

ТЗ на ЛК и мобильное приложение для химчистки

01/04/2016

1 Описание проекта

Рабочее название проекта - cleanmeupscotty. Состоит из двух частей - личного кабинета пользователя (представляет собой веб-сайт на Битриксе) и двух мобильных приложений (или одним с разными доступами и разным функционалом).

subsectionЛичный кабинет (сайт) Суть заключается в следующем - в личный кабинет (ЛК) на сайт поступает информация от сервера, который обрабатывает данные ПО химчистки. Например о статусе определенной вещи. Также видна некая статистика - сколько вещей было сдано, сколько денег потрачено, сколько бонусных баллов заработано.

1.1 Мобильное приложение для клиента

Дублирует основной функционал личного кабинета (выводит данные сервера химчистки). Дополнительно позволяет вызывать водителя к текущему местоположению для того, чтобы он забрал вещи.

1.2 Мобильное приложение для водителя

Позволяет принимать заказ от клиента. Отправляет данные на сервер химчистки.

2 Схема работы

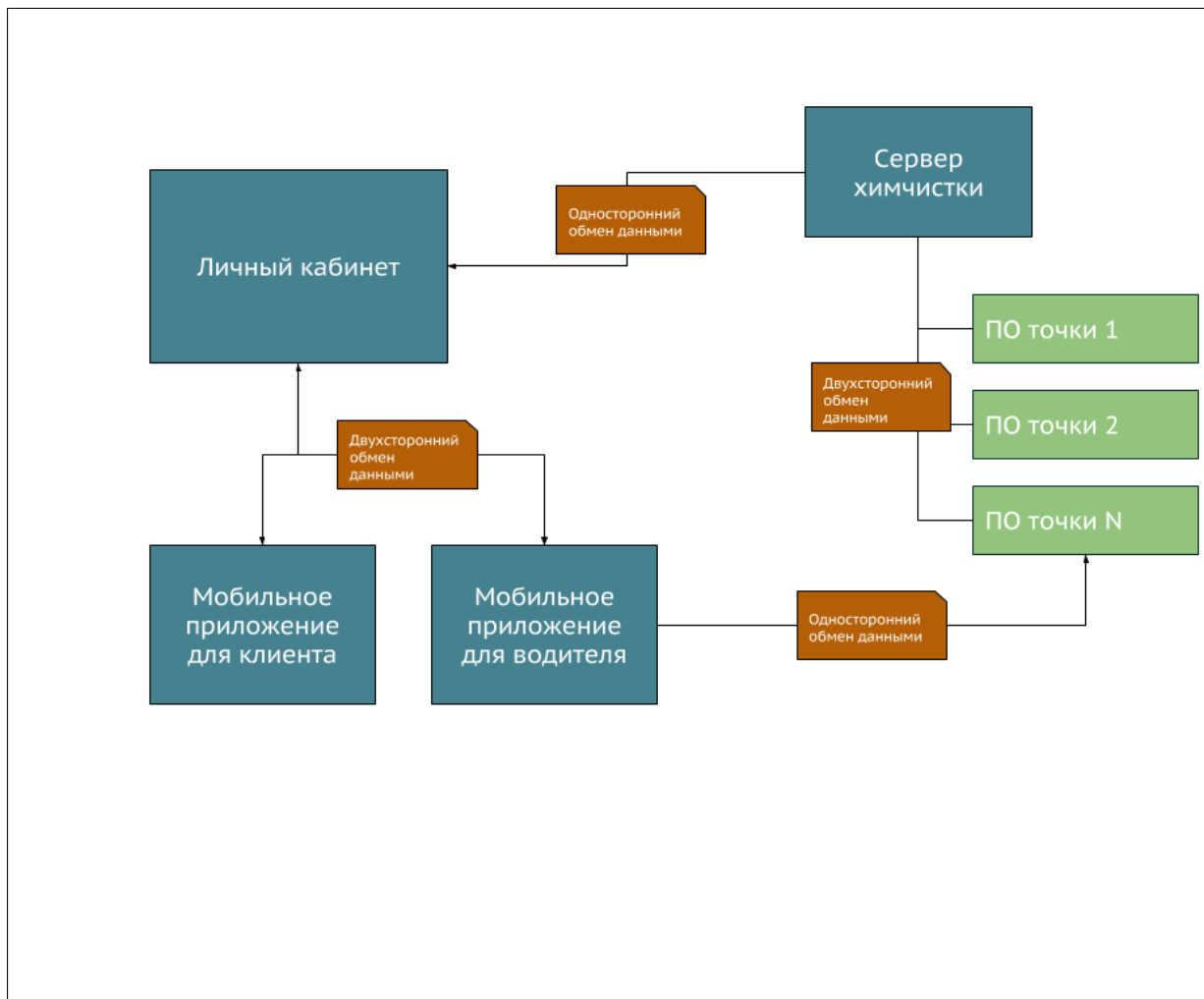


Рис. 1: Схема связей в общей системе

Из точек приема 1...N поступают данные о заказах на общий сервер. Сервер отправляет данные также и обратно (эта сторона не касается проекта, данный обмен налажен на стороне ПО химчистки).

У сервера есть интерфейс приложения (application interface, API), с помощью которого можно запрашивать данные с сервера. Эти данные показываются клиенту в его личном кабинете (на сайте или в мобильном приложении).

Заказывая сбор вещей на дом, клиент вызывает водителя, который забирает грязные вещи. Водитель через свое приложение отмечает что он их принял.

Нужно ли? Водитель разве не просто отдает вещи в одной из точек приема? Тогда нижняя стрелка не нужна

3 Параметры

3.1 Личный кабинет

3.1.1 Запрос

К серверу химчистки отправляется GET-запрос, который содержит в себе уникальный идентификатор клиента. Пример запроса:

http://api.cleanmeupscotty.ru/params?clientid=7817236178

3.1.2 Ответ

От сервера химчистки приходят данные в формате XML или JSON. Перечень данных:

- Кол-во вещей в обработке
- Накопленные баллы
- Общее кол-во сданных вещей
- Перечень вещей в обработке (для каждой вещи получаем уникальный айди от химчистки, наименование, статус по химчистке, цену на
- Общий перечень всего сданных вещей
- Статус клиента

Пример ответа может выглядеть так:

```
1 {"client": {
2   "id": "7817236178",
3   "status": "vip",
4   "current_items": 3,
5   "bonus_miles": 9351,
6   "clothes_current": {
7     "cloth": [
8       {"id": 4125124, "value": "Platje", "status": "clean", "price": 490},
9       {"id": 89172387, "value": "Rubashka", "status": "being cleaned", "price": 199},
10      {"id": 171273, "value": "Kostjum", "status": "preparation", "price": 2500}
11    ],
12    "furniture": [
13      {"id": 7887182, "value": "Pokrivalo on divana", "status": "preparation", "price": 4900}
14    ]
15  },
16  "clother_all": {
17    "listing_all_clothes": ...
18  }
19 }}
```

3.2 Мобильное приложение клиента

Мобильное приложение является полным дублем веб-личного кабинета и показывает в удобном интерфейсе те же данные (см. пункт 3.1.2). Дополнительно мобильное приложение позволяет вызвать водителя для сбора вещей, которые необходимо отдать в чистку.

3.2.1 Запросы

Мобильное приложение основано на веб-личном кабинете, соответственно мобильное приложение "общается" с сайтом, а тот, в свою очередь с сервером ПО химчистки. Дополнительно к общему функционалу у мобильного приложения есть уникальный функционал - заказ водителя.

3.2.1.1 Вызов водителя При нажатии на кнопку у пользователя есть выбор двух опций - время и место. Время по умолчанию стоит "сейчас". Пользователь может альтернативно выбрать определенное время. Место стоит по умолчанию "здесь" и выбирает текущее местоположение пользователя, альтернативно пользователь может выбрать местоположение.

Для местоположения - так как сервис рассчитан на Москву выбираем только улицу, дом и квартиру.

3.3 Мобильное приложение водителя

Данное приложение получает заявки от мобильных приложений клиентов.

3.3.1 Схема работы

Водитель может принять заказа или проигнорировать его. После того, как он принял заказ должен открываться навигатор и показывать маршрут к месту назначения клиента. По прибытии и забора вещей водитель должен отметить в системе, что он забрал вещи (или не забрал, в этом случае нужно указать причину). Если навигатор отдал назад ответ, что водитель приехал и он не нажал ни одну из кнопок, то экран должен мигать красным.

3.3.1.1 Прием заказа Заказ попадает к водителям в определенном радиусе от клиента. Всегда проверяем по шагу, равному 1км. Если водитель(и) в радиусе 1км от местоположения клиента (или адреса заказа) не отреагировали (или таковых вообще нет), то ищем в радиусе 2км и т.д. Можно поставить лимит 200км, что бы приложение не уходило в бесконечный цикл. Увеличиваем шаг каждые 5 минут.

Пример - клиент размещает заказ на свое текущее местоположение. Приложение ищет водителей, находит одного в радиусе 1км. Тот занят, так как везет обратно вещи другому клиенту. Через некоторое время приложение расширяет радиус поиска до 2км. Там приложение находит двух водителей. Тот из них, который ближе к точке клиента получает запрос быстрее. Каждому водителю в текущем радиусе показываем запрос с 30и секундным задержанием, чем дальше он от точки назначения. Первый водитель ближе к клиенту, но он пропустил сообщение. Через 30 секунд запрос отображается второму водителю, который сразу подтверждает. Он и получает заказ, нажимает "поехали" и включается навигация к точке клиента.

3.3.1.2 Навигация Имеет смысл использовать стороннюю, гугл или яндекс навигаторы например. От них необходим ответ, что по GPS координатам дивайс прибыл к цели.