросы) и экземпляр класса Favourite favourite1 (сохранённый понравившийся товар).

2.4.9. Диаграмма развертывания

Рисунок 9. Диаграмма развёртывания

На диаграмме развёртывания представлены физические средства, необходимые для развёртывания системы. За front-end часть приложения отвечает мобильное устройство с операционной системой Android, на котором установлено приложение S-Mart. Мобильное устройство взаимодействует с веб-сервером, который отвечает за back-end часть приложения и базу данных.

3. Основная часть

Заключение

Список источников

Приложения