

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**(ФГБОУ ВО «АмГУ»)**

Факультет математики и информатики  
Кафедра информационных и управляющих систем  
Направление подготовки 09.03.01 – Информатика и вычислительная техника  
Направленность (профиль) образовательной программы Автоматизированные  
системы обработки информации и управления

ДОПУСТИТЬ К ЗАЩИТЕ  
Зав. Кафедрой  
\_\_\_\_\_ А.В. Бушманов  
« \_\_\_\_\_ » \_\_\_\_\_ 2022г

**БАКАЛАВРСКАЯ РАБОТА**

на тему: Разработка чат-бота для работы с обращениями граждан в МФЦ на  
основе машинного обучения

Исполнитель  
студент группы 853об

\_\_\_\_\_  
(подпись, дата)

А.М. Кочарян

Руководитель  
доцент, канд. физ.-мат. наук

\_\_\_\_\_  
(подпись, дата)

В.В. Еремина

Консультант по безопасности  
и экологичности  
доцент, канд. техн. наук

\_\_\_\_\_  
(подпись, дата)

А.Б. Булгаков

Нормоконтроль  
инженер кафедры

\_\_\_\_\_  
(подпись, дата)

В.Н. Адаменко

Благовещенск 2022



## РЕФЕРАТ

Бакалаврская работа содержит 70 с., 17 рисунков, 6 таблиц, 3 приложения, 21 источник.

### БАЗА ДАННЫХ, МАШИННОЕ ОБУЧЕНИЕ, ИНФОРМАЦИОННАЯ СИСТЕМА, АРІ, ЧАТ-БОТ, НЕЙРОННЫЕ СЕТИ

Объектом исследования выступает Многофункциональный центр предоставления государственных и муниципальных услуг. Цель работы состоит в проектировании информационной системы с чат-ботом в качестве интерфейса для интерактивной коммуникации с гражданами.

Для достижения поставленной цели в работе используются исследование предметной области и сбор данных для выявления высоконагруженных отделов учреждения и недостатков классического подхода обслуживания. На основе полученных с помощью этих методов данных становится возможным проектирование структуры системы и создание базы данных с использованием свободно-реляционной системой управления базами данных PostgreSQL. Далее происходит разработка программного обеспечения с использованием и фреймворка для разработки разговорного искусственного интеллекта с открытым исходным кодом Deepravlov и механизма Telegram Bot API, позволяющего использовать Telegram-мессенджеры для взаимодействия с пользователем.

Результатом выполнения данной работы является программный продукт чат-бот для работы с обращениями граждан Многофункционального центра предоставления государственных и муниципальных услуг и сопровождающая база данных системы.

## НОРМАТИВНЫЕ СЫЛКИ

В настоящей бакалаврской работе использованы ссылки на следующие стандарты и нормативные документы:

ГОСТ 2.105-95 ЕСКД Нормоконтроль

ГОСТ 19.004-80. ЕСПД Термины и определения

ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам

ГОСТ 2.052 2015 Единая система конструкторской документации. Электронная модель изделия. Общие положения

ГОСТ 7.32-91 (ИСО 5966-82) Отчет о научно-исследовательской работе. Структура и правила оформления.

ГОСТ Р 50948-2001 Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности

ГОСТ Р 57589-2017 Аддитивные технологические процессы. Базовые принципы. Часть 2. Материалы для аддитивных технологических процессов. Общие требования

ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация.

ГОСТ 12.1.007-76 Система стандартов безопасности труда (ССБТ). Вредные вещества. Классификация и общие требования безопасности

ГОСТ Р ИСО 1503-2014 Эргономика. Требования к пространственной ориентации и направлениям движения органов управления

СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Чат-бот – компьютерная программа, обеспечивающая интерактивное взаимодействие с пользователем для выявления их потребностей для дальнейшего их удовлетворения;

API – специализированный протокол, служащий для описания взаимодействия компьютерных программ друг с другом;

Мессенджер – программное обеспечение для мгновенной передачи текстовых сообщений, фото и видео;

Нейронная сеть – математическая модель сетей нервных клеток живого организма;

RSA – криптографический алгоритм с открытым ключом, который базируется на факторизации простых чисел;

Искусственный интеллект – системы, выполняющие задачи, традиционно считающиеся творческими для человека

## Содержание

Введение	8
1 Анализ предметной области	11
1.1 Цели и задачи организации МФЦ	11
1.2 Использование виртуального ассистента в качестве интерфейса информационной системы	13
1.3 Машинное обучение и компьютерная лингвистика	14
2 Угрозы информационной безопасности приложения	17
2.1 Встроенная защита Telegram и криптографический протокол MTProto	17
2.2 Сбор персональных данных	18
3 Обучение нейронной сети	20
4 Разработка информационного продукта	25
3.1 Выбор средств разработки	25
3.1.1 Библиотеки для разработки нейронных сетей	25
3.1.2 Использование стороннего API	27
3.1.3 СУБД	30
3.1.4 Сбор актуальной информации	32
3.2 Проектирование информационной системы	33
3.3 Разработка БД	35
3.3.1 Инфологическое проектирование БД	35
3.3.2 Логическое и физическое проектирование БД	38
3.3.3 Создание БД	39
3.4 Разработка информационного продукта	39
3.4.1 Элементы теории конечных автоматов	39

3.4.2 Форма регистрации	40
3.4.3 Определение услуги по обращению	42
3.4.4. Вопросно-ответная система	44
3.4.5 Описание работы программы	46
5 Безопасность и экологичность	48
5.1 Безопасность	48
5.1.1 Анализ потенциальных опасностей	48
5.1.2 Мероприятия по обеспечению безопасности	48
5.2 Экологичность	51
5.3 Чрезвычайные ситуации	53
Заключение	56
Библиографический список	57
Приложение А	59
Приложение Б	64
Приложение В	66

## ВВЕДЕНИЕ

По данным Министерства экономического развития Российской Федерации активность запросов в электронном виде на получение государственных услуг увеличилось на 500 тыс. в неделю. Обращения граждан преимущественно однотипные, поэтому рассматривается вариант автоматизации формирования пользовательского сценария предоставления услуг.

Также данная проблема поднималась в рамках Международного конкурса по решению глобальных социальных задач с помощью AI & Data, согласно которому, по данным Платформы обратной связи и Инцидент-менеджмента растет число однотипных вопросов по справочной информации, а увеличение постов в социальных сетях не позволяют пользователю прозрачно находить нужную информацию.

В данной работе будет представлена разработка чат-бота для работы с обращениями граждан в МФЦ на основе машинного обучения.

Разработка программ для выявления потребностей пользователей и их удовлетворения началась еще в середине двадцатого века. Разработчики уже тогда видели перспективу в автоматическом общении с пользователем. Преимущество такого взаимодействия организации и пользователя состоит в упрощении процессов получения, обработки и передачи актуальной информации в оперативные сроки. Тогда же, английский математик Алан Тьюринг предложил идею эмпирического теста, который должен пройти идеальный чат-бот, обладающий искусственным интеллектом. Интерпретация данного теста, опубликованного в статье «Вычислительные машины и разум» в журнале Mind в 1950 году, звучит следующим образом: «Человек взаимодействует с одним компьютером и одним человеком. На основании ответов на вопросы он должен определить, с кем он разговаривает: с человеком или компьютерной программой. Задача компьютерной программы – ввести в заблуждение человека, заставив сделать неверный выбор».

На данный момент, чат-бот представляет собой информационный продукт, работающий в мессенджере, при помощи которого пользователь получает актуальную информацию по соответствующему вопросу в рамках предоставляемых ботом услуг.

Сфера разработки чат-ботов тесно связана с обработкой естественного языка (NLP), поэтому данные виртуальные собеседники образуют класс программ с естественного-языковым интерфейсом. NLP решает широкий круг задач по обработке текстов на естественном языке, и на данный момент этот раздел искусственного интеллекта и компьютерной лингвистики произвел революцию в задачах машинного перевода, классификации текстов, извлечении именованных сущностей (NER), вопросно-ответных системах, аннотировании текста. Перечисленные системы можно использовать в разработке виртуального собеседника.

В рамках данной работы предполагается разработать информационную систему с чат-ботом в качестве интерфейса для интерактивной коммуникации с гражданами, нуждающимися в услугах МФЦ, предоставляющую в любое время суток:

- Возможность регистрации в системе и безопасного взаимодействия;
- Удобный доступ к актуальным новостным лентам организации;
- Обратную связь и оповещения о записи на прием;
- Автоматическое определение требуемой услуги по обращению с помощью технологий машинного обучения;
- Запись на услугу в выбранное отделение с визуализацией интерактивной карты маршрута с учетом пробок;
- Возможность находить ответы на задаваемые пользователем вопросы в новостях и документах организации с помощью технологий глубинного обучения.

Предполагается, что внедрение в организацию системы с естественного-языковым интерфейсом, способной выполнять перечисленные выше услуги,

позволит снизить нагрузку на операторов кол-центра и повысить качество их работы без дополнительного найма персонала. Также такая система позволит повысить скорость предоставления услуг.

Целевая аудитория данного продукта – пользователи услуг МФЦ, имеющие доступ к мессенджеру Telegram. В дальнейшем планируется перенести логику приложения в социальные сети, такие как, например, VK и Одноклассники, либо интегрировать с сайтом МФЦ.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Цели и задачи организации МФЦ

В 2005 году в Российской Федерации была принята Концепция административной реформы качественного изменения системы предоставления государственных услуг населению, декларирующая принципы полной прозрачности операций, упорядочивания процесса посредством электронной очереди, экспертной консультации. В конце 2008 года в стране было запущено уже более двадцати многофункциональных центров предоставления услуг.

Многофункциональный центр – это организация, предоставляющая государственные и муниципальные услуги по технологии «одного окна», таким образом концентрируя множество услуг в одном месте, что снижает время вынужденного обращения граждан. Типичными услугами, предоставляемыми МФЦ, являются оформление паспорта, регистрация брака, оплата сборов и пошлин, предусмотренных законодательством и т. д.

Один из ключевых событий информатизации данной сферы – это разработка интернет-портала «Портал государственных услуг Российской Федерации».

В основе МФЦ лежит единая автоматизированная система, ориентированная на процесс, построенная на принципах безопасности информации и защиты личных пользовательских данных.

При описании предметной области требуется рассмотреть и построить организационную структуру МФЦ, которая изображена на рисунке А.1. Данная схема показывает, как организация устроена и представляет собой простую линейную структуру, характеризующуюся наличием определенной цепи инстанций. Из схемы видно, что во главе МФЦ стоит директор, которому подчиняются руководители бухгалтерии, отдела приема и обработки документов, отдела информационных технологий, отдела по правовым и организационным вопросам.

Для дальнейшего выполнения анализа и проектирования необходимо построить внешний и внутренний документооборот.

Диаграмма внешнего документооборота представляет собой контекстную диаграмму, построенную в нотации DFD.

Внешний документооборот представлен на рисунке А.2. Глядя на диаграмму внешнего документооборота, можно сделать вывод о том, что МФЦ связан с несколькими внешними сущностями:

- Заявитель;
- Министерство экономического развития РФ;
- IT-компания;
- Региональный орган исполнительной власти или Орган местного самоуправления.

МФЦ получает приказы от Министерства экономического развития Российской Федерации. Заявитель обращается в МФЦ и делает запрос на услугу. МФЦ, в свою очередь, делает запрос на выполнение услуги в Региональный орган исполнительной власти и Органы местного самоуправления. После выполнения услуги МФЦ отправляет Заявителю отчет о выполнении услуги. IT-компания поставляет ПО в МФЦ, а также участвует в поддержке информационной структуры организации.

При описании внутреннего документооборота были описаны основные функции каждого подразделения организации и рабочего места, а также охарактеризованы хранилища данных и основные документы, циркулирующие внутри предприятия. Внутренний документооборот представлен на рисунке А.3.

Запросы от заявителей через различные каналы связи собираются в хранилище, после чего информация о консультации перенаправляется в Отдел обслуживания и Центр телефонных обращений. После происходит запись на прием через Терминал очереди и заявителю выдается талон. Из схемы видно, что основная нагрузка приходится на Отдел обслуживания и Центр телефон-

ных обращений, которые занимаются обработкой запросов заявителей. Помимо этого, зачастую определение услуги требует отдельной консультации с сотрудником МФЦ. Вопрос о содержании данных отделов, призванных бесперебойно и качественно осуществлять прием и обработку вопросов граждан с помощью различных каналов связи, стоит наиболее остро. Как показал анализ, затраты на содержание данных штатных единиц будут возрастать, и это без учета того, что call-центр и отдел обслуживания традиционно считаются самыми многочисленными подразделениями в организации. Назначение разрабатываемой системы – это автоматизация труда сотрудников данных отделов.

## **1.2 Использование виртуального ассистента в качестве интерфейса информационной системы**

Для создания чат-ботов используется технология API (Application Programming Interface) либо REST API (Representational State Transfer), который разрабатывается со стороны мессенджеров и социальных сетей. Преимущества чат-ботов состоит в следующем:

- Круглосуточное сервисное обслуживание: данный показатель повышает качество обслуживания и успешность бизнеса;
- Эффективное взаимодействие с клиентами: чат-боты не перегружают аудиторию информацией и предоставляют релевантную информацию по запросам пользователей;
- Экономность: чат-боты являются одноразовой инвестицией, способствующая сокращению траты на персонал;
- Легкость в эксплуатации.

Однако следует отметить, что функционал бота, работающего на основе инструкций, довольно ограничен и зачастую он предназначен для ответа на фиксированный вопрос. Однако многие отрасли используют чат-ботов для оптимизации операционных процессов и обслуживания в организации.

### **1.3 Машинное обучение и компьютерная лингвистика**

Машинное обучение можно представить областью науки, изучающей способы извлечения закономерностей из ограниченного числа примеров. Система машинного обучения не программируется в явном виде. На вход система получает многочисленные объекты, которые имеют отношение к решаемой задаче, преобразовывая данные в значимые результаты. Таким образом происходит обучение представлению исходных данных, максимально приближающему выход к ожидаемому результату. Раздел машинного обучения, характеризующийся многослойностью в представлении данных, называется глубоким обучением. В данном случае происходит автоматический анализ последовательных уровней все более значимых результатов. Данные многослойные представления изучаются с применением моделей – нейронных сетей. Следует отметить, что область глубоких нейронных сетей автоматизировала выбор признаков, значимых для решения задачи, из множества доступных во время обучения. Данную задачу до этого решал исследователь.

В общем случае существует три категории машинного обучения:

– Обучение с учителем: машина учится по выбранным данным; обычно, данные отмечаются людьми, используется для решения двух типов задач – регрессии (прогнозирование непрерывных значений, например, стоимость дома или погоду на улице в градусах) и классификации (прогнозирование дискретные значения, например классификация что-либо по категориям);

– Обучение без учителя: обнаружение закономерностей в данных и их группировка, существует два типа задач – кластеризации и поиска ассоциативных правил;

– Обучение с подкреплением: требует использования системы вознаграждения/штрафа, причем модель не имеет информации о системе, но при этом может производить некие действия, влияющие на систему.

Бурный рост доступной текстовой информации, связанный с появлением сети Интернет, а также развитие технологий машинного обучения, спо-

способствовали прогрессу в области автоматической обработки текстов (NLP) и компьютерной лингвистики. Перед построением алгоритма, текстовые данные необходимо соответствующим образом изменить. Для этого используется стандартная последовательность стадий преобразования текста, называемая пайплайном:

- Привести текст к единому регистру;
- Провести очистку текста от лишних символов (знаки пунктуации, не очень значимые по смысловой нагрузке слова, html/xml-разметка и т. д.);
- Провести токенизацию текста, то есть разбить монолитный блок текста на атомарные составные части: абзацы, предложения, слова, n-граммы символов;
- Провести разметку слов по частям речи;
- Привести слова в тексте к нормальной форме: лемматизация (приведение слова к словарной форме), стемминг (нахождение основы слова);
- Провести процесс преобразования текста в цифровое представление – векторизацию, с использованием одной или комбинации из нескольких существующих моделей: частотной, тематической, дистрибутивной.

На данный момент, NLP задачи, такие как распознавание речи, информационный поиск, машинный перевод, генерация текста и т. д., используют следующие архитектуры глубоких нейронных сетей:

- LSTM (Long short-term memory) – рекуррентные нейронные сети с долгой краткосрочной памятью;
- GRU (Gated Recurrent Units) – еще одна модификация рекуррентных нейронных сетей на основе механизма вентиляей;
- ELMO – нейронная сеть на основе LSTM, но она сочетает две параллельные LSTM, одна из которых идет слева направо, а вторая – справа налево
- Transformer – архитектура, основанная на encoder части, состоящий как из RNN, так и из CNN-слоев и decoder части, состоящий из RNN-слоев, а также attention-слоев, которые позволяет каждому входному вектору слова взаимо-

действовать с векторами других слов;

– BERT (Bidirectional Encoder Representations from Transformers), основанная на идеях ELMO и Transformer, а также на идее обучения с подкреплением: обучаясь, она решает задачу предсказания части замаскированных случайным образом слов.

## 2 УГРОЗЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИЛОЖЕНИЯ

### 2.1 Встроенная защита Telegram и криптографический протокол MTProto

Telegram позиционируется как один из наиболее безопасных мессенджеров, который невозможно взломать, а данные пользователей надежно защищены с помощью передовых методов шифрования.

В системе обмена сообщениями Telegram для шифрования используется криптографический протокол MTProto. В основе протокола лежит оригинальная комбинация симметричного алгоритма шифрования AES (в режиме IGE), протокола Диффи-Хеллмана для обмена 2048-битными RSA-ключами между двумя устройствами и ряда хеш-функций. Протокол допускает использование шифрования end-to-end (сквозное шифрование) с опциональной сверкой ключей. Во время сквозного шифрования исходящее сообщение преобразуется в определенную кодовую комбинацию и затем декодируется на конечном хосте. Следует отметить, что ключи расшифровки хранятся на устройствах пользователей Telegram, а не на сервере, что позволяет исключить риск потери данных в случае каких-то взломов.

При первом запуске приложения пользователь вводит номер своего телефона, на который приходит пятизначный код его подтверждения. После ввода кода запускается алгоритм протокола авторизации:

- Клиент направляет запрос на сервер со случайно сгенерированной последовательности битов, длиной 128;
- Сервер отправляет ответ из другой последовательности, число  $n$  и цифровую подпись публичного ключа RSA;
- Клиент раскладывает  $n$  на два простых числа  $p$  и  $q$ , причем  $p < q$ , и выбирает ключ для отправленной с сервера цифровой подписи;
- Клиент генерирует строку, длиной 256 бит, далее собирает и шифрует с помощью выбранного ключа три случайные строки, числа  $p$ ,  $q$  и  $n$ , исполь-

зую алгоритм RSA и отправляет серверу;

- Сервер отправляет параметры протокола  $g$ ,  $p$ ,  $g_a$  Диффи-Хеллмана, которые зашифрованы алгоритмом AES-256 в режиме IGE;

- Клиент выбирает закрытый ключ и вычисляет открытый ключ, который в зашифрованном виде отправляется на сервер и общий открытый ключ.

## **2.2 Сбор персональных данных**

В разрабатываемой системе необходимо проводить регистрацию пользователей для записи на услугу. Для этого разработан модуль сбора персональных данных через самого бота Telegram.

Однако следует иметь в виду, что для сертификации приложения сбор персональных данных должен соответствовать требованиям Закона о персональных данных №152-ФЗ, согласно которому персональные данные – это информация, относящаяся к определенному, или определяемому физическому лицу, а обработка персональных данных – это совокупность действий, совершаемых с использованием средств автоматизации или без использования таких средств с персональными данными, включая сбор, запись, систематизацию, накопление, хранение, уточнение (обновление, изменение), извлечение, использование, передачу (распространение, предоставление, доступ), обезличивание, блокирование, удаление, уничтожение персональных данных.

С 1 сентября 2015 года в России вступило в силу положение, обозначенное законом ФЗ-242, которое обязывает операторов персональных данных обрабатывать и хранить персональные данные россиян с использованием баз данных, размещенных на территории РФ.

Следует отметить, что согласно КоАП РФ Статья 13.12. «Нарушение правил защиты информации» – использование несертифицированных информационных систем, баз и банков данных, а также несертифицированных средств защиты информации, если они подлежат обязательной сертификации влечет наложение административного штрафа, а также нарушение требований о защите информации (за исключением информации, составляющей государ-

ственную тайну), установленных федеральными законами и принятыми в соответствии с ними иными нормативными правовыми актами Российской Федерации, за исключением некоторых предусмотренных случаев, влечет наложение административного штрафа.

Поэтому было решено разработать и вынести отдельную форму регистрации вне экосистемы Telegram.

### 3 ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Такие системы работают на основе глубоких нейронных сетей. Для понимания работы такой системы необходимо рассмотреть принцип обучения систем искусственного интеллекта.

Задача машинного обучения – это построение алгоритма  $a: X \rightarrow Y$ . В данном случае  $X$  – это пространство объектов, а  $Y$  – пространство ответов. Для формализации соответствия алгоритма ожиданиям вводится функционал качества. Функционал качества измеряет качество работы алгоритма и зафиксирован при поиске оптимального алгоритма. Такой процесс называется обучением.

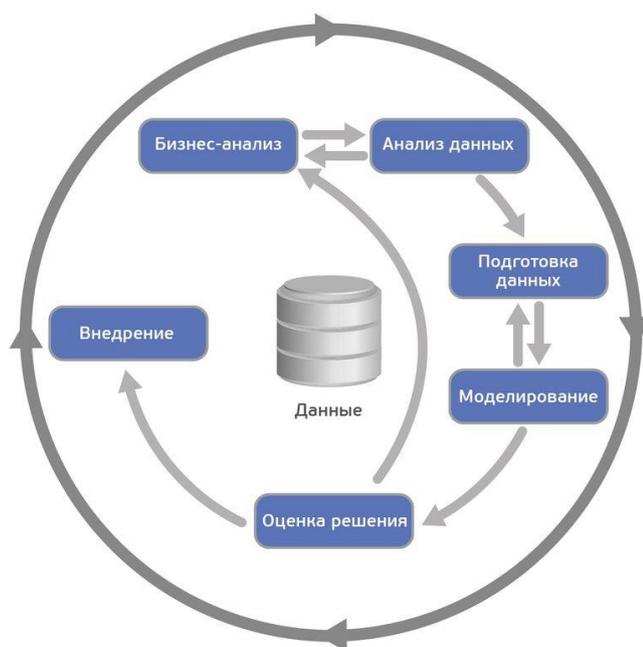


Рисунок 1 – Жизненный цикл модели машинного обучения

Жизненный цикл машинного обучения представлен на рисунке 1. На первом этапе необходимо провести анализ предметной области, далее – подготовить данные. Подготовка заключается в сборе и нормализации данных, а также в конструировании новых признаков. Следующим этапом является моделирование, состоящее в выборе алгоритма, обучении модели и оценки ре-

зультатов. Далее следует оценка решения, внедрения и мониторинга.

Один из ключевых этапов в жизненном цикле модели машинного обучения является оценка ее эффективности на данных, которые модели до этого не встречались. Существует два основных подхода для получения надежных оценок эффективности модели:

– Holdout cross-validation, при котором происходит разбиение набора данных на «обучающий» и «тестовый» наборы. Обучающий набор используется для обучения модели, а тестовый набор используется для оценки ее эффективности обобщения. Обычное разделение при использовании данного подхода происходит в соотношении 80% на 20%, в пользу обучающего набора;

– K-fold cross-validation, при котором набор данных случайным образом разделен на «k» групп. Одна из групп используется в качестве тестового набора, а остальные используются в качестве обучающего набора. Модель обучена обучающему набору и использована на тестовом наборе. Затем процесс повторяется до тех пор, пока каждая уникальная группа не была использована в качестве тестового набора.

Процесс обучения модели машинного обучения является оптимизационным процессом, который можно решить с помощью градиентных методов, таким как градиентный спуск, который состоит в итерации следующего выражения до сходимости:

$$\omega^{(k)} = \omega^{(k-1)} - \eta_k \nabla Q(\omega^{(k-1)}), \quad (1)$$

где  $\omega$  – начальный набор параметров;

$Q(\omega)$  – значение функционала ошибки;

$\eta_k$  – длина шага,  $k$  – номер итерации (шага);

$-\nabla$  – обозначение антиградиента, указывающий в сторону наискорейшего убывания функции в конкретной точке.

Такой подход в обучении используется также и для многоуровневых архитектур моделей машинного обучения, являющихся предметом исследований в сфере глубокого обучения. Однако для изменения значения параметров  $\omega$  для подгонки модели используется алгоритм обратного распространения ошибки, при котором происходит выражение частной производной функционала ошибки по параметру  $\omega$ . Такое выражение показывает, насколько быстро изменяется функционал ошибки при изменении параметров  $\omega$ . На рисунке 2 представлен принцип данного алгоритма в полносвязной нейронной сети.

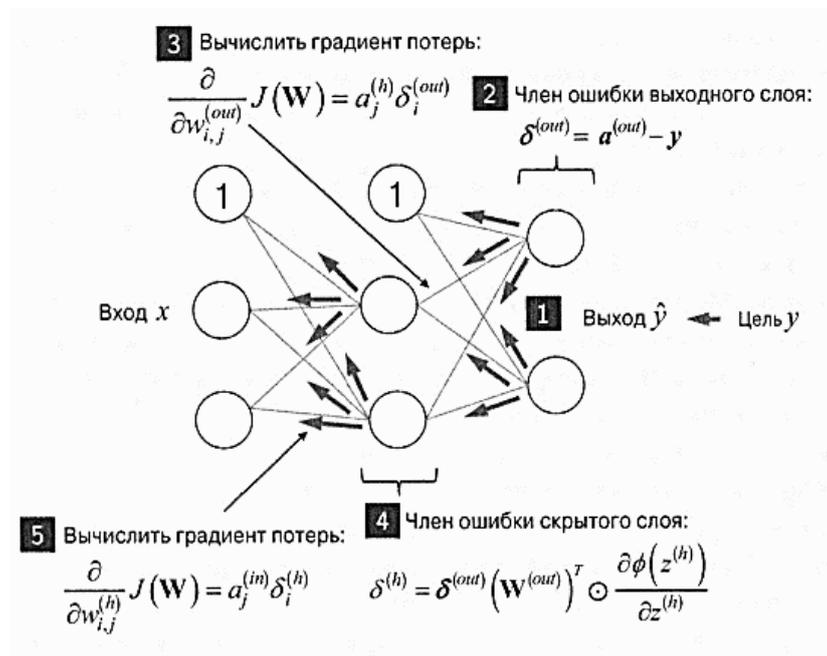


Рисунок 2 – Визуализация обратного распространения ошибки

Алгоритм обратного распространения ошибки можно описать следующим образом:

- Инициализировать параметры нейронной сети  $\omega$  случайными значениями, установить функцию  $a$  активации для выходного слова и подать на вход обучающий пример  $x$ ;

- Вычислить прямое распространение для всех слоев нейронной сети:

$$z_j^l = \sum_k \omega_{jk}^l a_k^{l-1} + b_j^l, \quad (2)$$

где  $z_j^l$  – взвешенный вход функции активации нейрона  $j$  в слое  $l$ ;

$a$  – функция активации;

$\omega$  – параметры;

$b$  – смещение;

– Вычислить выходную ошибку:

$$\delta^L = \nabla_a Q \odot \sigma'(z^L), \quad (3)$$

где  $\nabla$  – градиент функции;

$Q$  – функционал ошибки;

$\sigma$  – функция активации;

– Вычислить обратное распространение

$$\delta^l = ((\omega^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l); \quad (4)$$

– Рассчитать градиент функционала ошибки для каждого  $l = L, \dots, 2$ :

$$\omega^l = \omega^l - \eta \sum_x \delta^{x,l} (a^{x,l-1})^T. \quad (5)$$

На данный момент предпочтительными моделями нейронных сетей для естественного языка являются контекстные архитектуры, такие как, например, двунаправленные рекуррентные нейронные сети, которые можно описать выражениями (6) – (8):

$$H_t^f = \sigma^f(W_x^f X_t + W_h^f H_{t-1}^f + b^f), \quad (6)$$

где  $W_x^f, W_h^f$  – матрицы весов;

$b^f$  – смещение;

$\sigma^f$  – функция активации;

$H_{t-1}^f$  – выход однонаправленной рекуррентной сети;

$X_t$  – матрица входа на шаге  $t$ .

$$H_t^b = \sigma^b(W_x^b X_t + W_h^b H_{t+1}^b + b^b), \quad (7)$$

где  $W_x^b, W_h^b$  – матрицы весов;

$b^b$  – смещение;

$\sigma^b$  – функция активации;

$H_{t+1}^b$  – выход однонаправленной рекуррентной сети;

$X_t$  – матрица входа на шаге  $t$ .

$$O_t = W^q H_t + b^q, \quad (8)$$

где  $W^q$  – матрица весов;

$b^q$  – смещение;

$\sigma^b$  – функция активации;

$H_t$  – конкатенированный вектор;

$O_t$  – выход сети на шаге  $t$ .

## 4 РАЗРАБОТКА ИНФОРМАЦИОННОГО ПРОДУКТА

### 3.1 Выбор средств разработки

#### 3.1.1 Библиотеки для разработки нейронных сетей

Для разработки моделей глубокого обучения используются программные библиотеки, в которых реализованы все основные архитектуры нейронных сетей, оптимизированные под параллельные вычисления на центральных и графических процессорах. Использование библиотек позволяет сделать переход от низкоуровневого программирования и оптимизации к экспериментам с нейронными сетями на высоком уровне абстракции. Исследователь может полностью сосредоточиться на сути нейронных сетей и изучать их поведение при разных конфигурациях. В таблице 1 представлены самые распространенные библиотеки глубокого обучения.

Таблица 1 – Обзор библиотек глубокого обучения

Название библиотеки	Описание
TensorFlow	Базовым языком является C++, но имеет API для Python. Разработан компанией Google. Вычисления выполняются с использованием графов потоков данных. Предлагает мощные средства мониторинга процесса обучения моделей и визуализации. Лидер промышленной разработки
Theano	Базовым языком является Python. Разработан университетом Монреаля. Вычисления с использованием графов потоков данных. Эффективная обработка тензоров. Вычисления выражаются NumPy – подобным синтаксисом. Имеет высокий входной порог.

Название библиотеки	Описание
PyTorch	Базовый язык Lua. Поддерживает API для Lua, Python, Java, C++. Разработан Ронаном Коллабертом. Имеет множество модульных элементов, которые легко комбинировать. Легко писать собственные типы слоев и работать на GPU. Имеет API разных уровней абстракции
CNTK	Базовый язык C++. Поддерживает API для C++, C#, Python, Java. Разработана Microsoft. Обеспечивает скорость обучения, сравнимую с TensorFlow, а на рекуррентных сетях превосходит его. Имеет API разных уровней абстракции
Caffe	Базовый язык C++. Поддерживает API для C++, Python. Разработан в центре компьютерного зрения и обучения Беркли. Имеет небольшую скорость относительно других библиотек
Keras	Библиотека верхнего уровня. Поддерживает Python. В качестве вычислительного back-end использует TensorFlow или Theano. Позволяет создавать и обучать нейронные сети на очень высоком уровне абстракции. Имеет низкий порог вхождения

Программная библиотека TensorFlow на данный момент является лидером промышленной разработки, которая поддерживает все типы архитектур нейронных сетей, имеет огромное сообщество, качественное руководство, а также большое количество готовых к использованию заранее обученных мо-

делей. TensorFlow свою сложность компенсирует тем, что использует библиотеку Keras как официальное и рекомендуемое API, а уровень вхождения у Keras – в один из самых низких.

Помимо этого, существуют мощные и удобные инструменты для решения задач классического машинного обучения: Scikit-learn, Xgboost, LightGBM, CatBoost.

Данные инструментальные средства используются в открытой программной библиотеке разговорного ИИ DeepPavlov, предоставляющий разработчикам:

- Набор предобученных моделей нейронных сетей для анализа текста и компонентов диалоговых систем;
- Возможности для создания и тестирования диалоговых моделей;
- Инструменты прикладной разработки и интеграции.

Также следует отметить библиотеку на языке программирования Python для обучения state-of-the-art, или SOTA (перспективные технологии) моделей в обработке естественного языка – Transformers. Разработкой библиотеки занимается компания HuggingFace. Transformers предоставляет API для использования таких архитектур трансформеров, как BERT, RoBERTa, GPT-2 или DistilBERT.

В данной работе используются библиотеки DeepPavlov и Transformers.

### **3.1.2 Использование стороннего API**

Программные компоненты в данной работе взаимодействуют друг с другом посредством API (программный интерфейс приложения), технологии, которая описывает наборы классов, процедур, функций, структур и констант, которыми одна компьютерная программа может взаимодействовать с другой программой. API реализуется программной библиотекой или сервисом операционной системы и используется для написания приложений. API библиотеки функций и классов включает описание сигнатур и семантики функций.

Использование API упрощает процесс программирования при создании приложений, абстрагирует базовую реализацию и предоставляет объекты, необходимые программисту, а также используется как средство интеграции приложений.

Да данный момент существует два основных инструмента API, с помощью которых можно задействовать сервисы Telegram - Telegram Bot API, служащий для разработки чат-ботов, и Telegram API, для разработки клиентов. Для их работы задействованы промежуточные сервера с HTTPS-интерфейсом (используется определённый набор HTTP-запросов, а также определение структуры HTTP-ответов, для выражения которых используют XML- или JSON-формат), шифрующий трафик и обеспечивает связь с Telegram API. Таким образом предоставляется возможность создавать программы, использующие интерфейс Telegram для выполнения кода на локальном сервере. Пользователь может взаимодействовать с ботом, отправляя им сообщения, команды, встроенные запросы.

Бот перманентно направляет на сервер и регулярно получает обновления. Процесс получения обновлений может происходить при помощи опросов polling, при котором происходит периодическое подключение к серверам Telegram для проверки новых обновлений или отправки сообщений (рисунок 3).

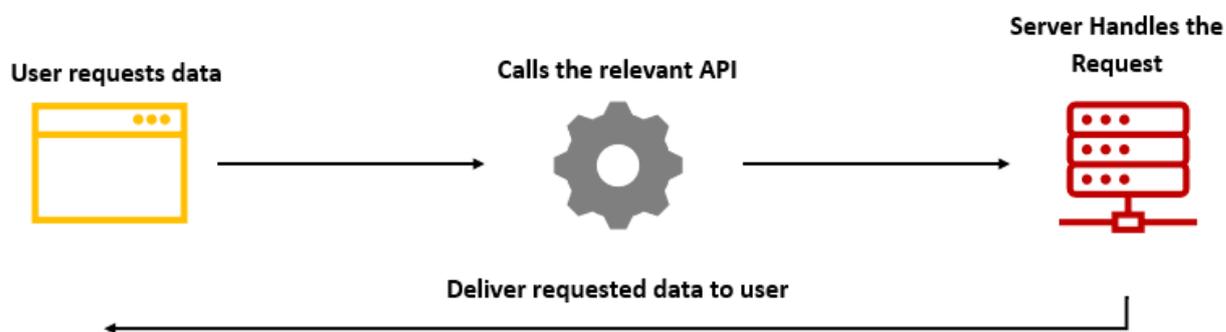


Рисунок 3 – Схема работы API-вызова в веб-приложении

Также можно использовать механизм, обратный API – Webhook, при котором всякий раз, когда приходит новое обновление для бота, сервер Telegram отправляет это обновление на указанный URL.

Подробнее Webhook можно представить как схему, состоящую из триггера событий, запускающий Webhook, Webhook-провайдера, отправляющий в стороннее приложение HTTP-запрос методом «POST», которое получает стороннее приложение и обрабатывает (рисунок 4).

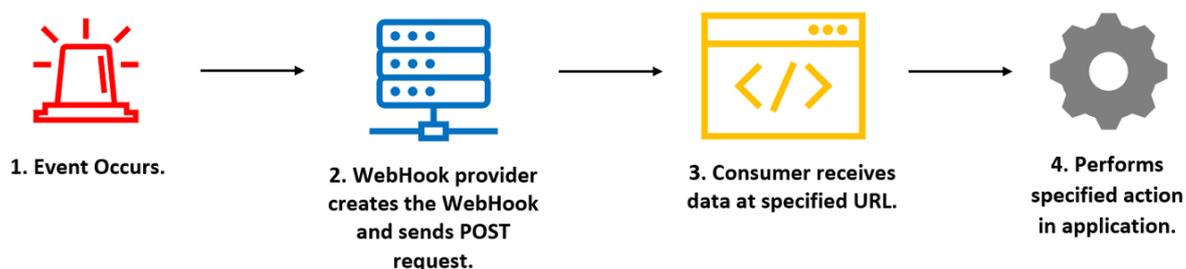


Рисунок 4 – Механизм работы Webhook в веб-приложении

Еще одним примером использования стороннего API в рамках разрабатываемого проекта является JavaScript API – программная библиотека для работы с картами в браузерах, и представляющая собой набор компонентов для размещения интерактивных Яндекс.Карт на сайте или веб-приложении. JavaScript API предоставляет возможности:

- Доступа к данным и технологиям Яндекса: встраивание на сайт или в приложение карту с поиском по топонимам и организациям, с возможностью строить маршруты и смотреть панорамы, а также с другими функциями, доступными на Яндекс.Картах;
- Гибкой настройки карты: настройка нужной логики взаимодействия пользователя с картой и определить, как эта карта будет выглядеть;
- Визуализации данных и аналитики: визуализация большого объёма информации или принятие решения на основе анализа геоданных.

### 3.1.3 СУБД

База данных – это логически смоделированное хранилище любых типов данных. СУБД – это приложение для управления базам данных различных форм, размеров, типов. В данном проекте используется реляционная модель работы с данными, которая определяет всю хранимую информацию как набор связанных записей и атрибутов в таблице. Каждый атрибут содержит свой тип информации, а каждая запись обладает уникальным ключом. Наиболее популярными РСУБД на данный момент являются SQLite, MySQL, PostgreSQL.

В таблице 2 представлены основные преимущества и недостатки названных РСУБД.

В данной работе используется РСУБД PostgreSQL как более функциональный вариант.

Для взаимодействия с базой данных была выбрана технология программирования, позволяющая преобразовать несовместимые типы моделей в ООП, в частности, между хранилищем данных и экземплярами классов в коде – объектно-реляционное отображение ORM.

Это позволило использовать упрощенную схему сохранения объектов в программе в реляционную базу данных и их извлечения.

Таблица 2 – Таблица преимуществ и недостатков РСУБД

РСУБД	Преимущества	Недостатки	Использование
SQLite	Файловая БД (упрощает перемещение), стандартизированная (использует SQL), подходит для разработки и тестирования	Отсутствие пользовательского управления, дополнительной настройки	Используется в встроенных, не предназначенных для масштабирования приложениях, системах доступа дисковой памяти

РСУБД	Преимущества	Недостатки	Использование
MySQL	Простота, многофункциональность, безопасность, мощность и масштабируемость, высокая скорость	Присутствует ограничение функциональности, некоторые операции реализованы ненадежно, застой в разработке	Используется при работе с распределёнными операциями, веб-сайтах, имеет высокую безопасность
PostgreSQL	Полная SQL-совместимость, поддерживается опытным сообществом, расширяемость, объектно-ориентированность	Низкая производительность в простых операциях, популярность, проблематичный хостинг	Используется в системах с высоким приоритетом надежности, сложными процедурами, не имеет проблем с интеграцией.

Помимо всего прочего, ORM обеспечивает полное разделение задач в программе и решает проблему «несоответствия», которая гласит о том, что объектные и реляционные модели не хорошо работают вместе.

Ключевой особенностью ORM является отображение, которое используется для привязки объекта к его данным в БД. ORM как бы создает «виртуальную» схему базы данных в памяти и позволяет манипулировать данными уже на уровне объектов. Отображение показывает, как объект и его свойства связаны с одной или несколькими таблицами и их полями в базе данных. ORM использует информацию этого отображения для управления процессом

преобразования данных между базой и формами объектов, а также для создания SQL-запросов для вставки, обновления и удаления данных в ответ на изменения, которые приложение вносит в эти объекты. Использование ORM в проекте избавляет разработчика от необходимости работы с SQL и написания большого количества кода, часто однообразного и подверженного ошибкам. Весь генерируемый ORM код предположительно хорошо проверен и оптимизирован, поэтому не нужно в целом задумываться о его тестировании.

Для использования технологии ORM на языке программирования Python существует фреймворк SQLAlchemy, который позволяет работать с базами данных MySQL, Oracle, SQLite, PostgreSQL.

### **3.1.4 Сбор актуальной информации**

Так как разрабатываемая система может использоваться, помимо регистрации на услуги, и для получения справочной и организационной информации, то первостепенным является вопрос актуальности информации, хранящейся в системе, и автоматизации процесса ее обновления.

Веб-скрейпинг еще со времен появления интернета стал незаменимым инструментом для автоматизированного сбора информации в интернете, позволяющий производить поиск необходимой информации на веб-ресурсах, копировать данные из Интернета, проводить мониторинг обновлений на сайтах.

Веб-скрейпинг реализуется с помощью программной кода, выполняющего GET-запросы на целевой сайт с последующим синтаксическим преобразованием полученных веб-страниц. В данной работе используется Python пакет BeautifulSoup, позволяющий проводить анализ HTML документов с целевого сайта с использованием древовидного синтаксического анализатора, преобразующий входную линейную последовательность лексем текстовых языков разметки (HTML и XHTML) в структуру данных – дерево, которое отражает синтаксическую структуру входной последовательности и подходит для дальнейшей обработки.

### 3.2 Проектирование информационной системы

В проектировании данного программного продукта были использованы диаграмма вариантов использования системы, диаграмма состояний системы, а также диаграмма развертывания, которые предлагаются стандартом UML. Данные схемы на рисунках упрощены для наглядности.

Диаграммы вариантов использования подходят для моделирования процессов и определения требований к системе. Данная диаграмма представлена на рисунке А.4. Из схемы видно, что через разрабатываемую систему заявитель имеет возможность получать новости организации, выставлять и получать уведомления на устройство о записях на прием, получать справочную и организационную информацию в документах и новостях организации, а также записаться на прием. В последнем случае встроена возможность автоматически определять услугу по обращению с помощью технологий машинного обучения, выбора отделения, даты и времени приема, а также получения интерактивной карты маршрута до отделения МФЦ.

Диаграммы состояний определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий. Диаграмма представлена на рисунке А.5.

В начальном состоянии система ожидает ввод пользователя. При вводе команды «/start» происходит поиск пользователя в базе с использованием уникального идентификатора чата. Если пользователь уже зарегистрирован, то происходит его авторизация, иначе для дальнейших действий ему необходимо зарегистрироваться. После система переходит в состояние ожидания команд пользователя. В этом состоянии ему предоставляется возможность завершить работу. Также есть возможность установить напоминание о записи, получить новости организации и задать вопрос либо записаться на услугу, после чего перейти в конечное состояние в контексте данного автомата. С помощью команды «/qa» система переходит в состояние «Получение новостей организа-

ции», во время перехода в которое происходит сбор новостей организации с целевого сайта, и затем в состояние «Ожидание вопроса», при котором система получает вопрос по новостям организации, затем генерирует ответ. В случае выбора команды «/admission», происходит переход в суперсостояние «Запись на прием». Пользователь последовательно переходит по состояниям автомата: выбирает город, описывает проблему, отбирает отделение, отправляет геолокацию и устанавливает удобное время приема. После происходит запись заявителя на прием и назначение сотрудника. Автомат переходит в состояние ожидания команд пользователя.

Диаграмма развёртывания системы изображена на рисунке А.6. С помощью этой диаграммы были визуализированы аппаратные узлы системы, каналы связи между ними размещения программных файлов. Из схемы видно, что на стороне клиента используется мессенджер Telegram. Он позволяет взаимодействовать с разрабатываемой системой через интерфейс чат-бота. Мессенджер непосредственно взаимодействует с серверами Telegram посредством протокола HTTP, связывая через Telegram Bot API легковесный клиент Telegram с сервером приложения. Используя Telegram Bot API, предоставляется возможным вызывать код на сервере приложения. Такой подход в случае изменения логики работы приложения не требует изменений на стороне клиента.

На сервере приложения запущен главный программный скрипт «main.py», а также запущен ORM-драйвер для работы с серверной базой данных PostgreSQL. Сервер приложения получает доступ к JavaScript API через промежуточный сервер, построенный на архитектуре REST.

Также сервер имеет доступ к вычислительным мощностям, в качестве которых выступают сервера с графическими процессорами фирмы Nvidia и программно-аппаратной архитектурой параллельных вычислений CUDA для увеличения вычислительной производительности. На них выполняются классификация обращений пользователя, а также ранжирование документов с помощью нейросетевых моделей для предоставления справочной информации.

### 3.3 Разработка БД

#### 3.3.1 Инфологическое проектирование БД

На данном этапе была построена семантическая модель предметной области, без ориентации на какую-либо конкретную СУБД и модель данных. Во время инфологического проектирования были выполнены следующие этапы:

- формирования набора сущностей;
- формирование спецификаций атрибутов по сущностям;
- обоснования установления связей.

А также были выделены следующие сущности:

- сущность «Заявитель» – хранит основную информацию о клиенте;
- сущность «МФЦ» – хранит основную информацию о МФЦ;
- сущность «Новость» – хранит сообщения о событиях в организации;
- сущность «Услуга» – хранит основную информацию об услуге, предоставляемой МФЦ.

Сущность «Сотрудник» – хранит основную информацию о сотруднике организации, который берет на себя ответственность за выполнение услуги

Для каждой сущности был выбран первичный ключ, который однозначно идентифицирует каждую запись таблицы. Для концептуальной инфологической модели необходимо установить связи между сущностями. Модель «сущность-связь» – модель данных, описывающая концептуальные схемы предметной области. На рисунке А.7 представлена инфологическая модель в нотации Crow's Foot. Из схемы видно, что сущность Заявитель обращается в МФЦ для получения услуги. Причем данная связь является единственной связью «многие-ко-многим». Сущность МФЦ размещает Новость на своем сайте. В МФЦ работает Сотрудник, который выполняет услугу для Заявителя.

Спецификация имеет вид таблицы, которая содержит 5 столбцов – наименование атрибута, тип данных, диапазон значений, пример атрибута. На таблицах 3 – 6 представлены данные спецификации.

Таблица 3 – Спецификация атрибутов сущности «Заявитель»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
Id	Число, однозначно определяющее пользователя	Integer	>0	2
Фамилия	Фамилия	String	-	Anna
Имя	Имя	String	-	Maria
Фотография	Фотография пользователя	Binary	-	-
Паспортные данные	Паспортные данные	String	-	12233456
Chat-id	Уникальный идентификатор чата	String	-	1234567789
Пол	Пол	String	M/F	M

Таблица 4 – Спецификация атрибутов сущности «МФЦ»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
Id	Число, однозначно определяющее организацию	Integer	>0	2
Субъект	Регион, где расположена организация	String	-	Moskovskay oblast
Город	Город, где расположена организация	String	-	Blagoveschensk
Адрес	Адрес, по которому находится организация	String	-	Lenina 144
Долгота и Широта	Географическая долгота и широта на карте	String	-	1223345645

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
Ссылка на официальный сайт	Ссылка на официальный сайт	String	-	<a href="https://yandex.ru">https://yandex.ru</a>
Ссылка на социальную сеть	Ссылка на социальную сеть	String	-	<a href="https://vk.com">https://vk.com</a>

Таблица 5 – Спецификация атрибутов сущности «Услуга»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
Id	Число, однозначно определяющее услугу	Integer	>0	2
Дата приема	Дата приема	Date	Date.now – Date.now + 60 * 60 * 24 * 30 сек	01.10.2021
Дата регистрации	Дата регистрации	Date	Date.now	01.10.2021
Статус приема	Выполнена ли услуга	Boolean	True/False	True
Описание	Описание услуги	String	-	Получение паспорта

Таблица 6 – Спецификация атрибутов сущности «Сотрудник»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример атрибута
Id	Число, однозначно определяющее сотрудника	Integer	>0	2
Фамилия	Фамилия	String	-	Anna
Имя	Имя	String	-	Maria
Фотография	Фотография пользователя	Binary	-	-
Паспортные данные	Паспортные данные	String	-	12233456
Должность	Должность сотрудника	String	-	Главный консультант
Отдел	Отдел, в котором работает сотрудник	String	-	Бухгалтерия
Пол	Пол	String	M/F	M

### 3.3.2 Логическое и физическое проектирование БД

С целью создания совокупности нормализованных отношений, в которых реализованы связи между объектами предметной области и выполнены все преобразования, необходимые для эффективной реализации, необходимо провести этап логического проектирования. Построенная логическая модель позволяет провести разработку физической модели. Ниже на рисунках А.8 и А.9 представлены логическая и физическая диаграммы в нотации IDEF1X.

На данном этапе были разрешены связи «многие-ко-многим» с помощью промежуточной сущности «Заявитель\_МФЦ», а также исходная сущность «Услуга» была разбита на две сущности: «Регистрация» и «Услуга». Также на данном этапе была проведена проверка базы данных на три нормальные формы.

### **3.3.3 Создание БД**

Для создания базы данных использовалась программная библиотека на Python SQLAlchemy, обеспечивающая работу с реляционными СУБД с помощью технологии ORM. Был создан отдельный каталог database, содержащий файлы «\_\_all\_models.py» (для импорта всех сущностей БД), содержащихся в «entity.py». Файл «db\_session.py» содержит функцию для подключения к предварительно созданной серверной базе данных postgresql, создания виртуальной базы и сессии для взаимодействия с ней.

Сами сущности объявлены как классы, представляющие собой шаблоны для создания объектов. Класс (шаблон) обеспечивает начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов. В данном случае поля класса будут преобразованы в атрибуты отношения.

Далее в файле «main.py» описан программный код для генерации и инициализации отношений БД и выполнении SQL-запросов через объектно-ориентированный интерфейс.

Таким образом происходит начальная инициализация БД. Такая технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

## **3.4 Разработка информационного продукта**

### **3.4.1 Элементы теории конечных автоматов**

При разработке чат-бота высокий приоритет получили модули регистрации заявителя и его дальнейшая запись на прием в отделение. Данные процессы были представлены в виде математической модели – вычислительной машины, и были реализованы с использованием абстрактных автоматов, имеющих свойство нахождения в некотором допустимом состоянии, и, при чтении символов входной цепочки, способных переходить из одного состояния в другое.

Конечный автомат можно представить как пятерку:

$$A = (S, X, Y, \delta, \rho), \quad (9)$$

где  $S$  – конечное множество состояний автомата;

$X, Y$  – конечные входной и выходной алфавиты;

$\delta: S \times X \rightarrow S$  – функция переходов;

$\rho: S \times X \rightarrow Y$  – функция выводов.

Реализации абстрактного автомата в данной работе происходила средствами Telegram API.

В качестве примера, можно прокомментировать автомат процесса регистрации пользователя.

Так, при вводе команды «/registration» конечный автомат начинает функционирование. После получения ответа пользователя, согласно определённой в программе функции переходов, происходит перевод автомата в новое состояние: так, получив утвердительный ответ на регистрацию, пользователю предлагается ввести свои паспортные данные, выбрать пол и т. д. После этого происходит запись в БД нового заявителя с возможностью записаться на услуги МФЦ.

Таким образом, в программном коде были реализованы алфавиты состояний автоматов, а также подмножества начальных и конечных их состояний.

### **3.4.2 Форма регистрации**

В программе реализована возможность регистрации через отдельную форму после перехода по ссылке, предоставляемого ботом. После ее заполнения выполняется POST-запрос, который передает данные пользователя на сервер. Полученные данные соответствующим образом обрабатываются и происходит запись данных заявителя в БД. Для разработки формы использовался фреймворк для создания веб-приложений Flask.

Дополнительно была реализована возможность загрузить фотографию

паспорта или его оцифрованную версию для того, чтобы не вводить данные вручную. В данном случае оцифрованный документ будет обработан с помощью алгоритмов компьютерного зрения для выделения участка изображения паспорта с машиночитаемыми зонами (MRZ), которые обеспечивают быстрое распознавание основных реквизитов паспорта. После выделенная зона передается в систему оптического распознавания символов (OCR), который выполняет перевод изображений текста в текстовые данные с помощью искусственных нейронных сетей.

Для выделения области с MRZ изначально происходило считывание файла изображения. Далее следовало преобразование цветного изображения в градации серого:

$$RGB \text{ to Gray: } Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \quad (10)$$

где  $Y$  – новое значение пикселя изображения;

$R$  – значение красного канала исходного изображения;

$G$  – значение зеленого канала исходного изображения;

$B$  – значение синего канала исходного изображения.

Результат преобразования подвергался сглаживанию с использованием размытия по Гауссу с ядром размерностью 3 на 3:

$$G_i = \alpha \cdot e^{-\frac{\left(i - \frac{ksize-1}{2}\right)^2}{2 \cdot \sigma^2}}, \quad (11)$$

где  $G_i$  – новое значение пикселя изображения;

$ksize$  – размер окна;

$\sigma$  – стандартное отклонение Гаусса;

$i = 0, \dots, ksize - 1$ .

Далее изображение проходило через серию морфологических преобра-

зований, такие как Black Hat, для выявления темных областей изображения на светлом тоне, с последующим применением оператора Собеля (12) для вычисления производных изображения, и Erode для удаления шумов.

$$dst = \frac{\partial^{orderx+ordery} src}{\partial x^{orderx} \partial y^{ordery}}, \quad (12)$$

где  $src$  – исходное изображение;

$dst$  – выходное изображение;

$orderx$  - порядок производной по направлению  $x$ ;

$ordery$  – порядок производной по направлению  $y$ .

В конце были выявлены все контуры в изображении и отсортированы.

### 3.4.3 Определение услуги по обращению

Одной из ключевых идей разработки системы поддержки клиента является использование актуальных исследований в области систем с естественно-языковым интерфейсом. Так, для определения услуги по обращению заявителя можно использовать алгоритмы преобразования текста в числовое описание с последующим вычислением меры сходства с базой таких предварительных векторов с размеченными услугами.

Для таких целей можно использовать статистику ранжирования релевантности документа в корпусе с учетом запроса клиента «term frequency – inverse document frequency», или Tf-Idf, позволяющая конвертировать текст в векторно-пространственную модель:

$$tfidf(t,d,D) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \cdot \log \frac{N}{|\{d \in D: t \in d\}|}, \quad (13)$$

где  $N$  – количество документов в корпусе;

$t$  – термин;

$d$  – документ,

$f_{t,d}$  – количество термина в документе.

База знаний в таком случае представляет из себя таблицу из двух колонок, одна из которых содержит текстовое описание услуги, предоставляемой МФЦ, другая – наименование услуги. Во время процесса обучения данной системы классификации, каждому описанию ставится свой вектор размерности словаря из всех слов набора данных, рассчитываемый с использованием меры Tf-Idf, которая ставит каждому слову в соответствие число, прямо пропорциональное частоте встречаемости в предложении и обратно пропорционально частоте во всем наборе. Тогда, текстовое обращение о проблеме пользователя поступает на вход данной системы и конвертируется в числовое представление. Далее происходит фильтрация записей в базе по описанию и выбирается наиболее подходящая услуга.

В качестве фильтра использована метрика схожести векторного представления документов – косинусное подобие:

$$\text{cosine similarity} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}, \quad (14)$$

где  $A_i$  и  $B_i$  компоненты векторов  $A$  и  $B$  соответственно.

Следует отметить, что на данный момент существуют более продвинутые способы преобразования текста в векторную модель. Одним из них является модель на основе нейронных сетей Word2Vec, оптимизационным критерием которой может выступать выражение:

$$\sum_{i=1}^n \log \text{SoftMax}_{w_i \in W} \left( \frac{1}{2k} \sum_{c \in C_i} \langle u_{w_i}, v_c \rangle \right) \rightarrow \max_{U, V}, \quad (15)$$

где  $w_i$  – слово в заданном контексте  $C_i$  из словаря  $W$ ;

$u_{w_i}$  – вектор предсказываемого слова;

$v_c$  – вектор предсказывающего слова;

$v_c \in \mathbb{R}^d$ ,  $U, V \in \mathbb{R}^{|W| \times d}$ ;

$$\text{SoftMax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}.$$

#### 3.4.4. Вопросно-ответная система

Для интерактивного взаимодействия с пользователем, а также для предоставления ему справочной информации, в программе реализована возможность ответа на задаваемый вопрос. Система проводит поиск ответа в новостной ленте организации, который получается с целевого сайта с помощью технологий веб-скрейпинга.

Данную задачу можно решить с помощью вопросно-ответных систем на основе архитектуры генерации признакового описания токенов BERT, точность которых согласно команде разработчиков deepravlov достигает 88,49 на метрике F1. В качестве критериев для обучения BERT служат MLM (16) и NSP (17):

$$\sum_S \sum_{i \in M(S)} \ln p(w_i | i, S, W) \rightarrow \max_W, \quad (16)$$

где  $M(S)$  – подмножество маскированных токенов из множества токенов предложения входного текста  $S$ ;

$p(w_i | i, S, W)$  – языковая модель, предсказывающая  $i$ -ый токен предложения  $S$  при ее параметрах  $W$ .

$$\sum_{(S, S')} \ln p(y_{SS'} | S, S', W), \quad (17)$$

где  $y_{SS'}$  – классификация пары предложений  $S'$  и  $S$ ;

$p(y_{SS'} | S, S', W)$  – вероятностная модель бинарной классификации пар  $(S, S')$ .

Также для повышения вариабельности ответов бота и естественности общения можно «переписать» его высказывания, сохраняя их смысл. Это достигается с использованием глубокой нейронной сети. При обучении для сравнения исходного и сгенерированного предложений, измерения семантического сходства предложений и естественности текстов можно воспользоваться метриками BLEU, описываемая формулами (18) – (19), косинусного сходства векторных моделей (similarity), описываемая формулой (20) и перплексией (PP), описываемая формулой (21) соответственно.

$$BLEU = BP \cdot \exp(\sum_{n=1}^N w_n \log p_n), \quad (18)$$

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r, \end{cases} \quad (19)$$

где  $p_n$  – точность для n-gram;

$w_n$  – веса для точности;

$BP$  – штраф за короткие варианты перевода;

$r$  – длина эталонного предложения;

$c$  – длина проверяемого предложения.

$$similarity = \frac{\sum_{i=1}^n N_i \cdot M_i}{\sqrt{\sum_{i=1}^n N_i^2} \cdot \sqrt{\sum_{i=1}^n M_i^2}}, \quad (20)$$

где  $N_i, M_i$  – элементы векторов N и M соответственно.

$$PP = \prod_x p(x)^{-p(x)}, \quad (21)$$

где  $p$  – вероятность слова  $x$ .

### 3.4.5 Описание работы программы

После запуска мессенджера бот приветствует пользователя. Чтобы воспользоваться возможностями программы, необходимо отправлять виртуальному ассистенту специальные команды. Они все указаны и описаны в выпадающем меню слева. После отправки команды «/registration» пользователь подтверждает процесс регистрации и вводит свои персональные данные (рис. Б.2). Некоторые из пользовательских сообщений бот удаляет сразу после ввода, например, паспортные данные. Также при отправке команды «/enter\_passport» пользователь получает ссылку на форму регистрации, где помимо ручного ввода данных можно загрузить оцифрованную версию паспорта, и уникальный идентификатор чата, который необходимо указать в форме. При отправке команды «/admission» начинается процесс записи на услугу. Заявителю предлагается выбрать город и адрес, которые извлекаются из базы данных и отображаются в виде inline-кнопок, подходящий из которых необходимо нажать (рис. Б.3). После описания проблемы, система автоматически определяет необходимую услугу, которую необходимо предоставить заявителю, выводит список необходимых документов либо ссылку на источник, где описаны необходимые документы. Далее с помощью inline-кнопок необходимо выбрать дату, месяц, число и время приема (рис. Б.4, Б.5). Если в это время нет записи на другого заявителя, и есть свободный штатный сотрудник, то происходит запись на услугу. Также, если пользователь сохранил свою геолокацию в системе с помощью команды «/get\_geo», бот дополнительно отправляет ссылку на сайт с интерактивной картой и маршрутом пути с учетом пробок. Пользователь имеет возможность выбрать способ пути (автомобиль, пешком, автобус), а также посмотреть расстояние и время, затраченное на путь. Пользователь может также установить таймер, через сколько ему должно прийти напоминание о записи, также узнать организационную информацию через вопросно-ответную систему с естественно-языковым интерфейсом, встроенную в систему. В последнем случае пользователь с помощью команды «/qa» пере-

дает вопрос по новостям. Бот находит ответ в новостной ленте организации, которую получает с целевого сайта и возвращает предложение, в которой находится ответ, а также ссылку на саму новость.

## 5 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

### 5.1 Безопасность

#### 5.1.1 Анализ потенциальных опасностей

Прежде всего, необходимо рассмотреть опасные и вредные факторы, воздействующие на пользователя ПЭВМ. Таковыми являются: несоответствие норме эмиссионных параметров от «мыши», клавиатуры, дисплея, системного блок; несоответствие норме визуальных параметров дисплеев при сертификации и испытаниях; нерациональная организация рабочего места; недостаточное освещение; высокие уровни шума; повышенные значения напряжения электрического и магнитного полей токов промышленной частоты 50 Гц (отсутствие заземления и т. п.). Данные факторы способны привести к следующим последствиям: проблемы со здоровьем (туннельный синдром кистевых суставов, нарушение осанки, ожирение, проблемы со зрением, гиподинамия, сердечно-сосудистые заболевания и др.), нарушения сна, повышенная утомляемость, депрессии и стрессы, возможно появление различных аллергических реакций.

#### 5.1.2 Мероприятия по обеспечению безопасности

Для предотвращения неблагоприятного воздействия на здоровье человека вредных факторов производственной среды и трудового процесса при работе с ПЭВМ в Российской Федерации приняты нормативы СанПиН 1.2.3685-21 и Санитарно-эпидемиологические правила.

Системный блок, клавиатура, монитор, мышь и другие возможные периферийные устройства должны располагаться таким образом, чтобы пользователь ПЭВМ был способен поворачиваться в пространстве, как в горизонтальном, так и в вертикальном направлениях. При фиксации своего положения пользователю должно быть удобно фронтальное наблюдение экрана монитора. Внешняя окраска обозначенных периферийных устройств и блоков ПЭВМ обязана быть спокойной, рассеивающей свет и в целом не раздражаю-

щей. Поверхность используемых устройств должна быть матовой, с коэффициентом отражения 0,4 – 0,6, а также не иметь блестящих деталей, способных создавать блики. Конструкция используемого монитора обязана предусматривать регулирование контрастности и яркости.

Помещения, в которых размещаются ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии со всеми техническими требованиями по эксплуатации. Рабочие места с ПЭВМ должны быть размещены таким образом, чтобы расстояние от экрана одного монитора до задней стороны другого составляло не менее 2 м, а расстояние между боковыми поверхностями мониторов составляло не менее 1,2 м.

Свет из любого окна не должен падать прямо на экран или на лицо пользователя. Крайне предпочтительно, чтобы источники освещения располагались слева от пользователя. Для регулирования уровня естественного освещения окна должны комплектоваться таким средством регулирования, как жалюзи.

В зависимости от особенностей работы, выполняемой пользователем ПЭВМ, искусственное освещение может быть общим равномерным или комбинированным – общее верхнее освещение должно дополняться местными осветительными элементами. Освещение дополнительными местными светильниками используется, если пользователь, помимо компьютера, работает с бумажными носителями информации. Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. Экран монитора должен находиться от глаз пользователя на расстоянии около 600 – 700 мм, но не менее чем 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Рабочая мебель, предназначенная для пользователей компьютерными устройствами, должна отвечать требованиям:

– высота рабочей поверхности стола должна быть способна к регулированию в пределах 680 – 800 мм; при отсутствии подобной возможности высота

рабочей поверхности должна быть равна 725 мм;

– рабочий стол должен иметь пространство для ног пользователя высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм;

– рабочий стул (кресло) должно иметь подъемно-поворотные механизмы, быть регулируемым по высоте и углам наклона сиденья и спинки, а также регулироваться по расстоянию спинки от переднего края сиденья, при этом регулировка любого параметра должна быть независимой от другого параметра, легко осуществляемой пользователем, а также иметь надежную фиксацию положения;

– рабочее место должно быть оборудовано подставкой для ног пользователя, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и регулировку по углу наклона опорной поверхности подставки до 20°; поверхность подставки должна быть рифленой, иметь по переднему краю бортик высотой 10мм;

– клавиатуру нужно располагать на поверхности рабочего стола на расстоянии 100 – 300 мм от края поверхности, обращенного к пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от столешницы.

В помещениях, оборудованных ПЭВМ, должна проводиться ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ПЭВМ.

В соответствии с нормативами, площадь на одно рабочее место пользователя ПЭВМ с монитором на базе электронно-лучевой трубки (ЭЛТ) должна составлять не менее 6 м<sup>2</sup>, с монитором на базе плоских дискретных экранов (жидкокристаллические, плазменные) – не менее 4,5 м<sup>2</sup>. При эксплуатации ПЭВМ с монитором на базе ЭЛТ (без вспомогательных устройств – принтер, сканер и др.), отвечающих требованиям международных стандартов безопас-

ности компьютеров, с продолжительностью работы менее 4-х часов в день допускается минимальная площадь 4,5 м<sup>2</sup> на одно рабочее место на каждого пользователя.

## **5.2 Экологичность**

Для производства одного ПЭВМ (системный блок, монитор, принтер) с общим весом 24 кг требуется на технологические расходы 240 кг ископаемого невозполнимого топлива для необходимых энергоносителей, 22 кг химических веществ и 1500 кг воды.

После окончания срока эксплуатации компьютерной техники, образуется лом, одна тонна которого содержит 480 кг черных металлов, 200 кг меди, 32 кг алюминия, 32 кг серебра, 1 кг золота, остальное – 33 элемента таблицы Менделеева Д.И. По данным Microelectronic and Computer Technology Corporation при изготовлении компьютерной техники используют бериллий, алюминий, титан, ванадий, хром, марганец, железо, кобальт, никель, медь, цинк, галлий, германий, мышьяк, селен, ниобий, рутений, родий, палладий, серебро, кадмий, индий, олово, барий, европий, тантал, платина, золото, ртуть, свинец, висмут, актиний.

Энергообеспечение стремительного роста компьютеризации мирового сообщества требует использования топлива для получения энергии, что приводит к следующим негативным явлениям: постоянно расходуются запасы невозобновляемых источников энергии или тех источников, которые возобновляются достаточно медленно, чтобы признать их невозобновляемыми на фоне темпов их потребления; получение энергии сопровождается ростом количества углекислого газа (СО<sub>2</sub>); образуется огромное количество загрязняющих веществ.

Для замедления разрушительных для экологии процессов компьютеризации необходимо использовать все возможные способы. В частности, стоит рассмотреть возможности создания «экологически чистых» вычислительных устройств.

Разработка и производство «экологически чистых» компьютеров могут быть обеспечены развитием нескольких направлений:

- разработка новых конструкторских решений и технологий в сфере программного обеспечения
- снижение эмиссионных факторов (электромагнитные поля, рентгеновские и ионизирующие излучения);
- сокращение номенклатуры элементов таблицы Д.И. Менделеева, используемых в процессе изготовления компьютерной техники;
- улучшения в сфере переработки отходов, что должно упростить переработку компьютерного лома, корректно утилизируя различные вредные вещества: бромосодержащие вещества и хлористо-фтористые соединения (кадмий, свинец, ртуть и т.д);
- разработка комбинаций различных материалов, которые позволят создавать дешевые электронные устройства памяти, которые совместят в себе органические материалы, электропроводящий полимерный материал и неорганические компоненты.

Согласно стандарту (EPA Energy Star VESA DPMS), монитор должен поддерживать три энергосберегающих режима – ожидание (Standby), приостановку (Suspend), и «сон» (off). Такие мониторы при простое вычислительного устройства переводятся в соответствующий режим с энергопотреблением, которое значительно ниже штатного при работе.

По оценке Агентства по защите окружающей среды США только соблюдение сертификации энергосбережения изделий программы Energy Star для офисного электрооборудования, купленного в мире до 2010 г. предотвращает ежегодное загрязнение, равное загрязнению 6,5 млн. автомобилей.

Наличие в составе материалов электронных вычислительных устройств фтористо-хлористых соединений углеводорода, бромосодержащих средств защиты от возгорания, а также наличие пластмасс отрицательно сказываются на состоянии экологии. Из пластмасс только 20 % может быть переработано,

остальное требует захоронения, что не является оптимальным сценарием работы с подобными веществами.

В процессе разборки ПЭВМ образуется медьсодержащий лом (в частности, используемые в устройствах провода), алюминий и его сплавы, оловянно-свинцовые припои. Металлический лом может быть переработан и переиспользован.

Стекла люминесцентных экранов электронно-лучевых трубок могут использоваться в качестве сырья для производства новых трубок и керамики.

В процессе разборки компьютерной техники пластмассы сортируют по видам термопласт (полистирол, поливинилхлорид, полиэтилен) и направляют на следующие виды переработок:

- прямая переработка отходов пластмасс ПЭВМ во вторичное сырье, материалы, изделия;
- термическое разложение;
- термическое обезвреживание;
- разработка биоразрушаемых пластмасс.

Важнейшей характеристикой отходов пластмасс является их энергетическая ценность. По химическому составу и теплоте сгорания пластмассы подобны основным ископаемым топливным элементам: природному газу, нефти, углям. Но перевод пластика в топливо не всегда возможен в связи с присутствием в них примесей, приводящих к образованию токсичных соединений при их сжигании в топливных системах.

Одной из перспективных технологий переработки отходов пластмасс является их использование в металлургическом производстве в качестве источника энергии и восстановителей.

### **5.3 Чрезвычайные ситуации**

При возникновении возгорания необходимо сообщить о чрезвычайной ситуации по телефону: 01; 112, обесточить все электрическое оборудование, обеспечить незамедлительную эвакуацию людей, принять меры по ликвида-

ции возгорания.

Для тушения пожара на начальной стадии применяются огнетушители. Так как в помещениях с ПЭВМ наиболее вероятные классы пожаров – «А» и «Е» (т.е. могут гореть в основном твердые вещества, горение которых сопровождается тлением – класс А; или возможны пожары, вызванные возгоранием электроустановок – класс Е), то использовать нужно порошковый и углекислотный огнетушители. Огнетушитель углекислотный ОУ-5 предназначен для тушения любых материалов, предметов и веществ, а также электроустановок, находящихся под напряжением до 1 кВ, применяется для тушения ПЭВМ и оргтехники. При пожаре поднести огнетушитель как можно ближе к огню, направить раструб в очаг пожара, сорвать пломбу (выдернуть чеку), открыть вентиль, нажать на пусковой рычаг, направить струю выходящего газа на огонь. Во время работы раструба нельзя держать рукой, т. к. он имеет очень низкую температуру. Огнетушитель порошковый ОП-5 предназначен для тушения твердых, жидких, газообразных веществ и электроустановок, находящихся под напряжением до 1 кВ, применяется для тушения ПЭВМ и оргтехники. При пожаре поднести огнетушитель к очагу загорания, выдернуть чеку, нажать на рычаг, направить шланг с распылителем на огонь.

Расстояние от возможного очага возгорания до места размещения огнетушителя не должно превышать 20 м, если ПЭВМ установлены в общественных зданиях и сооружениях.

Для исключения паники и быстрой и безопасной эвакуации людей (при возможном задымлении помещений и коридоров), у дверных проемов, выключателей, рубильников, по пути возможной эвакуации для быстрого обнаружения шкафов с первичными средствами пожаротушения и т.п. следует размещать фотолюминесцентные эвакуационные знаки. Комплекс помещений вычислительных центров должен иметь не менее двух самостоятельных эвакуационных выходов.

Для автоматического обнаружения пожаров могут быть использованы любые извещатели. Основные требования к ним состоят в том, чтобы они реагировали на определенный параметр среды. Ручные извещатели предназначены для передачи информации о пожаре с помощью человека и должны размещаться на высоте 1,5 м от уровня пола. Автоматические пожарные извещатели о пожаре, за исключением световых, устанавливаются в помещении на потолке.

## ЗАКЛЮЧЕНИЕ

В процессе работы над проектом был проведен анализ предметной области. На основании этого были выделены особенности устройства многофункционального центра и недостатки функционирования данного учреждения и классического подхода в обслуживании.

Посредством проведенных работ процесс реализации базы данных и программного продукта являлся последовательным и определенным. Применение case-средств при построении диаграмм и моделей в промежуточных этапах помогло в принятии грамотных решений.

Была проделана работа по анализу методов и средств решения проблем автоматизации высоконагруженных отделов учреждения. Была спроектирована информационная система, включающая в себя определение требований, описание структуры на логическом и физическом уровнях. В результате проделанной работы была разработана информационная система для Многофункционального центра в виде взаимодействующего с пользователем чат-бота, позволяющего проводить регистрацию заявителя, его запись на прием, оперативно отвечать на возникающие вопросы используя естественно-языковой интерфейс. Для разработки данной системы использовалась программная библиотека разговорного искусственного интеллекта с открытым исходным кодом Depravlov и механизм Telegram Bot API, позволяющего использовать Telegram-мессенджеры для взаимодействия с пользователем.

Параллельно были изучены и использованы передовые технологии разработки программного обеспечения: объектно-ориентированная парадигма, объектно-реляционное отображение, веб-скрейпинг. Особое внимание в данной работе было уделено технологиям машинного обучения, результаты работы которых в последнее время называют искусственным интеллектом.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Документация python-telegram-bot [Электронный ресурс] – Режим доступа: <https://python-telegram-bot.org/>. – 25.04.2022.
- 2 Документация deepravlov [Электронный ресурс] – Режим доступа: <https://deepravlov.ai/>. – 25.04.2022.
- 3 Документация transformers [Электронный ресурс] – Режим доступа: <https://huggingface.co/docs/transformers/index>. – 25.04.2022.
- 4 Документация opencv [Электронный ресурс] – Режим доступа: <https://docs.opencv.org/4.x/>. – 25.04.2022.
- 5 Дейтел Пол, Дейтел Харви. Python: Искусственный интеллект, большие данные и облачные вычисления. – СПб.: Питер, 2020. – 864 – с.
- 6 Жеребцова, Ю. А., Чижик, А. В. Сравнение моделей векторного представления текстов в задаче создания чат-бота // Вестник Новосибирского Государственного Университета. 2020. – № 3 – С. 16-34.
- 7 Информационный портал habr [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/446530/> – 25.04.2022.
- 8 Информационный портал habr [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/otus/blog/574624/> – 25.04.2022
- 9 Информационный портал habr [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/ods/blog/329410/> – 25.04.2022
- 10 Информационный портал habr [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/487358/> – 25.04.2022.
- 11 Клеппман М. Высоконагруженные приложения. Программирование, масштабирование, поддержка. – СПб.: Питер, 2018. – 640 с.
- 12 Ключева, И.А. Современные возможности и примеры внедрения машинного обучения // Оригинальные исследования. – 2021. – № 7 – С. 12-32.
- 13 Луридас, Панос. Алгоритмы для начинающих: теория и практика для разработчика – Москва: Эксмо, 2018. – 608 с.

- 14 Маккини У. Python и анализ данных / пер. с англ. А.А. Слинкина. – М.: ДМК Пресс, 2020. – 540 с.
- 15 Новиков Б.А. Основы технологий баз данных: учебное пособие / Б.А. Новиков, Е.А. Горшкова, Н.Г. Графеева; под ред. Е.В. Рогова. – 2-е изд. – М.: ДМК Пресс, 2020. – 582 с.
- 16 Осипов Д.Л. Технологии проектирования баз данных. – М.: ДМК Пресс, 2019. – 498с.
- 17 Рашка, Себастьян, Мирджалили, Вахид. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2, 3-е изд.: Пер. с англ. - СПб.: ООО "Диалектика", 2020. - 848 с.
- 18 Сибуя М., Ямамото Т. Алгоритмы обработки данных: Пер. с япон. – М.: «Мир», 1986. – 218 с.
- 19 Филиппова, А.С. Взаимодействие с чат-ботами на основе машинного обучения // Альманах научных работ молодых ученых университета ИТМО. – 2018. – С. 365-366.
- 20 Чио К., Фримэн Д. Машинное обучение и безопасность / перс. с англ. А. В. Снастина. – М.: ДМК Пресс, 2020. – 388 с.
- 21 Kocharyan A.M. Customer support system of a multifunctional public services center based on machine learning. // Collection of Scientific Papers based on the results of an XVII international scientific conference Scientific achievements of the third millennium (February 15, 2022, New York, USA).



## Продолжение ПРИЛОЖЕНИЯ А

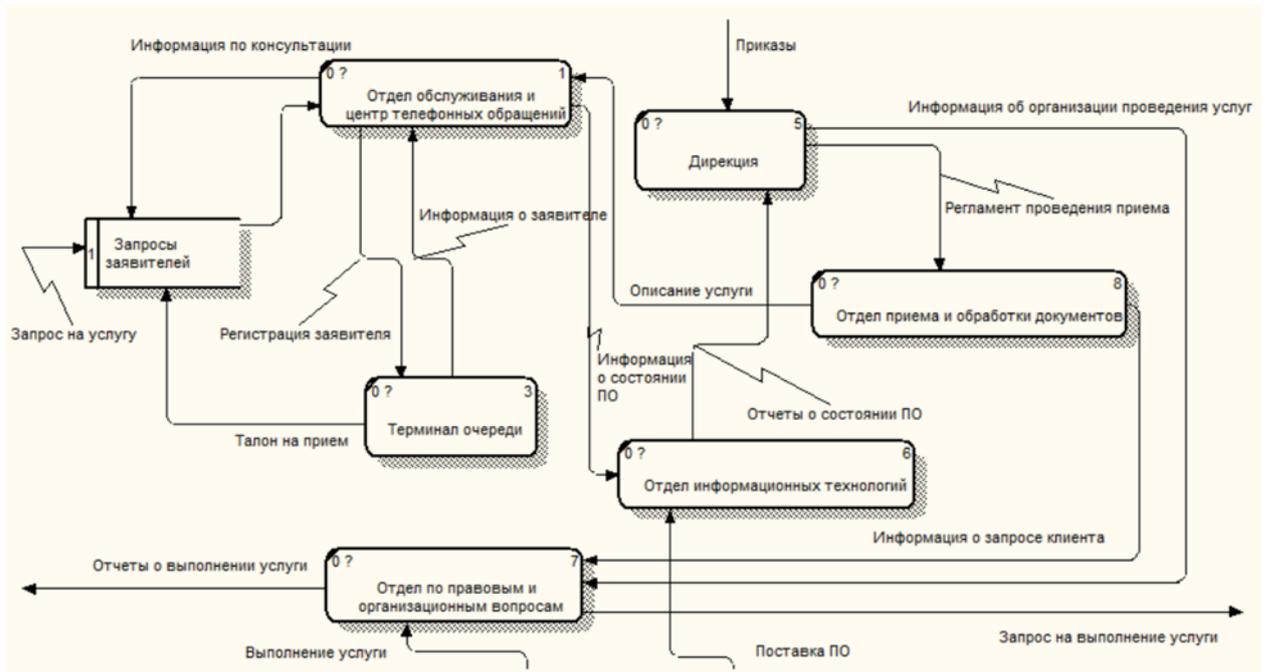


Рисунок А.3 – Внутренний документооборот



Рисунок А.4 - Диаграмма вариантов использования системы

## Продолжение ПРИЛОЖЕНИЯ А

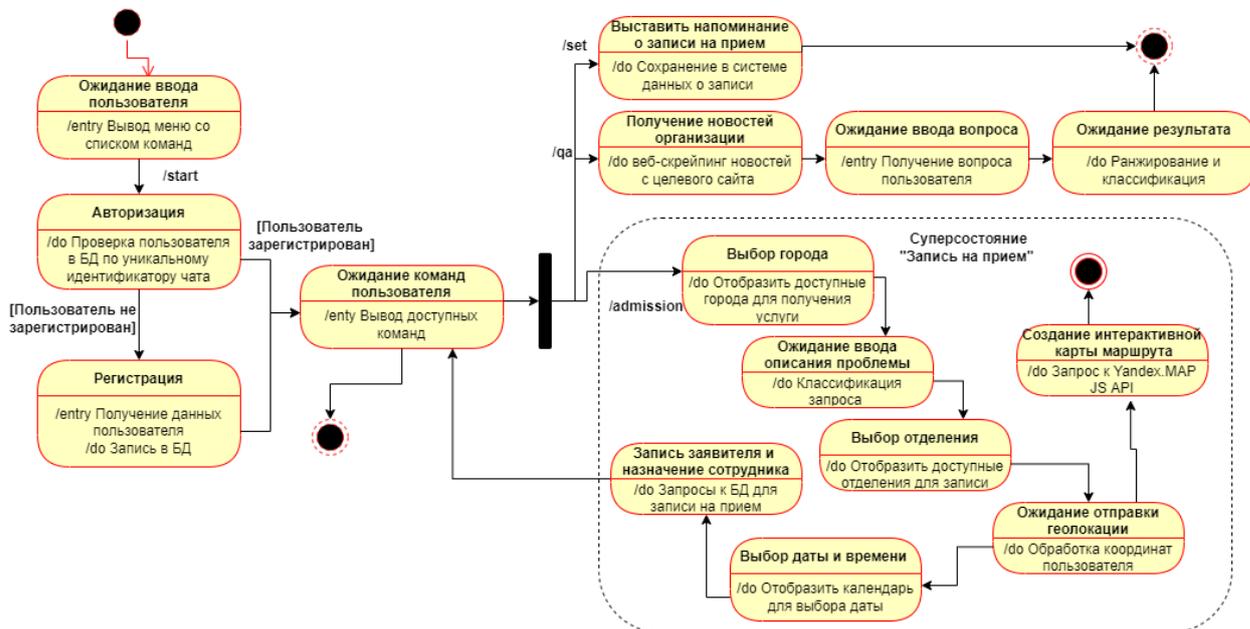


Рисунок А.5 - Диаграмма состояний системы

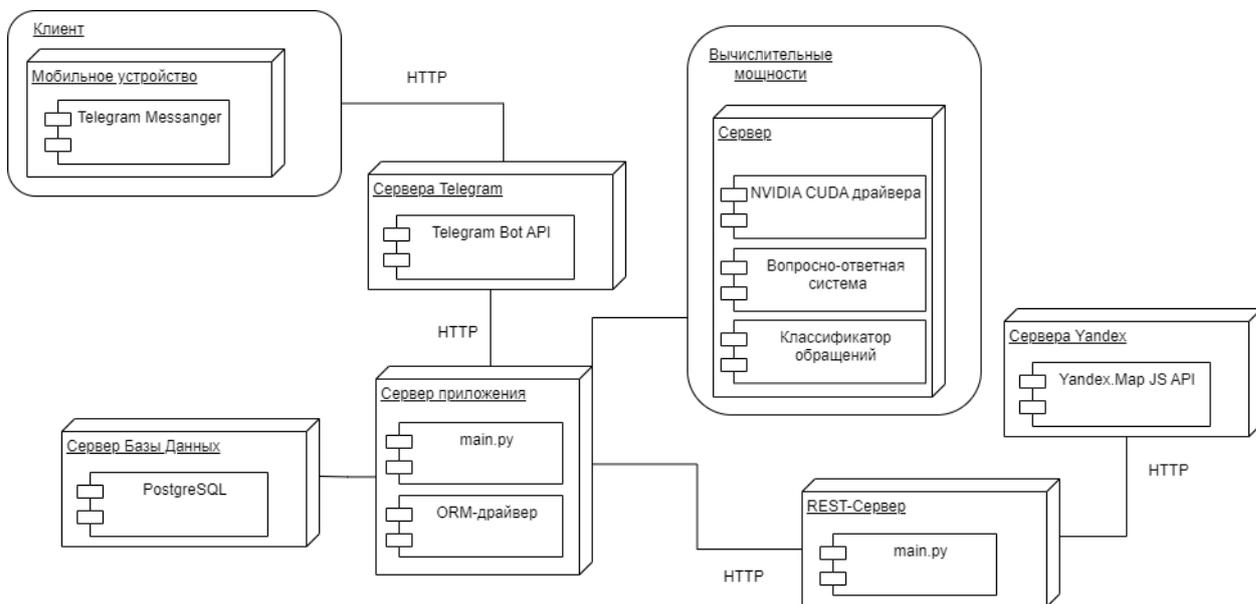


Рисунок А.6 - Диаграмма развертывания системы

## Продолжение ПРИЛОЖЕНИЯ А

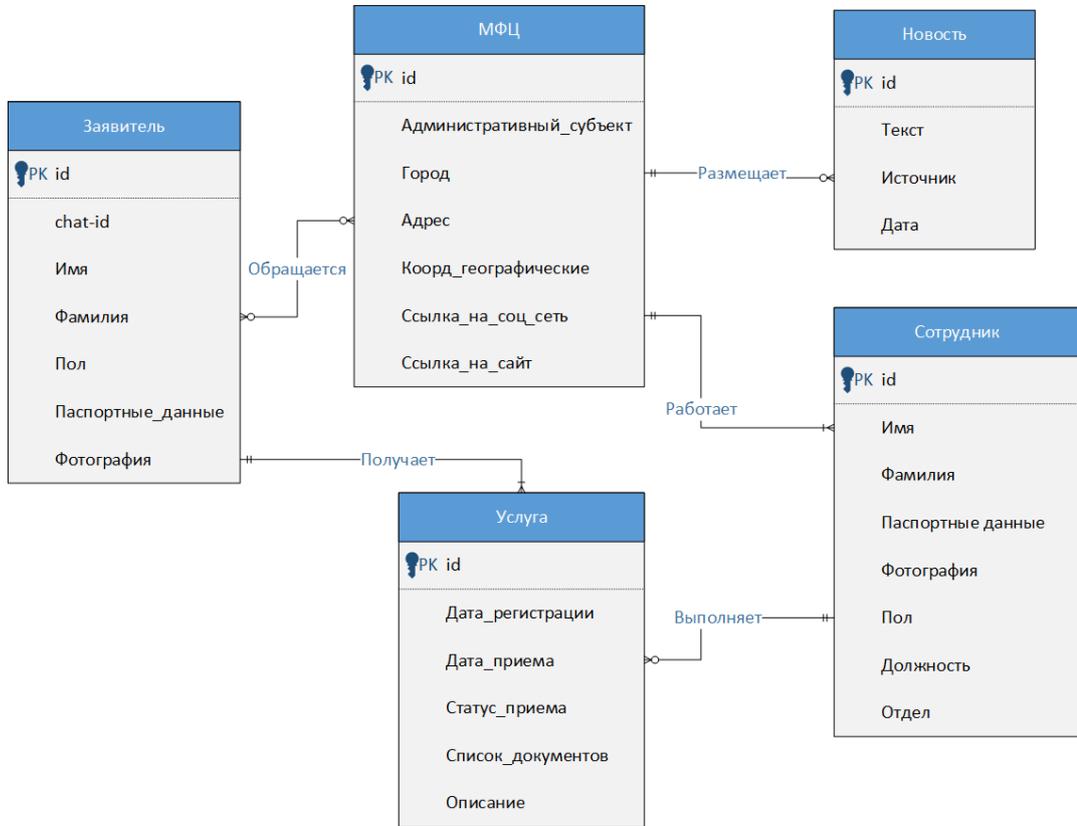


Рисунок А.7 – Инфологическая модель

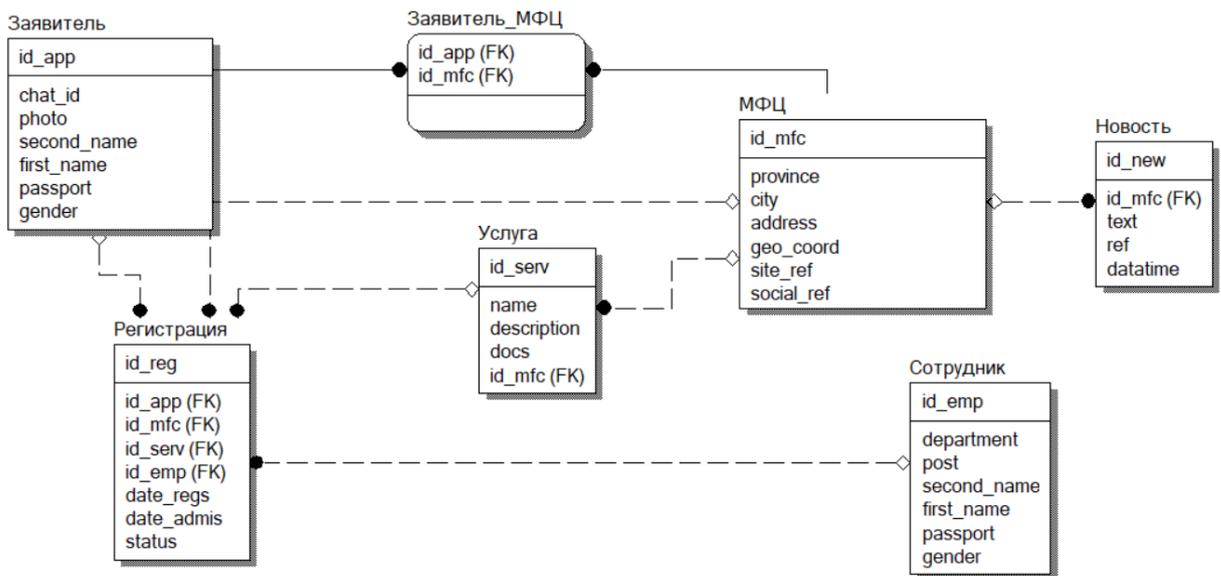


Рисунок А.8 – Логическая модель

## Продолжение ПРИЛОЖЕНИЯ А

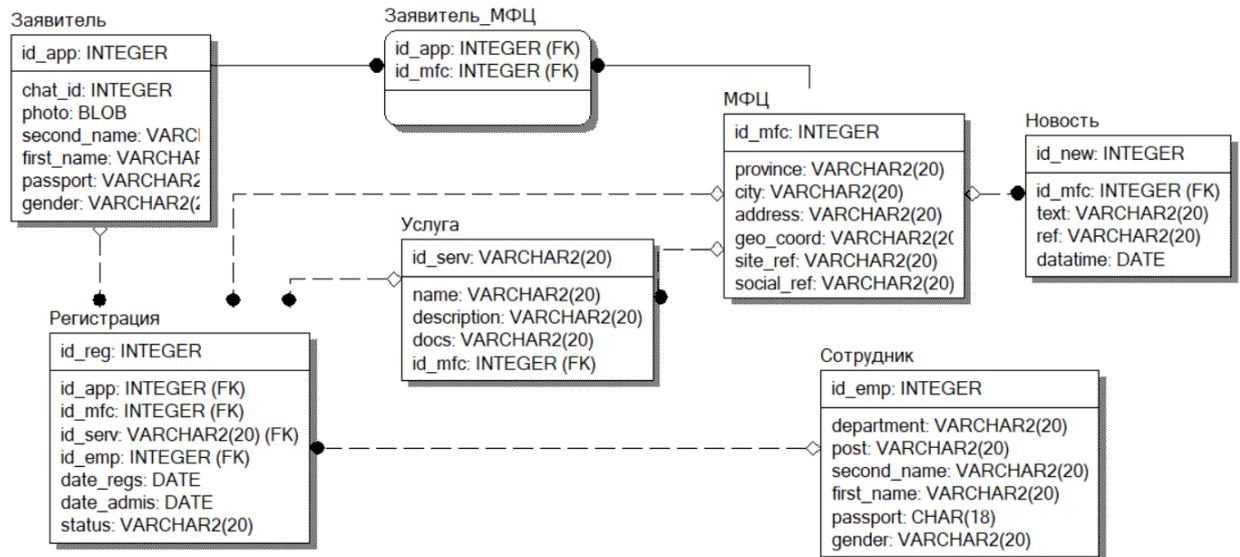


Рисунок А.9 – Физическая модель

## ПРИЛОЖЕНИЕ Б

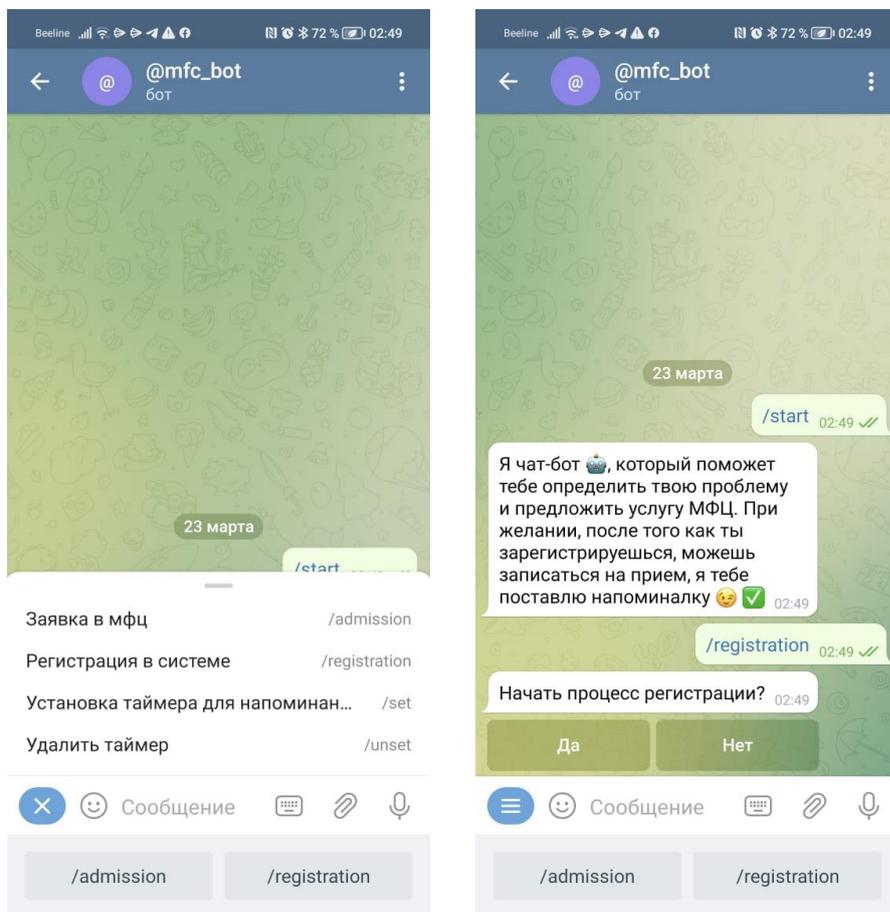


Рисунок Б.2 – Команды бота и приветствие

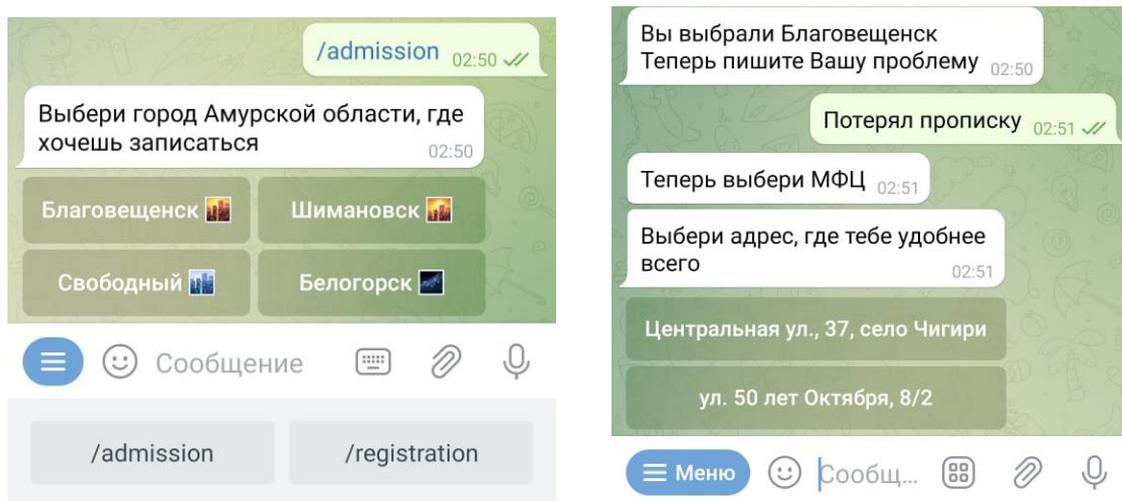


Рисунок Б.3 – Запись на прием

## Продолжение ПРИЛОЖЕНИЯ Б

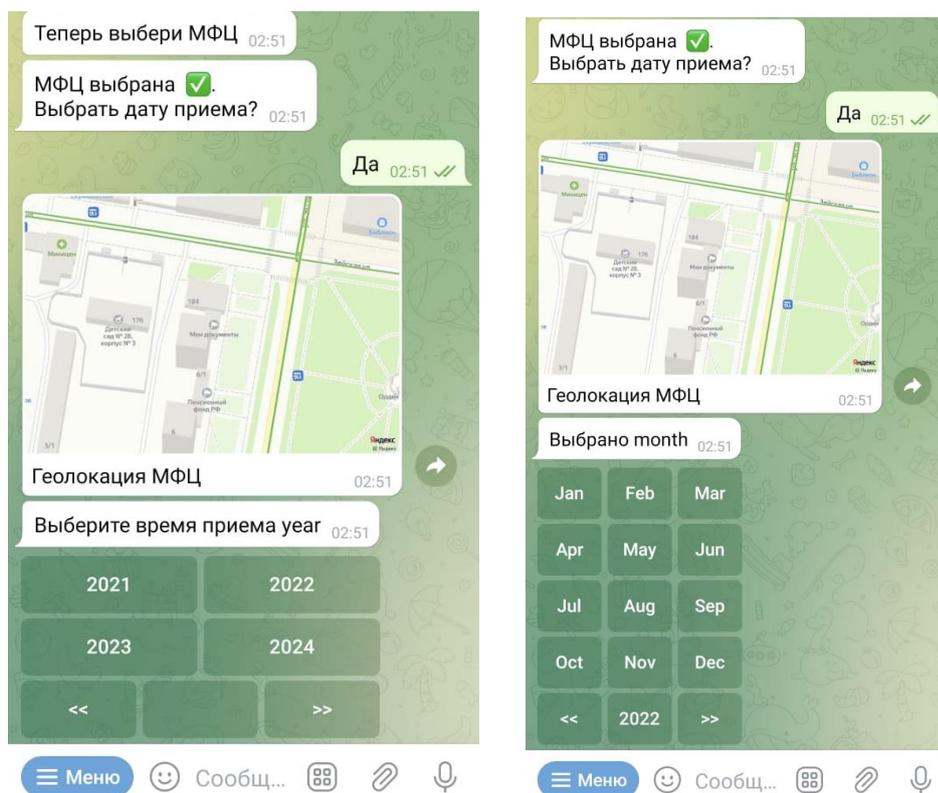


Рисунок Б.4 – Выбор года и месяца при записи на услугу

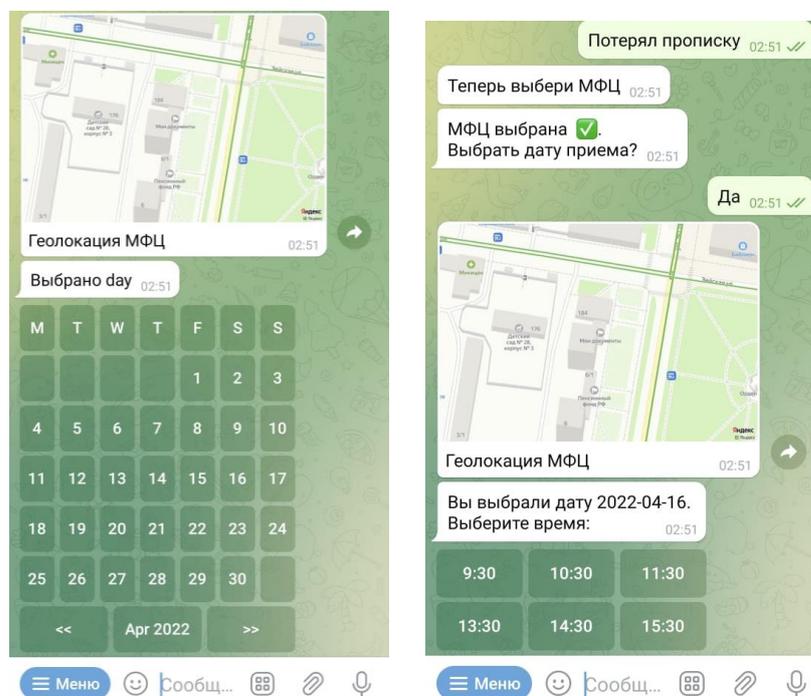


Рисунок Б.5 – Выбор числа месяца и времени при записи на услугу

# ПРИЛОЖЕНИЕ В

## Техническое задание

### **1. Введение**

#### **1.1. Наименование программы**

Разработка информационной системы «Чат-бот для работы с обращениями граждан в МФЦ»

#### **1.2. Кратка характеристика области применения**

Чат-бот предназначен для использования в мессенджерах для получения услуги или оперативной консультации.

### **2. Основание для разработки**

#### **2.1. Основания для проведения разработки**

Основанием для проведения разработки является выполнение курсовой работы по дисциплине «Проектирование АСОИУ».

#### **2.2. Наименование и условное обозначение темы разработки**

Наименование темы разработки – «Чат-бот для работы с обращениями граждан в МФЦ»

Условное обозначение темы разработки – «Чат-бот для работы с обращениями граждан в МФЦ»

### **3. Назначение разработки**

#### **3.1. Функциональное назначение программы**

Получение оперативной и быстрой консультации или запись на услугу.

#### **3.2. Конечные пользователи программы**

Конечными пользователями программы должны являться пользователи услуг МФЦ, имеющие доступ к социальным сетям и мессенджерам.

### **4. Требования к программе**

#### **4.1. Требования к функциональным характеристикам**

##### **4.1.1. Требования к составу выполняемых функций**

## Продолжение ПРИЛОЖЕНИЯ В

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- 1) Регистрация пользователя;
- 2) Авторизация пользователя;
- 3) Получение справочной информации;
- 4) Запись на услугу;
- 5) Получение услуги.

### **4.1.2. Требование к временным характеристикам**

Обработка ввода информации и показа уведомления не должны превышать 2 сек.

## **4.2. Требования к информационной и программной совместимости**

### **4.2.1. Требования к информационным структурам и методам решения**

Пользовательский интерфейс должен быть интуитивно понятным и содержать подсказки.

### **4.2.2. Требования к исходным кодам и языкам программирования**

Исходные коды программы должны быть реализованы на языке Python. В качестве интеграционной среды разработки программы должна быть использована среда PyCharm community.

### **4.2.3. Требования к программным средствам, используемым программой**

Основа для системы – серверная часть, в которой будет храниться база данных и обрабатываются запросы пользователя

### **4.2.4. Требования к защите информации и программ**

В системе должен быть обеспечен надлежащий уровень защиты информации в соответствии с законом о защите информации и программного комплекса в целом от несанкционированного доступа – «Об информации, инфор-

## Продолжение ПРИЛОЖЕНИЯ В

матизации и защите информации» РФ N 24-ФЗ от 20.02.95.

### **4.3. Специальные требования**

Программа должна обеспечивать высокую защиту данных и удобный и быстрый просмотр необходимой информации посредством отчетов.

### **5. Требования программной документации**

#### **5.1. Предварительный состав программной документации**

Состав программной документации должен включать в себя:

- 1) Техническое задание;
- 2) Спецификация;
- 3) Текст программы;
- 4) Описание программы;
- 5) Программу и методики испытаний;
- 6) Пояснительная записка;
- 7) Ведомость эксплуатационных документов;
- 8) Формуляр
- 9) Описание применения;
- 10) Руководство системного администратора;
- 11) Руководство пользователя.

#### **5.2. Специальные требования к программной документации**

Специальные требования к программной документации не представляются.

### **6. Технико-экономические показатели**

#### **6.1. Ориентировочная экономическая эффективность**

Ориентировочная экономическая эффективность пока что не рассчитывается.

### **7. Стадии и этапы разработки**

#### **7.1. Стадии разработки**

## Продолжение ПРИЛОЖЕНИЯ В

Разработка должна быть проведена в 3 стадии:

- 1) Разработка технического задания;
- 2) Рабочее проектирование;
- 3) Внедрение.

### **7.2. Этапы разработки**

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

- 1) Разработка программы;
- 2) Разработка программной документации;
- 3) Испытания программы.

### **7.3. Содержание работ по этапам**

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- 1) Постановка задачи;
- 2) Определение и уточнение требований к техническим средствам;
- 3) Определение требований к программе;
- 4) Определение стадий, этапов и сроков разработки программы и документации на нее;
- 5) Выбор языков программирования;
- 6) Согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями

## Продолжение ПРИЛОЖЕНИЯ В

ГОСТ 19.101-77 и требованием п. «Предварительный состав программной документации» настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

- 1) Разработка, согласование и утверждение программы и методики испытаний;
- 2) Проведение приемно-сдаточных испытаний;
- 3) Корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена подготовка и передача программы и программной документации в эксплуатацию.