

# Техническое задание

## Концепция проекта

### Общее описание

Данный софт, позволяет визуализировать в одном месте статистику по рекламным кампаниям из разных источников и складывать её между собой, получая при этом общие значения по всем источникам.

По полученным данным можно собрать дашборд - отчет, доступный для просмотра по сгенерированной ссылке.

В отчете используется статистика по периоду, выбранному при формировании отчета.

Доступ к дашборду осуществляется по ссылке.

Кроме того, софт даёт возможность обернуть существующую ссылку для отслеживания переходов по урлу (Сгенерировав при этом новую) и получить ссылку на вызов пикселя 0x0 для отслеживания ивентов.

Отслеживаемые события записываются в ту же базу, где хранится основная информация.

Почти весь софт построен на Open-Source - коде. Проблем в реализации быть не должно.

### Основные цели проекта

Цель 1. Сейчас сотрудники тратят много времени и сил на выгрузку данных с разных рекламных интерфейсов и сведении статистики в одном месте. Нужно автоматизировать этот процесс, чтобы облегчить их работу

Цель 2. Дать возможность клиенту самостоятельно получать статистику по рекламной кампании в современном интерфейсе.

### Целевая аудитория

Целевой аудиторией для использования сервиса является внутренний пользователь - трафик и аккаунт-менеджеры.

Целевой аудиторией для просмотра дашборда является клиент, купивший рекламное размещение у нашей фирмы.

## **Адаптация под девайсы**

Интерфейс адаптирован под десктоп - версию.

## **Поддержка после разработки**

Требуется сделать интерфейс так, чтобы он требовал к себе как можно меньше внимания после запуска.

В компании мало ресурсов под разработку и администрирование для технической поддержки.

После окончания разработки нужно оставить инструкции по тому, как поддерживать продукт и всю документацию о разработке.

# **Требуемые работы**

## **Подключение и настройка сервера (Amazon AWS)**

Изначально нет никаких серверных мощностей, с которыми мы работаем.

Для работы интерфейса требуется подключить AmazonAWS. (Должен подойти middle-large инстанс на **Elastic cloud 2**)

Опыт работы приветствуется

## **Установка готовых приложений**

На AWS требуется установить два готовых приложения, которые будут работать в связке

### **MySQL**

В качестве базы данных используется MySQL.

### **SuperSet**

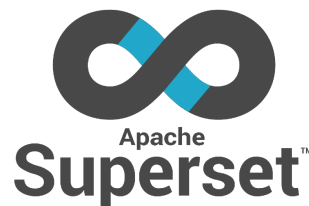
Данные, хранящиеся в MySQL будут читаться и визуализироваться через Apache SuperSet.

Бекенд приложения написан на Python

apache/superset

A modern, enterprise-ready business intelligence web application. Gallery View Dashboards Slice & dice your data  
Query and visualize your data with SQL Lab Visualize geospatial

 <https://github.com/apache/superset>



Его бекенд нужно расположить так же на AmazonAWS, там же, где лежит БД. Фронтенд - на нашем домене.

## Синхронизация различных данных в MySQL



Внимание! Этот пункт наиболее размытый, потому что на данный момент не до конца понятна структура поступающих данных. Скорее всего это будет выгрузка с разбивкой по дням по 4-5 полям.

В MySQL данные будут поступать из стороннего сервиса Renta. Этот сервис получает данные с API рекламных кабинетов и кладет в стороннюю бд (В данном случае - нашу).

Полученные данные из разных рекламных кабинетов, в основном, несут одну и ту же информацию, но названия передаваемых полей может отличаться. Требуется настроить бекенд так, чтобы он собирал все полученные данные в одной коллекции и синхронизировал друг с другом по названию полей.

## Доработки Superset.

Для полноценного закрытия наших задач требуется доработать базовый функционал SuperSet

SuperSet популярный интерфейс с обширным комьюнити. В большинстве случаев легко найти полноценные гайды по доработкам определенных моментов.

## White-Labeling

White-Labeling подразумевает брендинг интерфейса таким образом, чтобы для стороннего наблюдателя, не знакомого с SuperSet он выглядел как наша собственная разработка

Требуется:

- Заменить логотип SuperSet на логотип нашей фирмы.
- Убрать из URL упоминания SuperSet

## **Общие доработки механик**

SuperSet позволяет пересохранять существующие дашборды. Мы используем эту механику для того, чтобы воспроизвести механику шаблонов, чтобы пользователи интерфейса могли максимально быстро создать шаблонный дашборд для новой рекламной кампании.

- При создании копии дашборда (dashboard/save as) автоматически создаются так же копии всех графиков (charts) внутри. Но они сохраняют исходное название, из-за чего не ясно, к какому дашборду относится какой график. Это может вызвать проблемы при редактировании графиков. Нужно, чтобы при копировании в названии новосозданного графика помечалось, что это копия. Пометка должна выглядеть как номер, добавляемый после основного названия. (График / График (1) / График (2))

## **Доработки, связанные с конструктором дашбордов (Dashboards/Charts)**

### **Доработка визуала графиков (Charts / Visualisation Type)**

Графики, сейчас используемые в SuperSet, хоть и информативные, но визуально устаревшие. Для демонстрации клиентам нужно использовать более современные визуально графики. (Пусть и менее информативные)

Требуется добавить дополнительные варианты для графиков типа:

- Line Chart
- Bar chart
- Table

- Для Table важно добавить мгновенную выгрузку таблицы в эксель по нажатию на одну кнопку.
- Doughnut chart
- Big Number with Trendline
  - Нужна небольшая визуальная доработка текущего решения, конкретно - поработать со шрифтами, сейчас цифры сливаются друг с другом.)

- Campaign Filter

Этот элемент нужен для быстрого создания шаблонов из готовых по алгоритму: Скопировал существующий дашборд - поменял в фильтре ID кампаний, которые будут использоваться.

Этот фильтр по механике и визуалу является копией существующего Visualisation type = Filter Box. Полностью дублирует весь функционал существующего фильтра (За одним исключением, он доступен для просмотра при наличии у аккаунта отдельного Permission, ниже информации будет больше)

По итогу в интерфейсе должно быть 2 типа фильтра .

Стандартный Filter Box и Campaign Filter могут находиться в одном дашборде и применяться одновременно по логике "И"

Все графики должны быть выполнены в едином цветовом исполнении, присущему нашему бренду и в единой визуальной стилистике.

### Рефереры графиков:

Vuexy - Vuejs, HTML & Laravel Admin Dashboard Template



♥ <https://pixinvent.com/demo/vuexy-vuejs-admin-dashboard-template/demo-3/>

Vue Element Admin

<https://panjiachen.github.io/vue-element-admin/#/dashboard>

Shards Dashboard Lite - Vue

 <https://designrevision.com/demo/shards-dashboard-lite-vue/blog-overview>

ArchitectUI - Vuejs & Bootstrap 4 & Vuetify Admin UI Dashboard Template

 <https://vue-dashboard-architectui-free.admin-dashboards.com/#/charts/chartjs>

Для всех добавленных графиков так же требуется убрать отображение названия графика, чтобы оно было недоступно для просмотра клиентом (Которое на данный момент располагается в левом верхнем углу каждого графика) в финальном дашборде

### **Доработка компонентов (Dashboards / Components)**

Компоненты - это виджеты, доступные для добавления в шаблон в меню редактирования шаблона.

Нужно добавить компоненты:

- Table (Не путать с Charts / Table)

Виджет, позволяющий загрузить xls/xlsx - таблицу на страницу.  
Желательно добавить Drag`n`Drop загрузку таблицы.



Загруженная таблица должна быть визуально идентична таблице с данными, добавляемой через Charts. Сторонний наблюдатель не должен заметить подмены.

Это делается для того, чтобы таблицу можно было исправить и загрузить вручную, не из базы.

- Image (Png/jpeg/gif)

### **Доработки ролей (Вкладка List Roles)**

SuperSet позволяет добавлять для ролей различные разрешения. Они в корне меняют пользовательский опыт.

Через них можно отключать доступ к вкладкам, датабазам и отображению отдельных элементов. Так как для быстрого создания шаблонов мы используем новый элемент Campaign Filter, который расположен в дашборде, нам требуется спрятать его от стороннего наблюдателя. Для этого:

- Добавить Permissions "Show Campaign Filter".

Если это разрешение отсутствует у пользователя - в его дашборде не отображается Campaign Filter. (Никак, совсем)

При этом фильтрация, выбранная в этом фильтре при создании дашборда, сохраняется.

- Добавить Permissions "work with audit"

Далее будет описание вкладки "Audit". Нужно добавить разрешение на доступ к ней. Оно подразумевает то, что пользователю без разрешения вкладка недоступна и она не отображается в интерфейсе. Пользователю с ролью, в которой указано данное разрешение, доступна вкладка целиком и разрешена работа с интерфейсом

## Переименование метрик

Метрики на фронте у клиента должны быть названы привычными, нетехническими обозначениями (Например, для метрики event\_impression - показ).

Полный список метрик и соотнесенных значений будет понятен после запуска БД и подключения первых интеграций

## Механизм трекинга кликов и событий

powergo/pytracking

This library provides a set of functions that provide open and click tracking when sending emails. This is particularly useful if you rely on an Email Service Provider (ESP) which does not

 <https://github.com/powergo/pytracking>



Не все рекламные кампании ведутся в кабинетах, к API которым у нас есть доступ.

Некоторые данные придется получать с помощью собственного трекинга, развернутого на нашей стороне.

Для этого требуется добавить вкладку, которая позволяет обернуть существующую кликовую ссылку, создав при этом новую ссылку. Новая ссылка является ссылкой нашего домена, по переходу на которую происходит мгновенный редирект на исходную ссылку, при этом к нам в базу записывается событие "click". Такая механика позволяет отследить все клики по рекламе, при этом не затрагивая API сторонних систем.

Кроме того в этой вкладке должен быть функционал, позволяющий сгенерировать ссылку на 0×0 пиксель, по вызову которого будет осуществляться событие "Impression"

Оба этих события в момент срабатывания должны быть записаны в MySQL в ту же коллекцию и в том же виде, как и остальные данные из сторонних рекламных кабинетов.

Позже пользователь будет использовать их при создании дашборда.

Сверху прикреплена библиотека, которая позволяет реализовать эту механику. Можно использовать её или другие аналогичные решения.

## **Интерфейс с ранее созданными ссылками (\* /audit/list/)**

Этот интерфейс выглядит аналогично существующим вкладкам Charts и Dashboards.

Ссылка на переход в интерфейс располагается там же, где и ссылка на переходов Dashboards и Charts.

Название вкладки - Audit

Поля таблицы интерфейса с созданными ссылками:

- Кнопки навигации
- Name
- Campaign Id
- Clicks

Количество кликов за весь период

- Impressions



Количество показов за весь период

- Creator
- Last Modified

## **Интерфейс создания/редактирования трекинговых кампаний (\* /audit/add/ и \* /audit/edit)**

Вкладка для настройки аудита. в ней доступны два текстовых поля:

Текстовое поле с вводом названия

Текстовое поле с вводом ссылки, которую надо обернуть.

## **Интерфейс просмотра трекинговых кампаний (\* /audit/"порядковый номер")**

Информация на этой вкладке - это две строки и фильтр по дате, позволяющий отследить, сколько данных было собрано за какой период.

Строка 1:

- Обернутая ссылка на клик (На нашем домене, которая делает редирект на ссылку, введенную при создании/редактировании и вызывает событие "Click"). Нужна здесь, чтобы пользователь мог скопировать её и вставить в сторонний интерфейс.
- Количество вызовов за выбранный в фильтре период, записанных в нашу базу.

Строка 2:

- Ссылка на 0x0 пиксель (На нашем домене, который при вызове вызывает событие "Impression"). Нужна здесь, чтобы пользователь мог скопировать её и вставить в сторонний интерфейс.
- Количество вызовов за выбранный в фильтре период, записанных в нашу базу.