Государственное бюджетное образовательное учреждение   
высшего образования Московской области  
Университет «Дубна»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**БАКАЛАВРСКАЯ РАБОТА**

**Тема:** Разработка программного модуля для экспорта данных из внешних источников в CRM-систему SugarCRM

**Ф.И.О. студента** Мартинович Александр Сергеевич

**Группа** 4252 **Направление подготовки** 09.03.04 Программная инженерия

**Направленность (профиль) образовательной программы** Разработка программно-информационных систем

**Выпускающая кафедра** распределенных информационных вычислительных систем

**Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**ст. преп. Смирнова Я.В.**/**

**Консультант (ы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**ст. преп. Михеев М.А.**/**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **/**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **/**

**Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**Асс. Юсупов И.И. **/**

**Выпускная квалификационная работа**

**допущена к защите «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.**

(дата)

**Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_** Кореньков В. В. \_ **/**

***г. Дубна, 2021***

***Государственное бюджетное образовательное учреждение   
высшего образования Московской области  
Университет «Дубна»***

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**УТВЕРЖДАЮ**

**Заведующий кафедрой**

**\_\_\_\_\_\_\_\_\_ /**Кореньков В. В. **/**

(Подпись) (Ф И О )

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.**

**З а д а н и е**

**на выпускную квалификационную работу – бакалаврскую работу**

**Тема** Разработка программного модуля для экспорта данных из внешних источников в CRM-систему SugarCRM

**Утверждена приказом № \_\_\_\_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ФИО студента** Мартинович Александр Сергеевич

**Группа** \_4252 **Направление подготовки** 09.03.04 Программная инженерия

**Направленность (профиль) образовательной программы** Разработка программно-информационных систем

**Выпускающая кафедра** распределенных информационных вычислительных систем

Дата выдачи задания \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата завершения бакалаврской работы\_\_\_\_\_\_\_\_\_\_\_\_

**Исходные данные к работе**

База данных студентов, их академические успехи, научные достижения и прочие успехи

**Результаты работы:**

1. Содержание пояснительной записки (перечень рассматриваемых вопросов)

Сбор данных из внешних источников.  
Форматирование данных для импорта.  
Импорт данных в CRM систему.

1. Перечень демонстрационных листов

Презентация PowerPoint

|  |  |
| --- | --- |
| **Консультант(ы)** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**ст. преп. Михеев М.А. **/**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **/**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **/** |

**Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /** ст. преп. Смирнова Я.В.**/**

**Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(дата)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(подпись студента)

Я, \_ Мартинович Александр Сергеевич\_, ознакомлен(а) с требованием об обязательности проверки выпускной квалификационной работы на объем заимствования. Все прямые заимствования из печатных и электронных источников, а также из защищенных ранее выпускных квалификационных работ, научных докладов об основных результатах подготовленной научно-квалификационной работы (диссертации), кандидатских и докторских диссертаций, должны иметь в работе соответствующие ссылки.

Я ознакомлен(а) с Порядком проверки на объем заимствования и размещения в электронно-библиотечной системе текстов выпускных квалификационных работ и научных докладов обучающихся, согласно которому обнаружение в выпускной квалификационной работе заимствований, в том числе содержательных, неправомочных заимствований является основанием для недопуска к защите выпускной квалификационной работы и отчисления из образовательной организации.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

подпись Фамилия И.О.

Аннотация

В данной бакалаврской работе решается задача упрощения работы с *CRM* системой для сотрудников “Абитуриент-центра”. В данный момент обработка и занесение данных об абитуриентах и их достижениях ведётся вручную. Результаты онлайн-опросов, которые проходят абитуриенты, собираются и заносятся в систему самими сотрудниками центра. Для упрощения данной задачи требовалось создать программный модуль, который автоматически следил бы за появлением новых ответов на опросы и заносил данные из этих ответов в *CRM* систему “Абитуриент-центра”.

Работа состоит из анализа предметной области, сбора требований, проектирования программного модуля, его разработки и внедрения.

Работа проводилась под научным руководством старшего преподавателя Я. В. Смирновой в 2021 году в институте Системного Анализа и Управления Государственного Университета «Дубна».

*Annotation*

*The aim of this bachelor’s thesis is to simplify working process for the staff of the Applicant Center. At this moment processing and recording applicant’s information requires manual labor. Results of online poll applicants take are being collected and recorded by the staff. To simplify this process, it was needed to develop module that would automatically detect new completed inquires and record data from them to «Applicant Center’s» CRM system.*

*This work consists of analysis of the problematics, collecting requirements, module designing, it’s development and implementation.*

*The work was carried out under the scientific supervision of senior teacher Y. V. Smirnova in 2021 at the Institute for Systems Analysis and Management at the State University «Dubna».*

**Оглавление**

[Введение 5](#_Toc74230445)

[Постановка задачи 6](#_Toc74230446)

[1 Глава Анализ предметной области 7](#_Toc74230447)

[1.1 Анализ информационной системы “Абитуриент-центра” 7](#_Toc74230448)

[1.2 Анализ внешних источников информации 9](#_Toc74230449)

[2 Глава Проектирование программного модуля 12](#_Toc74230450)

[2.1 Требования 12](#_Toc74230451)

[2.2 Выбор средств реализации 12](#_Toc74230452)

[2.3 Подробное описание функциональных возможностей приложения 12](#_Toc74230453)

[2.4 Инструкция для оформления и настройки опросов 19](#_Toc74230454)

[3 Глава Реализация 24](#_Toc74230455)

[Заключение 27](#_Toc74230456)

[Список литературы 28](#_Toc74230457)

[Интернет-ресурсы 28](#_Toc74230458)

# Введение

В 2018 году у “Абитуриент-центра” Государственного Университета «Дубна» появилась *CRM* система *SuiteCRM*, цель которой была помочь оптимизировать взаимодействие с абитуриентами. Система работала, но в своем стандартном состоянии она была настроена на работу в сфере бизнеса. К тому же в ней не хватало нужных “Абитуриент-центру” инструментов автоматизации.

В 2020 году центр предложил студентам-выпускникам в качестве дипломных проектов настроить систему для работы над конкретными задачами в предметной области “Абитуриент-центра”, а также наладить взаимодействие с другими внешними системами, используемыми для работы сотрудниками центра. Одним из студентов, взявшихся за эту работу, был я, и мне досталась вторая задача. В рамках этой работы мне нужно будет оптимизировать процесс работы сотрудников “Абитуриент-центра” с *CRM* системой, обеспечив автоматическое взаимодействие системы с сервисом *Google* *Forms*, предназначенным для проведения опросов, викторин, тестов.

# Постановка задачи

**Цель:** оптимизация процесса работы сотрудников “Абитуриент-центра” с *CRM* системой.

**Исходные данные:** сайт *CRM* системы, база данных абитуриентов, их академические успехи, научные достижения.

**Ожидаемый результат:** функционирующий модуль, который автоматически заносит данные об абитуриентах.

**Критерий оценки результата:**

* сведение временных затрат сотрудников «Абитуриент-центра» по импорту данных в систему к нулю;
* снижение времени обработки ответа на анкету и его импорта в базу сокращено до 10 секунд;
* 100% исключение ошибки;
* 100% исключение потери данных в силу человеческого фактора при импорте.

# Глава Анализ предметной области

## Анализ информационной системы “Абитуриент-центра”

Система по работе с базой абитуриентов “Абитуриент-центра” основана на *CRM*-платформе *SuiteCRM* - системе управления взаимоотношениями с клиентами (*CRM* *System*), основанной на открытом исходном коде.

Само определение *CRM* расшифровывается как *Customer* *Relationship* *Management*. Такие системы с помощью автоматизации процессов помогают эффективнее выстраивать диалог с клиентом, не допуская ошибок в работе [2]. По своей сути – это *Excel* таблицы, но с большим количеством удобного функционала. Например, при щелчке на имя клиента открывается удобная карточка, в которой содержится вся информация о клиенте и история взаимодействия с этим ним – от первого звонка, знакомства до прекращения отношений с ним. Вот небольшой список того, что может *CRM* система (см. рис.1).

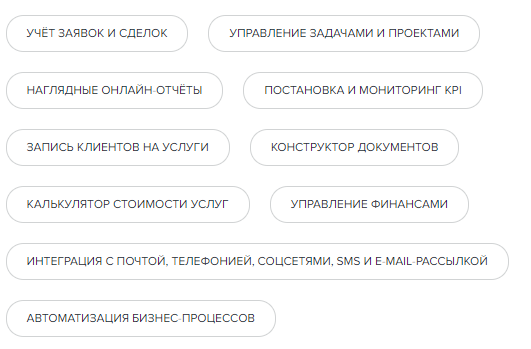


Рис. 1. Список возможностей CRM системы

Автоматизация в *CRM* системах позволяет выполнять некоторую рутинную работу быстро и чётко, сводя ошибки «человеческого фактора» к минимуму. Система может формировать документы по заданному шаблону, ставить задачи менеджерам на различных этапах взаимодействия с клиентом, рассылать *sms* и *email*, создавать отчёты по всем показателям, рассчитывать стоимость услуг через встроенный калькулятор и отслеживать важные даты.

Благодаря таким системам менеджеры могут работать точнее и эффективнее, а значит лучше и прибыльнее для компании.

В основном *CRM* системы используются для управления бизнесом, но, благодаря своему широкому функционалу, они могут также использоваться и в других сферах. “Абитуриент-центр” использует подобную систему для удобного учёта абитуриентов, школ, учителей, мероприятий, проводимых для них, и анкет, а также для взаимодействия с ними.

Для работы центром была выбрана система *SuiteCRM* – бесплатное ответвление от популярной *SugarCRM*. Одноимённая компания-разработчик *SugarCRM* с самых первых версий и вплоть до 6.5 выпускала несколько версий системы: бесплатную (*Community* *edition*) и коммерческие (*Professional*, *Enterprise*, *Ultimate*). В коммерческих было закономерно больше возможностей, чем в бесплатной. И начиная с 6.5 версии компания прекратила развитие версии Community, иногда лишь исправляя найденные ошибки. Другая же компания, *SalesAgility*, активно разрабатывала модули для бесплатной версии, особенностью которых была схожесть на модули из платных версий продукта. В 2013 году компания выпустила первую версию *SuiteCRM*, которая была основана на последней версии *SugarCRM* *Community* *Edition* и дополнена собственными модулями, что делало её очень близкой по функциональности с *Professional* версией *SugarCRM*. Когда в 2014 году *SugarCRM* официально объявила о прекращении развития своей бесплатной версии, *SalesAgility* заняла вакантное место, объявив, что их продукт всегда будет бесплатный и с открытым исходным кодом. В этой системе были следующие дополнительные модули:

Дополнительные модули *SuiteCRM* от *SalesAgility*:

* *Advanced* *OpenTheme* (Тема оформления)
* Advanced OpenSales (Продажи)
* Advanced OpenWorkflow (Процессы)
* Advanced OpenReports (Отчёты)
* Advanced OpenEvents (События)
* *Advanced* *OpenPortal* (Интеграция обращений с внешним порталом)
* *Advanced* *OpenDiscovery* (Полнотекстовый поиск)
* *Reschedule* (Отложенные звонки)

Дополнительные модули *SuiteCRM* от сторонних разработчиков:

* *SecuritySuite* (Группы пользователей)
* Google Maps (Карты Google)

Сама система *SuiteCRM* состоит из модулей, каждый из которых представляет собой какую-либо функциональную сторону *CRM*: Абитуриенты, Контакты, Школы и т.д (см. рис.2).

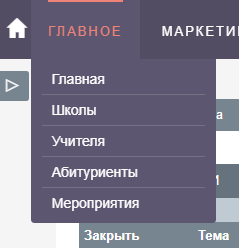


Рис. 2. Модули CRM системы

Эти модули управляют взаимоотношениями с клиентами. Многие из модулей связаны между собой. Например, просматривая информацию об абитуриентах, система также покажет связанную с ними информацию о школах и мероприятиях. Информацию можно редактировать, создавать и удалять [3]. Пользователи с правами администратора могут контролировать доступ других пользователей к модулям, а также создавать новые модули или редактировать имеющиеся.

## Анализ внешних источников информации

В рамках этой работы требуется экспортировать данные из внешних источников в модуль системы, связанный с абитуриентами. Информация будет поступать из анкет, которые составляются сотрудниками “Абитуриент-центра” в преддверии различных мероприятий для школьников и абитуриентов. В этих анкетах запрашиваются такие следующие данные:

* ФИО;
* адрес электронной почты;
* номер мобильного телефона;
* название школы;
* номер класса;
* адрес проживания.

Также в систему будут заноситься данные о направлении мероприятия, его название и достижения конкретного абитуриента на этом мероприятии.

Анкеты составляются в информационной системе *Google* *Forms*. Во время создания очередного опроса, в системе можно настраивать оповещения о новых ответах, указывать кому они будут отправляться и при каких условиях (см. рис.3).

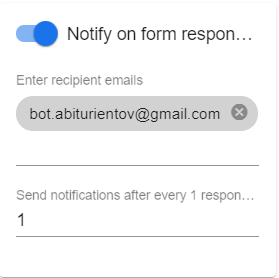


Рис. 3. Настройка оповещений

Ответы на анкеты автоматически сохраняются в выделенной для этого таблице *Excel*, которая лежит в облачном хранилище автора анкеты. Чтобы получать доступ другим пользователям, например ботам, нужно настроить таблицу, добавив туда адрес электронной почты этих лиц [1] (см. рис.4). В случае с ботами и автоматическими скриптами нужно создать для них сервисные учётные данные в *Google* *Developers* *Console*.

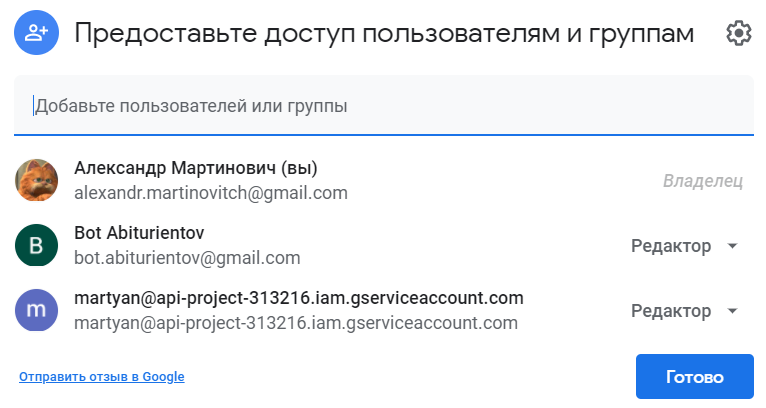


Рис. 4. Предоставление доступа к таблице ответов

После всех этих процедур можно будет с помощью программного кода обращаться к данным таблицы, которая придёт вместе с оповещением, предварительно аутентифицировавшись в системе. Для доступа к таблице, следует указать путь до неё в виде гиперссылки на облачное хранилище и саму таблицу.

# Глава Проектирование программного модуля

## Требования

**Требование** – условие или возможность, которой должна соответствовать система.

Данная бакалаврская работа связана с другой бакалаврской работой моей одногруппницы Рассохацкой Натальи. Ей требовалось спроектировать информационное представление данных *CRM* системы “Абитуриент-центра”. И поскольку моя работа предполагает взаимодействие с данными в этой системе, требования формировались на основе результатов работы Натальи.

Функциональные требования:

* Считывание данных из электронных таблиц.
* Форматирование данных согласно их представлению в *CRM* системе.
* Занесение данных в *CRM* систему.

Системные требования

* Операционная система *Windows* *XP* и выше.

## Выбор средств реализации

Программный модуль будет написан с помощью языка программирования Python в среде разработки от *JetBrains* – *PyCharm*. Для связи с системой *SuiteCRM* будет использоваться библиотека *PySugarCRM.* Для получения доступа к таблицам – *gspread* и *oauth2client*, а для получения доступа к оповещениям от *Google Forms* будут использоваться библиотеки *imaplib* и *email*.

## Подробное описание функциональных возможностей приложения

Приложение разработано на языке *python*, потому что нам необходимо работать с данными, для чего лучше всего подходит именно этот язык. Он очень популярен в науке и в коммерческой сфере. Этому способствует простота языка и множество разнообразных библиотек с готовыми инструментами, которые лежат в открытом доступе (см. рис.5).

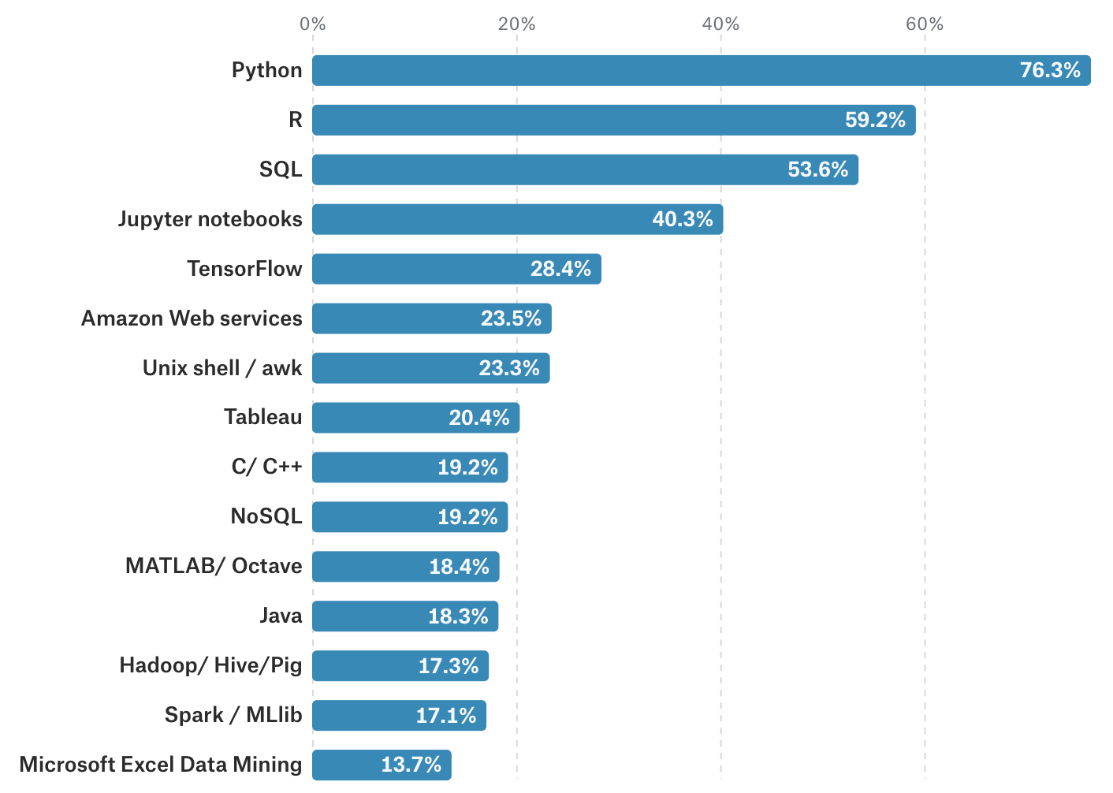


Рис. 5. Диаграмма сравнения популярности языков программирования для работы с данными

Приложение представляет из себя автоматический скрипт, хранящийся и работающий на сервере университета, который раз в определённый период (1 час) срабатывает, чтобы собрать новые данные из ответов на анкеты и поместить их в *CRM* систему “Абитуриент-центра”.

Последовательность функций:

1. Авторизация на сервере *imap.google.com.*
2. Поиск непрочитанных писем от *Google Forms.*
3. По списку писем проводится поиск ссылок на вложенные таблицы с ответами на опросы.
4. Аутентификация на сервере *docs.google.com.*
5. Скачивание содержимого таблиц.
6. Форматирование данных таблиц.
7. Отправка данных на сервер *CRM* системы.

Её также удобно описывать с помощью диаграммы последовательностей (см. рис.6).

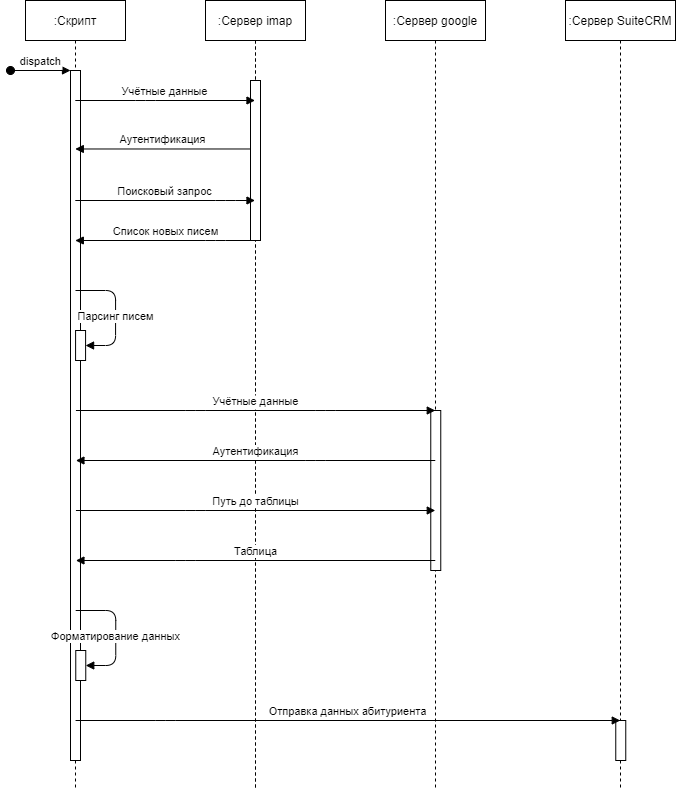


Рис. 6. Схема последовательности действий приложения

Для описания работы приложения используются диаграммы *IDEF0.* С точки зрения *IDEF*0, система — совокупность взаимодействующих работ или функций, преобразующих поступающий поток информации в выходной.

При каждом завершении абитуриентами анкеты, система анкетирования *Google Forms,* используя настроенное встраиваемое расширение, отправляет письмо-оповещение на почту бота. В этом письме, помимо оповещения о завершённой анкете, содержится информация об общем количестве ответов на анкету, а также вложение в виде ссылки на *Excel* таблицу, лежащую в облачном хранилище автора анкеты, которая содержит все собранные ответы. Все данные для авторизации на сервере почты, облачном хранилище и в *CRM* системе “Абитуриент-центра” прописаны в коде программы и не изменяются (см. рис.7). Приложению в том числе требуется интернет-соединения для авторизации в разных системах и для экспорта/импорта данных из/в них.

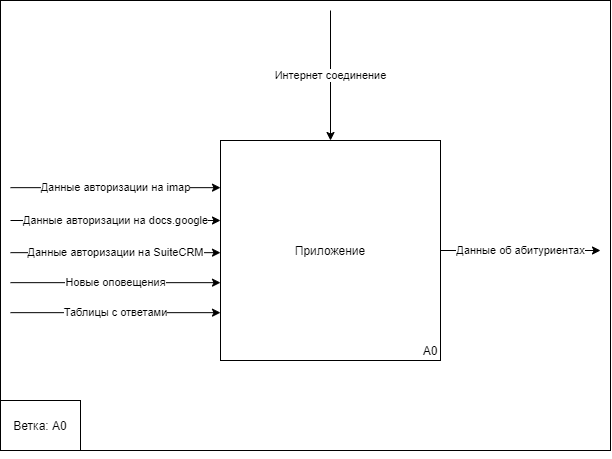


Рис. 7. Диаграмма IDEF0

Для отправки данных об абитуриентах, которые они занесли в анкеты, в CRM систему “Абитуриент-центра”, их сначала нужно достать из таблицы, доступ к которой можно получить из письма с оповещением, а также отформатировать (см. рис.8).

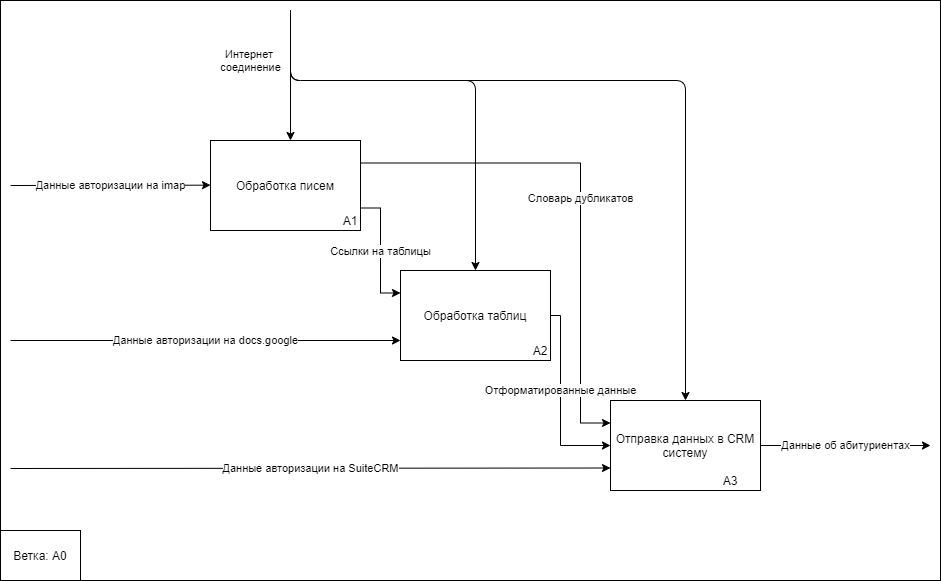


Рис. 8. Диаграмма декомпозиции процесса А0

Для начала работы с письмами необходимо аутентифицироваться на сервере *imap.gmail.com*, используя соответсвенные учётные данные. После этого у нас открывается доступ к почтовому ящику, где мы можем найти новые письма-оповещения от *Google Forms,* а также посчитать сколько оповещений приходило от разных опросов. Последние данные нам понадобятся впоследствии для точного разграничения новых ответов и старых. В отсортированных письмах нам нужно найти ссылки на таблицы, в которых хранятся ответы на анкеты. В результате выполнения процесса А1 мы получаем словарь дубликатов оповещений и ссылки на соответствующие таблицы (см. рис.9).

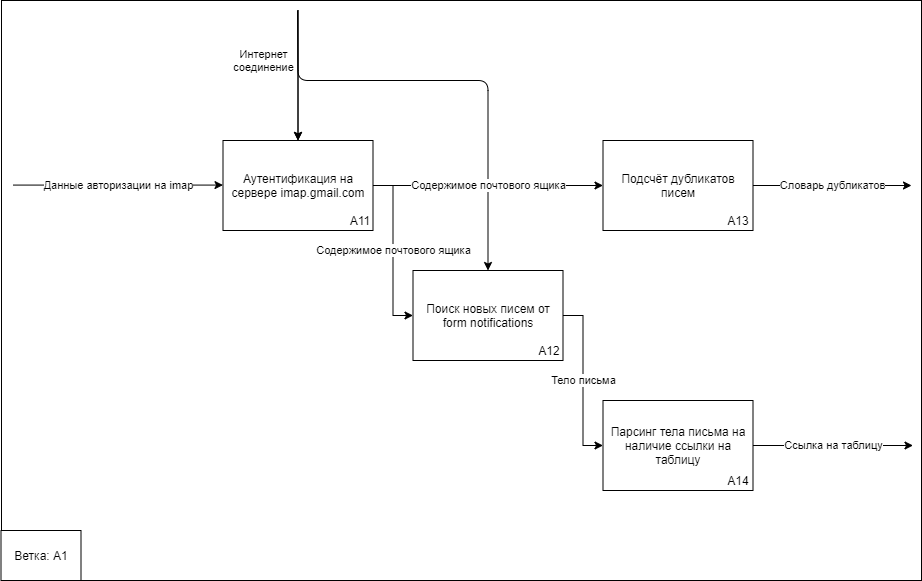


Рис. 9. Диаграмма декомпозиции процесса А1

Получив адреса, по которым лежат таблицы, мы аутентифицируемся в облачном хранилище, используя учётные данные сервисного аккаунта, который был настроен в сервисе *Google Developers* [5]. Теперь мы можем сохранить данные из этих таблиц и отформатировать их таким образом, чтобы к ним было удобно обращаться при занесении их в *CRM* систему (см. рис.10).

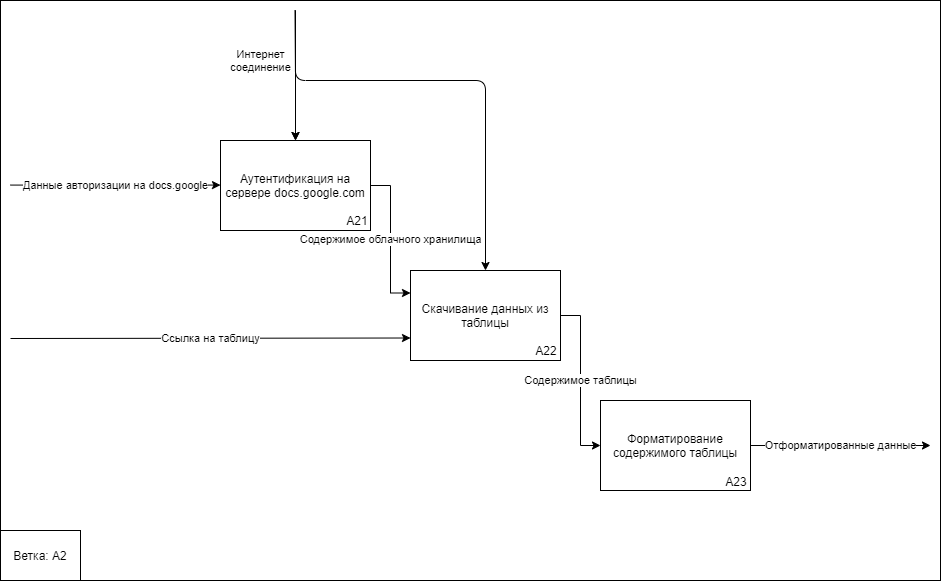


Рис. 10. Диаграмма декомпозиции процесса А2

Теперь, имея отформатированнные данные таблиц, а также словарь дубликатов оповещений об ответах на различные опросы, мы можем, предварительно аутентифицировавшись с учётными данными пользователя-бота CRM системы, сформировать данные и отправить их (см. рис.11).

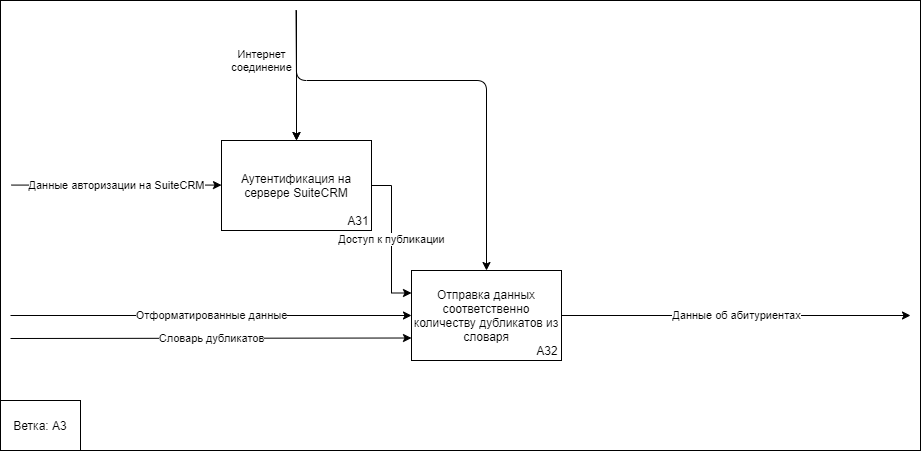


Рис. 11. Диаграмма декомпозиции процесса А3

Данные формируются согласно формату модуля абитуриентов. В данном модуле содержится большое количество полей (см. рис.12). Но нам не нужно заполнять их все, большинство из них содержит данные, которые нам не могут быть известны из ответов на анкеты, и могут изменяться в процессе работы с конкретным абитуриентом, например, поле *do\_not\_call* заполняется уже после связи сотрудника “Абитуриент-центра” с абитуриентом. Или поле *progress*, в которое заносятся достижения абитуриента на конкретном мероприятии. На момент получения ответа на анкету, он только зарегистрировался, и никаких достижений заработать ещё не мог.

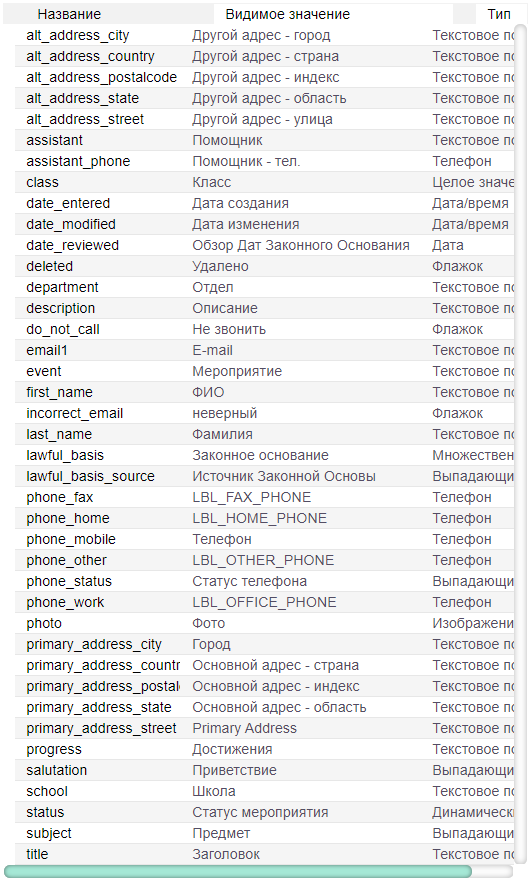


Рис. 12. Список полей модуля абитуриентов

А вот поля, которые мы можем заполнить на основании ответов на анкеты (см. рис.13).

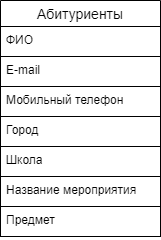


Рис. 13. Список доступных для заполнения полей

Так как мы заносим данные автоматически, используя программу, нужно соблюсти ряд условий для корректной обработки данных этой программой. Условия будут представлены в виде инструкции по составлению и настройке опросов в системе *Google Forms*.

## Инструкция для оформления и настройки опросов

Для того, чтобы ответы абитуриентов корректно обработались, необходимо составить опрос, имеющий чёткую структуру и точные названия разделов. На рисунке представлена минимальная конфигурация для опроса (см. рис.14). Форму можно дополнять, добавляя разделы после стандартных, если специфика мероприятия, для которого готовится опрос, предполагает уточнение этой информации, а также настраивая ограничения ответов на вопросы.

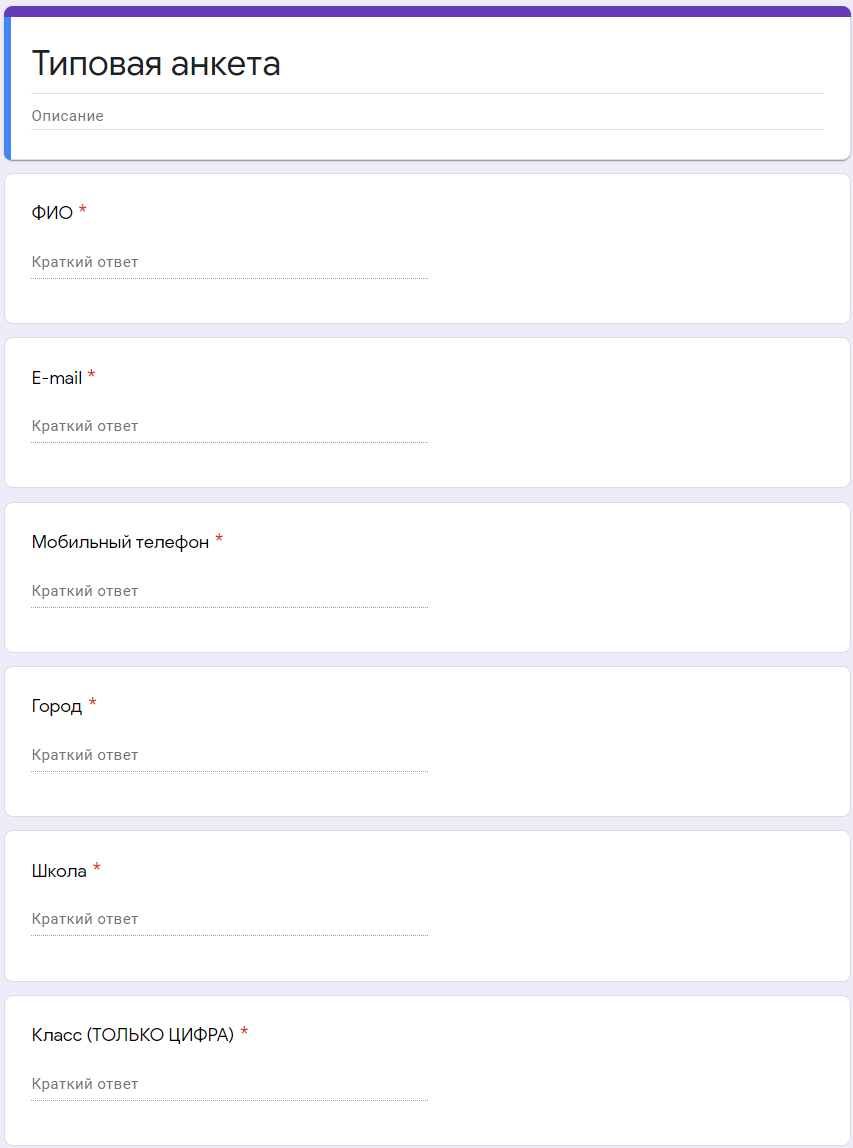


Рис. 14. Базовый набор вопросов

После составления опроса необходимо произвести его настройку. Сначала нужно перейти на вкладку «Ответы» (см. рис.15).

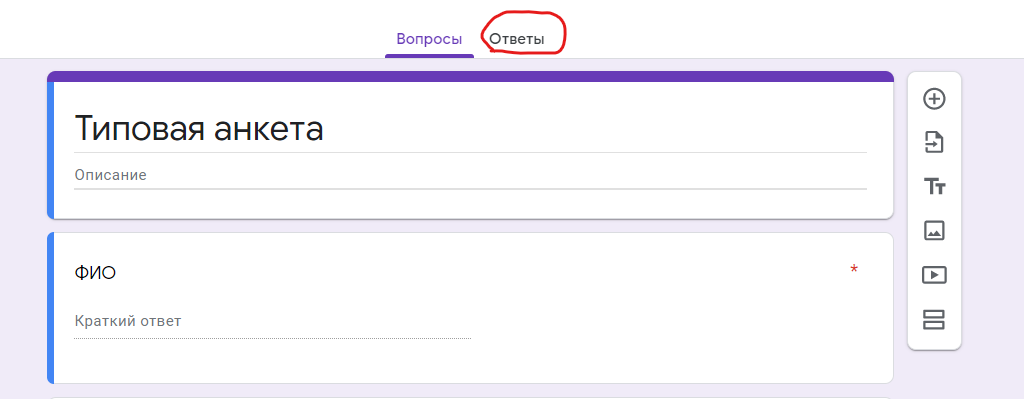


Рис. 15. Переход на вкладку «Ответы»

Затем в настройках нужно выбрать опцию «Сохранять ответы…» (см. рис.16).

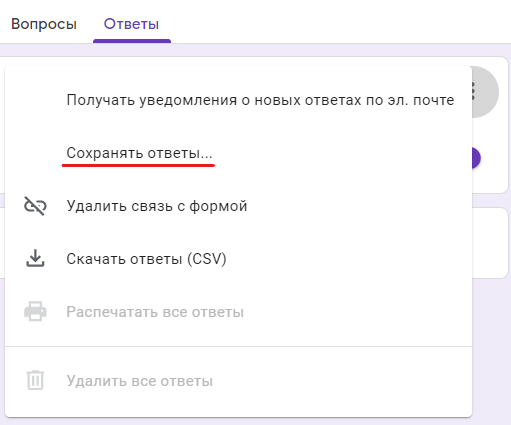


Рис. 16. Переход к настройке сохранения ответов

Во всплывшем окне выбрать опцию «Новая таблица» и назвать эту таблицу строго в формате «\*Название мероприятия\* \*Название предмета\*» (см. рис.17). Если предметов больше одного, необходимо объединить их названия символом «-» или «\_». Важно, чтобы в названии было только два слова, разделённых одним пробелом. Из названия таблицы берутся данные о мероприятиях и предметах, которые тоже автоматически заносятся в *CRM* систему.

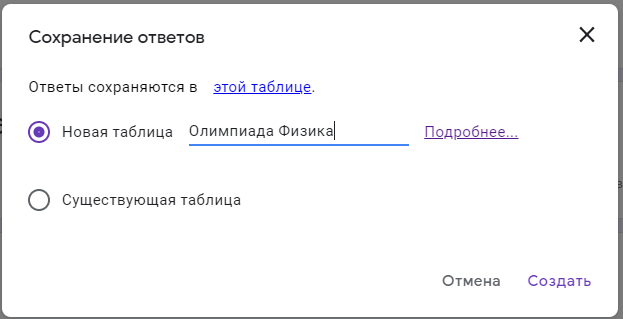


Рис. 17. Настройка сохранения ответов

Далее нужно настроить оповещения о прохождении абитуриентом анкеты. Для этого необходимо открыть меню расширений и выбрать *Form Notifications* (см. рис.18), а затем нажать кнопку «*Open*» во всплывшем окне (см. рис.19).

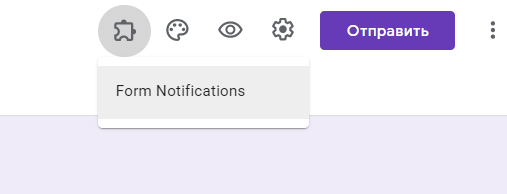


Рис. 18. Меню расширений

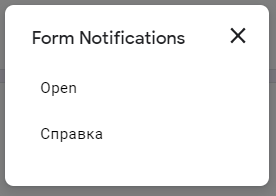


Рис. 19. Окно расширения

Расширение необходимо настроить следующим образом (см. рис.20). В графе recipient – получатель, нужно указать почту бота, который будет получать оповещения и обрабатывать их, и в графе ниже нужно указать число «1»: оповещение будет приходить после каждого завершения анкеты.

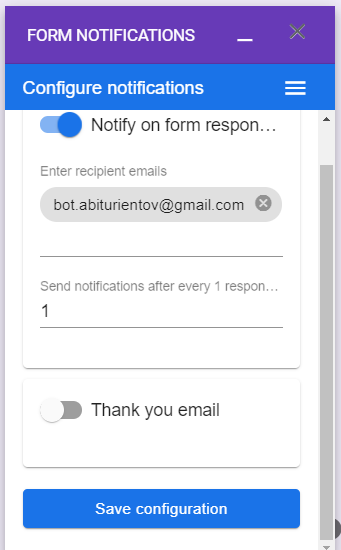


Рис. 20. Настройка расширения

Далее необходимо перейти в таблицу ответов, нажав на зелёную иконку таблицы для настройки доступа к этой таблице бота и сервисного аккаунта (см. рис.21).

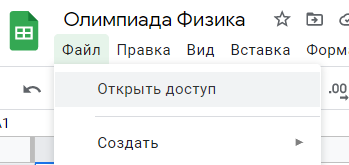


Рис. 21. Переход к настройке доступа к таблице

В открывшемся меню нужно ввести адреса почты бота (*bot.abiturientov@gmail.com*) и сервисного аккаунта (*martyan@api-project-313216.iam.gserviceaccount.com*), либо выбрать из списка, если настройка уже предварительно проводилась (см. рис.22).

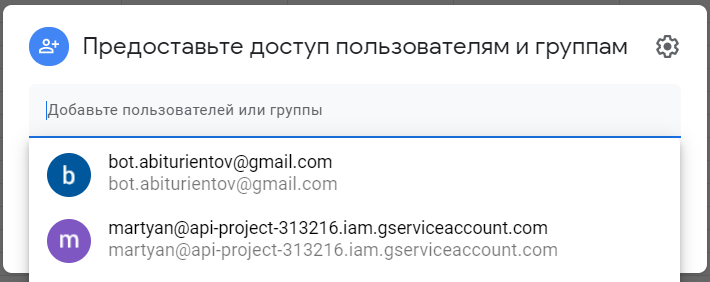


Рис. 22. Предоставление доступа для обработки данных

И теперь, проведя все эти настройки можно выкладывать опрос для абитуриентов на любых платформах, где “Абитуриент-центру” удобно. В будущем планируется перенести скрипт на сервер университета, где он, получая оповещения от новых опросов, будет так же, как и всегда, обрабатывать данные и импортировать их в *CRM* систему.

# Глава Реализация

Для реализации приложения использовался ряд сторонних библиотек, предназначенных для связи с различными сервисами, обработки данных и управления ходом выполнения программы (см. рис.23).

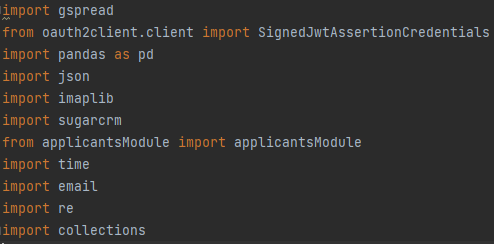


Рис. 23. Список используемых библиотек и классов

Приложение представляет из себя бесконечный цикл с задержкой, в котором раз в какой-то период (1 час) выполняются действия необходимые для получения, обработки и отправления данных. Сначала мы, используя учётные данные бота, созданного специально для этой программы, аутентифицируемся на сервере *imap.gmail.com*, чтобы получить доступ к почтовому ящику бота, куда приходят оповещения о пройденных анкетах. В ящике мы ищем новые непрочитанные письма от *Form Notifications* – расширения *Google Forms*, а также подсчитываем их количество (см. рис.24).



Рис. 24. Получение новых оповещений

Далее в каждом найденном письме, нам необходимо обратиться к телу письма и с помощью регулярного выражения, обнаружить там ссылку на вложение – таблицу с ответами на опросы и записать в список. Затем мы подсчитываем количество дубликатов для каждой ссылки на таблицу. Это нужно для того, чтобы знать, сколько людей ответило на анкету за последний час. Впоследствии мы будем из таблицы ответов считывать столько последних строк, сколько мы насчитали дубликатов (см. рис.25).

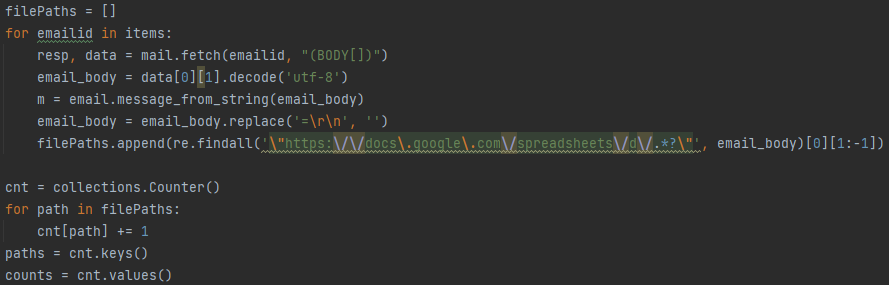


Рис. 25. Обработка писем

Теперь, когда у нас есть ссылки на таблицы, лежащие в облачном хранилище автора анкеты, для каждой из них мы аутентифицируемся в *docs.google.com* и скачиваем её содержимое. Далее мы форматируем данные из таблицы: измененяем названия столбцов на более удобные и преобразуем таблицу из формата *DataFrame* в список списков, представляющих из себя строки таблицы [4] (см. рис.26).

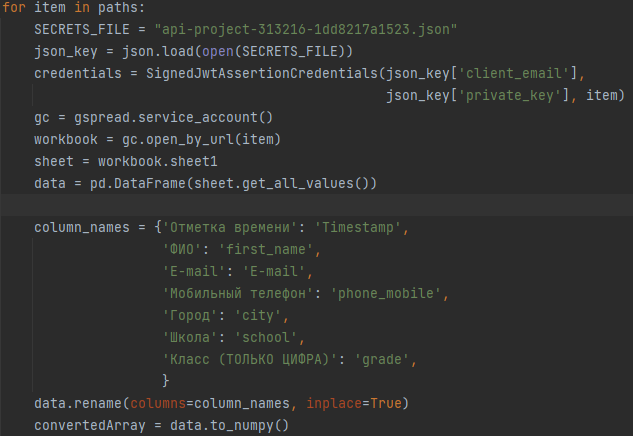


Рис. 26. Обработка таблиц

Следующим шагом будет занесение данных в *CRM* систему “Абитуриент-центра”. Для этого надо аутентифицироваться, используя учетные данные специального пользователя системы. А затем, в зависимости от ранее подсчитанного количества дубликатов оповещений, сформировать структуры данных, соответствующие формату модуля абитуриентов в *CRM* системе, используя отформатированные данные таблиц, и отправить их (см. рис.27).

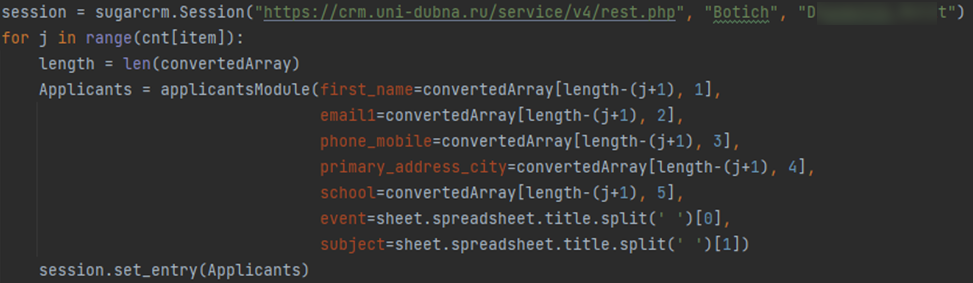


Рис. 27. Отправка данных в *CRM* систему

После проделывания этих шагов, программа приостанавливает свою работу с помощью метода *sleep* на один час, и повторяется заново. В случае, если новых ответов на анкеты не было, и оповещения не пришли, цикл завершается на первом шаге, описанном на рисунке 24.

# Заключение

В ходе работы была проанализирована предметная область. Были изучены принципы работы нескольких систем и сервисов, а также документации библиотек, которые были использованы для разработки. На основе этих данных были выдвинуты функциональные требования к приложению: новые ответы на анкеты должны автоматически обнаруживаться, обрабатываться и заноситься в *CRM* систему “Абитуриент-центра”.

Были сформированы требования к структуризации и настройке будущих опросов: они должны содержать необходимый минимум чётко названных полей-вопросов, должна быть настроена рассылка оповещений и открыт доступ сервисным аккаунтам к таблице, содержащей ответы.

Проект был разработан и реализован по заказу “Абитуриент-центра” Государственного Университета «Дубна» для оптимизации рабочего процесса сотрудников центра за счёт автоматизации «ручной» работы с базами абитуриентов.

На данный момент скрипт может работать на машине сотрудников “Абитуриент-центра” и требует ручного включения, но в будущем планируется перенести его на сервер университета для полностью автоматической работы.

# Список литературы

## Интернет-ресурсы

1. Chris Moffitt Collecting Data with Google Forms and Pandas [Электронный ресурс] — Электрон. текст. — 2015. — Режим доступа: https://pbpython.com/pandas-google-forms-part1.html, свободный (дата обращения: 17.05.2021).
2. СRM-система – что это такое? [Электронный ресурс] — Электрон. текст. — 2015. — Режим доступа: https://salesap.ru/crm\_sistemy\_chto\_eto/, свободный (дата обращения: 19.05.2021).
3. SuiteCRM Documentation Site [Электронный ресурс] — Электрон. текст. — 2019. — Режим доступа: https://docs.suitecrm.com/, свободный (дата обращения: 22.05.2021).
4. Pandas documentation [Электронный ресурс] — Электрон. текст. — 2021. — Режим доступа: https://pandas.pydata.org/pandas-docs/stable/index.html, свободный (дата обращения: 25.05.2021).
5. Google Cloud Documentation [Электронный ресурс] — Электрон. текст. — 2021. — Режим доступа: https://cloud.google.com/docs, свободный (дата обращения: 21.05.2021).
6. Ryanss Python client for SugarCRM API [Электронный ресурс] — Электрон. текст. — 2016. — Режим доступа: https://pypi.org/project/sugarcrm/#description, свободный (дата обращения: 27.05.2021).