**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Стажер-исследователь научной лаборатории моделей и методов вычислительной прагматики наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. А. Киселев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |
|  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. |  | | **ИНТЕРАКТИВНЫЕ ДАШБОРДЫ ДЛЯ АНАЛИТИКИ ТРАЕКТОРИЙ ПОЛЬЗОВАТЕЛЕЙ**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.13 01-1-ЛУ**  Исполнитель  Студент группы БПИ172  \_\_\_\_\_\_\_\_\_\_\_\_/А.Н. Карпин/  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. | |
|  |  |

**УТВЕРЖДЕН**

**RU.17701729.04.13 01-1-ЛУ**

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

**ИНТЕРАКТИВНЫЕ ДАШБОРДЫ ДЛЯ АНАЛИТИКИ ТРАЕКТОРИЙ ПОЛЬЗОВАТЕЛЕЙ**

**Пояснительная записка  
  
RU.17701729.04.13 01-1**

**Листов 25**

# **АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программного комплексу «RetentionBoards» («Интерактивные дашборды для аналитики траекторий пользователей»), предназначенной для визуализации и проведения аналитики пользовательских данных.

В разделе «Введение» указано наименование программного комплекса, и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программного комплекса, эксплуатационное назначение программного комплекса и краткая характеристика области применения программного комплекса.

В разделе «Технические характеристики» содержатся следующие подразделы:

– постановка задачи на разработку программного комплекса;

– описание алгоритмов, архитектуры и функционирования программного комплекса с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программного комплекса с другими программами;

– описание и обоснование выбора метода организации входных и выходных данных;

– описание и обоснование выбора состава технических и программных средств.

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

Настоящий 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].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

**Содержание**

[**АННОТАЦИЯ** 2](#_Toc37600224)

[**1.** **ВВЕДЕНИЕ** 5](#_Toc37600225)

[**1.1** **Наименование программного комплекса** 5](#_Toc37600226)

[**1.2** **Документы, на основании которых ведется разработка** 5](#_Toc37600227)

[**2.** **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ** 6](#_Toc37600228)

[**2.1** **Назначение программного комплекса** 6](#_Toc37600229)

[2.1.1 Функциональное назначение 6](#_Toc37600230)

[2.1.2 Эксплуатационное назначение 6](#_Toc37600231)

[**3.** **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ** 7](#_Toc37600232)

[**3.1** **Постановка задачи на разработку программного комплекса** 7](#_Toc37600233)

[**3.2** **Описание алгоритма и функционирования программного комплекса** 7](#_Toc37600234)

[3.2.1 Описание алгоритма программного комплекса 7](#_Toc37600235)

[3.2.2 Структура аналитики 9](#_Toc37600236)

[3.2.3 Возможные взаимодействия программного комплекса с другими программами 12](#_Toc37600237)

[**3.3** **Описание и обоснование выбора метода организации входных и выходных данных** 12](#_Toc37600238)

[3.3.1 Описание метода организации входных и выходных данных 12](#_Toc37600239)

[3.3.2 Обоснование выбора метода организации входных и выходных данных 13](#_Toc37600240)

[**3.4** **Описание и обоснование выбора состава технических и программных средств** 13](#_Toc37600241)

[3.5 Архитектура программного решения 14](#_Toc37600242)

[3.5.1 Архитектура основного веб-приложения 14](#_Toc37600243)

[3.5.2 Архитектура аналитического приложения 15](#_Toc37600244)

[3.5.3 Архитектура инфраструктуры 15](#_Toc37600245)

[**4.** **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ** 16](#_Toc37600246)

[**4.1** **Предполагаемая потребность** 16](#_Toc37600247)

[**4.2** **Экономические преимущества разработки по сравнению с отечественными аналогами** 16](#_Toc37600248)

[**5.** **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 17](#_Toc37600249)

[**ТЕРМИНОЛОГИЯ** 18](#_Toc37600251)

[**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ** 19](#_Toc37600253)

[**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ** 20](#_Toc37600255)

[**СХЕМА БАЗЫ ДАННЫХ** 24](#_Toc37600257)

[**ДИАГРАММА USE-CASE** 25](#_Toc37600259)

[**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** 2](#_Toc37600260)

# **ВВЕДЕНИЕ**

## **Наименование программного комплекса**

Наименование программного комплекса – «Интерактивные дашборды для аналитики траекторий пользователей».

Наименование программного комплекса на английском языке – «Interface Dashboards for Analytics of User Trajectories».

Краткое наименование программного комплекса – «RetentionBoards».

## **Документы, на основании которых ведется разработка**

Разработка ведется на основании приказа Национального исследовательского университета "Высшая школа экономики" № 2.3-02/1004-01 от 10.04.2020.

# **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**

## **Назначение программного комплекса**

### **Функциональное назначение**

Программный комплекс будет применяться с целью анализа поведения пользователей в программных продуктах. Программный комплекс позволяет визуализировать аналитические данные в виде графов о перемещении пользователей на основе загружаемых в систему файлов с событиями, полученными системой аналитики. На основе этих данных система также строит матрицу переходов, отражающую вероятность следующего действия пользовать, и кластеризует пользователей по модели их поведения, позволяя рассмотреть каждый кластер в отдельности.

### **Эксплуатационное назначение**

Программный комплекс будет применяться для продуктовых исследований, UX-исследований, маркетинговых исследований в IT-компаниях. Конечными пользователями продукта могут быть менеджеры продукта, продуктовые и бизнес аналитики, маркетологи, руководители и менеджеров проектов, UX-исследователи и QA-специалисты.

# **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**

## **Постановка задачи на разработку программного комплекса**

Тема работы: «Интерактивные дашборды для аналитики траекторий пользователей». Цель разработки программного комплекса: предоставить конечному пользователю систему, которая может визуализировать данные аналитики и кластеризовать пользователей в зависимости от их поведения

Основные функции разрабатываемой программного комплекса:

1. Загрузка файлов с данными аналитики в систему
2. Расчет матрицы переходов для набора данных
3. Визуализация матрицы переходов для набора данных
4. Кластеризация пользователей по их поведению в программном продукте
5. Визуализация кластеров
6. Построение графа перемещения пользователя по каждому кластеру
7. Управление учетными записями пользователей.

## **Описание алгоритма и функционирования программного комплекса**

### **Описание алгоритма программного комплекса**

Система представляет из себя микросервисное веб-приложение с базой данных, кэшем и очередью сообщений. Общая схема работы системы приведена ниже (рис. 1).

A picture containing computer

Description automatically generated

Рисунок 1 – Схема работы системы

#### **Описание общей схемы работы системы**

1. Пользователь открывает главную страницу
2. Проводит логинизацию в системе или регистрацию в случае необходимости
3. Загружает в систему csv файл с данными событий аналитики
4. Введенные данные сохраняются в базе данных и начинается параллельнай расчет матрицы переходов
5. Создается матрица переходов
6. Создается диаграмма кластеров
7. Создаются графики по каждому кластеру

### **Структура аналитики**

#### **События аналитики**

Событие аналитики представляют из себя запись о действии определенного пользователя в определенное время.

Пример выгрузки событий аналитики:

onboarding\_welcome\_screen,1538397432261002000,80f4a724cd5c5a1a4d187fa59e2aa1e3

onboarding\_\_chooseLoginType,1538397440902004000,80f4a724cd5c5a1a4d187fa59e2aa1e3

onboarding\_privacy\_policyShown,1538397440956005000,80f4a724cd5c5a1a4d187fa59e2aa1e3

onboarding\_login\_Type2,1538397440966006000,80f4a724cd5c5a1a4d187fa59e2aa1e3

Так, событие onboarding\_welcome\_screen обозначает, что пользователь с уникальным идентификатором 80f4a724cd5c5a1a4d187fa59e2aa1e3 увидел экран Welcome в процессе онбординга в 1538397432261002000 такое время по системе UNIX. Аналогично onboarding\_privacy\_policyShown пользователь посмотрел политику приватности и тд.

#### **Траектория пользователя**

Траектория пользователя представляет последовательность событий аналитики в определенных временных точках. Например, события из пункта 3.2.1.2 образуют траекторию onboarding\_welcome\_screen, onboarding\_\_chooseLoginType, onboarding\_privacy\_policyShown, onboarding\_login\_Type2, что соответствует действиям пользователя «Увидел приветственный экран», «Выбрал тип логина», «Увидел политику приватности», «Выбрал второй тип логина».

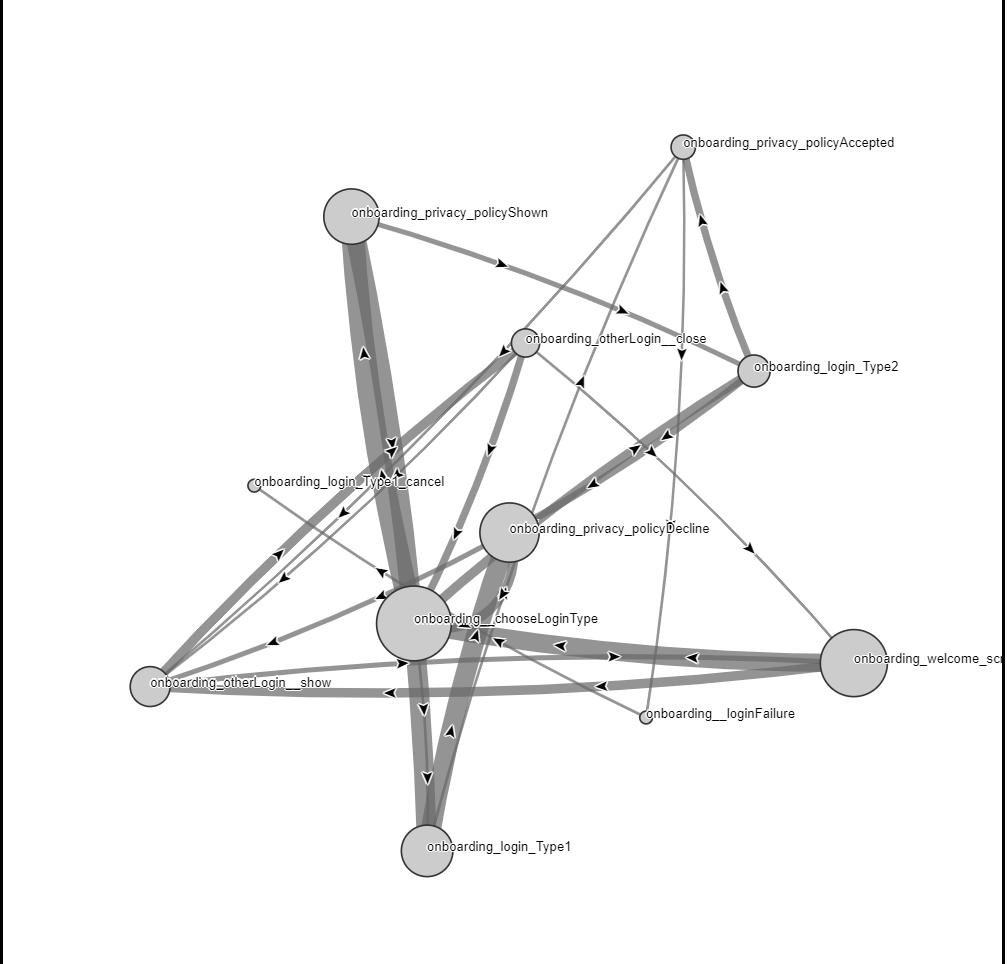


Рисунок 2 - Пример графа траекторий пользователей

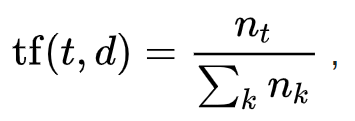
#### **Кластеризация пользовательских траекторий**

Траекторию пользователя можно представить как точку с помощью непрерывных представлений, например, [TF-IDF](http://datameetsmedia.com/bag-of-words-tf-idf-explained/). После преобразования траектория становится точкой в пространстве, где по осям отложена нормированная встречаемость в траектории различных событий и переходов между ними. Затем эту точку можно спроецировать на плоскость с помощью [TSNE](https://www.quora.com/How-does-t-SNE-work-in-simple-words). TSNE — преобразование, снижает размерность пространства до 2 осей и по возможности сохраняет относительные расстояния между точками. Соответственно, становится возможным на плоской карте, образной проекционной карте траекторий, изучать как располагались между собой точки разных траекторий.

#### **TF-IDF**

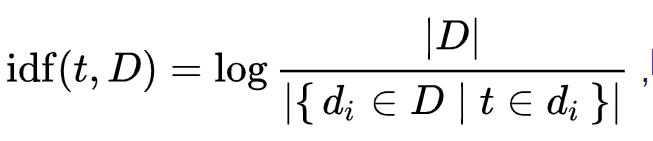
TF-IDF - это статистическая мера, используемая для оценки важности слова в контексте [документа](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82), являющегося частью коллекции документов или [корпуса](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BF%D1%83%D1%81%D0%BD%D0%B0%D1%8F_%D0%BB%D0%B8%D0%BD%D0%B3%D0%B2%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0). Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.

TF - отношение числа вхождений некоторого слова к общему числу слов документа. Таким образом, оценивается важность слова в пределах отдельного документа.

{\displaystyle \mathrm {tf} (t,d)={\frac {n\_{t}}{\sum \_{k}n\_{k}}}}

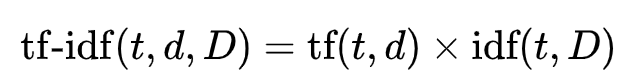
где n - число вхождений слова t в документ, а в знаменателе общее число слов в данном документе.

IDF - инверсия частоты, с которой некоторое слово встречается в документах коллекции. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF.

{\displaystyle \mathrm {idf} (t,D)=\log {\frac {|D|}{|\{\,d\_{i}\in D\mid t\in d\_{i}\,\}|}}}  ,[[2]](https://ru.wikipedia.org/wiki/TF-IDF#cite_note-2)

где |D| — число документов в коллекции; — число документов из коллекции D, в которых встречается t.

Таким образом, мера TF-IDF является произведением двух сомножителей:

{\displaystyle \operatorname {tf-idf} (t,d,D)=\operatorname {tf} (t,d)\times \operatorname {idf} (t,D)}

#### **t-SNE**

Алгоритм t-SNE состоит из двух главных шагов. Сначала t-SNE создаёт [распределение вероятностей](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9) по парам объектов высокой размерности таким образом, что похожие объекты будут выбраны с большой вероятностью, в то время как вероятность выбора непохожих точек будет мала. Затем t-SNE определяет похожее распределение вероятностей по точкам в пространстве малой размерности и минимизирует [расстояние Кульбака — Лейблера](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5_%D0%9A%D1%83%D0%BB%D1%8C%D0%B1%D0%B0%D0%BA%D0%B0_%E2%80%94_%D0%9B%D0%B5%D0%B9%D0%B1%D0%BB%D0%B5%D1%80%D0%B0) между двумя распределениями с учётом положения точек.

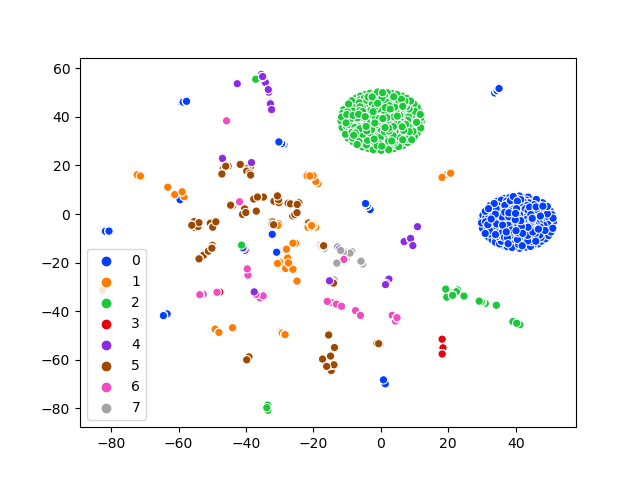


Рисунок 3 - Пример отображения кластеров пользователей

### **Возможные взаимодействия программного комплекса с другими программами**

Программный комплекс не взаимодействует ни с какими внешними программами.

## **Описание и обоснование выбора метода организации входных и выходных данных**

### **Описание метода организации входных и выходных данных**

Программный комплекс должна принимать на вход .csv файл, содержащий данные о событиях в формате event\_name, event\_timestamp, user\_pseudo\_id, где event\_name – название события в системе аналитики, event\_timestamp – идектификатор времени происхождения события, user\_pseudo\_id – уникальный идектификатор пользователя в системе аналитики.

Результаты работы программного комплекса доступны для просмотра через интерфейс пользователя, а также сохраняются в базу данных системы.

### **Обоснование выбора метода организации входных и выходных данных**

В качестве источника данных служат файлы .csv, что позволяет быстро импортировать данные, не реализовывая обработку стримов данных и сервисов потоковой обработки данных.

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

Для правильного функционирования программного комплекса требуется компьютер, обладающий следующими техническими характеристиками:

1. Процессор с тактовой частотой 2 ГГц или более и 2 ядрами. Минимальная тактовая частота процессора — 1 ГГц;
2. Рекомендуется 8 ГБ ОЗУ или более. Минимальный объём – 4 ГБ;
3. Не менее 8 ГБ свободного места на жёстком диске, на котором хранится программный комплекс;
4. Мышь Microsoft или совместимое указывающее устройство.
5. Монитор с разрешением 1024x768 или выше.
6. Клавиатура
7. Интернет-соединение скоростью не менее 5Мб/сек

Состав программных средств, необходимых для работы системы:

Для эксплуатации:

Операционная система с поддержкой выхода в сеть интернета, стабильное интернет-соединение скоростью не меньше 500 Кбит/сек, а также браузер аналогичный Google Chrome последних трех версий.

Для развертывания:

Операционная система Windows 10 Pro или более поздняя версия Windows. Должен быть установлен Docker Community Edition.

### **Архитектура программного решения**

Приложение реализует микросервисную архитектуру. Данный вид архитектуры характеризуется низкой связанностью и высокой масштабируемостью, так как нагрузку можно распределить между несколькими копиями одного сервиса.

A picture containing screen, black, monitor, dark

Description automatically generated

Рисунок 4 – Архитектура системы

### **Архитектура основного веб-приложения**

Основное веб приложение написано на основе фреймворка Django 3.0.5,

Приложение имеет трехслойную архитектуру, в которую входят слои работы с моделью, слой сервисов и бизнес-логики, и слой презентации. архитектурной модели MVT (Model – View – Template). Микросервис отвечает за взаимодействие с пользователями, за регистрацию и логинизацию, хранение информации о наборах данных и управление очередями сообщений с помощью фреймформа Celery. Запросы кэшируются в Redis Cache.

Основным способом взаимодействия с базой данных SQL Django ORM - ORM (Object-Relational Mapping) фреймворк, созданный Microsoft, позволяющий сопоставлять таблицы БД с классами модели программы. Основными моделями программы являются:

* User – характеризует пользователя системы;
* Event – событие аналитики;
* Eventset – набор данных;

Все модели являются «Rich Domain Model», то есть включают в себя всю логику и валидацию и не полагаются на отдельные сервисы по обработке данных, предоставляя более объектно-ориентированный подход.



Рисунок 5 – Архитектурная модель MVT

### **Архитектура аналитического приложения**

Основой аналитического приложения также является фреймворк Django 3.0.5 и библиотека Retentioneering, разработанная М. Годзи, А. Зайцевым и Д. Киселевым, созданная для решения задачи построения траекторий пользователей и их кластеризации.

### **Архитектура инфраструктуры**

Инфраструктура приложения базируется docker-compose файлах, позволяющих развернуть приложение, как в dev, так и в production среде. В Docker производится развертывание основного веб приложения, аналитического приложения, базы данных PostrgeSQL, двух хранилищ данных Redis, сервиса Celery для основного приложений, сервиса Сelery для веб приложения и сервиса мониторинга очередей Flower.

Инфраструктура реализуем механизм Publish – Subscribe шаблон обмена сообщениями, при котором издатели отправляют сообщения подписчикам.

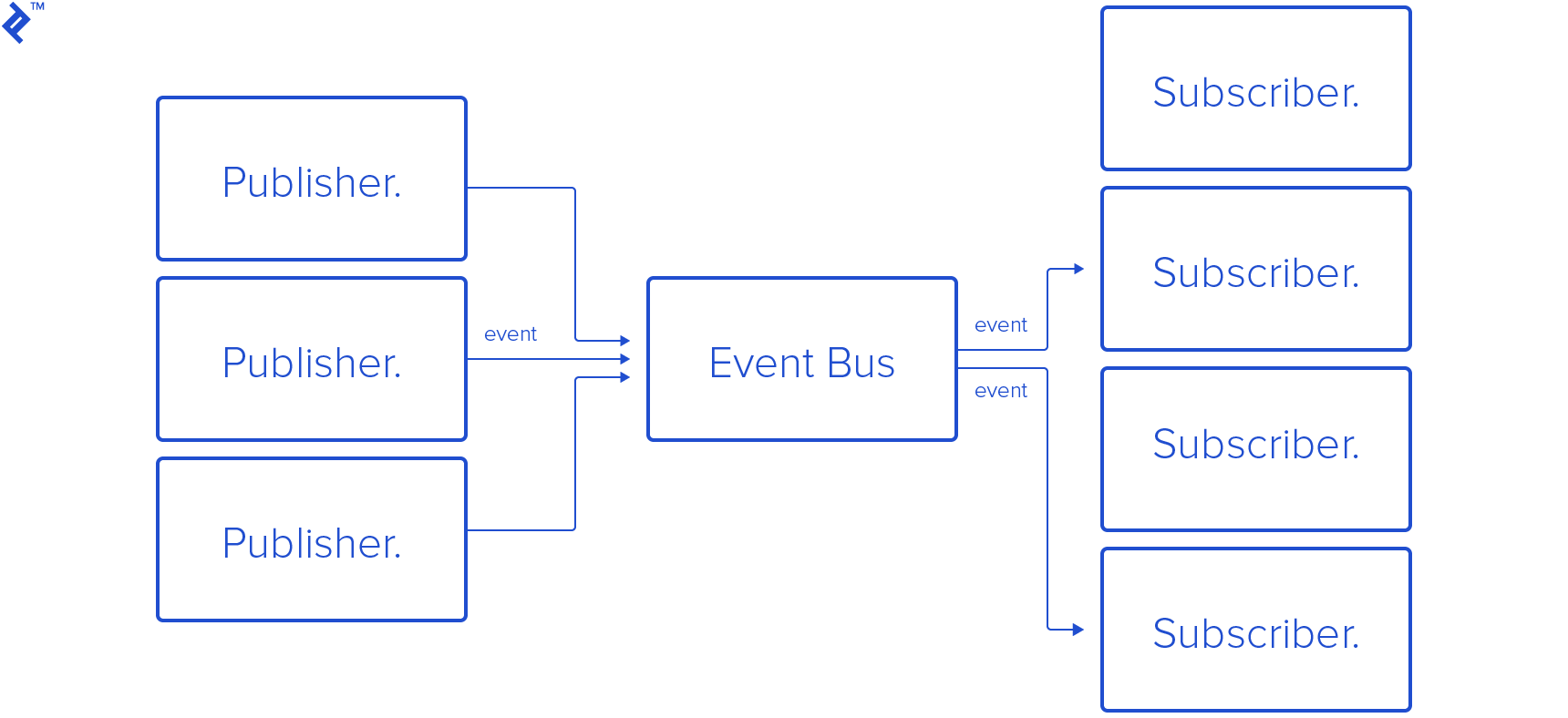


Рисунок 6 – Механизм Publish Subscribe

В программном комплексе механизм Publish – Subscribe реализован через библиотеку Celery с реализацией 3 очередей сообщений. Результат выполнения задач хранится в базе данных Redis.

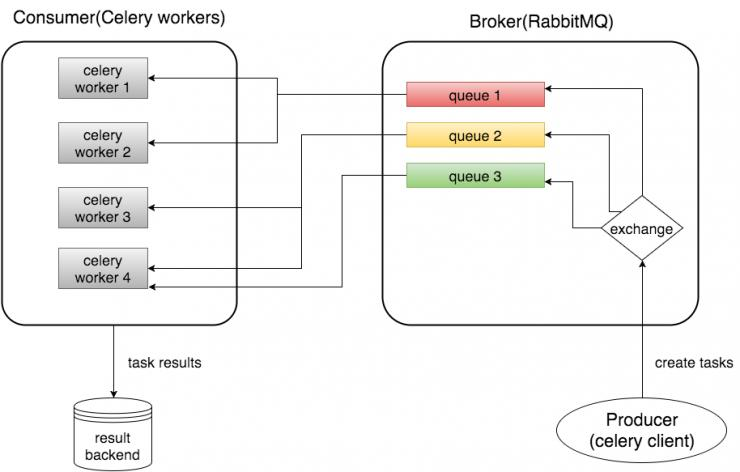


Рисунок 6 – Механизм Publish Subscribe в программном комплексе

# **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

## **Предполагаемая потребность**

Данный продукт будет интересен менеджерам продукта, продуктовым и бизнес аналитикам, маркетологам, руководителям и менеджерам проектов, UX-исследователям и QA-специалистам, а также, преподавателям и студентам, изучающим менеджмент и аналитику, и желающим проводить исследования в этой области.

## **Экономические преимущества разработки по сравнению с отечественными аналогами**

Программами аналогами, можно существующие системы аналитики такие как Google Analytics и Яндекс Метрика, однако они не поддерживают функции кластеризации пользователей и построения графов перемещения пользователей.

# **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. Разработка Микросервисных приложений на C# Документация [Электронный ресурс]. Режим доступа: <https://aka.ms/microservicesebook>, свободный. Дата обращения: многократно 1.01. -10.04.2020.
11. Retentioneering Документация [Электронный ресурс]. Режим доступа: <https://retentioneering.github.io/retentioneering-tools/_build/html/index.html>, свободный. Дата обращения: многократно 1.01. -10.04.2020.
12. Django Документация [Электронный ресурс]. Режим доступа: <https://docs.djangoproject.com/en/3.0/>, свободный. Дата обращения: многократно 1.01. -10.04.2020.
13. Celery Документация [Электронный ресурс]. Режим доступа: <http://docs.celeryproject.org/en/latest/index.html>, свободный. Дата обращения: многократно 1.01. -10.04.2020.

# **ПРИЛОЖЕНИЕ 1**

# **ТЕРМИНОЛОГИЯ**

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| **Система аналитики** | Система, которая позволяет собирать, хранить и обрабатывать события аналитики |
| **Событие аналитик** | Действие пользователя в приложении, характеризуемое названием, пользователем и временем |
| **Матрица переходов** | Матрица вероятности следующего события в зависимости от предыдущего |
| **Граф перемещений** | Визуализация матрицы перехода в виде графа |
| **Поведение пользователя** | Порядок действий осуществленным пользователем в программном продукте выраженная как последовательность событий аналитики |

# **ПРИЛОЖЕНИЕ 2**

# **ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

**Классы модуля dashboards**

*Таблица 1 - Классы модуля dashboards*

|  |  |
| --- | --- |
| **Класс** | **Назначение** |
| UploadForm | Форма загрузки файла |
| EventsetStatus | Enum статуса обработки |
| Eventset | Модель набора данных |
| Event | Модель события |
| EventsetView | View управления набором данных |
| ExperimentView | View управления обработанными данными |
| GraphView | View управления графов |

**Классы модуля users**

*Таблица 2 - Классы модуля users*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| User | Модель пользователя |
| Profile | Прокси модель пользователя |
| LoginForm | Форма логинизации |
| RegisterForm | Форма регистрации |
| MainView | View главного экрана |
| LoginView | View логинизации |
| RegisterView | View регистрации |
| LogoutView | View выхода пользователя |

# **ПРИЛОЖЕНИЕ 3**

# **ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ**

**Описание членов класса UploadForm**

*Таблица 3 – Описание полей класса UploadForm*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| name | Public | CharField | Название файла |
| file | Public | FileField | Файл |

**Описание членов класса Eventset**

*Таблица 4 – Описание методов класса Eventset*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| processing | Public | Bool | * - | Статус обработки |
| events\_per\_user | Public | Integer | - | Количество событий на пользователя |
| href | Public | String | - | Ссылка |
| get\_eventset\_list | Public | List | User | Список наборов данных по пользователю |

*Таблица 5 – Описание полей класса Eventset*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| user | Public | ForeignKey | Пользователь |
| name | Public | CharField | Имя |
| status | Public | CharField | Статус |
| step\_matrix\_id | Public | CharField | Id задачи построения матрицы переходов |
| tsne\_task\_id | Public | CharField | Id задачи расчета tsne |
| events | Public | IntegerField | Количество событий |
| users | Public | IntegerField | Количество пользователей |
| unique\_events | Public | IntegerField | Количество уникальных событий |

**Описание членов класса Event**

*Таблица 6 – Описание полей класса Event*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| event\_name | Public | CharField | Название события |
| event\_timestamp | Public | CharField | Timestamp события |
| user\_pseudo\_id | Public | CharField | ID пользователя |
| event\_set | Public | ForeignField | Eventset события |

**Описание членов класса ExperimentView**

*Таблица 7 – Описание методов класса ExperimentView*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| get | Public | HttpResponse | * - | View дашборда |

**Описание членов класса GraphView**

*Таблица 8 – Описание методов класса GraphView*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| get | Public | HttpResponse | * - | View графа |

**Описание членов класса EventsetsView**

*Таблица 9 – Описание методов класса Eventsset*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| get | Public | HttpResponse | * - | View набора событий |
| post | Public | HttpResponse | * - | Создание нового набора событий |

**Описание членов класса User**

*Таблица 10 – Описание полей класса User*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| Username | Public | CharField | Имя пользователя |
| Password | Public | CharField | Пароль |

**Описание членов класса Profile**

*Таблица 11 – Описание полей класса Profile*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| user | Public | ForeignKey | Внешний ключ пользователя |

**Описание членов класса MainView**

*Таблица 12 – Описание методов класса MainView*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| get | Public | HttpResponse | * - | View главной страницы |

**Описание членов класса LoginView**

*Таблица 13 – Описание методов класса LoginView*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| post | Public | HttpResponse | * - | View логинизации |

**Описание членов класса RegisterView**

*Таблица 14 – Описание методов класса RegisterView*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| post | Public | HttpResponse | * - | View регистрации |

**Описание членов класса LogoutView**

*Таблица 15 – Описание методов класса LogoutView*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| get | Public | HttpResponse | * - | View выхода из аккаунта |

**Описание членов класса LoginForm**

*Таблица 16 – Описание полей класса LoginForm*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| username | Public | CharField | Имя пользователя |
| password | Public | CharField | Пароль |

**Описание членов класса RegisterForm**

*Таблица 17 – Описание полей класса RegisterForm*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| username | Public | CharField | Имя пользователя |
| password | Public | CharField | Пароль |
| Password2 | Public | CharField | Пароль повторение |

# **ПРИЛОЖЕНИЕ 4**

# **СХЕМА БАЗЫ ДАННЫХ**

A screenshot of a computer

Description automatically generated

Рисунок 7 – Схема базы данных

# **ПРИЛОЖЕНИЕ 5**

# **ДИАГРАММА USE-CASE**

A close up of a device

Description automatically generated

Рисунок 8 – Диаграмма прецендентов использования

# **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 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |