

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К**  
**ИНДИВИДУАЛЬНОМУ ДОМАШНЕМУ ЗАДАНИЮ**

**по дисциплине «Введение в нереляционные базы данных»**

**Тема:** Анализ данных Google-календарей пользователя - сопоставление и статистика

Студентки гр. 5381

Преподаватель

Боровикова О.В.

Буздина М.А.

Кочнева О.Р.

Заславский М.М.

Санкт-Петербург

2018

## ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студентки: Боровикова О.В., Буздина М.А., Кочнева О.Р.

Группа 5381

Тема работы : Анализ данных Google-календарей пользователя - сопоставление  
и статистика

Исходные данные:

Файлы календарей ( < >.ics) экспортированные из Google-календаря

Содержание пояснительной записки:

«Содержание», «Введение», «Этапы разработки» «Заключение», «Список  
использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 15.09.2018

Дата сдачи реферата: 13.12.2018

Дата защиты реферата: 20.12.2018

Студентки

Преподаватель

Боровикова О.В.

Буздина М.А.

Кочнева О.Р.

Заславский М.М.

## **АННОТАЦИЯ**

В данной работе описываются этапы разработки web-приложения для анализа Google-календарей пользователя. Основное внимание приходится на взаимодействие приложения с нереляционной базой данных, в которой хранятся данные о календаре пользователя.

В результате разработано действующее приложение.

## **SUMMARY**

This project describes the steps involved in developing a web application for analyzing a user's Google calendars. The focus is on the interaction of the application with a non-relational database, which stores data about the user's calendar.

As a result, a valid application has been developed.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1. Качественные требования к решению .....	6
2. Сценарии использования .....	6
3. Модель данных .....	9
4. Разработанное приложение .....	14
ЗАКЛЮЧЕНИЕ .....	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	17
ПРИЛОЖЕНИЕ А .....	18
Инструкция для пользователя.....	18
ПРИЛОЖЕНИЕ В .....	20
Снимки экрана приложения .....	20

## ВВЕДЕНИЕ

В настоящее время планирование времени является одним из важных аспектов деловых людей, поэтому существует множество приложений, позволяющих так или иначе планировать свой день. Одним из таких приложений является Google-календарь, который позволяет быстро организовать встречу или мероприятие. Но порой возникают проблемы с выбором свободного дня для назначения встречи, приходится просматривать свой календарь вдоль и поперёк, чтобы отыскать такой день. Или же иногда интересно выяснить кто чаще из пользователей приглашает на мероприятия. Таким образом, вопрос об анализе данных календаря (получение некоторой статистики) является довольно актуальным.

Данный вопрос и является задачей, необходимо провести анализ данных Google-календарей пользователя (сопоставление и статистика), используя нереляционную базу данных (MongoDB) для хранения и обработки информации. В качестве решения, предлагается разработать Web-приложение, которое будет позволять анализировать календарь пользователя и выводить соответствующие результаты на экран в виде таблиц или диаграмм.

# 1. Качественные требования к решению

## 1.1. Текущие

Использование для хранения данных нереляционной БД (MongoDB)

Загрузка статических данных и их анализ

Экспорт загруженных данных в файл формата .json

Графическое представление полученных результатов

## 1.2. Перспективные

Синхронизация с Google-календарями (нестатическая загрузка данных)

Возможность выбора и удаления календаря из базы.

# 2. Сценарии использования

## 2.1. Макет UI

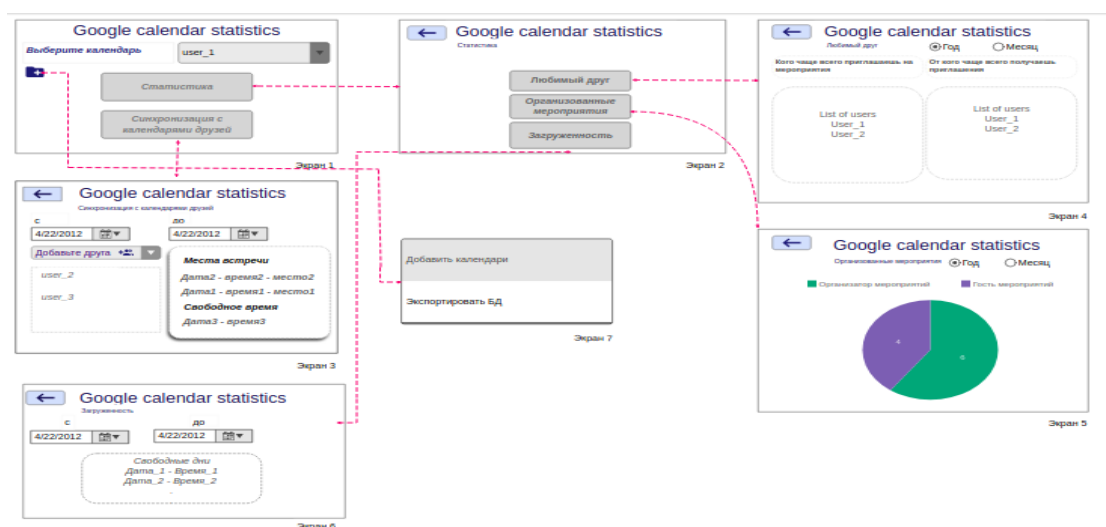


Рис. 1

## 2.2. Сценарии использования для задачи

### I. Импорта данных:

Для начала работы необходимо импортировать хотя бы свой календарь.

На экране 1 необходимо нажать иконку «обзор», в выпавшем окне выбрать файл для загрузки и нажать «ОК».

После чего требуется нажать кнопку «импорт в базу данных» (также стоит добавить календари друзей).

Таким образом вы успешно загрузили свой календарь в БД. Для дальнейшей работы необходимо выбрать нужный календарь из выпадающего списка и нажать кнопку «Выбрать пользователя».

### II. Анализ данных

Для анализа календаря пользователь может перейти от 1 экрана к экранам 2, 3.

При нажатии кнопки '**статистика**' переходим на экран 2.

- При нажатии кнопки '**Любимый друг**' переходим на экран 4.
  - Отражают статистические данные в виде двух списков.
  - Есть возможность увидеть данные по различным временным отрезкам (месяц, год).
- При нажатии кнопки '**Организованные мероприятия**' переходим на экран 5.
  - Отражают статистические данные в виде круговых диаграмм.
  - Есть возможность увидеть данные по различным временным отрезкам (месяц, год).

- При нажатии кнопки **'Загруженность'** переходим на экран 6.
  - Отражают статистические данные в виде списка с вариантами свободного времени.
  - Есть возможность увидеть данные в различные временные интервалы.

При нажатии кнопки **'синхронизация с календарем друга'** переходим на экран 3.

- Необходимо выбрать как минимум одного друга для сопоставления календарей (Календарь друга может быть импортирован и на этом экране) и временной период, нажать кнопку **«получить результат»**.
- Далее на экране будут отображаться данные в виде двух списков. Первый – когда и где вы имеете общие мероприятия (если таковые имеются), второй – «окна» в календаре, чтобы можно было назначить встречу.

### III. Экспорт данных

Для того чтобы экспортировать данные, которые находятся в БД, пользователю необходимо нажать **'Экспорт бд Users'** или **'Экспорт бд Events'** на экране 1.

## **2.3. Вывод**

В данном проекте будут преобладать операции «чтения», так как задача направлена на анализ данных, т.е. вывод полученной информации в разных видах.



### 3. Модель данных

#### 3.1. Нереляционная модель данных

##### I. Графическое представление.

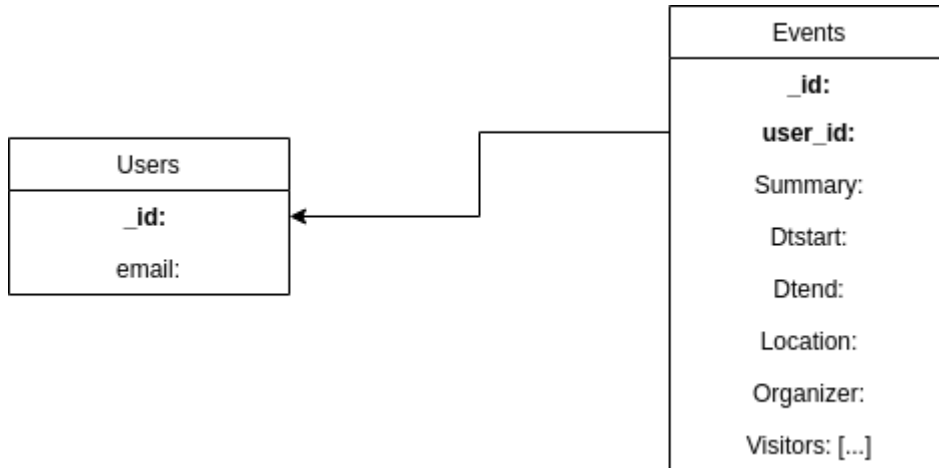


Рис. 2

##### II. Описание назначений коллекций, типов данных и сущностей.

Существует 2 Коллекции. Первая коллекция хранит данные о всех пользователях, загруженных в базу. Вторая коллекция хранит данные о всех мероприятиях всех пользователей. Связь коллекций осуществляется ссылочно по `_id` пользователя.

##### ■ 1 Коллекция users :

- `_id`: "user\_id" - идентификатор пользователя(автоматически)
- email: "something@gmail.com" – email

##### ■ 2 Коллекция events:

- `_id`: "event\_id" - идентификатор мероприятия(автоматически)

- user\_id: "user\_id" - идентификатор пользователя(владельца календаря) из коллекции 1
- summary: "Event's name" - название мероприятия
- dtstart: "DataFormat" - дата и время начала мероприятия
- dtend: "DataFormat" - дата и время окончания мероприятия
- location: "Location" - место проведения мероприятия
- organizer: "something@gmail.com" - email организатора мероприятия
- visitors:
  - email: "something@gmail.com" - email гостей мероприятия

*III.* Оценка удельного объема информации, хранимой в модели (сколько потребуется памяти, чтобы сохранить объекты, как объем зависит от количества объектов)

*Описание типов используемых данных и объем занимаемой памяти*

- 1 Коллекция users:
  - \_id: (Object ID) - 12 байт
  - email: (String) - 32 байта
- 2 Коллекция events:
  - \_id: (Object ID) - 12 байт
  - user\_id: (String) - 32 байт
  - summary: (String) - 32 байт
  - dtstart: (Date) - 8 байт
  - dtend: (Date) - 8 байт

- location: (String) - 32 байт
- organizer: (String) - 32 байт
- visitors: (Array of String)
- email: (String) - 32 байт

#### *Удельный объем используемых данных*

- Пусть совокупность полей ("\_id","email") занимает n памяти, m - количество пользователей; совокупность полей ("\_id","user\_id","summary","dtstart","dtend","location","organizer","visitors") занимает k памяти, p – количество мероприятий

- Тогда при условии того, что:

$$n = (32+12) \text{ байт}$$

$$\text{среднее количество пользователей } m = 10$$

$$k = (12+32+32+8+8+32+32+32*f) \text{ байт,}$$

$$\text{где } f = 5 - \text{среднее количество visitor}$$

$$\text{среднее количество мероприятий } p$$

- Тогда хранение данных будет занимать

$$S=S1+S2=m \times n + k \times p = 10 \times 44 + 316 \times 700 = 440 + 316 * p \text{ (байт)}$$

IV. Запросы к модели, с помощью которых реализуются сценарии использования.

Гистограмма организованные мероприятия (экран 5).

`db.users.find({email:"myemail@gmail.com"})` - находим \_id

`db.events.find({user_id: "_id", organizer: "myemail@gmail.com"}).count()` - подсчет организованных мероприятий

db.events.find({user\_id: "\_id").count() - подсчет всего мероприятий

### 3.2. Аналог модели данных для СУБД

#### 1. Графическое представление

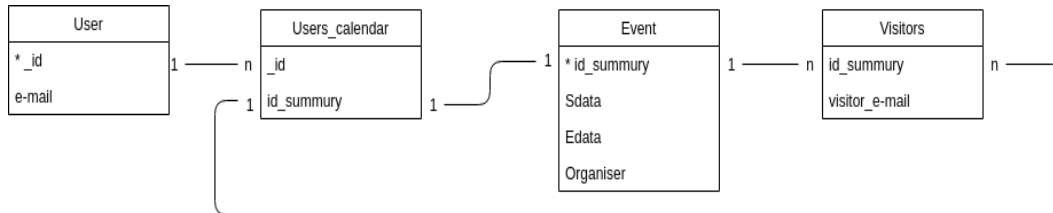


Рис 3.

#### 2. Описание назначений коллекций, типов данных и сущностей

Реляционная модель бд состоит из следующих таблиц:

- User:
  - `_id_calendar` - длинное целое (ключ)
  - `e-mail` - `varchar(25)`
- User's\_calendar (ключ)
  - `_id` - длинное целое
  - `_id_summury` - длинное целое Ключом является данная пара значений
- Event
  - `_id_summury` - длинное целое
  - `Place` - `varchar(100)`
  - `Sdata` - дата/время
  - `Edata` - дата/время
  - `Organiser` - `varchar(25)`
- Visitors

- `_id_summury` - длинное целое (ключ)
- `visitors_e-mail` - `varchar(25)`

3. Оценка удельного объема информации, хранимой в модели (сколько потребуется памяти, чтобы сохранить объекты, как объем зависит от количества объектов)

- Пусть  $m = 10$  - количество пользователей,  $p$  - количество мероприятий,  $f = 5$  - количество посетителей.
- `User` :  $(25+4) * m = 290$  байт
- `User's_calendar` :  $(4+4) * m = 80$  байт
- `Event` :  $(4+10+10+100+25) * m * p = 1\,490 * p$  байт
- `Visitors` :  $(25+4) * m * p * f = 1450 * p$  байт
- $S = 370 + 2940 * p$  байт

4. Запросы к модели, с помощью которых реализуются сценарии использования

Гистограмма организованные мероприятия (экран 5)

- Подсчет количества мероприятий, где пользователь был организатором

```
select count(table1.Organiser) from
(select user's_calendar.[_id_summury], Event.organiser as table1 from
user's_calendar, Event
where user's_calendar.[_id_summury]=Event.[_id_summury])
where table1.Organiser = user.e-mail
group by table1.organiser
```

- Подсчет количества приглашений на мероприятия

```
select count(visitors.visitors_e-mail) from visitors
```

```
where visitors.visitors_e-mail = user.e-mail  
group by visitors.visitors_e-mail
```

### 3.3. Сравнение моделей

Данные занимают больший объём при использовании реляционной СУБД.

- $SQL - S = 370 + 2940 * p$  байт
- $MongoBD - S = 440 + 316 * p$  байт

NoSQL требуется меньший объём, благодаря отсутствию дублирования информации.

Запросы на mongodb-языке являются более громоздкими, чем на языке SQL.

- 2 запроса необходимо для SQL: сложность –  $O(n)$
- 3 запроса необходимо для NoSQL : сложность –  $O(n+2*p)$ , где  $n$  – кол-во пользователей, а  $p$  – кол-во мероприятий

Количество запросов в mongodb требуется примерно столько же, сколько и у SQL, но в mongodb они еще и более сложные. Так как данная программа требует хранения большого объема данных, то можно сделать вывод, что NoSQL - модель предпочтительнее SQL.

## 4. Разработанное приложение

Данное web-приложение направленно на анализ данных Google-календаря. В нём содержится 6 экранов, 4 из которых представляют результаты анализа (статистику и сопоставление). Экран 1 и 2 являются отправными. В приложении реализованы сервер и клиент. Осуществлено взаимодействие с базой данных MongoDB. Все запросы производятся со стороны сервера.

Пользователь может узнать:

- Сколько мероприятий он создал и в какие был приглашён за определённое время;
- С каким другим пользователем он чаще пересекается
- В какое время он сможет встретиться с выбранным пользователем
- Своё свободно время в определённый период времени

Пользователь может как импортировать данные, так и экспортировать их.

Схема экранов приложения:

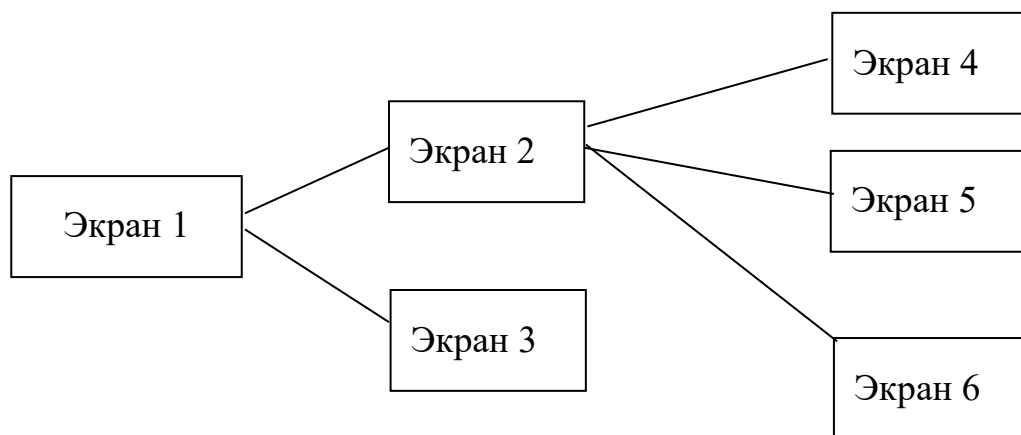


Рис 4. Схема 1.

Используемые технологии:

python 3.6, pip3, virtualenv, html

mongodb

фреймворк Flask

Ссылка на приложение:

[https://github.com/moevm/nosql2018-google\\_calendar](https://github.com/moevm/nosql2018-google_calendar)

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения работы было создано клиент-серверное приложение для анализа Google-календарей пользователя, которое позволяет ему лучше планировать своё свободное время, получать интересную статистику, получать информацию о возможных встречах с выбранными людьми.

При создании приложения были изучены особенности работы нереляционных баз данных.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. URL: <https://docs.mongodb.com/> (Дата обращения: 9.10.2018)
2. URL: <http://flask.pocoo.org/docs/1.0/> (Дата обращения: 10.10.2018)

# ПРИЛОЖЕНИЕ А

## Инструкция для пользователя

1. Для начала работы необходимо импортировать хотя бы свой календарь.
  - На экране 1 необходимо нажать иконку «обзор», в выпавшем окне выбрать файл для загрузки и нажать «ОК».
  - После чего требуется нажать кнопку «импорт в базу данных» (также стоит добавить календари друзей).
  - Таким образом вы успешно загрузили свой календарь в БД. Для дальнейшей работы необходимо выбрать нужный календарь из выпадающего списка и нажать кнопку «Выбрать пользователя».
2. Далее пользователь может перейти от 1 экрана к экранам 2, 3.

*При нажатии кнопки 'статистика' переходим на экран 2.*

- При нажатии кнопки 'Любимый друг' переходим на экран 4
  - Отражают статистические данные в виде двух списков
  - Есть возможность увидеть данные по различным временным отрезкам (месяц, год).
- При нажатии кнопки 'Организованные мероприятия' переходим на экран 5
  - Отражают статистические данные в виде круговых диаграмм
  - Есть возможность увидеть данные по различным временным отрезкам (месяц, год).
- При нажатии кнопки 'Загруженность' переходим на экран 6
  - Отражают статистические данные в виде списка с вариантами свободного времени

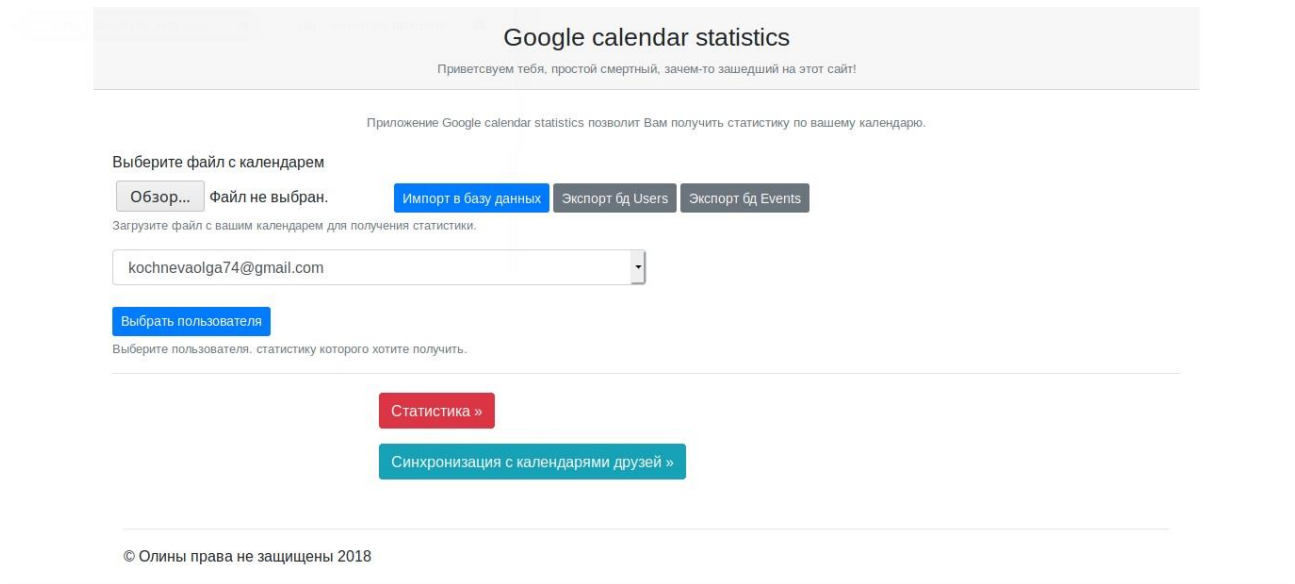
- Есть возможность увидеть данные в различные временные интервалы.

*При нажатии кнопки 'синхронизация с календарем друга' переходим на экран 3*

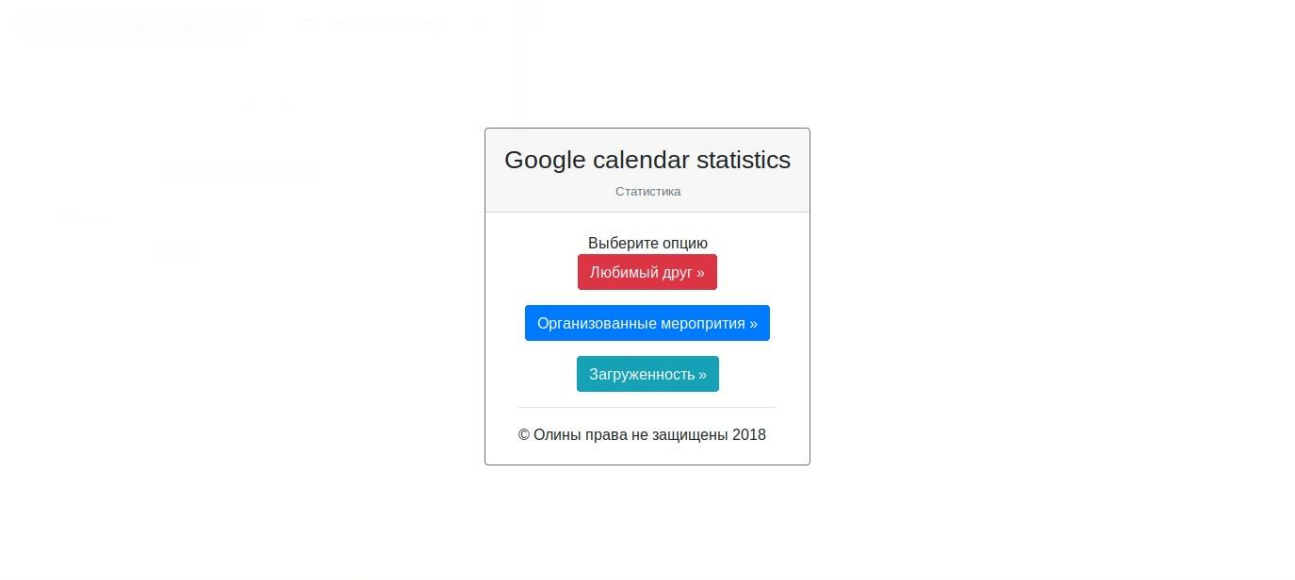
- Необходимо выбрать как минимум одного друга для сопоставления календарей (Календарь друга может быть импортирован и на этом экране) и временной период, нажать кнопку «получить результат»
  - Далее на экране будут отображаться данные в виде двух списков. Первый - когда и где вы имеете общие мероприятия (если таковые имеются), второй - "окна" в календаре, чтобы можно было назначить встречу.
3. Для того чтобы экспортировать данные, которые находятся в БД, пользователю необходимо нажать 'Экспорт бд Users' или 'Экспорт бдEvents' на экране 1.

# ПРИЛОЖЕНИЕ В

## Снимки экрана приложения



Экран 1



Экран 2



Google calendar statistics

Любимый друг

Временной промежуток

Год

Месяц

Получить статистику

Кто чаще всего приглашает на мероприятия

Кого чаще всего приглашаешь на мероприятия

mariyabuuu@gmail.com: 71

olchick0923@gmail.com: 19

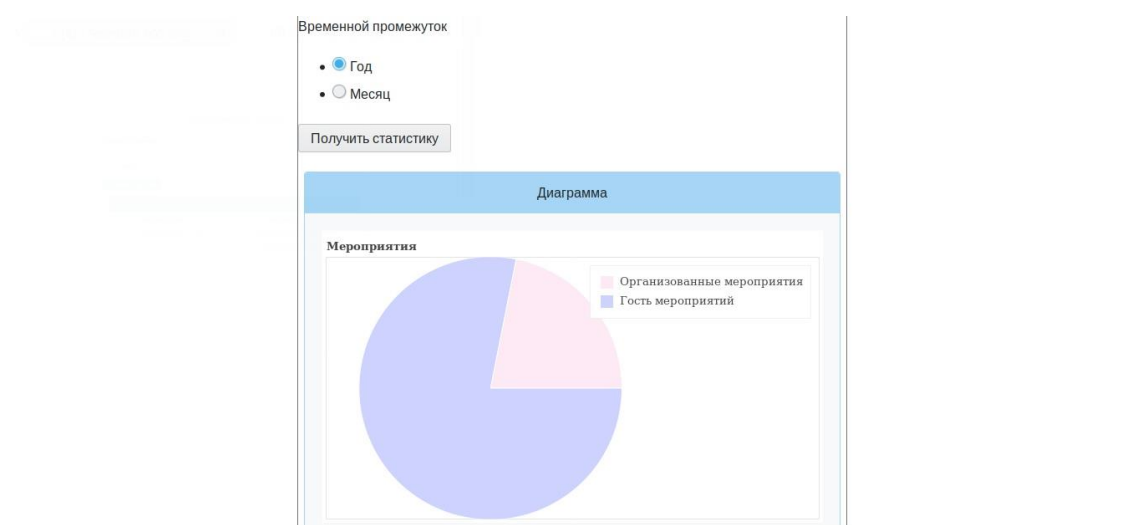
mariyabuuu@gmail.com: 1

olchick.borovikova@yandex.ru: 1

mariyabuzdina@yandex.ru: 1

© Олины права не защищены 2018

Экран 4



Экран 5

Google calendar statistics

Загруженность

С 26.11.2018 По 01.12.2018

Получить результат

Свободное время в выбранный промежуток времени

- 26-11-2018 00:00 - 26-11-2018 06:00
- 26-11-2018 13:00 - 27-11-2018 02:00
- 27-11-2018 03:30 - 27-11-2018 11:00
- 27-11-2018 19:30 - 28-11-2018 07:30
- 28-11-2018 10:15 - 29-11-2018 14:45
- 29-11-2018 21:30 - 30-11-2018 08:30
- 30-11-2018 09:30 - 01-12-2018 00:00

© Олины права не защищены 2018

Экран 6