

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра "Кибербезопасность, обработка и хранение информации"

Камаш К. К.

Администрирование облачных серверов и систем управления  
взаимоотношениями с клиентами

**Дипломная работа**

Специальность 5В070300 – Информационные системы

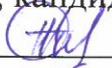
Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра "Кибербезопасность, обработка и хранение информации"

**ДОПУЩЕН К ЗАЩИТЕ**  
Заведующая кафедрой  
«Кибербезопасность, обработка  
и хранение информации»,  
PhD, кандидат технических наук  
 Сейлова Н. А.  
« 13 » 05 2019.

**Дипломная работа**

На тему: «Администрирование облачных серверов и систем управления  
взаимоотношениями с клиентами»

Специальность 5В070300 – «Информационные системы»

Выполнил:

Камаш Кайсар Кайратулы

Рецензент,  
PhD, старший научный сотрудник  
 Козбакова А. Х.

« 13 » мая 2019.

Научный руководитель,  
Доцент, кандидат технических  
наук

 Косников В. А.  
« 13 » мая 2019.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра "Кибербезопасность, обработка и хранение информации"

5B070300 – Информационные системы

**УТВЕРЖДАЮ**

Заведующая кафедрой  
«Кибербезопасность, обработка  
и хранение информации»,  
PhD, кандидат технических наук

 Сейлова Н. А.  
«13» 05 2019.

**ЗАДАНИЕ**

**на выполнение дипломной работы**

Обучающемуся Камаш Кайсар Кайратұлы

Тема: «Администрирование облачных серверов и систем управления  
взаимоотношениями с клиентами»

Утверждена приказом Ректора Университета № 1162-б от «16» 10 2018.

Срок сдачи законченной работы «13» 05 2019.

Исходные данные к дипломной работе: Облачные технологии и их решения в бизнес – процессах, системы управления взаимоотношениями с клиентами, обзор литературы, основанные на теоретических данных.

Краткое содержание дипломной работы:

- а) Современное состояние развития информационных систем предприятий;
- б) Модели, методы, алгоритмы эффективного функционирования информационной системы ТОО “Leo Group Services”;
- с) Информационное и программное обеспечение информационной системы ТОО “Leo Group Services”.

Перечень графического материала: представлены 15 слайдов презентации работы  
Рекомендуемая основная литература: из 10 наименований

**ГРАФИК**  
подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Анализ и обзор облачных технологии и CRM - систем.	08.01.2019 - 08.02.2019.	
Разработать функциональную структуру, обобщенный алгоритм функционирования систем; Предложить алгоритмы решения функциональных задач.	08.02.2019 - 13.03.2019.	
Разработка программной части	13.03.2019 - 20.04.2019.	

**Подписи**  
консультантов и нормоконтролера на законченную дипломную работу с указанием относящихся к ним разделов работы

Наименование разделов	Консультанты, И.О.Ф (уч. степень, звание)	Дата подписания	Подпись
Программная часть	Бауыржан М. Б., магистр технических наук	8.05.19г	
Нормоконтролер	Киселева О.В., PhD	8.05.19	

Научный руководитель

Косников В. А.

Задание принял к исполнению обучающийся

Камаш Қ. Қ

Дата

«08» мая 2019г.

## РЕЦЕНЗИЯ

Дипломная работа  
(наименование вида работы)

Камаш Кайсар Кайратулы  
(Ф.И.О обучающегося)

5B070300 – Информационные системы

На тему: Администрирование облачных серверов и систем управления взаимоотношениями с клиентами

Выполнено:

- а) программная часть на 5 страниц
- б) общий объём работы 30 страниц

Дипломная работа предназначена для улучшения процесса автоматизации в компаниях и для удаленного взаимодействия с системой управления взаимоотношениями с клиентами.

Основной части дипломной работы были собраны теоретические аспекты, а именно были приведены понятие и сущность облачных технологии, их видов и их ключевых факторов. Было разработано функциональная структура систем по взаимоотношению с клиентами с облачными решениями.

Была описана разработка информационного и программного обеспечения информационной системы ТОО “Leo Group Services”.

## ОЦЕНКА РАБОТЫ

Дипломная работа выполнена на оценку 85 % и считаю что автору степень бакалавра по специальности 5B070300 – «Информационные системы».

Рецензент

PhD, старший научный сотрудник  
Козбакова А. Х.

«13» мая 2019 г.

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на дипломную работу «Администрирование облачных серверов и систем управления взаимоотношениями с клиентами»

(наименование вида работы)

Камаш Қайсар Қайратұлы

(Ф.И.О. обучающегося)

по специальности 5В070300 – «Информационные системы»

(шифр и наименование специальности)

Тема: «Администрирование облачных серверов и систем управления» является на сегодняшний день актуальной практически для любого предприятия или организации, связанной с бизнесом. В первой главе дипломной работы Камаш К.К. проанализировал современное состояние облачных технологий и систем взаимоотношениями с клиентами с точки зрения их реализации для ТОО «Leo Group Services». Результатом анализа явилась постановка задач дипломной работы.

Во второй главе дипломной работы Камаш К.К. предложил функциональную структуры информационной системы и разработал алгоритмы ее функционирования, которые реализованы в третьей главе. В четвертой главе дипломной работы дано описание технической и программной реализации, а также показаны перспективы внедрения и развития информационной системы.

Во время работы над информационной системой Камаш К.К. проявил достаточный уровень квалификации по специальности 5В070300 – «Информационные системы», полученный им во время учебы, который позволил ему справиться с поставленными задачами. Следует отметить, что чрезмерное увлечение самостоятельностью не всегда приводило к желаемому результату в процессе обсуждения дипломной работы с научным руководителем.

Научный руководитель

ассоциированный профессор, кандидат технических наук, доцент  
(должность, уч. степень, звание)

В.А. Косников Косников В.А.  
(подпись)

«13» мая 2019г.

**Протокол анализа Отчета подобия**

**заведующего кафедрой / начальника структурного подразделения**

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Камаш Кайсар

**Название:** Администрирование облачных серверов и систем управления взаимоотношениями с клиентами

**Координатор:** Вячеслав Косников

**Коэффициент подобия 1:**1,7

**Коэффициент подобия 2:**0

**Тревога:**0

**После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:**

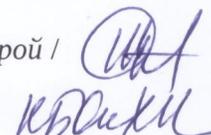
- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....  
.....  
.....  
.....  
.....

Дата 13.05.192

Подпись заведующего кафедрой /  
начальника структурного подразделения



Окончательное решение в отношении допуска к защите, включая обоснование:

.....  
.....  
.....  
.....  
.....

.....  
Дата 13.05.19г.

.....  
Подпись заведующего кафедрой /   
начальника структурного подразделения 

## Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Камаш Кайсар

**Название:** Администрирование облачных серверов и систем управления взаимоотношениями с клиентами

**Координатор:** Вячеслав Косников

**Коэффициент подобия 1:** 1,7

**Коэффициент подобия 2:** 0

**Тревога:** 0

**После анализа Отчета подобия констатирую следующее:**

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....  
.....  
.....  
.....  
.....  
.....

13.05.19

Дата

*[Handwritten Signature]*

Подпись Научного руководителя



Университет:	Satbayev University
Название:	Администрирование облачных серверов и систем управления взаимоотношениями с клиентами
Автор:	Камаш Кайсар
Координатор:	Вячеслав Косников
Дата отчета:	2019-05-04 16:41:37
Коэффициент подобия № 1:	<b>1,7%</b>
Коэффициент подобия № 2:	<b>0,0%</b>
Длина фразы для коэффициента подобия № 2:	<b>25</b>
Количество слов:	4 851
Число знаков:	39 608
Адреса пропущенные при проверке:	
Количество завершенных проверок:	38

>> Самые длинные фрагменты, определенные, как подобные

№	Название, имя автора или адрес гиперссылки (Название базы данных)	Автор	Количество одинаковых слов
1	URL_ <a href="https://o7planning.org/ru/11385/introduction-to-python">https://o7planning.org/ru/11385/introduction-to-python</a>		16
2	АНАЛИЗ ПОТЕНЦИАЛЬНОЙ ВОЗМОЖНОСТИ РЕАЛИЗАЦИИ ПРОЕКТА «ЭНЕРГИЯ БУДУЩЕГО» В РАМКАХ МЕЖДУНАРОДНОЙ ВЫСТАВКИ ЭКСПО-2017 Мырхеева Д. Н. NARXOZ (NEU) (Кафедра Экономика природопользования (ШПН))		12
3	URL_ <a href="https://rabotanur.kz/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80+%D0%BF%D0%BE+%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B5+%D1%81+%D0%BA%D0%BB%D0%B8%D0%B5%D0%BC">https://rabotanur.kz/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80+%D0%BF%D0%BE+%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B5+%D1%81+%D0%BA%D0%BB%D0%B8%D0%B5%D0%BC</a>		12
4	URL_ <a href="https://rabotanur.kz/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80+%D0%BF%D0%BE+%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B5+%D1%81+%D0%BA%D0%BB%D0%B8%D0%B5%D0%BC">https://rabotanur.kz/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80+%D0%BF%D0%BE+%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B5+%D1%81+%D0%BA%D0%BB%D0%B8%D0%B5%D0%BC</a>		12
5	URL_ <a href="https://nauchforum.ru/studconf/science/11/37126">https://nauchforum.ru/studconf/science/11/37126</a>		11
6	Analiza możliwości zarządzania bazą danych przechowywaną w chmurze Politechnika Lubelska (Politechnika Lubelska)	Rafał Gózdź	10
7	URL_ <a href="https://o7planning.org/ru/11385/introduction-to-python">https://o7planning.org/ru/11385/introduction-to-python</a>		6
8	URL_ <a href="https://o7planning.org/ru/11385/introduction-to-python">https://o7planning.org/ru/11385/introduction-to-python</a>		5

>> Документы, в которых найдено подобные фрагменты: из RefBooks

Не обнаружено каких-либо заимствований

>> Документы, содержащие подобные фрагменты: Из домашней базы данных

Не обнаружено каких-либо заимствований

>> Документы, содержащие подобные фрагменты: Из внешних баз данных

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобия № 2

№	Название (Название базы данных)	Автор	Количество одинаковых слов (количество фрагментов)
1	АНАЛИЗ ПОТЕНЦИАЛЬНОЙ ВОЗМОЖНОСТИ РЕАЛИЗАЦИИ ПРОЕКТА «ЭНЕРГИЯ БУДУЩЕГО» В РАМКАХ МЕЖДУНАРОДНОЙ ВЫСТАВКИ ЭКСПО-2017 Мырхеева Д. Н. NARXOZ (NEU) (Кафедра Экономика природопользования (ШПН))		12 (1)
2	Analiza możliwości zarządzania bazą danych przechowywaną w chmurze Politechnika Lubelska (Politechnika Lubelska)	Rafał Gózdź	10 (1)

>> Документы, содержащие подобные фрагменты: Из интернета

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобия № 2

№	Источник гиперссылки	Количество одинаковых слов (количество фрагментов)
1	URL_ <a href="https://o7planning.org/ru/11385/introduction-to-python">https://o7planning.org/ru/11385/introduction-to-python</a>	27 (3)
2	URL_ <a href="https://rabotanur.kz/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80+%D0%BF%D0%BE+%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B5+%D1%81+%D0%BA%D0%BB%D0%B8%D0%B5%D0%BC">https://rabotanur.kz/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80+%D0%BF%D0%BE+%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B5+%D1%81+%D0%BA%D0%BB%D0%B8%D0%B5%D0%BC</a>	24 (2)
3	URL_ <a href="https://nauchforum.ru/studconf/science/11/37126">https://nauchforum.ru/studconf/science/11/37126</a>	11 (1)

## ANNOTATION

The program implemented for this diploma work is intended to improve the automation process in companies and to interact remotely with customer relationship management system.

As a basis for the research of the diploma work, data on cloud computing and their use in enterprises are presented.

The program used the Python programming language, the PyCharm integrated development environment, the AmoCRM system API, and the text format for reading stored JSON data.

The program provides remote access to cloud programs, search for tasks of employees related to CRM, and the creation of reports on the customer database.

## АННОТАЦИЯ

Программа, реализованная для данной дипломной работы, предназначена для улучшения процесса автоматизации в компаниях и для удаленного взаимодействия с системой управления взаимоотношениями с клиентами.

В качестве основы исследования дипломной работы приводятся данные по облачным вычислениям, их использование в предприятиях.

В ходе программы использовались язык программирования Python, интегрированная среда разработки PyCharm, системный интерфейс API AmoCRM, текстовый формат для чтения сохраненных данных JSON.

Программа обеспечивает удаленный доступ к облачным программам, поиск задач сотрудников, связанных с CRM, и создание отчетов по базе данных клиентов.

## АҢДАТПА

Осы дипломдық жұмыста құрылған бағдарлама компаниядағы процесстерді автоматтандыруға үлесін қосып, клиенттермен қарым-қатынасты басқару жүйесімен қашықта байлансуға арналған.

Негіз ретінде дипломдық жұмыстағы зерттеулерінде бұлттық есептеулер туралы ақпараттар, есептулердің компанияларда қолданылуы үлгілері алынған.

Бағдарламаны құру барысында Python программалау тілі, PyCharm интеграцияланған құрылыс ортасы, AmoCRM жүйесінің API интерфейсы, JSON мәтіндік алмасу форматы қолданылды.

Бағдарлама бұлттық бағдарламалармен қашықта байланысу, CRM жүйесімен қызмет атқаратын жұмысшылардың тапсырмаларын, клиенттер базасы туралы мәліметтер есептер форматында алу қызметтерін атқарады.

## СОДЕРЖАНИЕ

Введение	7
1 Современное состояние развития информационных систем предприятий	8
1.1 Обзор облачных решений в информационных системах предприятия	8
1.2 Исследование технологий управления взаимоотношениями с клиентами	11
1.3 Постановка задач	15
2 Модели, методы, алгоритмы эффективного функционирования информационной системы ТОО “Leo Group Services” по взаимоотношению с клиентами	16
2.1 Функциональная структура информационной системы	16
2.2 Обобщенный алгоритм функционирования систем	17
2.3 Алгоритмы решения функциональных задач	18
3 Информационное и программное обеспечение информационной системы ТОО “Leo Group Services”	20
3.1 Описание информационной обеспечения	20
3.2 Структура программного обеспечения	22
4 Реализация и внедрение	26
4.1 Описание технической и программной реализации	26
4.2 Перспективы развития	27
Заключение	28
Список терминов и сокращений	29
Список использованной литературы	30
Приложение	31
Приложение А	

## ВВЕДЕНИЕ

Администрирование облачных серверов и систем управления взаимодействия с клиентами является одним из основных задач в организациях использующих облачные технологии. На сегодняшний день, большинство компании на территории Республики Казахстан скептически относятся к внедрению своей инфраструктуры бизнеса облачные технологии.

Обусловлено это во многих случаях в непонимании самой облачной технологии, по соображениям безопасности, в затратах при переходе к облачным технологиям.

Актуальность данной дипломной работы определяется тем, что по анализу прошедших лет и оценке мирового рынка специалистами, в частности из корпорации Microsoft и Gartner, ожидается в перспективе неизбежный переход к облачным технологиям. В связи с этим необходимо заранее дать определение насчет своих ожиданий от облачных технологий.

Целью данной дипломной работы является – улучшение и внедрение облачных решений в процессах автоматизации взаимоотношения с клиентами, а также разработка функциональную структуру систем взаимоотношения с клиентами.

Задачами, для достижения данной поставленной цели являются:

- рассмотрение процессов облачных технологии и систем управления с взаимоотношениями с клиентами, используемых в компаниях в целом;
- разработать функциональную структуру систем по взаимоотношению с клиентами с облачными решениями;
- разработать обобщённый алгоритм функционирования систем;
- предложить алгоритмы решения функциональных задач;
- разработать архитектуру информационного и программного обеспечения систем;
- выбрать комплекс технических и программных средств для реализации систем;
- внедрить систему в функциональную структуру информационной системы по взаимоотношению с клиентами.

Дипломная работа состоит из введения, четырех глав, заключения, списка использованной литературы и приложений. В первой главе собраны теоретические аспекты, а именно были приведены понятие и сущность облачных технологии, их видов и их ключевых факторов. Во второй главе, анализируется, модели, методы и алгоритмы информационной системы ТОО “Leo Group Services” по взаимоотношению с клиентами. На третьей главе, описывается разработка информационного и программного обеспечения информационной системы ТОО “Leo Group Services”. На последней главе, описывается реализация разработки и дальнейшие перспективы. В заключении были подведены выводы, обобщающие итоги по данной дипломной работе.

# **1 Современное состояние развития информационных систем предприятий**

## **1.1 Обзор облачных решений в информационных системах предприятия**

К современному и дальнейшему развитию информационных систем предприятий по большому счету повлияло облачные технологии. С развитием рынка сервисов, технологии работающих через сети Интернет, облачные провайдеры стали предлагать свои решения не только физическим лицам, но и предприятиям, организациям разного типа – малого, среднего и крупного бизнеса.

Облачные вычисления являются расширенной технологией для запуска либо для поддержания жизнедеятельности предприятий. Вместо того, чтобы запустить приложения на персональный компьютер или через локальную вычислительную сеть, они работают на коллективном режиме реального времени. Это – ориентированное самообслуживание; для использования любого приложения, которое работает в облаке, необходимо авторизоваться и в дальнейшем входить в систему, настроить его согласно потребительскому использованию и начать использовать его. Увеличивается число организаций различных отраслей выполняющих собственные задачи в приложениях находящихся в облаке, таких как CRM, ERP, HR и изготовленные на заказ приложения. "Облачная" технология следует модели Pay as You Go – с оплатой по мере использования, тем самым снижая затраты на расходы. За использование облачных приложений, потребитель не будет платить за все аппаратное и программное обеспечение или за обширную конфигурацию и за обслуживание полного технологического стека. Облако обеспечивает больше масштабируемый, более надежный и более безопасный сервис. И периодически облачные решения обновляются облачными поставщиками для упрощения новых возможностей, безопасности, улучшения производительности в автоматическом режиме. Характеристики "облачной" технологии различаются следующим образом:

**Самообслуживание по запросу (On-demand Self Service):** клиент может воспользоваться любым контрактным вычислительным ресурсом, такими как вычислительная мощность, объем памяти или прикладные программы у поставщика услуг самостоятельно, а именно без участия человека.

**Широкополосный сетевой доступ (Broad Network Access):** К вычислительным ресурсам можно получить доступ где угодно в любое время с любым стандартным устройством, которое может получить доступ к сети.

**Объединение ресурса (Resource Pooling):** вычислительные ресурсы поставщика собраны для обеспечения ограниченного сервиса. Объединенные ресурсы могут быть географически распространены через несколько дата-центров. Вычислительные ресурсы поставщика совместно используются

несколькими потребителями. Ресурсы динамично присвоены потребителям в зависимости от спроса.

**Эластичность (Rapid Elasticity):** Клиент может запросить больше ресурсов, когда это необходимо, и заблокировать их, когда это не требуется. С точки зрения клиента ресурсы неограниченны. Клиент платит только за общий объем использованных ресурсов.

**Измеряемая услуга (Measured Service):** Системы облачных вычислений являются адаптивными системами. Они автоматически сбалансируют загрузки и оптимизируют использование ресурсов. Потребитель может анализировать и контролировать использование ресурсов управления, таким образом, узнавая прозрачность расходов.

#### 1.1.1 Модели услуг "облачной" технологии и "облачного" развертывания

Существует три основных категории моделей облачного сервиса, по которым сервисы занимают места в информационных системах предприятия. Используются они следующим образом:

**Программное обеспечение как сервис (SaaS):** Вместо того чтобы устанавливать программное обеспечение на компьютере клиента и обновлять его с помощью регулярных исправлений, частых обновлений версий и т. д., Такие приложения, как обработка текстов, CRM (управление взаимоотношениями с клиентами), ERP (планирование ресурсов предприятия), становятся доступными (размещаются) через Интернет для потребления конечного пользователя.

**Платформа как сервис (PaaS):** Вместо покупки лицензий на программное обеспечение для таких платформ, как операционные системы, базы данных и промежуточное программное обеспечение, эти платформы, а также наборы для разработки программного обеспечения (SDK) и инструменты (такие как Java, .NET, Python, Ruby on Rails) доступны через Интернет.

**Инфраструктура как сервис (IaaS):** Это относится к материальным физическим устройствам (необработанным вычислениям), таким как виртуальные компьютеры, серверы, устройства хранения данных, передача данных по сети, которые физически расположены в одном центральном месте (центр обработки данных), но к ним можно получить доступ и использовать их через Интернет с использованием систем аутентификации входа через любого терминала или устройства.

Развертывание облачного сервиса может быть в любом из следующих видов моделей:

**Публичное облако:** Это доступно от стороннего поставщика услуг через Интернет и очень экономически эффективный для малого и среднего бизнеса для развертывания IT-решений. Например, Google Apps, Microsoft Azure.

**Частное облако:** Этим управляют в организации и подходит для крупных предприятий (управляемый в стенах предприятий).

**Коллективное облако:** Это использует и управляет группа предприятий, которые совместно использовали общие интересы.

**Гибридное облако:** Это - комбинация публичного облака и частного облака.

### 1.1.2 Преимущества облачных вычислений

Облачные вычисления предоставляют для малого и среднего бизнеса такие преимущества, как:

**Гибкость:** облачные сервисы могут быстро встретить бизнес-потребность любой организации путем предоставления различных услуг.

**Без предварительной оплаты:** услуги по облачным вычислениям обычно с оплатой по мере использования, таким образом, существует непотребность в оплачиваемой авансом стоимости инфраструктуры. Так как облачные вычисления намного быстрее для развертывания, компании (малые и средние предприятия) имеют крайние затраты на запуск проекта и вероятную продолжающуюся работу расходы.

**Увеличенное сотрудничество:** Повышение числа сотрудников облачных вычислений путем добавления и разрешения прав для всех сотрудников, чтобы синхронизироваться и работать над документами и общими приложениями одновременно от их собственного места. Это позволяет им следовать за коллегами и записями для получения очень важных обновлений в режиме реального времени.

**Автоматические обновления программного обеспечения:** поставщики облачного решений делают обслуживание сервера включая обновления программного обеспечения, обновления системы защиты, освобождая время и ресурсы их клиентов для различных других задач.

**Контроль документооборота:** Если компания не использует облако, рабочие должны отправить файлы с одной стороны к другому по электронной почте. Это означает, что только один человек может работать над файлом за один раз и тот же документ будет дублирован в миллионах форматов и имен.

**Безопасность:** Компании, хранящие все в облаке, могут получать доступ к данным даже если что-либо произойдет с машиной.

**Работа сервисами удаленном виде:** Облачные вычисления позволяют сотрудникам работать отовсюду. Такая эластичность положительно влияет на баланс продолжительности службы рабочих знаний и производительность.

**Аварийное восстановление:** Когда компании начинают работать на облачных сервисах, им нет необходимости иметь сложные планы аварийного восстановления комплекса устройства, потому что поставщики облачного сервиса решают большинство проблем очень быстрым способом.

**Конкурентоспособность:** "Облачные" технологии предоставляют доступ малых и средних предприятий к технологии промышленного класса путем обеспечения различных ERP-решений. Это также позволяет малым и средним компаниям действовать быстрее чем крупные конкурентов с физическими серверами и другими решениями.

Исходя от этих моделей, характеристик и преимуществ, используются облачные сервисы в компаниях для решений своих задач общего назначения и для решения задач специализированного назначения. Задачами общего назначения являются ежедневные процессы от сотрудников предприятия, такие как отправка электронных писем, составление отчетов и плана работы на

определенный период, и так далее. У каждого сотрудника и у каждого отдела есть свои ежедневные процессы. При решении этих задач общего назначения среди организации большой популярностью используются облачные решения от корпорации Google.

Облачная телефония предоставляет решения общего назначения для бизнес-коммуникаций для всех типов фирм с простой в использовании моделью с оплатой по мере роста. Он служит платформой «все в одном», которая отвечает всем коммуникационным потребностям предприятий посредством голосовой связи, вызовов, автосекретаря, интерактивного голосового ответа и многих других функций, которые помогают фирмам лучше понимать своих клиентов, быстрее разрешать их запросы и повышать производительность агентов. Благодаря этому технологическому прогрессу в коммуникации, предприятия могут увеличить свое присутствие на рынке и привлечь потенциальных клиентов. Потребители могут связаться с фирмой, выбирая наиболее удобные средства связи с предприятиями, предоставляемые через облачную телефонию. Это помогает сэкономить время и ресурсы как для клиентов, так и для бизнеса, и помогает создать долгосрочные отношения между ними.

## **1.2 Исследование технологий управления взаимоотношениями с клиентами**

Управление взаимоотношениями с клиентами (CRM) является комбинацией методов, стратегий и технологий, используемые компаниями, чтобы управлять и проанализировать взаимодействия клиентов и данные в течение жизненного цикла клиентов, с целью улучшения обслуживания и отношений между клиентами и помогающий в удержании их, на ведущем росте объема продаж. К данному жизненному циклу относятся – привлечение, удержание, продажа, лояльность. Итогом применения CRM-систем будет являться повышение конкурентоспособности компании, а также увеличение её прибыли.

Системы CRM компилируют данные о клиентах через различные каналы или точки контакта между клиентом и компанией, которая могла включать веб-сайт компании, телефон, живой чат, прямую рассылку, рекламные материалы и социальные сети. Системы CRM могут также дать подробную информацию о персональных данных клиентов, истории покупки, их предпочтения и проблемы.

На наиболее базовом уровне программное обеспечение CRM консолидирует информацию о клиенте и документы в единственную базу данных CRM, таким образом, что пользователи системы могут более легко получить доступ и управлять ими. Со временем много дополнительных функций были добавлены к системам CRM для использования их более эффективным способом. Некоторые из этих функций включают различные потребительские взаимодействия по электронной почте, телефону, социальным сетям или другим каналам взаимодействия с клиентами; в зависимости от системных возможностей, автоматизируя различные рабочие процессы, такие как задачи,

календари и предупреждения; и давая менеджерам способность отслеживать эффективность и производительность своей работы на основе информации зарегистрированной в системе.

#### 1.2.1 Компоненты CRM - систем

Основными компонентами полнофункциональных CRM – систем являются:

**Маркетинг автоматизации:** инструменты CRM с возможностями маркетинговой автоматизации могут автоматизировать повторяющиеся задачи для улучшения маркетинговой деятельности в различных точках жизненного цикла предприятий. Например, поскольку перспективы продаж входят в систему, она могла бы автоматически отправить коммерческие предложения, обычно по электронной почте или через социальные сети, с целью заключения договора, тем самым делая продажи своего продукта.

**Автоматизация процессов продаж (SFA, “Sales Force Automation”):** инструменты автоматизации процессов продаж отслеживают потребительские взаимодействия и автоматизируют определенные деловые функции цикла продаж, которые необходимы для следования примерам и привлечения и получения новых клиентов.

**Автоматизация центра взаимодействия с клиентом:** Разработанный для сокращения утомительных аспектов задания менеджера по продажам, автоматизация “контакт-центра” может включать записанное заранее аудио, которое помогает в потребительском решении проблем и распространении информации. Различные программные инструменты, которые интегрируются с настольными инструментами менеджера, могут обработать потребительские запросы для сокращения времени вызовов и упростить процессы обслуживания клиентов.

**Технология геолокации или услуги на основе определения местоположения:** Некоторые системы CRM включают технологию, которая может создать географические маркетинговые кампании на основе физических местоположений клиентов, иногда интегрирующихся с популярными основанными на местоположении приложениями GPS. Технология геолокации может также использоваться в качестве сетей или инструмента управления контактами для нахождения потенциальных продаж на основе местоположения.

**Автоматизация рабочего процесса:** системы CRM помогают компаниям оптимизировать процессы, оптимизировав ежедневные рабочие нагрузки, позволяя сотрудникам сфокусироваться на более сложных, высокоуровневых задачах.

**Управление персоналом (HRM):** системная справка CRM отслеживает информацию о сотруднике, такую как контактная информация, анализы производительности и эффективности в компании. Это позволяет отделу человеческих ресурсов эффективнее управлять внутренними трудовыми ресурсами.

Аналитика: Аналитика в CRM помогает создать лучшие уровни удовлетворенности потребителя путем анализа пользовательских данных, и помощь создают предназначенные маркетинговые кампании.

### 1.2.2 Типы технологии CRM - систем

На сегодняшний день, основными поставщиками систем управления в взаимоотношения с клиентами является такие компании как: Oracle, Salesforce, Microsoft, SAP и Oracle. Другие поставщики популярны среди малого и для среднего бизнеса, но эти четыре имеют тенденцию быть выбором для крупных корпораций. Что касается казахстанского рынка, наиболее известными CRM-системы для предприятий малого и среднего, иногда и крупного бизнеса являются: «AmoCRM», «SugarCRM», «1С:CRM Стандарт», «ASoft CRM», «Monitor CRM», «BPMonline CRM», «Zoho CRM», «Salesforce CRM», «Teamlab CRM», «1С-Битрикс», «Quick Sales CRM», «Sales Expert CRM», «FreshOffice CRM», «RegionSoft CRM».

Типы предлагаемой технологии CRM:

Собственный (Локальный) CRM: эта система возлагает ответственность администрирования, управления, безопасности и обслуживания базы данных и информации о компании с помощью программного обеспечения CRM. С этим подходом покупки компании лицензируют заранее вместо того, чтобы покупать ежегодные подписки от облачного поставщика CRM. Программное обеспечение находится на собственных серверах компании, и потребитель системы взаимоотношения управления с клиентами принимает дальнейшую стоимость любых обновлений.

Также обычно требуется длительный процесс установки для полной интеграции данных компании. Компании со сложными потребностями в CRM могут извлечь выгоду из локального развертывания.

Облачный CRM: С облачным CRM, также известным как SaaS (программное обеспечение как сервис) или CRM по запросу, данные будут храниться во внешней, удаленной сети, к которой сотрудники могут получить доступ в любое время, в каком-либо месте, где существует интернет соединение, иногда со сторонним поставщиком услуг, наблюдающим за установкой и обслуживанием. Быстрые возможности относительно простого развертывания облака обращаются к компаниям с ограниченными технологическими экспертными знаниями или ресурсами. Компании могли бы рассмотреть облачный CRM как более экономически эффективную опцию. Поставщики облачных систем управления взаимодействия с клиентом, взимают плату с пользователя по подписке и предлагают возможность ежемесячных или ежегодных платежей. Безопасность данных является первоочередной задачей для компаний с помощью облачных систем, поскольку компания физически не управляет устройством хранения данных и обслуживанием его данных. Если облачный поставщик обанкротился или получен другой компанией, данные предприятия могут ставиться под угрозу или потеряны. Проблемы совместимости могут также возникнуть, когда данные первоначально перемещены от внутренней системы компании до облака.

CRM с открытым исходным кодом. Система CRM с открытым исходным кодом делает исходный код общедоступным, что позволяет компаниям вносить изменения без каких-либо затрат для компании, использующей систему. CRM-системы с открытым исходным кодом также позволяют добавлять и настраивать каналы передачи данных в каналах социальных сетей, помогая компаниям стремиться улучшить практику социальных CRM. Платформы CRM с открытым исходным кодом, такие как OroCRM, SuiteCRM и SugarCRM предлагают альтернативы собственным платформам от Salesforce, Microsoft и других поставщиков.

Принятие любого из этих методов развертывания CRM зависит от бизнес-потребностей компании, ресурсов и целей, поскольку у каждого типа предлагаемой технологии CRM различные ценовые диапазоны лицензии.

Системы управления взаимоотношениями с клиентами помогут для любых предприятий в таких случаях, как:

- Получение стандартизированную базу данных клиентов или сделок компании;
- Мониторинг эффективности и качества работы менеджеров и других сотрудников отдела продаж в любое время;
- Получение развернутой статистики и отчетов аналитики об эффективности работы с сделками и входящими звонками, а также о заполнении различного рода заявок, которые могут оставлять клиенты на сайте компании;
- Планировка повышения качества работы менеджер и остальных сотрудников отдела продаж;
- Разработка стратегии бизнеса, бизнес-плана.

Можно выделить следующие основные возможности современных CRM-систем на казахстанском рынке:

- Быстрый доступ к актуальной информации о клиентах предприятия;
- Оперативность обслуживания клиентов;
- Полная или частичное автоматизация возможных схем взаимодействия с клиентами предприятия;
- Получение аналитики, всех необходимых отчетных данных предприятия;
- Возможность автоматизации однотипных задач для снижения операционных затрат менеджеров предприятия и мониторинга эффективности менеджеров.

При выборе CRM-системы необходимо обращать внимание на следующие основные факторы:

- Гибкость, т.е. соответствие функциональных возможностей системы целям бизнеса и стратегиям определенной компании;
- Открытый исходный код системы;
- Необходимое число рабочих мест;
- Стоимость системы;
- Стоимость внедрения и сопровождение системы;
- Возможность интеграции с другими корпоративными

информационными системами;

- Соответствие CRM-системы техническим требованиям;
- Доступность услуг по внедрению CRM-системы и её поддержки в конкретном регионе;
- Функциональная составляющая CRM-системы.

### 1.2.3 Проблемы CRM – систем

Для всех продвижений в технологии CRM, без надлежащего управления, система CRM может стать больше, чем стандартизированная база данных, в которой хранится информация клиента. Наборы данных должны быть соединены, распределены и организованы так, чтобы пользователи CRM могли легко получить доступ к информации, в которой они нуждаются.

Компании могут изо всех сил пытаться достигнуть единственного представления клиента, если их наборы данных не соединены и не организованы в единственной панели инструментов или интерфейсе. Проблемы также возникают, когда системы содержат дублирующиеся данные о клиентах или устаревшую информацию. Эти проблемы могут привести к снижению качества обслуживания клиентов из-за времен долгого ожидания во время телефонных вызовов, неправильного обращения случаев технической поддержки и других проблем.

Системы CRM работают лучше всего, когда компании проводят время от времени очистку существующих данных о клиентах для устранения дубликата и неполных записей, прежде чем они добавят данные CRM с внешними источниками информации.

## 1.3 Постановка задач

В обзорах облачных решений в информационных системах предприятия и в исследовании технологий управления взаимоотношениями с клиентами анализ показал преимущества и вариативность решений от облачных вычислений и тем самым необходимо решить следующие задачи ТОО “Leo Group Services”:

- Разработать функциональную структуру систем по взаимоотношению с клиентами с облачными решениями;
- Разработать обобщенный алгоритм функционирования систем;
- Предложить алгоритмы решения функциональных задач;
- Разработать архитектуру информационного и программного обеспечения систем;
- Выбрать комплекс технических и программных средств для реализации систем;
- Внедрить систему в функциональную структуру информационной системы по взаимоотношению с клиентами.

## 2 Модели, методы, алгоритмы эффективного функционирования информационной системы ТОО «Leo Group Services» по взаимоотношению с клиентами

### 2.1 Функциональная структура информационной системы

Основой информационной системы являются ее «функциональные компоненты» – модели, методы и алгоритмы получения управляющей информации.

Функциональная структура информационной системы является общей совокупностью функциональных внутренних подсистем, комплексов задач для обработки информации, создающих функции системы управления. В системе управления предприятий выделяются самостоятельные подсистемы функционального и организационного уровня управления.

По взаимоотношению с клиентами, функциональная структура данной информационной системы состоит из комплексов используемых сервисов и задач, из функциональных компонентов, методов принятия решений и моделей управления. На данном этапе, функциональная структура информационной системы по взаимоотношению с клиентами ТОО «Leo Group Services» начинается с методов привлечения клиентов и конечной итогом является успешная взаимоотношение с клиентом – продажа продукта.

Краткая характеристика о компании. Компания LeoLeo.me (ТОО «Leo Group Services») - образовательный центр работающая на рынке зарубежных образовательных программ с 2012 года.. Компания предоставляет программы по высшему образованию за рубежом, выбирая наиболее подходящий вариант для желающих отправиться на обучение. Помимо консультационных услуг LeoLeo.me сопровождает студентов на всех этапах: от подачи документов до зачисления в учебное заведение. В центре есть 5 направлений деятельности:

- Образование за рубежом;
- Гранты, стипендии, стажировки;
- Дистанционное образование;
- Языковые курсы;
- Трудоустройство за границей.

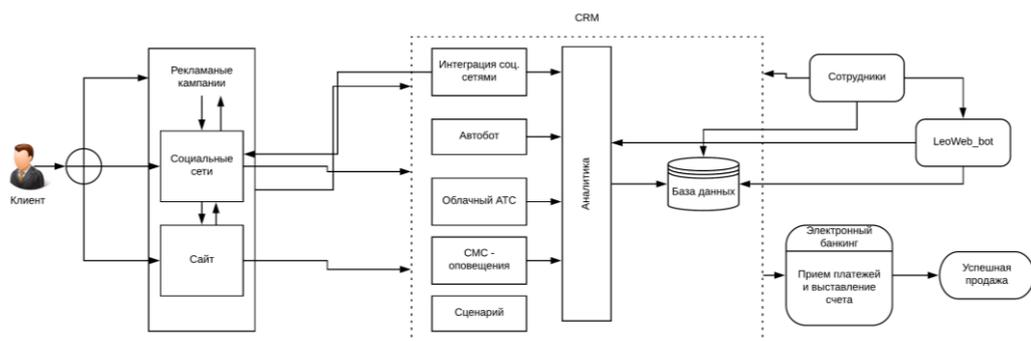


Рисунок 2.1 - Функциональная структура ИС по взаимоотношению с клиентами

## 2.2 Обобщенный алгоритм функционирования систем

Исходя от рисунка 2.1 - функциональной структуры ИС, есть 3 основных этапа взаимодействия клиента:

- алгоритм администрирования сайта и взаимодействия физического клиента с сайтом компании;
- алгоритм администрирования CRM - системы для взаимодействия между физическим клиентом и менеджером отдела продаж;
  - а) алгоритм взаимодействия физического клиента с CRM;
  - б) алгоритм взаимодействия менеджера отдела продаж с физическим клиентом;
- алгоритм приема платежей с выставлением счета.

Алгоритм администрирования сайта и взаимодействия физического клиента с сайтом компании:

Взаимодействие потенциальных клиентов с сайтом компании в основном состоит в создании рекламных кампаний через сервисы онлайн маркетинга Google Adwords, Facebook Ads, Yandex Direct и показа рекламных объявлений, созданных от этих онлайн сервисов. Показ рекламных кампаний осуществляется созданием рекламных объявлений с учетом интересов, запросов, выбором географических и возрастных критериев потенциальных критериев, а также выбором типа и маркетинговой цели для рекламных кампаний, их бюджета и ставки, местом показа и конечного сайта, после клика которого потенциальный клиент оставит заявку. Результатом успешного взаимодействия с клиентом с сайтом, является отправление заявки в CRM – систему.

Алгоритм администрирования CRM - системы для взаимодействия между физическим клиентом и менеджером отдела продаж:

На данном этапе, создается своеобразная база – “воронки”, для хранения данных о клиентах. После создания «воронки», создается интеграция – подключение «воронки» к сайту, по которому клиенты оставляют заявку. Внутри этих воронок создаются разделы - “этапы”, которые будут показывать статус заинтересованности клиентов. При изменении статусов клиентов менеджерами отдела продаж, либо администратором, запускаются автоматизированные процессы.

Алгоритм взаимодействия физического клиента с CRM:

После взаимодействия клиента с сайтом компании и успешной заполнения заявки, заявка в виде сделки создается в системе управления взаимодействиями с клиентами - AmoCRM. Данном этапе взаимодействия, только что появившейся сделке автоматически назначается ответственный на эту сделку – один из менеджеров отдела продаж и приклепляется за ним автоматическая задача до конца дня, оповещающая о том, что нужно проконсультировать клиента. При этапе консультации, дается подробная информация о продукте, заинтересовавшего клиента и выясняется дальнейшая взаимодействие с клиентом, выставляется возможная оценка степени заинтересованности. Исходя от полученной информации от консультации сделка клиента переводится на один из следующий вариативностей этапов воронки.

Алгоритм взаимодействия менеджера отдела продаж с физическим клиентом:

Данный алгоритм показывает этапы взаимодействия сотрудника отдела продаж с клиентом, заявка которой, только что поступила после взаимодействия клиента с сайтом компании.

Алгоритм приема платежей с выставлением счета:

Данный алгоритм является наиболее эффективным способом оплаты за продукт или за услуг для клиентов, находящихся в других городах или странах, для которых визит к компании или прямая оплата платежей является невозможным или проблематичным из-за множества факторов. Электронный банкинг позволяет выставлять индивидуальные для оплаты счета и принимать платежей мультивалютным образом.

### **2.3 Алгоритмы решения функциональных задач**

Алгоритм взаимодействия физического клиента с сайтом компании:

Шаг 0. Начало. Создание рекламных кампании на платформах Google Adwords, Facebook Ads, Yandex Direct. Настройка рекламных кампании и запуск на различных платформах.

Шаг 1. Поиск клиентом продукта компании через поисковые системы

Шаг 2. Показ потенциальному клиенту рекламное объявление по запросу (Ключевые слова)

Шаг 3. Взаимодействие клиента с рекламным объявлением, переход на сайт компании.

Шаг 4. Заполнение клиентом заявки (анкеты) находящийся на посадочной странице сайта компании.

Шаг 5. Конец.

Алгоритм администрирования CRM - системы для взаимодействия между физическим клиентом и менеджером отдела продаж:

Шаг 0. Начало.

Шаг 1. Создание воронки. Создание этапов воронки.

Шаг 2. Интеграция с сайтом.

Шаг 3. Настройка этапов воронки. Добавление автоматизированных процессов.

Шаг 4. Конец.

Алгоритм взаимодействия физического клиента с CRM:

Шаг 0. Начало.

Шаг 1. Оповещение клиенту об успешном заполнении сделки и голосовое приветствие от автобота.

Шаг 2. Получение физическим клиентом консультации от ответственного менеджера отдела продаж за клиента через облачную телефонию.

Если у физического клиента есть возможность встречи с ответственным менеджером, то на Шаг 3, отрицательном случае на Шаг 4.

Шаг 3. Условное заинтересованность от клиента, получение дополнительной информации при встрече с ответственным менеджером в офисе компании и ознакомление с договором.

Шаг 4. Получение клиента электронного либо физического договора через электронную почту/курьера.

Шаг 5. Оплата клиентом выставленного индивидуального счета через физический либо облачный банкинг.

Шаг 6. Конец

Алгоритм взаимодействия менеджера отдела продаж с физическим клиентом:

Шаг 0. Начало.

Шаг 1. Поступление заявки на CRM. Автоматическое назначение ответственного менеджера и создание задачи.

Шаг 2. Звонок клиенту через облачную телефонию.

Шаг 3. Использование менеджером сценария звонка через облачную платформу.

Шаг 4. Выставление статуса клиента:

Шаг 4.1. Статус – “50%”

Шаг 4.2. Статус – “Рано обратившиеся”

Шаг 4.3. Статус – “Недозвон”

Шаг 4.4. Статус – “Условный отказ”

Шаг 4.5. Статус – “Закрыто не реализовано”

Шаг 5. Выставление задач на будущее.

Шаг 6. Приглашение клиента на встречу при помощи облачной АТС, e-mail и смс-оповещения.

Шаг 7. Отправка выставленного счета для оплаты.

Шаг 8. Выставление статуса клиента – “Успешно реализовано”

Шаг 9. Конец.

Алгоритм приема платежей с выставлением счета:

Шаг 0. Начало.

Шаг 1. Авторизация в облачном банке. Вход в личный кабинет.

Шаг 2. Генерация индивидуального счета.

Шаг 3. Заполнение данных клиента.

Шаг 4. Выставление счета клиенту через e-mail, социальные сети.

Шаг 5. Оплата платежа клиентом. Получение клиентом квитанции о платеже. Уведомления облачным банкингом в личном кабинете и e-mail об успешном платеже.

Шаг 6. Конец.

### 3 Информационное и программное обеспечение информационной системы ТОО “Leo Group Services”

#### 3.1 Описание информационного обеспечения

Система управления взаимоотношениями с клиентами – AmoCRM, используемая в ТОО “Leo Group Services” представляет собой реляционную базу данных, где хранятся все данные о клиентах компании. Данные клиента хранятся в двух основных сущностях, взаимосвязанных между собой. Ниже приведены описания и назначения таблиц.

Таблица leads, одна из основных сущностей системы, хранит сведения об индивидуальных сделках с клиентами, составленных администратором CRM – системы, либо сотрудниками отдела продаж. Состоит из предустановленного набора полей и дополнительных, создаваемых администратором аккаунта. Каждая сделка может быть прикреплена к одному и более контакту (contact\_id) или не прикреплена ни к одному. Каждой сделке может быть задан ответственный (responsible\_user\_id) для разграничения прав доступа между сотрудниками аккаунта.

Сделка обладает статусом (status\_id), который обозначает положение сделки в жизненном цикле (бизнес-процесс). Он должен быть обязательно присвоен сделке. Структура таблицы приведена в таблице 3.1.

Таблица 3.1 – Описание таблицы leads

ПОЛЕ	ОПИСАНИЕ
id	уникальный идентификатор сделки
name	название сделки
created_by	идентификатор пользователя создавшего сделку
created_at	время и дата создания
updated_at	время и дата изменения сделки
status_id	идентификатор этапа цифровой воронки, на котором находится данная сделка
responsible_user_id	идентификатор ответственного пользователя
pipeline_id	идентификатор цифровой воронки, в которой находится данная сделка
contact_id	идентификатор контакта, прикрепленной к данной сделке

Таблица contacts, одна из основных сущностей системы, состоит из предустановленного набора полей и дополнительных, создаваемых администратором аккаунта CRM - системы. Каждый контакт может участвовать в одной и более сделке (leads\_id) или может быть вообще не связан ни с одной.

E-mail контакта и телефон используются как уникальные идентификаторы в связке с другими системами. К примеру, именно в события контакта попадает

информация о совершенных звонках, о e-mail-переписке. Каждому контакту может быть задан ответственный для разграничения прав (responsible\_user\_id) доступа между сотрудниками аккаунта. Структура таблицы приведена в таблице 3.2.

Таблица 3.2 – Описание таблицы contacts

ПОЛЕ	ОПИСАНИЕ
id	уникальный идентификатор клиента
name	название контакта
responsible_user_id	идентификатор ответственного пользователя
created_by	идентификатор пользователя создавшего контакт
created_at	время и дата создания
updated_at	время и дата изменения контакта
account_id	идентификатор аккаунта на котором создан контакт
group_id	идентификатор группы в которой состоит пользователь ответственный за данный контакт
leads_id	идентификатор сделки, которая прикреплена к данному контакту

Таблица tasks. Задача должна обязательно иметь ответственного и дату (число и время). Также задача может быть связана со сделкой или контактом, но не обязательно, она может быть не связана ни с каким объектом. Самая основная сущность системы считается задачи. Структура таблицы приведена в таблице 3.3.

Таблица 3.3 – Описание таблицы tasks

ПОЛЕ	ОПИСАНИЕ
id	уникальный идентификатор задачи
responsible_user_id	идентификатор ответственного пользователя
created_at	дата создания данной задачи
updated_at	дата последнего изменения данной задачи
account_id	идентификатор аккаунта
group_id	идентификатор группы в которой состоит пользователь имеющей отношение к задаче
element_type	тип привязываемого элемента (1 - контакт, 2- сделка, 3 - компания, 12 - покупатель)
is_completed	задача завершена или нет
task_type	тип задачи
complete_till_at	дата до которой необходимо завершить задачу

Специальная программа – бот, выполняющая действия (авторизация, отправка отчетности) автоматически и/или по определенному промежутку времени под названием “LeoWeb\_bot” представляет собой взаимодействие, удаленный доступ к системе управления взаимоотношениями с клиентами компании через программный интерфейс приложения (API), предоставленной самой CRM-системой.

Источником информации в данном ИС является реляционная база данных CRM – системы - AmoCRM. Прием информации, хранение и обработка информации, вывод информации выполняются через аппаратную и программную часть ИС (Системное, инструментальное, прикладное программное обеспечение). Потребителем информации является пользователь – от сотрудника до директора компании.

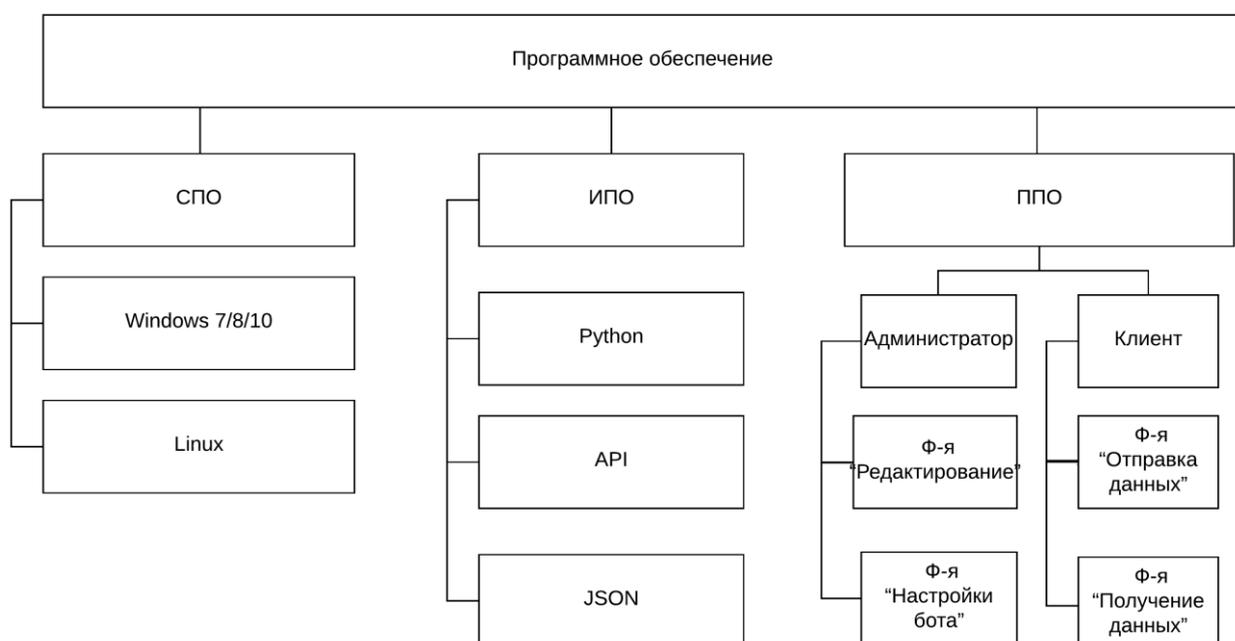


Рисунок 3.1 - Структура программного обеспечения

### 3.2 Структура программного обеспечения

Структура программного обеспечения представлена на рисунке 3.1. Она состоит из:

- Системное программное обеспечение;
- Инструментальное программное обеспечение;
- Прикладное программное обеспечение.

#### 3.2.1 Системное программное обеспечение

Специальная программа – бот, выполняющая действия (авторизация, отправка отчетности) автоматически и/или по определенному промежутку времени под названием “LeoWeb\_bot” реализована для рабочей станции Windows 7/8/10 и для других операционных систем macOS, Linux, Android, iOS,

Windows Phone, так как “LeoWeb\_bot” реализована через кроссплатформенный мессенджер Telegram. Интерфейс и платформа данного бота находится внутри мессенджера. Установлен данный бот на облачном сервере компании на базе ОС Linux.

### 3.2.2 Инструментальное программное обеспечение

Программа-бот реализована на языке программирования python, используемого на стороне веб-приложения – бота, для соединения с API CRM – системы, с использованием запросов GET, POST.

Python - это интерпретируемый объектно-ориентированный язык программирования высокого уровня с динамической семантикой. Его встроенные структуры данных высокого уровня в сочетании с динамической типизацией и динамическим связыванием делают его очень привлекательным для быстрой разработки приложений, а также для использования в качестве скриптового или связующего языка для соединения существующих компонентов. Простой, легкий в освоении синтаксис Python подчеркивает удобочитаемость и, следовательно, снижает стоимость обслуживания программы. Python поддерживает модули и пакеты, что способствует модульности программы и повторному использованию кода. Интерпретатор Python и обширная стандартная библиотека доступны в исходном или двоичном виде для всех основных платформ и могут свободно распространяться.

Интерфейс прикладного программирования (API) обеспечивает абстракцию для проблемы и определяет, как клиенты должны взаимодействовать с программными компонентами, которые реализуют решение этой проблемы. Сами компоненты обычно распространяются в виде библиотеки программного обеспечения, что позволяет использовать их в различных приложениях. По сути, API-интерфейсы определяют многократно используемые строительные блоки, которые позволяют включать функциональные модули в приложения конечных пользователей.

API можно написать для себя, для других инженеров в вашей организации или для сообщества разработчиков в целом. Это может быть всего одна функция или несколько сотен классов, методов, свободных функций, типов данных, перечислений и констант. Его реализация может быть с открытым исходным кодом. Важной базовой концепцией является то, что API - это четко определенный интерфейс, который предоставляет конкретную услугу другим частям программного обеспечения.

Современное приложение, как правило, строится поверх многих API, некоторые из которых также могут зависеть от других API. Например, приложение для просмотра изображений может использовать API для загрузки изображений GIF, и этот API сам может быть построен на API более низкого уровня для сжатия и распаковки данных. API веб-службы используют формат JSON, чтобы сделать данные доступными для сторонних приложений.

JSON стал выбором по умолчанию для форматирования данных для использования в интерактивных веб-приложениях. Он имеет очень легкий синтаксис и его очень легко изучать, читать и писать.

JSON является выбором по умолчанию для передачи данных в Интернете. Он превзошел форматы, такие как XML и CSV, благодаря своей простоте и независимости от языка.

### 3.2.3 Прикладное программное обеспечение

Программа – бот “LeoWeb\_bot” на стороне администратора включает в себя список функции, настроек от самого программы – бота мессенджера Telegram, BotFather:

Функция “Редактирование”. Данная функция позволяет редактировать данные о боте. Функции вызываются написанием определенных команд внутри бота мессенджера Telegram, BotFather.

Команды:

- setname – редактирование имени бота;
- setdescription - редактирование описания бота;
- setabouttext - редактирование информации о боте;
- setuserpic - редактирование фотографии профиля бота;
- setcommands - редактирование список команд;
- deletebot - удаление бота;

Функция “Настройки бота”: Данная функция позволяет настраивать конфиденциальность, запросы и другие данные бота:

Команды:

- token – генерация токена авторизации;
- revoke - отозвать токен доступа;
- setinline - включение встроенного режима бота;
- setinlinegeo - переключение встроенных запросов о местоположения;
- setinlinefeedback - изменение настроек обратной связи;
- setjoingroups – настройки приватности бота;
- setprivacy - переключение режима конфиденциальности в группах мессенджера Telegram.

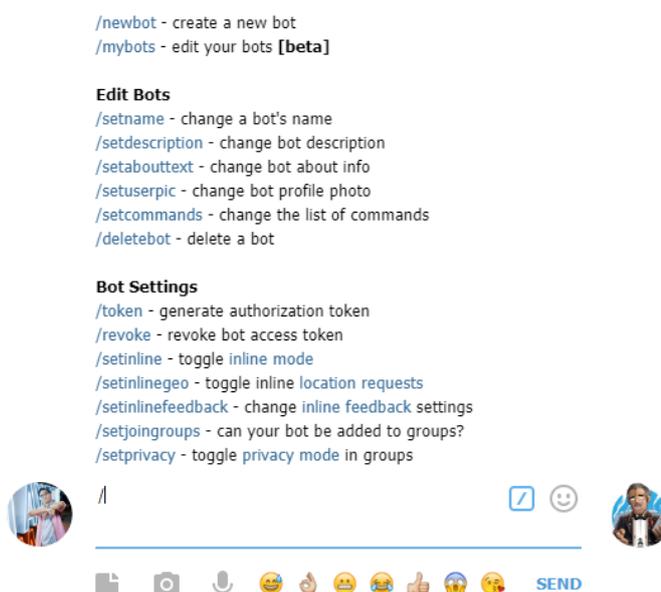


Рисунок 3.2 – Интерфейс администрирования

Интерфейс прикладного программирования (API) обеспечивает абстракцию для проблемы и определяет, как клиенты должны взаимодействовать с программными компонентами, которые реализуют решение этой проблемы. Сами компоненты обычно распространяются в виде библиотеки программного обеспечения, что позволяет использовать их в различных приложениях. По сути, API-интерфейсы определяют многократно используемые строительные блоки, которые позволяют включать функциональные модули в приложения конечных пользователей.

API можно написать для себя, для других инженеров в вашей организации или для сообщества разработчиков в целом. Это может быть всего одна функция или несколько сотен классов, методов, свободных функций, типов данных, перечислений и констант. Его реализация может быть с открытым исходным кодом. Важной базовой концепцией является то, что API - это четко определенный интерфейс, который предоставляет конкретную услугу другим частям программного обеспечения.

Программа – бот “LeoWeb\_bot” на стороне клиента включает в себя:

Функция “Отправка данных”. Эта функция позволяет авторизоваться к CRM – системе в удаленном формате.

Функция “Получение данных”. Эта функция позволяет просматривать и получать данные о задачах, сделках, заявках внутри CRM – системы.

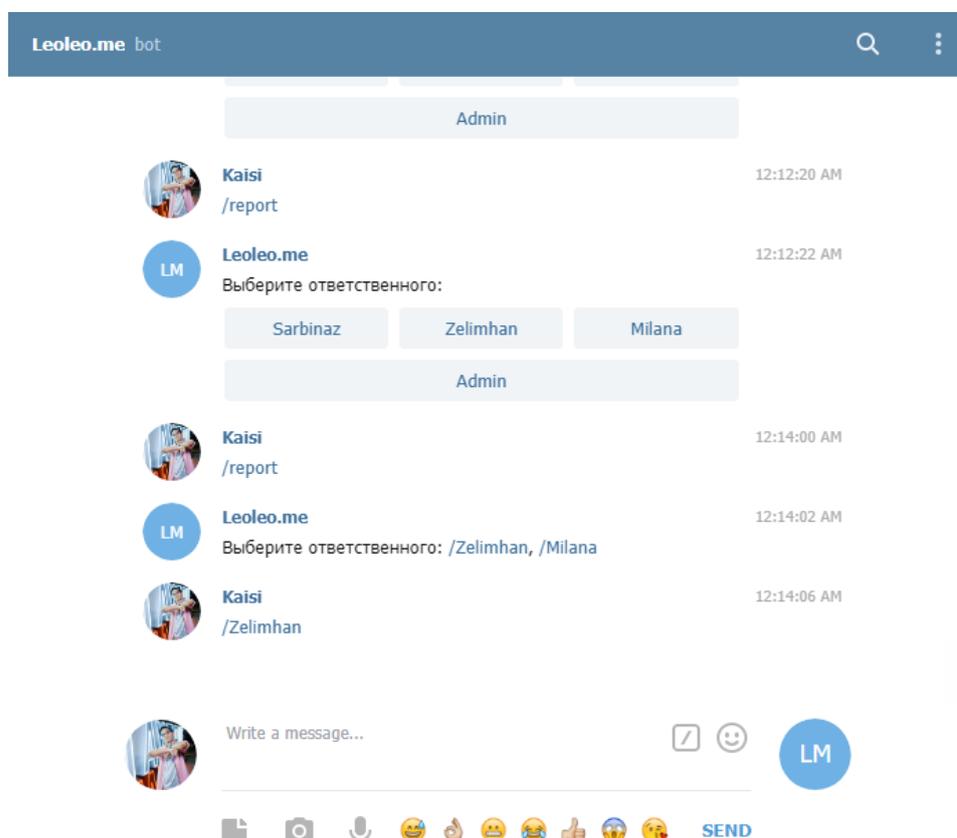


Рисунок 3.3 – Интерфейс пользователя

## 4 Реализация и внедрение

### 4.1 Описание технической и программной реализации

Специальная программа – бот, выполняющая действия (авторизация, отправка отчетности) автоматически и/или по определенному промежутке времени под названием “LeoWeb\_bot” реализована на интегрированной среде разработки PyCharm, для языка программирования Python.

PyCharm – это кроссплатформенная интегрированная среда разработки языка программирования Python, с доступными версиями для Windows, macOS и Linux. PyCharm поддерживает версии 2.x и 3.x поколения языка программирования Python. В зависимости приобретения версии лицензии, данный IDE (Integrated Development Environment - интегрированная среда разработки) имеет такие функции:

- Интеллектуальный редактор Python
- Графический отладчик и тестовый бегун
- Навигация и рефакторинг
- Проверка кода
- Научные инструменты
- Веб-разработка
- Веб-фреймворки Python
- Возможности удаленной разработки
- Доступ к базам данных и поддержка SQL

Разработка и отладка данного дипломного проекта осуществлены такими техническими средствами, как:

- процессор Intel Core i5 2500MHz;
- объем оперативной памяти RAM 8192 Mb;
- жесткий диск объемом 1000 Gb;
- сетевой адаптер Intel wireless-AC9560;
- видеоадаптер NVIDIA GeForce GTX 950M;
- стандартные клавиатура и мышь.

Файл main.py является первоначальным файлом, с которого начинается работа функций бота “LeoWeb\_bot”. Следующий файл Auth.py, запускает функцию авторизации пользователя. Файл tasks.py позволяет при введении команды пользователем предоставить отчетность о задачах. Файл nowtime.py позволяет при подключении системе, подгружать локальное время для корректного вывода отчетности. Описание файла main.py с привязкой к строкам текста имеет следующий вид:

- 1 - 10 подключение модулей;
- 2 – 16 подключение (активизация) файлов к общему модулю.

Описание файла Auth.py с привязкой к строкам текста имеет следующий вид:

- 1 – 3 подключение модулей;
- 5 – 7 переменные для выполнения авторизации;
- 9 – 11 переменная для хранения данных, отправленный веб-сервером AmoCRM;
- 13 – 25 функция для реализации авторизации личному кабинету AmoCRM;
- 26 – 27 вывод данных, при успешной авторизации;

Описание файла Tasks.py с привязкой к строкам текста имеет следующий вид:

- 1-8 подключение модулей;
- 9-11 функция для хранения данных, отправленный веб-сервером AmoCRM на отдельный файл формата JSON;
- 13-15 переменные – параметры, необходимые для запроса к веб-серверу;
- 17-18 переменная;
- 20-22 обращение к ссылке;
- 26-37 функция – сортировка полученных данных от веб-сервера AmoCRM;
- 39-41 чтение данных, сохраненном на отдельном файле формата JSON;
- 43-52 сортировка и вывод данных пользователю.

Описание файла Nowtime.py с привязкой к строкам текста имеет следующий вид:

- 1-3 подключение модулей;
- 4-8 вывод локального времени.

## **4.2 Перспективы развития**

Перспективами развития специальной программы – бота состоит в дальнейшем усовершенствовании функциональной части, в частности использование всех доступных запросов от интерфейса прикладного приложения (API) от CRM – системы; добавление многофункциональности – интеграция бота со сайтом компании, возможность взаимодействия с клиентами; усовершенствование использования интерфейса прикладного приложения для последующих новых реализации ботов с интеграцией другими возможными сервисами;

Отдельной от усовершенствовании программы, перспективой развития имеется и в коммерческих целях. На рынке, каталогах интеграции CRM – систем, ассортимент идентичных программ не большой, а спрос со временем только возрастает.

## ЗАКЛЮЧЕНИЕ

При входе обзора и исследования облачных технологии было выяснено применение облачных решений в информационных системах предприятия. А также их характеристики, типы, различных видов моделей по типу развертывания, использования, возможные проблемы использования.

Первой главе было рассмотрено процессы облачных технологии и систем управления с взаимоотношениями с клиентами, используемых в компаниях в целом.

На второй главе исходя от полученных данных на первой главе и включая практические знания, полученные входе работы на ТОО “Leo Group Services”, были решены и внедрены в данной организации данные задачи:

- Разработана функциональная структура системы;
- Разработана обобщенный алгоритм функционирования системы по взаимоотношению с клиентами с облачными решениями;
- Предложены алгоритмы решения функциональных задач;

На третьей и на четвертой главе было показано реализация программы бота, для удаленного доступа к системе управления взаимодействиями с клиентами. Тем самым были решены такие задачи, как:

- Разработана архитектура информационного и программного обеспечения системы;
- Выбран комплекс технических средств для реализации системы.
- Произведено внедрение системы в структуру функционирования ТОО “Leo Group Services”.

## ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

CRM - (англ. Customer relationship management) Система управления взаимоотношениями с клиентами.

HR - (англ. Human Resources) Управление персоналом.

ERP - (англ. Enterprise Resource Planning), планирование ресурсов предприятия.

API - (англ. Application Programming Interface), интерфейс создания приложений.

## Список использованной литературы

- 1 Риз Д. - «Облачные вычисления (Cloud Application Architectures)», 2011.
- 2 Peter Mell, Timothy Grance - «The NIST Definition of Cloud Computing », 2012.
- 3 А.Г. Ерохин, Е.А. Фролова - «Применение облачных технологий в компаниях малого и среднего бизнеса», 2015. [стр 61-63]
- 4 О.С. Чаплыгина, В. В. Фролов - «Облачные CRM – системы, том 2» 2017г.
- 5 Коряковский Андрей Валерьевич - «Перспективные направления развития информационных систем предприятий», 2016.
- 6 Носов А. Л. « Информационные системы в управлении организацией // Научно-методический электронный журнал «Концепт» ». – 2016. – № 6
- 7 Рамиль Кинзябулатов - «CRM. Подробно и по делу», 2018.
- 8 Мартин Редди, «API Design for C++ », 2011.
- 9 Рейтц Кеннет, Шлюссер Тан - « Автостопом по Python », 2017.
- 10 Daniel Jacobson, Dan Woods, Greg Brail - «APIs: A Strategy Guide », 2011 .

## Приложение А

### Текст программного кода

```
Файл main.py:
from flask import Flask
from flask import request
from flask import jsonify
from App.Auth import auth, amo_domain, amo_user, amo_key, state
import requests
import json
import re
import telebot
from App.Tasks import j

TOKEN = '73317175603:AAFuN-PqjvuO5hVk34l219a4YSrRIgOKzfA'
bot = telebot.TeleBot(TOKEN)
app = Flask(__name__)
URL='https://api.telegram.org/bot73317175603:AAFuN-
PqjvuO5hVk34l219a4YSrRIgOKzfA/'
def write_json(data, filename='answer.json'):
    with open(filename, 'w') as f:
        json.dump(data, f, indent=2, ensure_ascii=False)
def get_updates():
    url = URL + 'getUpdates'
    r = requests.get(url)
    return r.json()

def send_message(chat_id, text=""):
    url = URL + 'sendMessage'
    answer = {'chat_id': chat_id, 'text': text}
    r = requests.post(url, json=answer)
    return r.json()
auth_result = auth(amo_user, amo_key)
success_num = 24167767
amo_id = 1387123
res = requests.get(amo_domain + '/api/v2/leads', cookies=state['cookies'],
params={'&responsible_user_id': amo_id, 'status': success_num},)
data1=res.json()
orig_deals = data1['_embedded']['items']

def format_deal(deal):
```

## Продолжение приложения А

```
    return {
        'id': deal['id']
    }
deal = format_deal(orig_deals[0])
def parse_text(text):
    pattern = r'\w+'
    user = re.search(pattern, text)
    return user

@app.route('/', methods=['POST', 'GET'])
def index():
    if request.method == 'POST':
        r = request.get_json()
        chat_id = r['message']['chat']['id']
        message = r['message']['text']

        if 'report' in message:
            send_message(chat_id, "Выберите ответственного: /Zelimhan,
/Milana, /Sarbinaz")
            if 'Zelimhan' in message:
                rud = 1387123
                send_message(chat_id, "Просроченные задачи у Зелимхана на
данный момент: " + str(j))
            if 'Milana' in message:
                rud = 1335123
                send_message(chat_id, "Просроченные задачи у Миланы на
данный момент: " + str(j))
            if 'Sarbinaz' in message:
                rud = 1332123
                send_message(chat_id, "Просроченные задачи у Миланы на
данный момент: " + str(j))

        if 'amocrm' in message:
            send_message(chat_id, text=deal['id'])
        #write_json(r)
        return jsonify(r)

    return '<h1>Hello!</h1>'

def main():
    pass
```

## Продолжение приложения А

```
if __name__ == '__main__':  
    app.run()
```

Файл Auth.py:

```
from pprint import pprint  
import requests  
import arrow  
  
amo_domain = 'https://leostudy.amocrm.ru' # логин AmoCRM  
amo_user = 'arai.n@mail.ru' # ключ API  
amo_key = 'bcaa755bb2333f94c0df3899821'  
  
state = {  
    'cookies': None  
}  
result = 'Авторизация прошла успешно!  
def auth(user, user_hash):  
    url = amo_domain + '/private/api/auth.php'  
    data = {  
        'USER_LOGIN': user,  
        'USER_HASH': user_hash  
    }  
    res = requests.post(url, data=data, params={'type': 'json'})  
    #print('status code', res.status_code)  
    if res.status_code == 200:  
        state['cookies'] = res.cookies  
        return result  
auth_result = auth(amo_user, amo_key)  
pprint(auth_result)
```

Файл tasks.py:

```
from App.Auth import auth, amo_user, amo_key, amo_domain, state  
import requests  
import pprint  
from App.nowtime import unixtime  
import json  
from App.main import  
from datetime import datetime  
import time
```

## Продолжение приложения А

```
def write_json(data, filename='amocrm.json'):
    with open(filename, 'w') as f:
        json.dump(data, f, indent=2, ensure_ascii=False)

auth_result = auth(amo_user, amo_key)
amo_id = 1179549
limit_rows = 50

dt = datetime.strptime('02.05.2019 00:00:00', '%d.%m.%Y %H:%M:%S')
date_create = int(time.mktime(dt.timetuple()))

res1 = requests.get(amo_domain + '/api/v2/tasks{ }', cookies=state['cookies'],
params={'&responsible_user_id':date_create,      'limit_rows':limit_rows,'type':'lead',
'filter[date_modify][from]': 1556794800},)
data = res1.json()
print(data)

orig_deals = data['_embedded']['items']

def format_deal(deal):
    date = (deal['complete_till_at'])

    return {
        #id: deal['id'],
        'is_completed': deal ['is_completed'],
        'element_id': deal['element_id'],
        'complete_till_at': date
    }

deals = [format_deal(d) for d in data['_embedded']['items']]
write_json(deals)

with open('amocrm.json') as f:
    pf = json.load(f)
print(len(pf))

i = 0
j=0

while i <= (len(pf)-1):
    if pf[i]['complete_till_at'] < unixtime and pf[i]['is_completed'] == 0:
        j=j+1
```

## Продолжение приложения А

```
    i=i+1
else:
    i=i+1
print(j)
```

Файл nowtime.py:

```
from datetime import datetime
import time
import pytz
tz = pytz.timezone('Asia/Almaty')
almaty_now = datetime.now(tz)
print(almaty_now)
unixtime = time.mktime(almaty_now.timetuple())
print(int(unixtime))
```