Техническое задание

проект KRESLO

2021

Оглавление

[1. Общее положение 2](#_Toc91677685)

[1.1. Роли пользователей 2](#_Toc91677686)

[1.2. Термины и определения 2](#_Toc91677687)

[1.3. Описание системы 2](#_Toc91677688)

[2. Frontend 3](#_Toc91677689)

[2.1. Экраны 3](#_Toc91677690)

[2.2. Дизайн 7](#_Toc91677691)

[3. Beckend 7](#_Toc91677692)

[3.1. База данных 7](#_Toc91677693)

[3.2. Движок 7](#_Toc91677694)

[3.3. Сервер 7](#_Toc91677695)

[3.4. AR ядро 7](#_Toc91677696)

[4. Описание Сервиса 7](#_Toc91677697)

[4.1. 7](#_Toc91677698)

[4.2. ER 8](#_Toc91677699)

[4.3. Аналитика 12](#_Toc91677700)

# Общее положение

Сервис должен обеспечивать стабильную работу приложения на мобильных устройствах с операционными системами Android (с версии 7) и iOS поддерживающих работу дополненной реальности (далее – AR).

Цель сервиса – обеспечить взаимодействие продавцов и покупателей мебели для удовлетворения своих потребностей. Помочь продавцам мебели реализовывать мебельную продукцию, а покупателям приобретать используя технологию визуального восприятия AR, а также функции онлайн покупок и услуг доставки.

Потребность продавцов мебели заключается в том, чтобы реализовывать мебельную продукцию предоставляя своим клиентам современный сервис качества услуг, увеличение объёма продаж и снижения расходов маркетинга на единицу продукции.

Потребность покупателей мебели заключается в приобретении мебели при минимально затраченном времени на выбор из огромного многообразия предложения. Получение доступа к товарам производителей мебели, не выходя из дома при сохранении аналогичного качества восприятия мебели подобно шоурумам.

## Роли пользователей

### Покупатель – физическое лицо, использующее Сервис с целью визуализации интерьера помещения и производящее выбор, оплату товара;

### Продавец – юридическое лицо, использующее Сервис с целью продаж представленного каталога продукции.

## Термины и определения

### Сервис – это мобильное приложение «KRESLO»

### Аналитик – представитель Сервиса обеспечивающий сбор и анализ данных с целью оптимизации работы Сервиса, пользователей системы сокращения затрат и извлечения дополнительной прибыли;

### Разработчик – представитель Сервиса, занимающейся разработкой, созданием, улучшением и обновлением Сервиса по заданию Заказчика;

### Заказчик – инициатор создания Сервиса, владелец прав на разработанный Сервис включая все улучшения и обновления, выполненные Разработчиком.

## Описание системы

После загрузки и установки мобильного приложения KRESLO пользователь получает возможность пользоваться сервисом без регистрации.

Первой страницей после запуска открывается [стартовая страница](#_Стартовая_страница) с приветствием, на которой показан логотип и название бренда KRESLO. Внизу страницы номер версии приложения. [Стартовая страница](#_Стартовая_страница) показывается на 3 секунды в это время происходит запуск приложения и загрузка [страницы ленты](#_Страница_ленты) с постов.

На [странице ленты](#_Страница_ленты) рандомно отображаются посты, но в порядке их новизны. Пользователь, двигая вверх и вниз прокручивает ленту и выбирает понравившуюся мебель на фотографии. Под фотографией отображается ряд иконок для реакции на пост: иконка лайка, переслать сообщение, добавить в избранное и поле оставить отзыв о товаре, а также иконка AR нажатие на которые приводит к переходу на экран AR. Под постом кратко представлена о товаре.

Практически на всех экранах приложения присутствует [«подвал» и «шапка»](#_Шапка/подвал).

Кликая по шапке поста на странице ленты, пользователь может перейти на [страницу продавца,](#_Страница_продавца) перейти в меню, корзину, выполнить поиск или отреагировать на пост.

При нажатии на кнопку AR запускается экран AR и пользователю предлагается отсканировать поверхность пола. Пользователь видит, как приложение рисует сетку на полу обозначающую поверхность на которой будет размещён объект. Сервис, самостоятельно определив поверхность пола заменяет на экране AR сетку поверхности на маркер объекта. Выбрав место для установки 3D объекта, пользователь нажимает на экране кнопку и происходит установка 3D модели мебели в AR. Пользователь может вращать и перемещать объект как удобно в соответствии с локацией.

Изменение положения горизонтальной и вертикальной раскладки допускается только для страницы AR экрана.

Выбрав понравившийся товар, производитель добавляет его в корзину. При переходе в корзину он видит товары, их стоимость и количество, а также общую сумму покупки. Нажимаю на кнопку оформить заказ он заполняет свои контактные данные, соглашается с политикой и отправляет заказ продавцу.

Опыт работы продавца мебели отличается от покупателя только тем, что у продавца есть учетная запись, к которой он получает доступ через сайт kreslo.app и может вносить необходимые дополнения и изменения (данный функционал относится к версии 3 и выше).

Продавец мебели самостоятельно связывается с покупателем и закрывает сделку.

# Frontend

## Экраны

Описание страниц, UI/UX дизайн могут изменяться в процессе тестирования и проработки. Дизайн и интерфейс сервиса должны быть интуитивно понятны и знакомы пользователю. Дизайн должен сочетать в себе уже знакомый пользователю опыт (функционал и расположение кнопок) и индивидуально выделять сервис KRESLO среди многообразия приложений.

### Стартовая страница

Это первая страница, которую видит пользователь после запуска приложения. Она служит заставкой на время ожидания загрузки приложения. В центре страницы разместить логотип и название бренда KRESLO. Ниже под брендом KRESLO приветствие. Внизу страницы номер версии приложения. Стартовая страница показывается на 3 секунды в это время происходит запуск приложения и загрузка [страницы ленты](#_Страница_ленты) с постов.

### Шапка/подвал

Шапка и подвал присутствуют на всех страницах кроме стартовой страницы и экрана AR.

Шапка находится под строкой состояния и состоит из двух элементов. В левой части кнопка «Бургер» нажатие на которую открывает страницу меню. В правой части логотип KRESLO.

Подвал находится в нижней части экрана и состоит из трех кнопок, слева на право: «Дом» (возвращает пользователя на страницу ленты), «Лупа» (открывает страницу поиска) и «Корзина» (открывает страницу корзина).

### Пост продавца

Состоит может содержать информацию только об одном товаре и состоит из четырех блоков:

* Фотография товара, показывающая внешний вид и стиль товара. Может содержать до 10 фотографий, свайпая влево и вправо можно листать фотографии. При наличии более одной фотографии справа от фотографии выглядывает узкая полоска следующей фотографии;
* Реакции на пост: лайк, оставить отзыв, переслать, запустить AR и добавить в избранное. Количество лайков и отзывов выводится счетчиком, а кнопка AR выделяется на общем фоне гармонично вписываясь в общий дизайн;
* Описание товара – описание, которое продавец добавляет к товару состоит из общего описания, наименования материалов, габаритных размеров мебели и цены товара;
* Отзывы – аналогично описанию товара свёрнут в стандартном положении и при нажатии на надпись «Отзывы» разворачивается тело блока.

Описание товара в стандартном состоянии свёрнуто и обозначается треугольником вправо слева от надписи «Описание товара». При нажатии на надпись описание товара вниз разворачивается тело блока описание товара, а треугольник поворачивается в вниз. При повторном нажатии на надпись блок описание товара сворачивается и треугольник возвращается.

Блок отзывов и всю связанную с ним логику выполнять в последующих версиях и обновлениях. Блок отзывов состоит из стандартного поля для написания отзыва и перечислением внизу уже оставленных отзывов в порядке новизны в верхней части блока.

### Страница ленты

Страница ленты состоит из двух блоков (сверху вниз):

* Продавец – полоса над фотографией, слева лого производителя, справа от лого наименование бренда мебели;
* Посты продавцов мебели. Посты чередуются рандомно друг за другом сверху вниз по новизне публикации.

Пользователь, прокручивая ленту постов выбирает понравившийся товар и совершает свой выбор. При возврате на страницу ленты с других страниц пользователь возвращается к тому посту, на котором он остановился.

### Страница продавца

На страницу продавца пользователь попадает нажатием на логотип или название бренда вверху поста в ленте постов.

Страница продавца состоит из четырех блоков (сверху вниз):

* Логотип продавца в верхней левой части экрана;
* Счетчики реакций пользователя: лайк, избранное, количества товаров, количества покупок;
* Контактные данные продавца: телефон, email, адрес нахождения. Если какие-либо данные не указаны, то данное поле не выводится;
* Фотографии постов в виде таблицы из трех столбцов. Всю страницу можно прокрутить вверх для отображения фотографий других товаров, не поместившихся на экране.

При нажатии на продавца на фотографию расположение фотографий меняется с табличного вида на ленту, при этом вернуться к табличному виду пользователь может использую кнопку «Назад» на ОС девайса. Лента постов на странице продавца отображает фотографии только данного продавца аналогично основной ленте.

### Страница поиска

Функцию поиска выполнять в последующих версиях и обновлениях.

Страница поиска состоит из двух блоков, вверху экрана полосой фильтр по типам мебели, внизу сгруппированные по типам товаров посты. Посты группируются в горизонтальный ряд, который можно сдвигать влево и вправо. Например, группа диваны в которой представлены посты товаров разных производителей типа «диван». Каждая группа имеет шапку с название типа мебели. Например, Диваны. Посты в каждой группе формируется рандомно. Группы располагаются друг под другом и могут сдвигаться вверх и вниз для поиска подходящей группы.

### Экран AR

Экран AR должен содержать строку состояния, шапку и подвал, кнопку добавить в корзину и кнопку выбора материала.

После запуска экрана AR автоматически запускается процесс сканирования плоскости поверхности. После определения поверхности Сервис предлагает пользователю улучшить качество сканирования путем перемещения телефона, в это время внизу экрана заполняется шкала. Заполнение шкалы сообщает пользователю о том, что сканирование завершено успешно и появляется кнопка «Ок» завершающая сканирование и запускающее процедуру работы с 3D моделью. При запуске процедуры работы с моделью появляется выбранная 3D моделью. Пользователь может нажатием на модель одним пальцем перемещать её влево, вправо, вверх и вниз по плоскости. Двумя пальцами пользователь может вращать модель вокруг своей вертикальной оси. При перемещении и вращении модели объект приподнимается над плоскостью образуя небольшую тень.

### Страница меню

Не описана

### Страница корзины

Не описана

### Страница оформления заказа

На странице должны быть поля ввода адреса (город, улица, дом, квартира), телефона, email. Все пункт должны быть обязательны для заполнения. Внизу пользователь должен поставить галочку согласия с политикой. После их заполнения кнопка «Оформить заказ» внизу экрана становится доступной для отправки заказа. После нажатия на кнопку «Оформить заказ» пользователю показывается сообщение «Спасибо. Информация о заказе отправлена вам на почту.»

## UX дизайн

Не описана

## UI дизайн

Не описана

# Beckend

## Общие требования к коду.

Приложение собрать на основе фреймворка Flutter c применением необходимых библиотек под операционную систему Android (начиная с версии 7) и iOs.

### Стандарты оформления кода.

Необходимо описывать в комментариях каждый метод и процедуру, что данный метод или процедура делает, параметры возвращаемого значения.

### Давать наглядные имена.

Использовать длинные наглядные имена наподобие complementSpanLength, чтобы помочь себе и коллегам понять код в будущем. Исключения составляют несколько важных переменных, используемых в теле метода, наподобие итераторов циклов, параметров, временных значений или результатов исполнения.

Несколько простых правил именования: имена классов и типов должны быть существительными, название метода должно содержать глагол, если метод определяет, является ли какая-то информация об объекте истинной или ложной, его имя должно начинаться с «is». Методы, которые возвращают свойства объектов, должны начинаться с «get», а устанавливающие значения свойств — «set».

### Комментарии и документирование.

По мере разработки кода записывать в комментариях роль каждого файла и класса, содержимое каждого поля класса и основные шаги сложного кода.

### Не повторяться.

Не использовать копирование и вставку кода. Вместо этого выделять общую часть в методе или класс (или макрос, если нужно), и использовать его с соответствующими параметрами. Избегать использования похожих данных и кусков кода.

Следует использовать следующие техники:

* Создание справочников API из комментариев;
* Автоматическая генерация Unit-тестов на основе аннотаций или соглашений об именовании;
* Получение структуры классов из базы данных (или наоборот).

### Проверять на ошибки и реагировать на них.

Методы могут возвращать признаки ошибки или генерировать исключения. Обрабатывать их, не полагаться на то, что диск никогда не заполнится, конфигурационный файл всегда будет на месте, приложение будет запущено со всеми нужными правами, запросы на выделение памяти всегда будут успешно исполнены, или что соединение никогда не оборвётся.

### Разделять код на короткие, обособленные части.

Каждый метод, функция или блок кода должны умещаться в обычном экранном окне (25-50 строк). Если получилось длиннее, разделить на более короткие куски. Даже внутри метода разделять длинный код на блоки, суть которых можно описать в комментарии в начале каждого блока.

Каждый класс, модуль, файл или процесс должен выполнять определённый род задач. Если часть кода выполняет совершенно разнородные задачи, то разделить его соответственно.

### Использование API Flutter и сторонних библиотек.

Изучить, какие функции доступны с помощью API фреймворка Flutter, а также что могут делать развитые сторонние библиотеки.

### Не переусердствуйте с проектированием.

Проектировать только то, что актуально сейчас. Код можно делать довольно обобщённым, чтобы он поддерживал дальнейшее развитие, но только в том случае, если он не становится от этого слишком сложным. Не создавать параметризованные классы, фабрики, глубокие иерархии и скрытые интерфейсы для решения проблем, которых даже не существует.

### Последовательность.

Делать одинаковые вещи одинаковым образом. При разработке метода, функциональность которого похожа на функциональность уже существующего, то использовать похожее имя, похожий порядок параметров и схожую структура тела. То же самое относится и к классам. Создавать похожие поля и методы, делать им похожие интерфейсы, и сопоставлять новые имена с уже существующими в похожих классах.

Код должен соответствовать соглашениям фреймворка Flutter, например, делать диапазоны полуоткрытыми: закрытыми (включающими) слева (в начале диапазона) и открытыми (исключающими) справа (в конце). Если для конкретного случая нет соглашений, то самостоятельно выполнить выбор и придерживайтесь его.

### Безопасность.

Следует предусмотреть переполнении буфера, кросс-сайтовых сценариев, SQL-инъекциях и прочих подобных проблемах.

### Структура данных и алгоритмы.

Совмещать сопровождаемость и эффективность, используя структуры данных и алгоритмы, которые даёт Flutter.

### Unit-тесты.

Обеспечить каждый значительный участок кода тестами, которые позволят проверить корректность его работы. Этот подход упрощает отладку, т.к. он позволяет обнаружить ошибки раньше.

### Сохранять код портируемым.

Если нет особой причины, не использовать функциональность, доступную только на определённой платформе. Не полагаться на то, что определённые типы данных (как integer, указатели и временные метки) будут иметь конкретную длину (например, 32 бита), потому что этот параметр отличается на разных платформах. Хранить сообщения программы отдельно от кода и на зашивайте параметры, соответствующие определённой культуре (например, разделители дробной и целой части или формат даты). (Соглашения нужны для того, чтобы код мог запускаться в разных странах, так что сделать локализацию настолько безболезненной, насколько это возможно).

### Код должен быть собираемым.

Простая команда должна собирать ваш код в форму, готовую к распространению. Команда должна позволять быстро выполнять сборку и запускать необходимые тесты. Для достижения этой цели необходимо использовать средства автоматической сборки наподобие [Make](https://en.wikipedia.org/wiki/Make_(software)), [Apache Maven](https://maven.apache.org/), или [Ant](https://ant.apache.org/). Должна быть установлена интеграционная система, которая будет проверять, собирать и тестировать код при любом изменении.

### Размещать всё в системе контроля версий.

Все элементы — код, документация, исходники инструментов, сборочные скрипты, тестовые данные — должны быть в системе контроля версий [GitHub](https://github.com/)

## База данных

Не описано

## Сервер

Не описано

API

## AR ядро

Чек лист программисту

Функционал сервиса по версиям

# Описание работы со статистическими данными

## Сбор данных

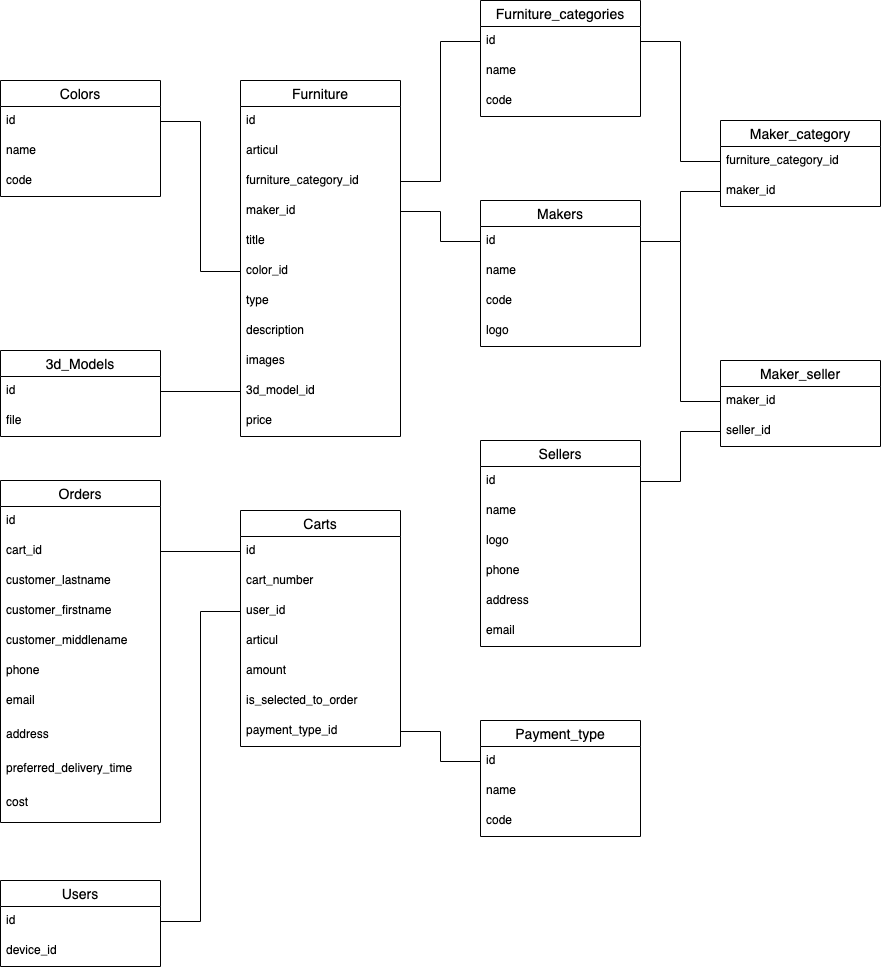
Должен осуществляться сбор данных: о выбранных способах оплаты, о городе доставки, о заказанном товаре. Хранение этих данных на сервере, для дальнейшего анализа.

## Визуализация данных

По способам оплаты должна строится круговая диаграмма. Должны отображаться города списком по убыванию количества заказов. Также для анализа требуется отображать 10 самых популярных товаров для заказа.

## Сторонние API для сбора статистических данных

## Схем базы данных



## Описание таблицы Furniture

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| articul | string | Артикул товара от продавца | Да |
| furniture\_category\_id | int | Категория мебели (диван, кресло и т.д.) | Да |
| maker\_id | int | Производитель | Да |
| title | string | Название мебели | Да |
| color\_id | int | Цвет | Да |
| type | string | Тип (например, стол квадратный ЛАКК и стол прямоугольный ЛАКК) | Нет |
| description | string | описание товара | Нет |
| images | [string] | ссылка на изображение | Да |
| 3d\_model | string | ссылка на 3д-модель | Да |
| price | float | цена | Да |

### Описание таблицы Colors

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| name | string | Название | Да |
| hex | string | цвет в HEX | Да |

### Описание таблицы Furniture\_categories

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| name | string | Название | Да |
| code | string | Кодовое название | Да |

### Описание таблицы Makers

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| name | string | Название | Да |
| code | string | Кодовое название | Да |
| logo | string | Лого в base64 | Нет |

### Описание таблицы Maker\_category

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| furniture\_category\_id | int |  | Да |
| maker\_id | int |  | Да |

### Описание таблицы Sellers

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| name | string | Название | Да |
| logo | string | Лого в base64 | Нет |
| phone | string | телефон | Да |
| address | string | адрес | Нет |
| email | string | email | Да |

### Описание таблицы Maker\_seller

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| seller\_id | int |  | Да |
| maker\_id | int |  | Да |

### Описание таблицы Users

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| device\_id | string | Идентификатор устройства | Да |

### Описание таблицы Orders

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| user\_id | int | Ссылка на пользователя | Да |
| cart\_number | string | ИД корзины, только где is\_selected\_to\_order=true | Да |
| customer\_lastname | string | Фамилия | Да |
| customer\_firstname | string | Имя | Да |
| customer\_middlename | string | Отчество | Нет |
| phone | string | телефон | Да |
| email | string | емайл | Да |
| address | string | адрес доставки | Да |
| preferred\_delivery\_time | datetimeoffset | Желаемые дата и время доставки | Да |
| cost | float | Стоимость заказа | Да |

### Описание таблицы Carts

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| user\_id | string | Идентификатор устройства | Да |
| cart\_number | string | Номер корзины | Да |
| articul | string | Артикул товара | Да |
| amount | int | Количество единиц товара | Да |

### Описание таблицы Payment\_type

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип поля | Описание | Обязательность |
| id | int |  | Да |
| name | string | Название | Да |
| code | string | Кодовое название | Да |

## Аналитика

### Формировать таблицу Carts по корзинам, которые создаются пользователями.

### Формировать таблицу Orders по оплаченным заказам, которые создаются пользователями.

### На экране “Способ оплаты” на каждый выбранный тип оплаты сохранять следующую информацию:

* user\_id
* payment\_type.id
* cart\_number

### На нажатие кнопки “Примерить в комнате” с [экрана](https://www.figma.com/proto/JH70ahrIjYalvf5iI5jFR0/AR-%D0%BC%D0%B5%D0%B1%D0%B5%D0%BB%D1%8C?node-id=12%3A224&viewport=588%2C-1695%2C0.19242721796035767&scaling=scale-down) сохранять следующую информацию:

* user\_id
* date\_time

### На нажатие кнопки “Добавить в корзину” с [экрана](https://www.figma.com/proto/JH70ahrIjYalvf5iI5jFR0/AR-%D0%BC%D0%B5%D0%B1%D0%B5%D0%BB%D1%8C?node-id=12%3A224&viewport=588%2C-1695%2C0.19242721796035767&scaling=scale-down) сохранять следующую информацию:

* user\_id
* articul
* date\_time

### На нажатие кнопки “Добавить в корзину” с [экрана](https://www.figma.com/proto/JH70ahrIjYalvf5iI5jFR0/AR-%D0%BC%D0%B5%D0%B1%D0%B5%D0%BB%D1%8C?node-id=12%3A272&viewport=588%2C-1695%2C0.19242721796035767&scaling=scale-down) сохранять следующую информацию:

* user\_id
* articul
* date\_time

### Дата и время запуска и выключения приложения (События запуска приложения и выключения приложения)

* user\_id
* date/time

### На выбор категории мебели (событие “Выбор категории” с указанием id категории)

* user\_id
* date\_time
* category\_id

### На выбор производителя (событие “Выбор производителя” с указанием, id производителя)

* user\_id
* date\_time
* maker\_id

### Во все события (кроме запуска приложения) добавить атрибут Session\_id - идентификатор сессии пользователя. Сессия = время от запуска приложения до завершения работы приложения.

### На переход в карточку товара (событие “Просмотр товара” с указанием, с какого экрана был совершен переход)

* user\_id
* date\_time
* furniture\_id
* previous\_screen

### Переход из карточки товара в каталог (событие “Выход в каталог”)

* user\_id
* date\_time
* furniture\_id