

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«Вычислительная техника»

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Магистерская программа: «Прикладной искусственный интеллект»

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

**ПРОГРАММНЫЙ МОДУЛЬ ОБРАБОТКИ
ЕСТЕСТВЕННОГО ЯЗЫКА ДЛЯ ОНЛАЙН-
КОНСУЛЬТАЦИЙ ПОЛЬЗОВАТЕЛЕЙ
ЭЛЕКТРОННОГО ДЕКАНАТА ЭИОС ПГУ**

Студент

Игошин Иван Валерьевич
(подпись, дата) (ФИО полностью)

Руководитель

А.М.Бершадский
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр

А.С.Бычков
(подпись, дата) (фамилия, инициалы)

Рецензент

зав. каф. Программного обеспечения ИИ ПГУ
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 28.05.2025 № 16)

Заведующий кафедрой

М.А. Митрохин
(подпись) (фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от

5 июня 2025 № 6)

Секретарь ГЭК

Е.И. Гурин
(подпись) (фамилия, инициалы)

Пенза 2025

УТВЕРЖДАЮ

Зав. кафедрой ВТ

«29» 09

М.А. Митрохин

2023 г.

ЗАДАНИЕ

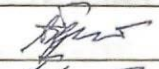
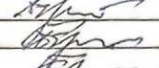
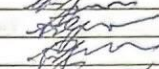
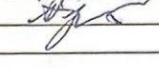
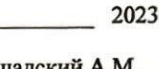
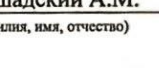
на выполнение магистерской работы

1. Студент гр. 23ВВИм1 факультета ВТ направление 09.04.01
Игошин Иван Валерьевич
(фамилия, имя, отчество)
2. Руководитель ВКР профессор кафедры САПР, ПГУ
(Должность и место работы)
Бершадский Александр Моисеевич
(фамилия, имя, отчество)
3. Время выполнения магистерской работы с _____ по _____
4. Тема магистерской работы Программный модуль обработки естественного языка для онлайн-консультаций пользователей электронного деканата ЭИОС ПГУ
Тема утверждена приказом ПГУ № 986/0 от 29.09.2023 г.
5. Задание на магистерскую работу: предмет и назначение разработки, функциональные, программные, технические требования и т.п.
 - 5.1. Разрабатываемый модуль предназначен для автоматизации процесса оказания консультативных услуг сотрудниками технической поддержки по вопросам взаимодействия с Электронным деканатом ЭИОС ПГУ различных групп пользователей.
 - 5.2 Требования к функциональности: Модуль позволяет:
осуществлять переход в мессенджер «Telegram» из интерфейса Moodle;
осуществлять идентификацию пользователей с целью безопасности;
принимать сообщения пользователей, и определять тип контекста входящего вопроса;
определять роль пользователя в системе для сохранения конфиденциальности инструкций по использованию интерфейсов Электронного деканата;
формирование и выдача сгенерированных нейронной сетью ответов на соответствующие вопросы пользователей.
 - 5.3 Требования к ИО: должна быть использована база данных LMS Moodle, являющаяся элементом Электронной информационной образовательной среды Пензенского государственного университета, и выполнена модификация (расширение структуры БД с поддержкой целостности данных).
 - 5.4 Требования к ПО: dbForge Studio for MySQL, XAMPP, Python, PHP, Telegram, GigaIDE, HTML
 - 5.5 Требования к аппаратному обеспечению: Операционная система Microsoft Windows 7 и выше;
6. Содержание пояснительной записки магистерской работы: перечень подлежащих разработке, расчетов, обоснований, описаний
 - 6.1 Анализ предметной области и аналогов, техническое задание
 - 6.2 Выбор средств разработки
 - 6.3 Структура подсистемы

6.4 Структура базы данных подсистемы

6.5 Разработка программного обеспечения

7. Календарный график выполнения магистерской работы

Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1. Анализ предметной области и аналогов, техническое задание	15 %	25.11.2023	
2. Выбор средств разработки	10 %	25.01.2024	
3. Структура подсистемы	15 %	25.04.2024	
4. Структура базы данных подсистемы	15%	25.07.2024	
5. Разработка ПО	35 %	25.03.2025	
6. Оформление ПЗ	10 %	30.05.2025	
7. Нормоконтроль		31.05.2025	

Дата выдачи задания

«29» 09 2023

Руководитель магистерской работы

 Бершадский А.М.
(фамилия, имя, отчество)

Задание к исполнению принял

«29» 09 2023

Студент

 Игошин И.В.
(фамилия, имя, отчество)

Магистерскую работу к защите допустить

Декан ФВТ  04.06.2025 / Л.Р. Фионова
(Подпись, дата)

Реферат

Пояснительная записка содержит 69 листа, 29 рисунков, 19 источников.

**ТЕХНИЧЕСКАЯ ПОДДЕРЖКА, ЭЛЕКТРОННАЯ
ИНФОРМАЦИОННАЯ ОБРАЗОВАТЕЛЬНАЯ СРЕДА, ЭЛЕКТРОННЫМ
ДЕКАНАТОМ, НЕЙРОННЫЕ СЕТИ, ЧАТ-БОТ, БАЗА ДАННЫХ, MYSQL,
PYTHON, TENSORFLOW**

Объектом разработки является модуль для оказания услуг технической поддержки в рамках взаимодействия с Электронным деканатом ЭИОС ПГУ.

Цель работы – увеличение качества и скорости оказаний услуг технической поддержки для пользователей ЭИОС ПГУ и повышения удобства работы с данной системой в целом. Модуль будет встроен в Электронную информационную образовательную систему Пензенского государственного университета.

При разработке были последовательно решены следующие задачи: изучение предметной области, анализ аналогов разрабатываемой системы, выбор средств разработки и обоснование их выбора, проектирование структуры разрабатываемого модуля и создания программного обеспечения для его функционирования.

В результате выполнения работы был разработан программный модуль для оказания услуг технической поддержки пользователям Электронного деканата информационной системы Пензенского государственного университета. Созданный модуль был разработан с помощью средства для проектирования и администрирования баз данных dbForge Studio for MySQL, среды разработки GigaIDE, мессенджера Telegram, языков программирования Python и PHP, а также библиотеки TensorFlow для языка Python.

					ПГУ 1.09.04.01.06.001 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Игошин И.В.		29.05.25	Программный модуль обработки естественного языка для онлайн- консультаций пользователей электронного деканата ЭИОС ПГУ Пояснительная записка	Лит.	Лист	Листов
Провер.		Бершадский А.М.		30.05.25			2	69
Н. Контр.		Бычков А.С.		30.05.25		гр. 23ВВИм1		
Утв.		Митрохин М. А.		30.05.25				

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ.....	4
ВВЕДЕНИЕ.....	5
1 Анализ предметной области и аналогов, техническое задание.....	8
1.1 Анализ предметной области.....	8
1.2 Техническое задание.....	9
1.3 Выбор способа решения задачи.....	11
1.4 Обзор аналогов разрабатываемой системы.....	13
2 Выбор средств разработки.....	16
3 Структуры подсистемы.....	25
4 Структура базы данных подсистемы.....	31
5 Разработка программного обеспечения.....	40
5.1 Интеграция Moodle и Telegram.....	40
5.2 Обучение модели нейронной сети.....	51
5.3 Тестирование ПО.....	54
ЗАКЛЮЧЕНИЕ.....	59
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	60
ПРИЛОЖЕНИЕ А Фрагмент обучающей выборки.....	63
ПРИЛОЖЕНИЕ Б Презентация.....	66

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

RHC – Рекуррентная нейронная сеть

ЭД – Электронный деканат

ЭИОС – Электронная информационная образовательная среда

HTTP – HyperText Transfer Protocol

IDE – Integrated development environment

JSON – JavaScript Object Notation

ML – Machine learning

URL – Uniform Resource Locator

VCS – Version Control System

ВВЕДЕНИЕ

В настоящее время в соответствии с ФГОС ВО университеты РФ используют в образовательном процессе средства автоматизации. В Пензенском государственном университете, на основе свободно распространяемого программного обеспечения с открытым кодом разработана электронная информационная образовательная среда (ЭИОС ПГУ), отвечающая всем необходимым требованиям. Ее пользователями являются все студенты университета, преподаватели и сотрудники ВУЗа. Важной задачей в подобной развивающейся информационной системе является правильное и удобное отображение информации и наличие согласованного интерфейса системы при взаимодействии с пользователем. В большинстве случаев такое взаимодействие налажено с помощью интерфейсов на веб – страницах.

Однако проектирование интерфейса, который отвечает вышеперечисленным требованиям, очень часто не является тривиальной задачей. К тому же каждая задача может требует своего отдельного интерфейса, а зачастую и не одного, если объём и сложность задачи велики. ЭИОС, как и любая другая информационная система, обладающая широким спектром функций, реализация которых предусмотрена с помощью интерфейсов, может создать непонимание у пользователей, как пользоваться определённым функционалом или почему некоторые действия им недоступны.

Поэтому к таким системам будут справедливы требования по обеспечению удобства пользования ею. Особенно остро этот вопрос встаёт в крупных университетах, где число пользователей достигает десятков тысяч человек, а спектр выполняемых ими действий весьма широк, и не оптимально и неудобно спроектированный интерфейс может затруднять выполнения своих непосредственных обязанностей для огромного количества людей, что в конечном итоге будет влиять на качество образовательных услуг для огромного количества обучаемых. Таким

образом, для обеспечения озвученных требований будет актуальна разработка подсистемы поддержки пользователей.

Актуальность данной работы также подкрепляется тем фактом, что помимо того, что существует необходимость создание модуля для консультации пользователей как такового, так ещё крайне существенными требованиями являются скорость работы модуля поддержки, поскольку в нём не будет никакой необходимости, если время его работы не будет существенно быстрее, чем скорость оказания услуг сотрудниками технической поддержки, и корректность его работы, что логично, ввиду того, что нет никакой практической пользы от программного продукта, который не может решать задачи для которых он предназначен.

Таким образом, можно сформулировать конечную цель работы: увеличение качества и скорости оказаний услуг технической поддержки для пользователей ЭИОС ПГУ и повышения удобства работы с данной системой в целом.

На решение проблем подобного рода направлены работы таких следующих деятелей: Галеев Д.Т., Титов Д.В., Волков Д.А., Култыгин О.П., Митина О.А., Жаров В.В., Мешкова Е.В., Фроленко Д.И., Курейчик В.В., Родзин С.И., Бова В.В., Фирова Д.В., Мукашев А.Ш., Байзакова С.М. и др.

Однако данные решения не способны покрыть потребности для выполнения текущей задачи проекта, что диктует необходимость дальнейшего изучения и усовершенствования методов машинного обучения и обработки естественного языка, а также их применения внутри образовательной среды.

Таким образом, объектом данной разработки является программный модуль для консультации пользователей Электронного деканата ЭИОС ПГУ.

Предмет разработки – нейронная сеть, умеющая распознавать естественный язык, а также формировать для пользователей ответы, соответствующие задаваемым ими вопросам.

В соответствии с целью, объектом, и предметом сформулированы задачи работы:

- анализ существующих аналогов,
- определить конечную форму разработанного продукта, через которую будет осуществлено взаимодействие с пользователями,
- определить средства разработки,
- сформировать базу знаний для обучения нейронной сети на основе наиболее популярных вопросов пользователей Электронного деканата на форумах для консультаций,
- обучение нейронной сети,
- тестирование разработанного модуля.

1 Анализ предметной области и аналогов, техническое задание

1.1 Анализ предметной области

Современный этап развития общества характеризуется активным внедрением цифровых технологий практически во все сферы жизни, включая образование. Цифровая трансформация оказывает существенное влияние на традиционные формы предоставления образовательных услуг, внедряя принципиально новые подходы к обучению, взаимодействию студентов и преподавателей, администрированию и техническому обслуживанию информационных систем. Этот тренд отражает глобальную тенденцию к повышению конкурентоспособности учреждений высшего образования и обеспечению качественной подготовки выпускников для рынка труда XXI века.

Современные университеты активно используют различные информационные системы для управления учебной деятельностью, академическими ресурсами, финансовыми потоками и инфраструктурой. Однако эффективное функционирование этих систем зависит от надежности и оперативности технического обслуживания, которое должно обеспечивать бесперебойную работу всей IT-инфраструктуры учебного заведения. Возникают важные вопросы относительно организации, структуры и функционирования службы технической поддержки, влияющей на успешность цифровой трансформации и повседневную жизнь учащихся и преподавателей.

Сегодня российские высшие учебные заведения стоят перед серьезными вызовами, связанными с глобальной конкуренцией, качеством подготовки специалистов и адаптацией к постоянно меняющимся экономическим и социальным условиям. Важнейшую роль играет способность оперативно реагировать на потребности студентов и преподавателей, устранять технические неполадки и поддерживать стабильность функционирования информационных систем. Таким образом, проблема своевременного и качественного технического обслуживания

приобретает особую актуальность, особенно в свете масштабных планов цифровизации высшей школы в нашей стране [1].

Исследование вопросов организации и оптимизации технической поддержки информационных систем университетов способно внести значительный вклад в решение конкретных задач, стоящих перед руководством и сотрудниками учебных заведений. Полученные выводы могут использоваться для разработки рекомендаций по улучшению структуры и механизмов функционирования службы технической поддержки, повышению квалификации сотрудников, внедрению инновационных решений и совершенствованию стандартов качества обслуживания. Таким образом, данная тематика соответствует общим интересам и нуждам современного высшего образования, что придает ей высокую актуальность и прикладную ценность [2].

Подводя итог, можно сказать, что исследование вопросов технической поддержки информационных систем университетов является крайне важным и востребованным направлением научных изысканий, имеющим большое практическое значение для дальнейшего развития и модернизации отечественной системы высшего образования.

1.2 Техническое задание

Цель работы

Увеличение качества и скорости оказаний услуг технической поддержки для пользователей ЭИОС ПГУ и повышения удобства работы с данной системой в целом.

Описание разрабатываемого продукта

Конечный продукт должен быть представлен в форме бота Telegram, управление которым осуществляет специально созданная под задачи проекта нейронная сеть. Разработанный бот будет осуществлять взаимодействие с пользователями в формате «вопрос-ответ».

Входные данные

Система должна принимать на вход текст, который пользователь вводит при обращении с ботом.

Требования к разрабатываемому продукту:

– требования к надёжности:

1) ресурс должен быть доступен пользователям в любое время суток каждый день без исключений;

2) ресурс должен функционировать без вмешательства технического специалиста;

3) учитывая специфику и сложность вопроса, а также потенциальные проблемы с подключением к Интернету, ответ на вопрос пользователя не должен занимать больше 30-ти секунд;

4) пользователь всегда должен получать какой-либо ответ от системы, т.е. диалог всегда должен завершаться сообщением от разработанного бота, вне зависимости от содержимого сообщения пользователя;

– требования к функциональности:

1) система должна формировать свой ответ пользователю исходя из анализа ключевых слов в сообщении пользователя;

2) ответ пользователю на его запрос должен присылать одним сообщением вне зависимости от объёма контента сообщения от системы;

3) на одно сообщение пользователя должно отправляться одно ответное сообщение;

4) ответ на вопрос должен сформулирован так, чтобы он был понятен пользователю «начального уровня»;

5) если на вопрос пользователя система не может дать соответствующий ответ, то в ответном сообщении должны содержаться контакты технической поддержки ЭИОС ПГУ;

– требования к этической составляющей взаимодействия с пользователем:

1) бот должен уметь распознавать и отличать прикладные сообщения от общих (например, «Привет», «До свидания», «Как дела?») и давать на них соответствующие ответы;

2) перед первым использованием система должна отправлять приветственное сообщение, в котором должно содержаться приветствие для пользователя;

3) в ответах бота пользователю не должна содержаться нецензурная лексика, а также система должна оповещать пользователя о недопустимости её применения, если он использовал её в своём сообщении.

Сроки реализации

Конечный продукт должен исправно и в полном объёме функционировать к 05.06.2025г.

1.3 Выбор способа решения задачи

В подобных разработках могут оказаться полезными боты, которые зарекомендовали себя как удачное решение и во многих сферах выполняют роль технической поддержки.

Однако круг вопросов, охватываемых в информационной системе высшего учебного заведения слишком велик, поэтому в данном случае применение нейросети будет адекватным решением.

Самым трудоёмким процессом при создании такой нейросети будет составление базы знаний, которая с появлением новых задач будет динамически расширяться.

Объёма такой базы хватит нейросети для её эффективного обучения и применения для консультирования учащихся и сотрудников университета по широкому спектру вопросов. Связь пользователей с указанной нейросетью будет осуществляться как раз через бот (например, Телеграмм-бот).

В многих случаях, в зависимости от организации общей информационной системы вуза, различные подсистемы того или иного университета решают кардинально разные задачи, поэтому в таких ситуаций возможно рассматривать вариант создания отдельных нейросетей и ботов

индивидуально для каждого компонента, особенно при наличии у каждой из подсистем очень объёмного спектра вопросов. Однако вариант с одной нейросетью, которая будет консультировать персонал и студентов абсолютно по всем вопросам, касающихся всей информационной системы университета, является более рациональным и оптимальным.

Использование нейросетей будет отвечать современным тенденциям по организации масштабных информационных систем.

Одной из самых весомых способностью нейронных сетей является способность самостоятельно «обучаться». Такие сети способны усиливать необходимые ассоциации, а менее важные, отбрасывать. Это позволяет нейронным сетям лучше распознавать закономерности, при этом «присутствие» человека в таком случае минимизируется.

Нейросети могут быть использоваться для обработки естественного языка. Это позволяет аппаратным средствам понимать язык людей и отвечать на человеческие запросы.

Это может быть особенно полезно в сфере обслуживания клиентов, где можно использовать нейронные сети для более точного и эффективного ответа на запросы клиентов. Его также можно использовать для создания контента, а также для отбора и анализа данных [2].

В предлагаемом решении для информационной системы университета, нейронная сеть должна будет выполнять задачу по обработке естественного языка, по результатам которой будет подобран наиболее подходящий по данной теме ответ. Ключевой задачей будет являться формирование не просто большой выборки, но, в том числе, такой, которая могла бы давать правильный ответ на заданный вопрос, даже несмотря на возможно не совсем корректный запрос к системе со стороны пользователя.

Таким образом, в качестве основной концепции разработки принято внедрение в информационную систему университета нейросети в качестве консультативного и вспомогательного её элемента.

1.4 Обзор аналогов разрабатываемой системы

В последние годы в российских университетах широкое распространение получила практика внедрения автоматизированных систем технической поддержки на основе искусственного интеллекта и чат-ботов. Подобные системы служат эффективным средством оперативного разрешения обращений студентов и преподавателей, минимизируя нагрузку на сотрудников технической поддержки и повышая общий уровень удовлетворённости пользователей услугами информационной системы университета.

Ниже приведён перечень ведущих российских вузов, внедривших или находящихся в процессе внедрения чат-ботов для решения технических задач и поддержки пользователей.

- Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ). В Высшей школе экономики функционирует специализированный чат-бот, встроенный в основную информационную систему вуза. Основная цель — быстрая обработка типовых запросов студентов, таких как восстановление паролей, навигация по расписанию занятий, подача документов на отсрочку сдачи экзаменов и помощь в поиске нужных материалов в электронной библиотеке [3],

- Санкт-Петербургский государственный университет (СПбГУ). Санкт-Петербургский госуниверситет разработал и внедрил механизм интеллектуальной поддержки пользователей на базе чат-бота. Студенты могут получать сведения о состоянии своей академической документации, правила доступа к закрытой информации, порядок прохождения аттестаций и экзаменационных сессий, а также устранение типичных технических препятствий, возникающих при использовании учебно-методических ресурсов [4],

- Московский физико-технический институт (МФТИ) активно внедряет новую технологию автоматизации технической поддержки, используя нейросети и обработку естественного языка. Модуль служит для

оперативной помощи студентам в вопросах аутентификации, восстановления учётных записей, загрузки файлов и заполнения заявочных документов на экзамены и зачёты [5],

– Южный федеральный университет (ЮФУ) внедрил модуль технической поддержки, основанный на технологиях искусственного интеллекта. Чат-бот доступен студентам и сотрудникам для получения оперативной информации по широкому кругу технических вопросов, таких как доступ к платформенным ресурсам, работа с мобильными устройствами и устранение небольших сбоев в информационной среде [6],

– Уральский федеральный университет имени первого президента России Б.Н. Ельцина (УрФУ) реализует комплекс мер по увеличению доступности технической поддержки посредством внедрения чат-бота. Пользователи могут получать детальные инструкции по самостоятельному исправлению простейших технических проблем, возникших при доступе к сетевым ресурсам университета, таким как интернет-сервисы, серверы печати и лабораторные устройства [7],

– Российский университет дружбы народов (РУДН) проводит эксперименты с внедрением специализированных модулей технической поддержки, основанных на технологиях искусственного интеллекта. Бот нацелен на обслуживание русскоязычных и иностранных студентов, помогая им решать стандартные технические вопросы, такие как оформление пропусков, запись на онлайн-консультации и устранение дефектов при входе в цифровую образовательную среду [8].

Данные системы поддержки направлены на различные области функционирования университетов, в том числе на работу в образовательном процессе. Однако, в связи с тем, что данные системы являются закрытыми и доступны только лицам, которые имеют доступ к образовательным ресурсам университета, проанализировать недостатки и преимущества аналогичных разработок.

Проблема анализа аналогов дополняется тем, что в данном проекте работа ведётся с системой управления обучением «Moodle», и более конкретная информация про взаимодействия интеллектуальных ассистентов с этой системой будет представлять наибольший интерес. Однако, не все университеты в открытом доступе располагают информацией о том, какие платформы они используют для обучения и организации работы вуза в целом, а закрытость и приватность данных систем делает подробный анализ невозможным.

Однако, в связи с тем, что в каждом конкретном университете информационные системы имеют свои особенности использования, отсутствие подробного анализа не является преградой для разработки поскольку все внедрённые решения будут актуальны только лишь для одного конкретного университета.

Таким образом, результат разбор аналогов показал невозможность подробного их анализа в связи с закрытостью информационных систем университета и привёл к выводу, что каждый разработанный подобный интеллектуальный помощник будет индивидуален и актуален только лишь для одного конкретного университета, что делает разработку подсистемы поддержки актуальной и необходимой.

2 Выбор средств разработки

Обработка естественного языка охватывает широкий спектр задач, начиная от базовых операций (токенизация, стемминга, лемматизация), заканчивая сложными аналитическими методами (анализ семантики, классификация, генерация текста). Современные библиотеки языка Python, выбранного в качестве основного при разработке модуля, предоставляют широкие функциональные возможности, позволяющие решать разнообразные задачи в рамках исследовательских работ и практических приложений.

– SpaCy. SpaCy позиционируется как производительная библиотека для промышленной обработки текста. Разработанная на языке программирования Python, она предлагает быстрый доступ к современным инструментам анализа текста, таким как:

1) предварительно обученные модели: SpaCy поддерживает большое число языков, включая русский. Модели позволяют быстро извлекать сущности (Named Entity Recognition, NER), анализировать структуру предложений (Dependency Parsing), определять части речи (Part-of-Speech Tagging) и многое другое;

2) производительность: благодаря оптимизации с использованием компилируемого расширения Cython, SpaCy обеспечивает высокую скорость обработки текстов даже на больших наборах данных;

3) простота внедрения: интеграция с существующими приложениями проста и эффективна, библиотека легко интегрируется в системы управления контентом, веб-сервисы и чатботы;

4) масштабируемость: приложения, разработанные на spaCy, успешно используются в реальных проектах промышленности и науки, обеспечивая стабильную работу с большими объёмами данных [9];

– Transformers (Hugging Face). Библиотека Transformers была разработана компанией Hugging Face и сосредоточена на архитектуре трансформеров, обеспечивающих передовую точность и гибкость в задачах

NLP. Трансформеры стали доминирующим методом в обработке естественного языка благодаря своей способности создавать высококачественные модели, способные обобщать знания и улучшаться на ограниченном объёме данных. Преимущества:

1) архитектуры моделей: Huggings Face предлагает широкий выбор заранее подготовленных моделей, таких как BERT, RoBERTa, GPT, XLNet и др., каждая из которых предназначена для конкретных типов задач, будь то понимание контекста, обработка вопросов и ответов, генерация текста или классификация документов;

2) интеграция с TensorFlow и PyTorch: это делает её совместимой практически с любыми инструментами глубокого обучения и позволяет разрабатывать проекты, основанные на нескольких фреймворках одновременно;

3) поддержка мультиязычности: большинство моделей являются кросс-лингвистичными, способными обрабатывать тексты на разных языках, включая русский;

4) Fine-Tuning: гибкая настройка моделей под собственные задачи и наборы данных помогает адаптировать существующие архитектуры под специфику конкретного проекта [10];

– TensorFlow. TensorFlow — один из ведущих инструментов для разработки нейронных сетей и методов глубокого обучения. Хотя сам по себе TensorFlow не предназначен исключительно для задач NLP, он широко используется для построения специализированных моделей и графов вычислительных процессов, связанных с обработкой текстов. Преимущества:

1) широкая экосистема: TensorFlow поддерживает огромное сообщество исследователей и практиков, создавая удобные интерфейсы для загрузки и обучения моделей, ведения экспериментальных проектов и мониторинга производительности;

2) оптимизация GPU/TPU: Встроенная поддержка аппаратных ускорителей позволяет существенно ускорить процесс обучения и тестирования моделей;

3) универсальность: может использоваться для множества задач, помимо NLP, включая компьютерное зрение, генерацию изображений и обработку временных рядов;

4) TF-Hub: позволяет загрузить готовые модули и встраивать их в свою систему, экономя ресурсы на разработке собственных компонентов. Несмотря на универсальность, TensorFlow требует значительных ресурсов для эффективного обучения и проектирования моделей, что ограничивает его использование для мелких задач NLP [11];

– NLTK (Natural Language Toolkit). NLTK является одним из старейших пакетов для обработки естественного языка на Python. Первоначально разработанный как инструмент для образовательных целей, NLTK постепенно стал популярной платформой для академических исследований и прототипирования новых подходов в NLP. Преимущества:

1) обширная документация: большое количество примеров и учебных материалов делают NLTK идеальным инструментом для изучения основ NLP;

2) разнообразие функций: NLTK включает инструменты для предварительной обработки текста (токенизация, удаление стоп-слов, стемминга), поддержки грамматического разбора (парсинга), анализа тональности и многих других стандартных задач;

3) модульность: каждый компонент NLTK легко заменяется собственными реализациями, позволяя расширять функциональность в зависимости от потребностей исследователя. Тем не менее, из-за своего образовательного характера, NLTK часто уступает другим пакетам в скорости и эффективности при промышленном применении [12];

– Gensim. Gensim получил популярность среди специалистов по обработке больших текстовых массивов благодаря своим мощным методам

анализа тем и векторному представлению слов (word embeddings). Эта библиотека отличается способностью эффективно обрабатывать большие объемы данных, обеспечивая высокие показатели точности в задаче определения сходства документов и категоризации текстов. Преимущества:

1) эффективность: за счёт использования эффективных структур данных и параллельных вычислений, Gensim показывает превосходную производительность при анализе больших текстовых корпусов;

2) алгоритмы векторизации: поддерживается создание и обновление векторизаторов слов, таких как Word2Vec и Doc2Vec, позволяющих качественно представлять смысл слов и документов в многомерном пространстве;

3) тематический анализ: одной из основных особенностей Gensim является способность эффективно выделять скрытые темы в тексте с помощью метода Latent Dirichlet Allocation (LDA). Gensim хорошо зарекомендовал себя в задачах классификации документов, выявления тенденций и создания рекомендаций, однако слабее справляется с задачами, требующими тонкого понимания смысла отдельных слов и выражений [13];

– FastText. Разработанный командой исследователей Facebook, FastText предлагает быстрые и эффективные методы векторного представления слов и классификации текстов. Одним из главных преимуществ FastText является его способность работать с малыми датасетами и ограниченными ресурсами, сохраняя высокий уровень точности. Преимущества:

1) быстроедействие: FastText известен своей скоростью обучения и высокой производительностью, что делает его отличным решением для быстрого прототипирования и минимального жизнеспособного продукта (MVP);

2) эффекты сублексикона: FastText учитывает морфологическую структуру слов, что улучшает качество работы с редкими словами и орфографически близкими терминами;

3) классификатор текстов: способность FastText эффективно справляться с задачами классификации документов делает его востребованным инструментом в системах фильтрации спама, детекции тональности сообщений и автоматизированного анализа отзывов пользователей. FastText демонстрирует отличную эффективность при работе с небольшим количеством меток и низко-ресурсными языками, такими как русский [14].

В по причине удобства использования, высокой популярности, и, следственно, большого количества примеров, для реализации текущего проекта была выбрана библиотека TensorFlow.

Также, для выполнения выпускной работы были выбраны следующие языки программирования и программные средства:

— Python — высокоуровневый интерпретируемый язык программирования общего назначения, известный своей лаконичностью, ясностью и удобством чтения. Создан голландским программистом Гвидо ван Россумом в конце XX века, Python стремительно набрал популярность и сегодня считается одним из самых востребованных языков программирования в мире. Основные характеристики и достоинства Python включают:

1) высокая читабельность и простой синтаксис: код на Python легко читать и понимать, что способствует быстрому освоению языка новичками и снижает порог входа в профессию программиста;

2) интерпретируемость. Программы на Python выполняются пошагово, что ускоряет цикл разработки и упрощает отладку;

3) многофункциональность: Python активно применяется в широком спектре областей, таких как веб-разработка, наука о данных, машинное обучение, автоматизация административных задач, сетевое программирование и геймдев;

4) богатая стандартная библиотека: стандартная библиотека Python включает обширный набор модулей, которые решают большинство распространённых задач без привлечения внешних зависимостей;

5) огромное сообщество и развитая экосистема: благодаря активному сообществу, вокруг Python сформировалась богатая инфраструктура инструментов и библиотек, позволяющая быстро решить любую техническую проблему;

6) многоплатформенность: Python одинаково уверенно чувствует себя на платформах Windows, macOS, Linux и Unix-подобных ОС, что гарантирует портативность и независимость от платформы;

7) объектно-ориентированное программирование: Python реализует мощную объектно-ориентированную парадигму, позволяя разработчику свободно управлять структурой приложения и многократно использовать код;

8) автоматическое управление памятью: в отличие от некоторых других языков, Python самостоятельно управляет динамическим распределением памяти, избавляя программиста от ручного освобождения ресурсов. Благодаря своей универсальности и дружелюбию к начинающим, Python активно изучают студенты университетов и колледжей, используют опытные разработчики для ускорения рабочего процесса и применяют крупные корпорации, такие как Google, Яндекс, Dropbox и Netflix, для разработки серверных частей, научных расчётов и сложных аналитических задач [15].

– DbForge Studio for MySQL — это комплексный инструмент, предназначенный для облегчения жизни разработчиков и администраторов баз данных MySQL. Данный продукт облегчает выполнение повседневных задач, связанных с созданием, управлением и обслуживанием базы данных MySQL, и делает возможным достижение лучших результатов с меньшими усилиями. Эта профессиональная утилита доступна абсолютно бесплатно и оснащена множеством полезных возможностей:

1) автоматизация ежедневных задач: DbForge Studio берёт на себя большую часть повторяющихся действий, освобождая разработчиков и администраторов от рутины;

2) улучшение рабочих процессов: предоставляет удобный интерфейс для всех аспектов работы с базой данных, помогая повысить эффективность труда;

3) повышение надёжности: уменьшаются риски ошибок и увеличивается стабильность создаваемых приложений и инфраструктуры [16];

– Telegram — популярный и универсальный мессенджер, доступный на множестве устройств и операционных систем, включая Android, iOS, Windows, macOS и Linux. Программа позволяет пользователям общаться друг с другом различными способами, отправляя текстовые, голосовые и видеосообщения, фотографии, стикеры и любые типы файлов [17]. Кроме того, Telegram предлагает ряд уникальных функций, делающих его удобным и полезным средством общения:

1) аудио- и видеозвонки: помимо обмена сообщениями, пользователи могут звонить друг другу и устраивать групповые звонки, общаясь голосом или через видеосвязь;

2) прямые эфиры и конференции: эти инструменты подходят для организации встреч, лекций, презентаций и массовых мероприятий онлайн;

3) хранение файлов: Telegram предоставляет неограниченное пространство для хранения файлов различного формата, что удобно для совместного пользования документами и медиаконтентом;

4) создание и использование ботов: Telegram позволяет создавать специальных роботов для автоматизации задач, отправки уведомлений, консультаций клиентов и выполнения других видов деятельности. Благодаря своему функционалу и простоте использования Telegram завоевал огромную аудиторию и продолжает оставаться одним из самых популярных способов коммуникации [17];

– Giga IDE — отечественная разработка российской компании «СберТех», созданная для повышения удобства и эффективности работы российских ИТ-специалистов. Этот продукт представляет собой полноценную среду разработки, предназначенную для удобной работы с исходным кодом и административного сопровождения многоплатформенных проектов [18]. Giga IDE характеризуется рядом значимых особенностей:

- 1) локализация: интерфейс и поддержка документаций выполнены на русском языке, что делает её идеальной для отечественных разработчиков;
- 2) работа с несколькими языками программирования IDE поддерживает многочисленные языки программирования, предоставляя унифицированный рабочий интерфейс для работы с разными технологиями;
- 3) настройка среды: широкий диапазон настроек позволяет индивидуально настроить рабочую среду под потребности каждого разработчика [18];

– XAMPP — бесплатное и открытое решение для локальной установки сервера веб-разработки, которое объединяет сразу несколько необходимых элементов инфраструктуры: Apache HTTP Server, MySQL-сервер базы данных, интерпретатор PHP и Perl, а также почтовый сервер Mercury Mail Transport System [19]. Основные особенности и преимущества XAMPP:

- 1) компактность и легкость установки: установочный файл занимает небольшой объем и устанавливается буквально за пару минут, не требуя особых усилий и технических навыков;
- 2) поддержка различных языков программирования: наряду с основными компонентами, включенными в состав XAMPP, доступно расширение и установка сторонних технологий, таких как Ruby on Rails, Node.js и Python;
- 3) кроссплатформенность: пакет XAMPP можно установить на любой современной операционной системе, будь то Windows, macOS или Linux, что обеспечивает свободу выбора рабочей среды;

4) безопасность и конфиденциальность: доступна возможность шифрования SSL/TLS для безопасной передачи данных, защиты паролей и аутентификации пользователей;

5) файловый менеджер и FTP-клиент: внутри пакета установлен модуль FileZilla, позволяющий управлять файлами и каталогами через встроенный клиент FTP. XAMPP идеально подходит для локального тестирования веб-приложений, учебного процесса и самостоятельного освоения веб-технологий. Он помогает разработчикам избежать трудностей с ручной настройкой сложного окружения и сосредоточиться непосредственно на создании качественного цифрового продукта [19].

3 Структуры подсистемы

Необходимо разработать программный модуль, основу которого составляет нейронная сеть, подготовленная для обработки естественного языка и обученная отвечать пользователям в соответствии с тематикой их вопроса. Круг вопросов, имеющих прикладной характер ограничен пространством Электронного деканата ЭИОС ПГУ. Для анализа структуры подсистемы и требований к её возможностям была создана диаграмма прецедентов, показанная на рисунке 1.

В реализуемой подсистеме будет 4 вида пользователей, функционал которых будет во многом схож, за исключением роли администратора. Отличия между остальными ролями будут проявляться не в функционале, а в тематике вопросов, на которые система будет выдавать ответ при наличии определённой роли.

Пользователям с ролью студента доступна возможность запрашивать инструкцию, где и как можно просмотреть свои оценки. Остальные компоненты Электронного деканата недоступны обычным студентом, поэтому более возможности пользователей с такими ролями никак не расширяются.

Пользователи с ролью Преподаватель могут позволить себе следующий функционал:

- запрос инструкции по просмотру оценок студентов. Сюда входят инструкции, которые включают в себя в каком интерфейсе системы необходимо просматривать оценки, как попасть в этот интерфейс и как им пользоваться,
- запрос инструкции по выставлению оценок. Модуль поддержки подскажет в каком интерфейсе и, соблюдая какие определённые действия, можно выставить оценки за преподаваемые пользователем дисциплины в подсистему Электронный деканат.

Пользователям с ролью Управляющий доступны все возможности пользователей с ролью Преподаватель, а также запрос инструкций по организации учебного процесса, в частности:

- запрос инструкций по работе с учебными программами: инструкции по работе с интерфейсами создания, редактирования, удаления и прочей работы с учебными программами университета,
- запрос инструкций по работе с дисциплинами: инструкции по работе с интерфейсами создания, редактирования, удаления и прочей работы с учебными дисциплинами университета,
- запрос инструкций по подписи преподавателей: инструкции по закреплению преподавателей за дисциплинами и, наоборот, откреплению от них,
- запрос инструкций по работе с академическими группами: инструкции по работе с интерфейсами создания, редактирования, удаления и прочей работы с академическими группами различных подразделений университета,
- запрос инструкций по работе со студентами: инструкции по работе с интерфейсами создания, редактирования, удаления, зачисления студентов в академические группы и прочей работы с учебными программами университета,
- запрос инструкций по работе с прочими компонентами.

Администратор системы является последней необходимой упоминания ролью в разрабатываемой системе, и он имеет следующие функции:

- настройка и поддержка системы: администратор отвечает за настройку и поддержку системы, что включает в себя установку обновлений, устранение неполадок и обеспечение бесперебойной работы системы,
- корректировка алгоритмов обучения: администратор также может корректировать алгоритмы обучения, что может включать в себя настройку параметров, оптимизацию процессов и внедрение новых технологий.

Основная функция разрабатываемой системы – это оказание технической поддержки пользователям Электронного деканата, поэтому следует поподробнее остановиться на алгоритме получения необходимых инструкций. Данный процесс, представленный на диаграмме последовательностей, показанной на рисунке 2.

Процесс взаимодействия пользователя с системой электронного обучения начинается с входа в интерфейс. Пользователь, столкнувшись с проблемой или вопросом, обращается к системе через Telegram-бот, который выполняет первичную проверку его идентификатора для обеспечения безопасности и подтверждения личности. После успешной аутентификации Telegram-бот предоставляет информацию о пользователе, что позволяет системе адаптировать взаимодействие под его индивидуальные потребности.

Следующим шагом является предоставление пользователю доступа к системе, что открывает перед ним возможности для взаимодействия. Пользователь описывает свою проблему или задает вопрос, который требует решения. Этот запрос передается в нейронную сеть, которая занимается распознаванием и классификацией вопросов. Нейронная сеть анализирует запрос, определяя его сущность и контекст, чтобы предоставить наиболее релевантный ответ.

Telegram-бот проверяет роль пользователя, чтобы убедиться, что он имеет право на получение запрашиваемой информации. После этого бот запрашивает у нейронной сети формирование ответа, который будет полезен и понятен пользователю. Нейронная сеть ищет контекст для ответа, используя предыдущие сообщения и данные, чтобы обеспечить точность и релевантность ответа.

Наконец, нейронная сеть формирует и выводит ответ, который предоставляется пользователю через интерфейс. Этот ответ может содержать инструкции, рекомендации или другую полезную информацию, которая помогает пользователю решить его проблему или получить необходимую информацию.

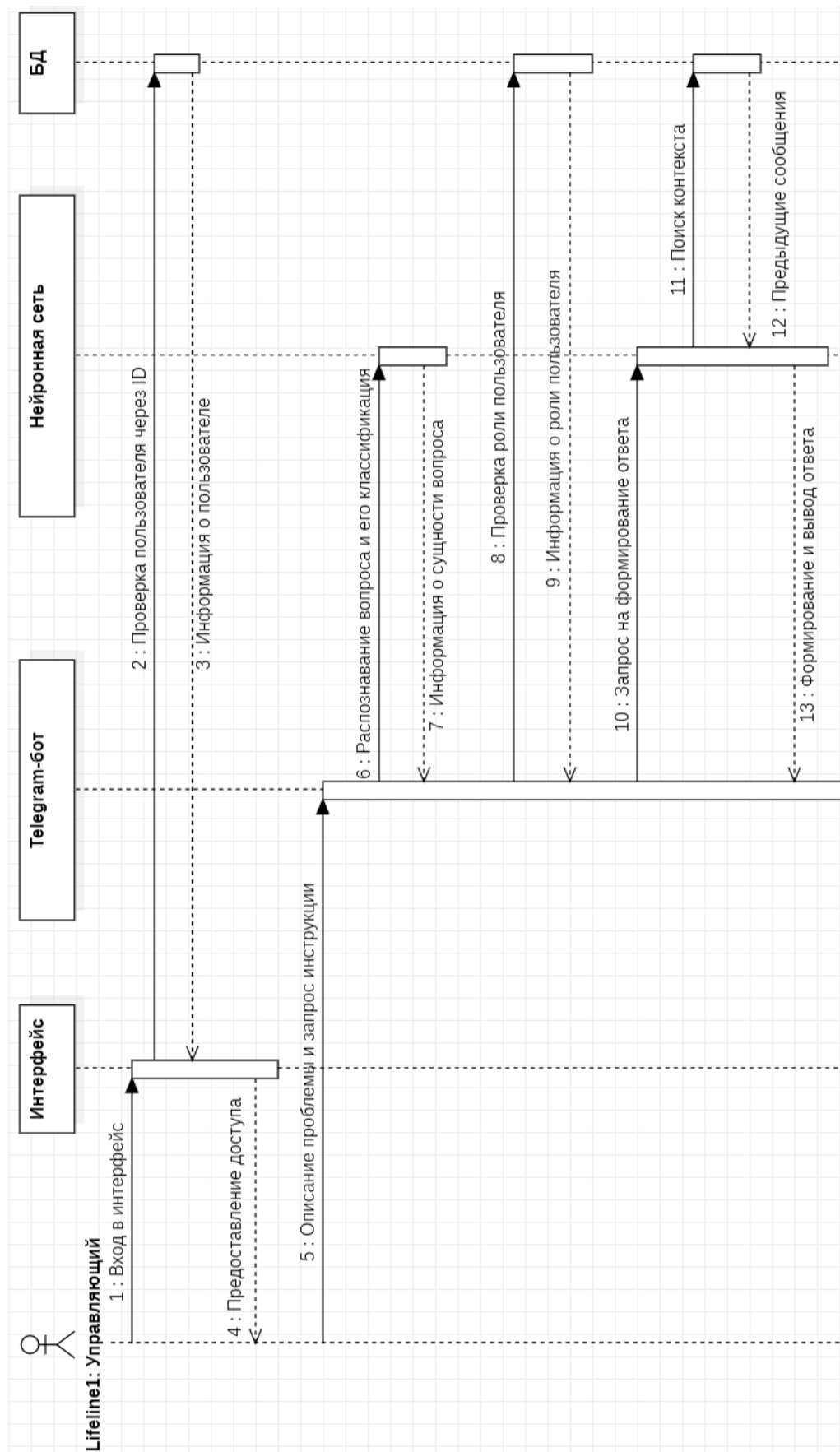


Рисунок 2 – Диаграмма последовательности при работе модуля

На рисунке 3 изображена структура подсистемы:

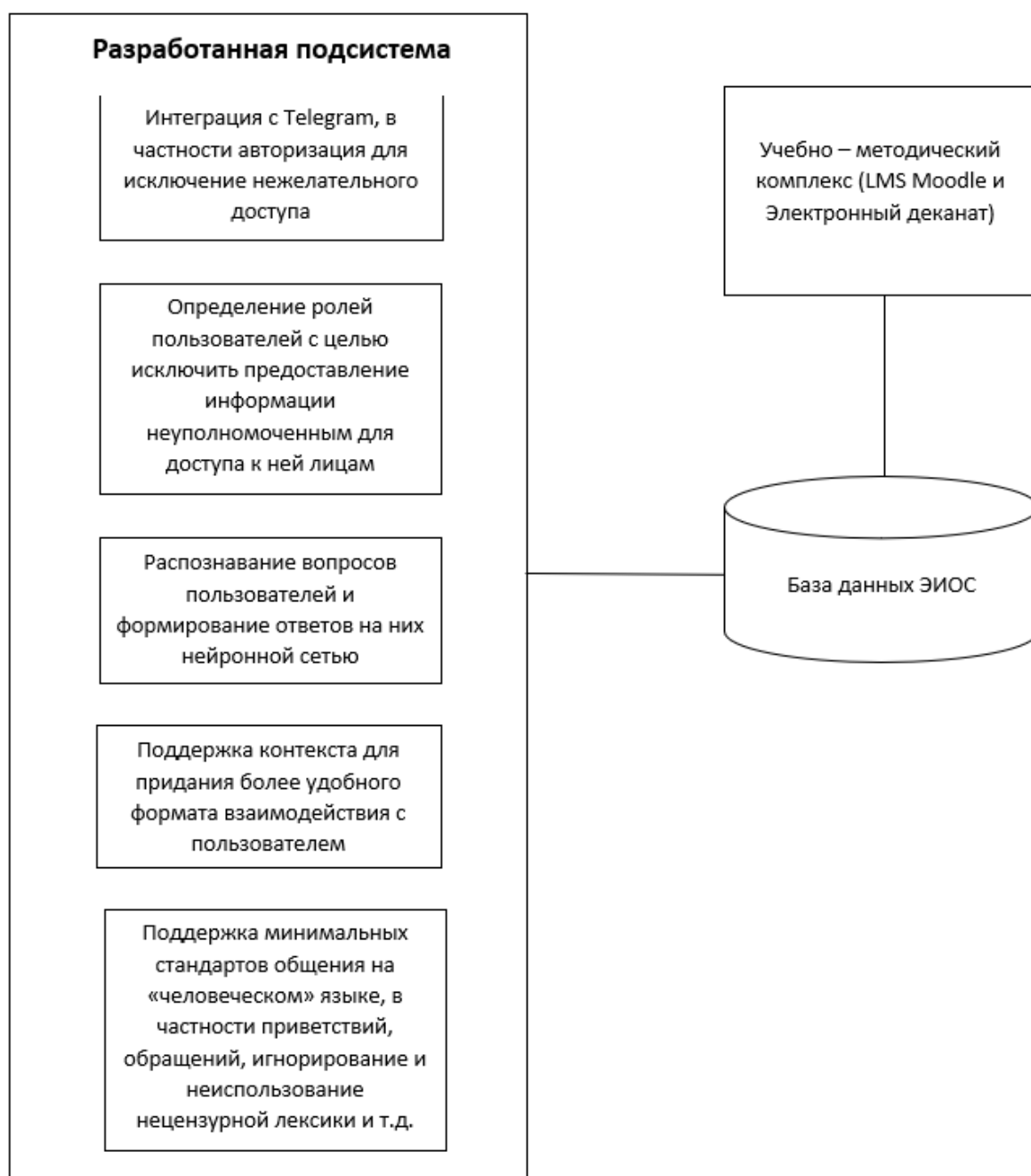


Рисунок 3 – Структура подсистемы

Таким образом, была разработана структура подсистемы, которая позволяет решить все поставленные перед системой задачи. Разработанные компоненты объединены в подсистему и интегрируются с информационной системой ПГУ через единую БД учебного портала, работающего на основе LMS Moodle.

4 Структура базы данных подсистемы

Для организации работы разрабатываемой подсистемы необходимо учесть потребность во внесении изменений в структуру базы данных. При этом часть функционала способно выполняться при помощи уже существующих таблиц баз данных LMS Moodle и Электронного деканата.

Для учёта контекста необходимо создать 3 новых таблицы в базе данных.

Таблица `mdl_tlq_type_content` создана для хранения типов контента (программы, дисциплины и т.д. записи). Описание полей:

- `id`: уникальный идентификатор типа контента (BIGINT), данное поле хранит идентификатор типа контента,
- `name_type`: название типа контента (VARCHAR), данное поле хранит название типа контента,
- `description`: описание типа контента (VARCHAR), данное поле хранит описание типа контента,
- `version`: тип данных `bigint(20)`, последняя дата изменения записи.

Таблица `mdl_tlq_session` создана для хранения сессий использования чат-бота (программы, дисциплины и т.д. записи). Описание полей:

- `id`: уникальный идентификатор сессии (BIGINT), данное поле хранит идентификатор сессии,
- `userid`: идентификатор пользователя (BIGINT), данное поле хранит идентификатор пользователя, связанного с сессией,
- `typecontextid`: идентификатор типа контекста (BIGINT), данное поле хранит идентификатор типа контекста сессии,
- `starttime`: время начала сессии (TIMESTAMP), данное поле хранит время начала сессии,
- `updatetime`: время обновления сессии (TIMESTAMP), данное поле хранит время последнего обновления сессии,

- `finishtime`: время окончания сессии (TIMESTAMP), данное поле хранит время окончания сессии,
- `is_actual`: флаг актуальности сессии (TINYINT), данное поле указывает, является ли сессия актуальной,
- `version`: тип данных `bigint(20)`, последняя дата изменения записи.

Таблица `mdl_tlq_message` создана для хранения сообщений чат-боту, отправляемых пользователями, и хранения ответов чат-бота. Описание полей:

- `id`: уникальный идентификатор сообщения (BIGINT), данное поле хранит идентификатор сообщения,
- `message`: текст сообщения (LONGTEXT), данное поле хранит текст сообщения,
- `sessionid`: идентификатор сессии (BIGINT), данное поле хранит идентификатор сессии, к которой относится сообщение,
- `is_user`: флаг пользователя (TINYINT), данное поле указывает, является ли сообщение от пользователя,
- `is_deleted`: флаг удаления сообщения (TINYINT), данное поле указывает, было ли сообщение удалено,
- `delete_time`: время удаления сообщения (TIMESTAMP), данное поле хранит время удаления сообщения,
- `version`: тип данных `bigint(20)`, последняя дата изменения записи.

Для функционирования подсистемы необходимо выделить из всей имеющейся базы данных только те таблицы, которые будут использовать компоненты реализуемой подсистемы. Схема базы данных подсистемы представлена на рисунке 4.

Подробное описание всех остальных таблиц и их полей, необходимых для функционирования разрабатываемого модуля будет описано в рамках текущего раздела.

Таблица `mdl_user` хранит информацию о пользователях системы. Описание полей:

- поле `id`, тип данных `bigint(10)`, данное поле хранит идентификатор пользователя,
- поле `confirmed`, тип данных `tinyint(1)`, определяет является ли данная учётная запись пользователя подтверждённой,
- поле `deleted`, тип данных `tinyint(1)`, определяет, является ли данная учётная запись пользователя удалённой,
- поле `suspended`, тип данных `tinyint(1)`, показывает, разрешено ли данному пользователю входить в систему,
- поле `username`, тип данных `varchar(100)`, хранит логин пользователя для входа в систему,
- поле `password`, тип данных `varchar(255)`, в данном поле хранятся пароли от учётных записей пользователей в зашифрованном формате,
- поле `idnumber`, тип данных `varchar(255)`, пользовательский идентификатор данной учётной записи,
- поле `firstname`, тип данных `varchar(100)`, в данном поле хранится имя пользователя,
- поле `lastname`, тип данных `varchar(100)`, в данном поле содержится фамилия пользователя,
- поле `email`, тип данных `varchar(100)`, хранит логин электронной почты пользователя,
- поле `address`, тип данных `varchar(255)`, в данном поле содержится адрес пользователя,
- поле `city`, тип данных `varchar(120)`, название города, которое укажет пользователь при регистрации,

- поле `country`, тип данных `varchar(2)`, название страны, которое укажет пользователь при регистрации,
- поле `lang`, тип данных `varchar(30)`, название языка, на котором система отображает контент для пользователя,
- поле `firstaccess`, тип данных `bigint(10)`, дата первого входа в систему данным пользователем,
- поле `lastaccess`, тип данных `bigint(10)`, дата последнего входа в систему данным пользователем,
- поле `description`, тип данных `longtext`, в данном поле хранится описание пользователя, которое он может оставить, редактируя свой профиль,
- поле `timecreated`, тип данных `bigint(10)`, дата регистрации учётной записи пользователя,
- поле `timemodified`, тип данных `bigint(10)`, дата последней модификации учётной записи пользователя,
- поле `middlename`, тип данных `varchar(255)`, отчество (второе имя) пользователя,
- поле `alternatename`, тип данных `varchar(255)`, альтернативное имя пользователя.

Таблица `mdl_role` хранит роли, которые могут быть у пользователей в курсах. Описание полей:

- поле `id`, тип данных `bigint(10)`, данное поле хранит идентификатор роли,
- поле `name`, тип данных `varchar(255)`, полное название роли,
- поле `shortname`, тип данных `varchar(100)`, краткое название роли,
- поле `description`, тип данных `longtext`, описание роли.

Таблица `mdl_role_assignments` необходима для связи роли, пользователя и контекста. Описание полей:

- поле `id`, тип данных `bigint(10)`, идентификатор записи,

- поле `roleid`, тип данных `bigint(10)`, идентификатор роли,
- поле `contextid`, тип данных `bigint(10)`, хранит идентификатор записи в таблице `mdl_context`,
- поле `userid`, тип данных `bigint(10)`, идентификатор пользователя,
- поле `timemodified`, тип данных `bigint(10)`, данное поле хранит в себе дату последней модификации записи (дата создания также хранится в этом поле),
- поле `modifierid`, тип данных `bigint(10)`, идентификатор пользователя, который создал запись.

Таблица `mdl_enrol` используется для регистрации на курсах. Описание полей:

- поле `id`, тип данных `bigint(10)`, идентификатор записи,
- поле `status`, тип данных `bigint(10)`, хранит статус записи (активно/неактивно),
- поле `courseid`, тип данных `bigint(10)`, идентификатор учебного курса, на который была совершена регистрация,
- поле `roleid`, тип данных `bigint(10)`, идентификатор роли,
- поле `timecreated`, тип данных `bigint(10)`, дата создания записи,
- поле `timemodified`, тип данных `bigint(10)`, дата последней модификации записи.

Таблица `mdl_user_enrolments` хранит пользователей, участвующих в курсах. Описание полей:

- поле `id`, тип данных `bigint(10)`, идентификатор записи,
- поле `status`, тип данных `bigint(10)`, содержит статус записи (активно/неактивно),
- поле `enrolid`, тип данных `bigint(10)`, идентификатор записи в таблице `mdl_enrol`,
- поле `userid`, тип данных `bigint(10)`, хранит идентификатор пользователя,

- поле `timestart`, тип данных `bigint(10)`, время начала активного участия пользователя в курса,
- поле `timeend`, тип данных `bigint(10)`, время окончания активного участия пользователя в курсе (по умолчанию всегда выставляется значение 2147483647, что соответствует последнему значению исчисления времени в UNIX – формате),
- поле `modifierid`, тип данных `bigint(10)`, данное поле хранит идентификатор пользователя, который последний модифицировал данную запись,
- поле `timecreated`, тип данных `bigint(10)`, дата создания текущей записи,
- поле `timemodified`, тип данных `bigint(10)`, дата последней модификации данной записи.

Таблица `mdl_d_departments` показывает подразделения института.
Описание полей:

- поле `id`, тип данных `bigint(20)`, идентификатор подразделения,
- поле `name`, тип данных `varchar(50)`, название подразделения,
- поле `code`, тип данных `varchar(50)`, код подразделения (аббревиатура),
- поле `leaddepid`, тип данных `bigint(20)`, идентификатор «родительского» подразделения,
- поле `date_born`, тип данных `bigint(20)`, дата создания записи о подразделении,
- поле `data_modify`, тип данных `bigint(20)`, дата последней модификации данных о подразделении,
- поле `managerid`, тип данных `bigint(20)`, идентификатор пользователя, создавшего запись о подразделении,
- поле `path`, тип данных `varchar(100)`, в этом поле содержится подразделение и все его «родители» (вышестоящие подразделения),

- поле `is_deleted`, тип данных `tinyint(1)`, показывает, является запись удалённой,
- поле `version`, тип данных `bigint(20)`, последняя дата изменения записи.

Таблица `mdl_d_student` используется связи персоны с академической группой. Описание полей:

- поле `id`, тип данных `bigint(20)`, идентификатор записи,
- поле `personid`, тип данных `bigint(20)`, идентификатор персоны,
- поле `groupid`, тип данных `bigint(20)`, идентификатор учебной группы,
- поле `start`, тип данных `bigint(20)`, время создания записи,
- поле `is_deleted`, тип данных `bool(1)`, флаг актуальности записи,
- поле `version`, тип данных `bigint(20)`, последняя дата изменения записи.

Таблица `mdl_d_person` используется для хранения информации о пользователях электронного деканата. Описание полей:

- поле `id`, тип данных `bigint(20)`, уникальный идентификатор персоны,
- поле `firstname`, тип данных `varchar(45)`, хранит имя пользователя,
- поле `lastname`, тип данных `varchar(45)`, хранит фамилию пользователя,
- поле `middlename`, тип данных `varchar(45)`, хранит отчество пользователя (при наличии такового),
- поле `mdluser`, тип данных `bigint(20)`, идентификатор персоны в таблице `mdl_user`,
- поле `born`, тип данных `bigint(20)`, дата рождения человека,
- поле `version`, тип данных `bigint(20)`, последняя дата изменения записи.

Таблица mdl_d_appointment используется для учёт должностных назначений. Описание полей:

- поле id, тип данных bigint(20), идентификатор записи,
- поле personid, тип данных bigint(20), идентификатор персоны,
- поле appointment, тип данных bigint(20), идентификатор типа назначения,
- поле createdate, тип данных bigint(20), время создания записи,
- поле version, тип данных bigint(20), последняя дата изменения записи.

В результате модифицирована структура базы данных с поддержкой ссылочной целостности: добавлены таблицы, которые необходимы для учёта контекста диалогов и сбора данных для дальнейшего обучения модели нейронной сети. Также добавлены необходимые связи с уже разработанными таблицами в БД с использованием СУБД MySQL.

5 Разработка программного обеспечения

5.1 Интеграция Moodle и Telegram

Важнейшей задачей при достижении целей разработки является интеграция чат-бота мессенджера «Telegram» и «Moodle». Первоначально необходимо попробовать поискать готовые решения и, если таковые имеются, проанализировать их на соответствие задачам, поставленным при разработке. Для поиска готовых решений необходимо перейти по ссылке <https://moodle.org>. По данному адресу располагается официальный сайт «Moodle». На главной странице находим ссылку, которая при нажатии на неё открывает список официальных плагинов для «Moodle», которые доступны для бесплатного скачивания (рисунок 5).

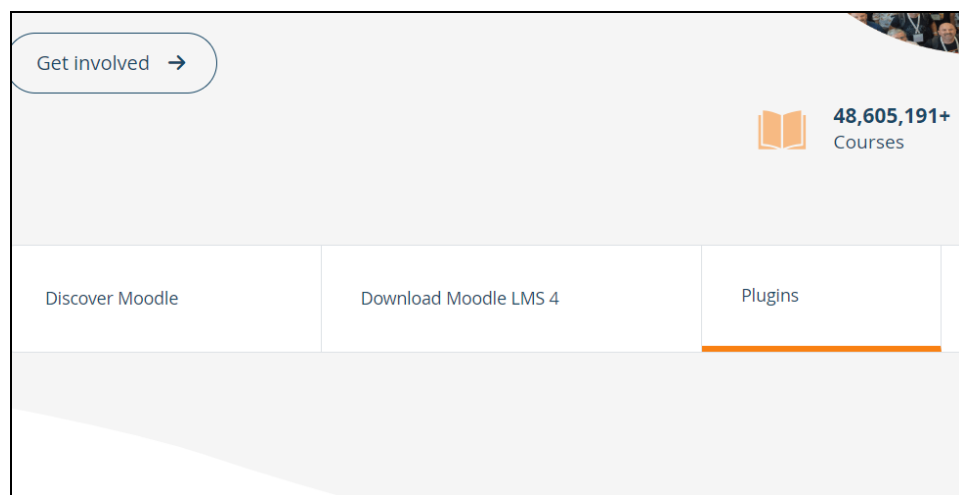


Рисунок 5 – Поиск плагинов

В поле поиска введём «Telegram», чтобы отобразить только те плагины, в которых есть какое-либо упоминание мессенджера «Telegram» (рисунок 6). Таковых плагинов среди всего списка обнаружилось четыре.

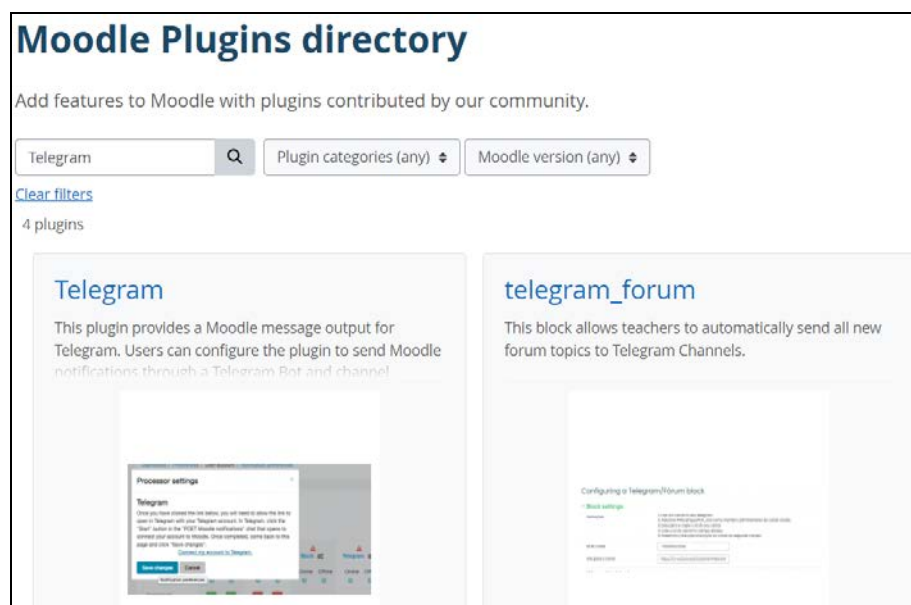


Рисунок 6 – Поиск плагинов

Данные плагины были проанализированы на совместимость с необходимой версией Moodle (3.9) и на соответствие потребностям разработки. 2 плагина – «Restriction by message» (рисунок 7) и «Course extra UI» (рисунок 8), предоставлять возможность использовать мессенджер «Telegram» в рамках прохождения учебных курсов на платформе Moodle. В рамках каждого курса есть возможность создавать каналы в «Telegram», что не соответствует потребностям, поставленным в рамках разработки. Данные плагины исключаются также в связи с их несовместимостью с требуемой версией, поскольку они разработаны для более поздних выпусков «Moodle».

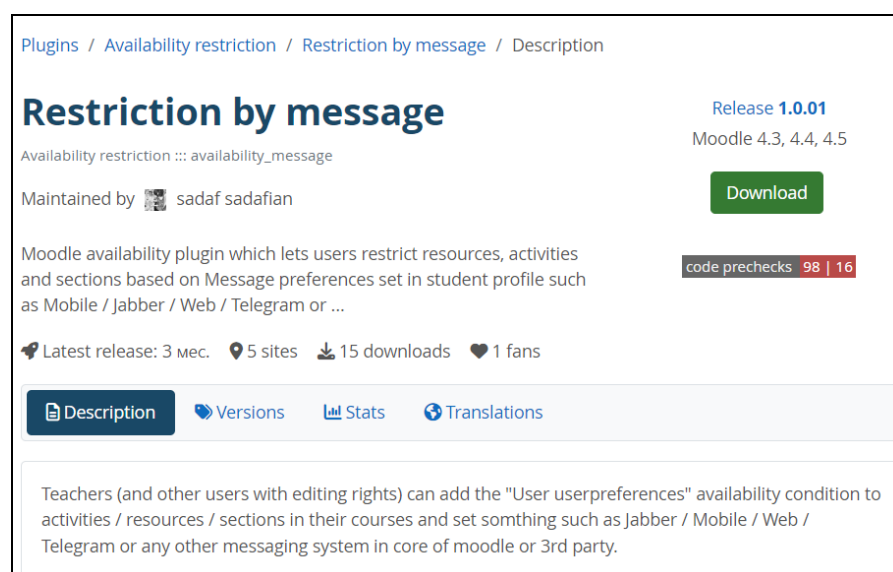


Рисунок 7 – плагин «Restriction by message»

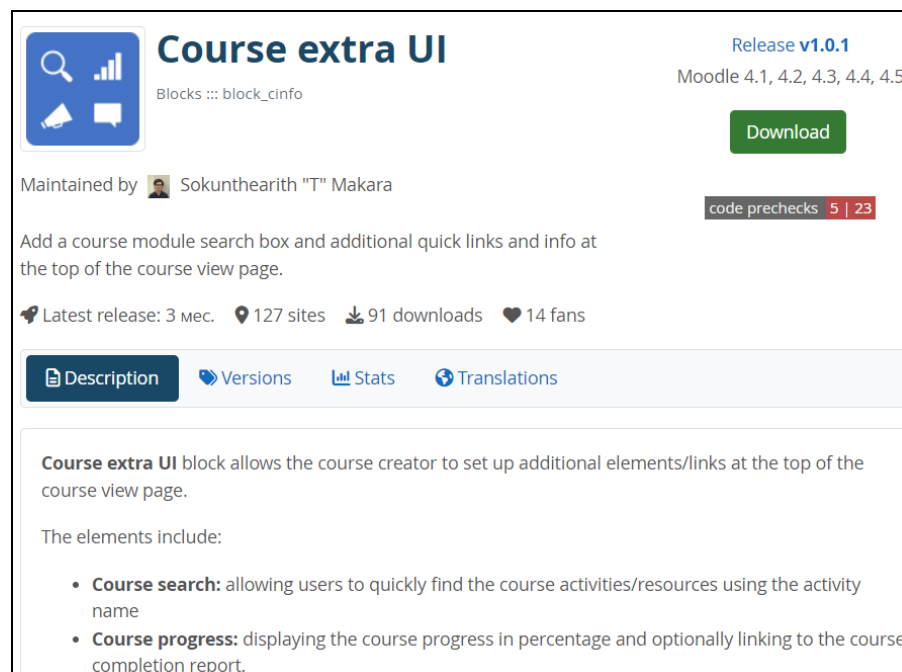


Рисунок 8 – плагин «Course extra UI»

Далее был исследован плагин «telegram_forum», в котором уже есть упоминание чат-бота (рисунок 9). Данный плагин также соответствует необходимой версии «Moodle». Но решения, представленные в текущем плагине, также не соответствуют решаемым задачам. Здесь чат-бот, создаваемый администратором текущей платформы, основанной на «Moodle», является администратором канала, который в мессенджере создаёт преподаватель для своего курса.

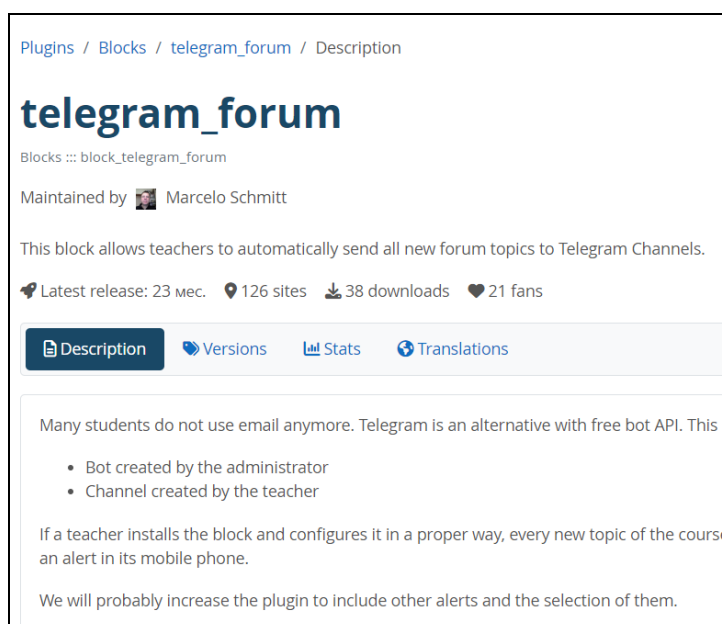


Рисунок 9 – плагин «Telegram forum»

Последний исследуемый плагин с названием «Telegram», оказался наиболее соответствующий заданным требованиям (рисунок 10). Данный плагин предоставляет чат-боту функционал отправлять пользователям сообщения, при от «Moodle» через мессенджер «Telegram». Неоспоримым преимуществом данного плагина является совместимость с необходимой версией «Moodle».

Информация, которую предоставляет чат-бот при работающем плагине не является необходимой для текущей разработке. Базовый функционал можно использовать в качестве дополнительных опций, поскольку определённым пользователям встроенные навыки плагина могут оказаться полезными при работе с «Moodle». Важным аспектом является интеграция чат-бота «Telegram» и текущей системы на базе «Moodle», что является важным фактором, влияющим на общую безопасность при использовании разрабатываемого продукта.

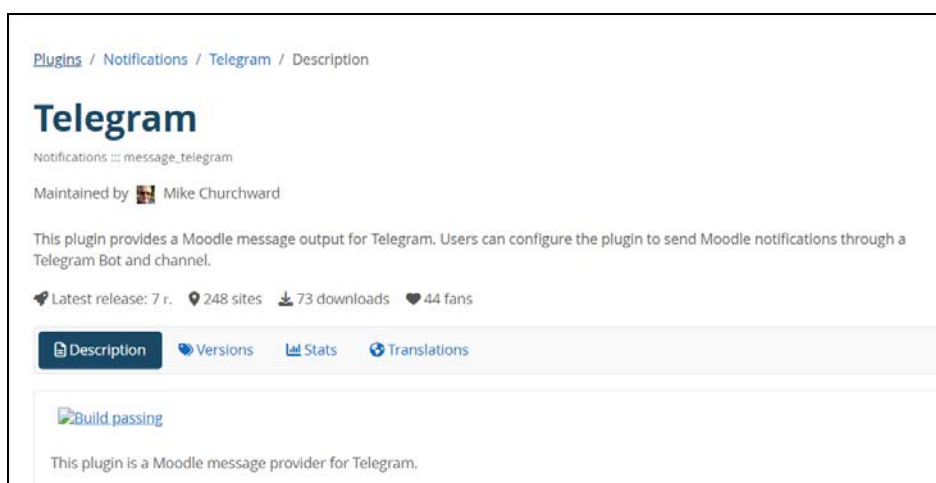


Рисунок 10 – плагин «Telegram»

Реализовать заданный проект можно и без плагинов для «Moodle», однако это может усложнить интеграцию с заданной системой на основе «Moodle», и это, в свою очередь, может отрицательно сказаться на безопасности использования разрабатываемого продукта. Ещё одной причиной предпочтительности готовых плагинов является возможное дальнейшее расширение функционала продукта. На момент разработки сложно предположить дальнейшие векторы развития разработки, однако

более «плотная» интеграция с «Moodle», может спровоцировать новые, более строгие требования к совместимости сервисов. Также следует отметить полезность выбранного плагина, некоторые опции которого действительно могут заинтересовать пользователей.

Для реализации интеграции чат-бота «Telegram» и «Moodle» первоначально необходимо установить выбранный плагин. Плагины для «Moodle» скачиваются в формате архива.

После скачивания плагина, для дальнейшей установки заходим в систему, основанную на «Moodle», с которой будет проводится интеграция. После авторизации необходимо перейти по ссылке «Администрирование», которая располагается на панели в левой части страницы (рисунок 11).

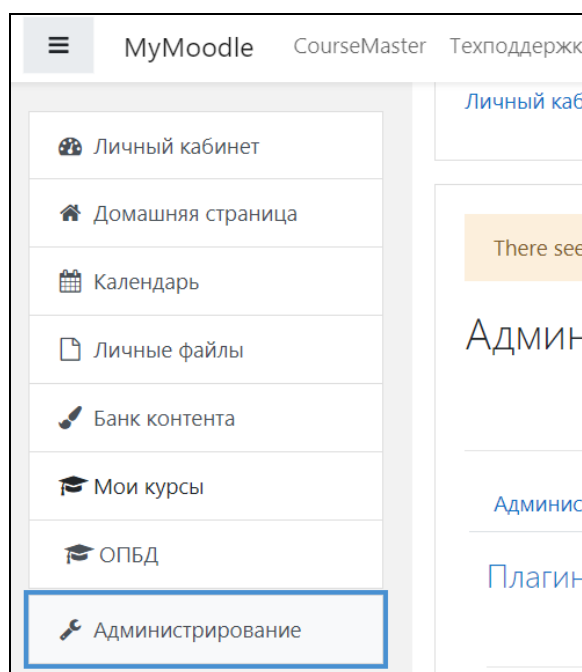


Рисунок 11 – Установка плагина

Далее, в блоке для администраторов, необходимо перейти на панель с плагинами и перейти по ссылке «Установка плагинов» (рисунок 12).

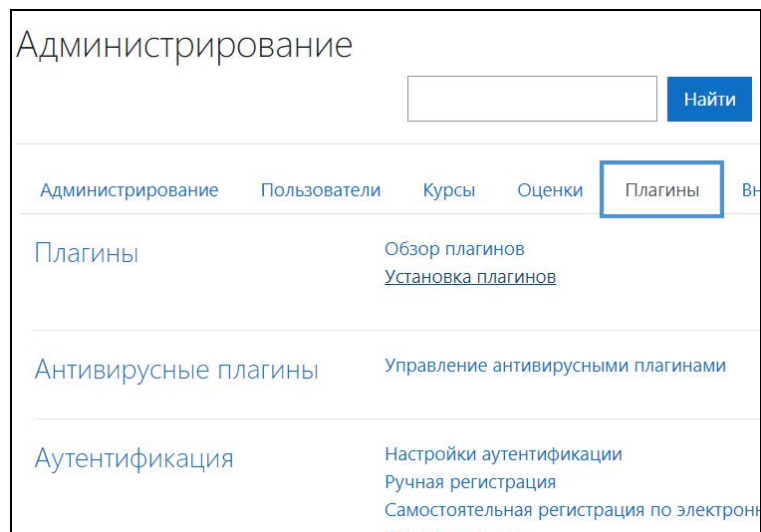


Рисунок 12 – Установка плагина

После перехода по ссылке откроется Установщик плагинов, в который необходимо загрузить скаченный ранее архив с необходимым плагином и после этого нажать на кнопку «Установить плагин из ZIP-файла» (рисунок 13).

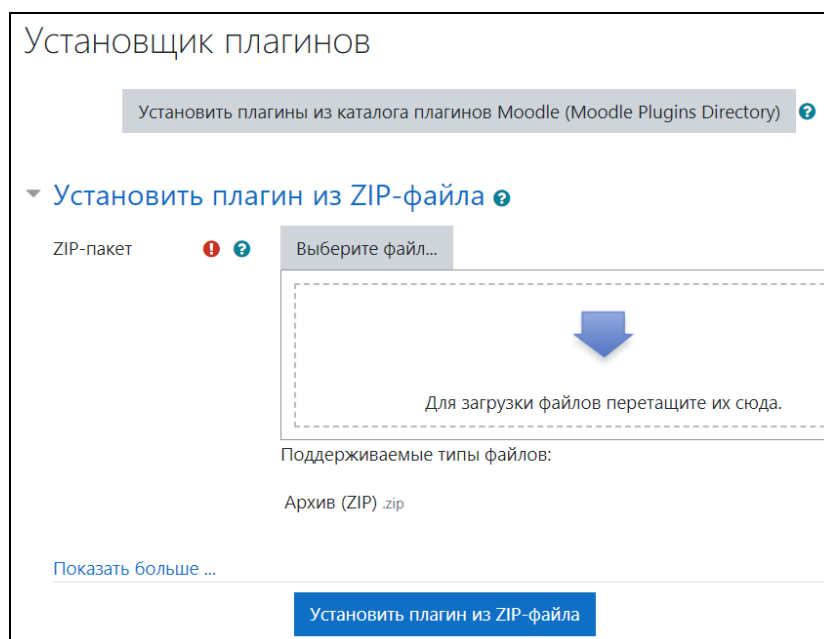


Рисунок 13 – Установка плагина

После успешной установки плагина, потребуется ввести токен чат-бота. Для его создания можно использовать чат-бот с названием «BotFather», предоставляющий широкий функционал для работы с чат-ботами мессенджера «Telegram» (рисунок 14).

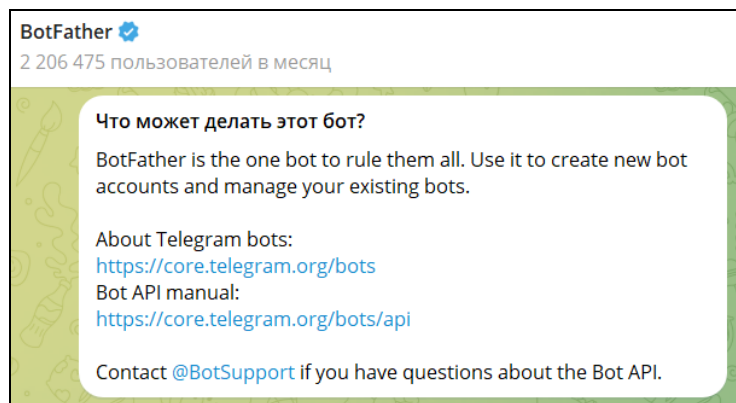


Рисунок 14 – создание чат-бота

При первом запуске данного бота, он предоставляет список команд, необходимых для создания и управления чат-ботами, созданными пользователем. Данные команды будут полезны при редактировании и при расширении функционала созданных ботов.

Для создания своего чат-бота необходимо ввести команду /newbot и, следуя запросам бота, ввести, соответственно, название бота, которое будет видно пользователям, которые будут взаимодействовать с созданным ботом, а также имя пользователя (рисунок 15). Имя пользователя необходимо, поскольку бот также, по своей сути, является пользователем мессенджера «Telegram», и данный атрибут необходим для идентификации среди всех пользователей.

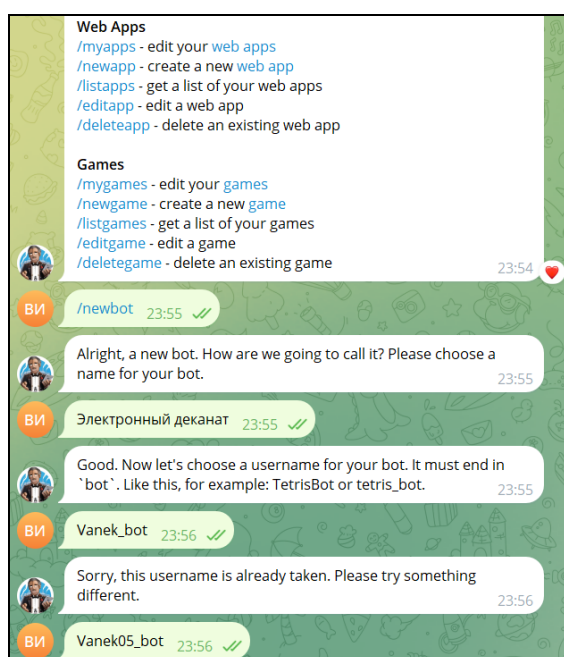


Рисунок 15 – создание чат-бота

При генерации создаваемого чат-бота, BotFather должен отправить в сообщении пользователю токен нового бота, который необходим для его интеграции (рисунок 16).

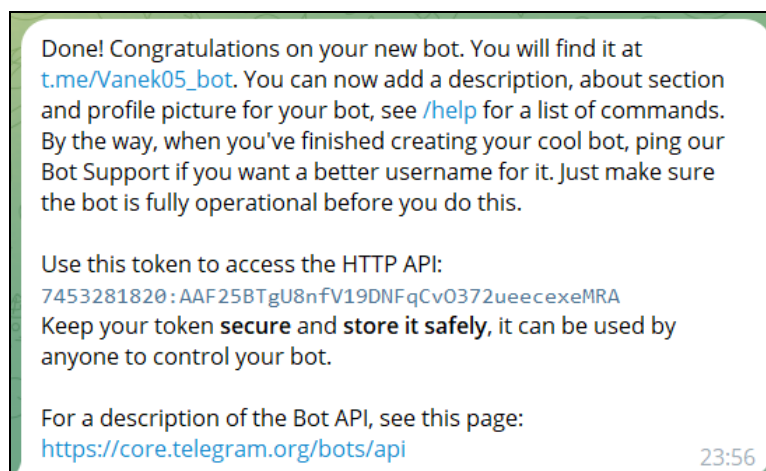


Рисунок 16 – создание чат-бота

Полученный токен необходимо вставить в поле «Токен бота для сайта» на странице настройки данного плагина (рисунок 17). Имя пользователя и название бота заполняются автоматически после нажатия на кнопку «Сохранить изменения».

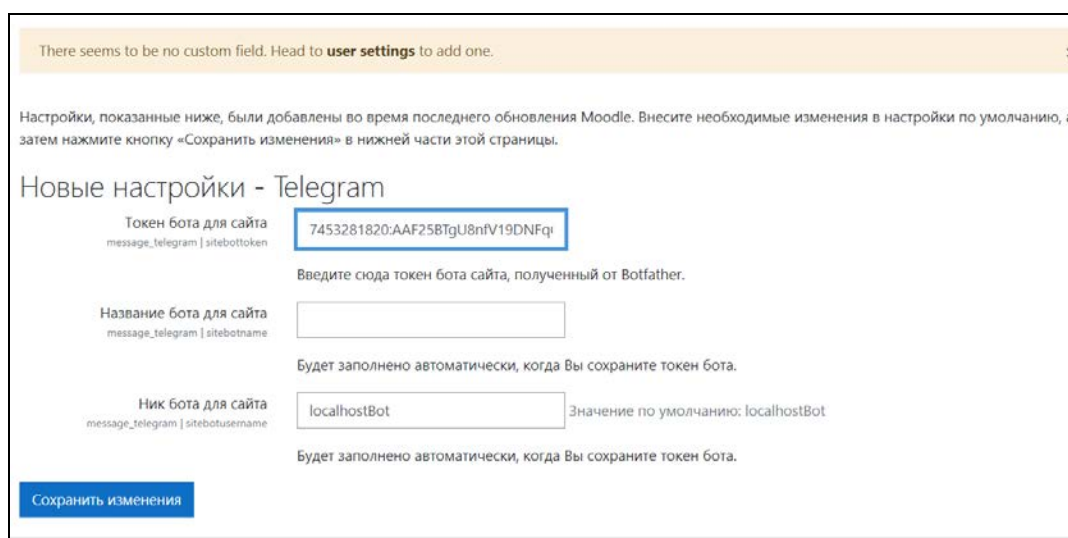


Рисунок 17 – интеграция чат-бота

Для настройки базового функционала установленного плагина необходимо нажать на иконку с уведомлениями и в открытом окошке нажать на иконку настроек (рисунок 18).

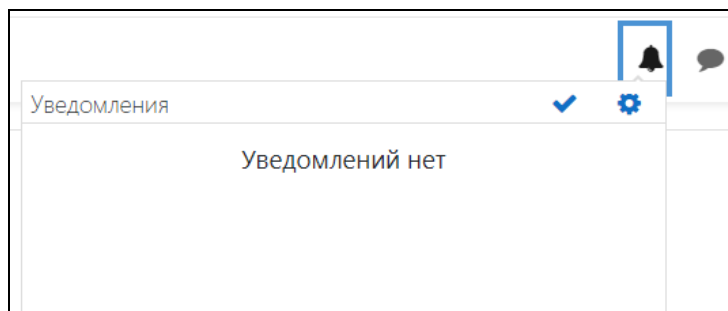


Рисунок 18 – проверка плагина

Установленный плагин позволяет чат-боту «Telegram» рассылать пользователям сопряжённой с ним системы получать от него различные сообщения. Имеется возможность для пользователя индивидуально выбирать какие типы сообщения будут приходить через мессенджер «Telegram» от созданного чат-бота (рисунок 19).

	Telegram 		Уведомление во всплывающем окне		Электронная почта 	
	На сайте ?	Не в сети ?	На сайте ?	Не в сети ?	На сайте ?	Не в сети ?
Задание	?	?	?	?	?	?
Уведомление о задании	Выкл.	Выкл.	Выкл.	Выкл.	Вкл.	Вкл.
Обратная связь	На сайте	Не в сети	На сайте	Не в сети	На сайте	Не в сети
Уведомления о получении ответов в элементах типа «Обратная связь»	Выкл.	Выкл.	Выкл.	Выкл.	Вкл.	Вкл.
Напоминание об «обратной связи»	Выкл.	Выкл.	Выкл.	Выкл.	Вкл.	Вкл.

Рисунок 19 – проверка плагина

Поскольку чат-бот по своей сути является обычным пользователем мессенджера, то по его уникальному имени его может найти любой пользователь «Telegram». Скрытие имени у бота доступно только при подключении платной подписки «Telegram», но в данной работе такие растраты рассматривать нецелесообразно, поэтому данный вариант автоматически отпадает. Но даже и он, не обеспечил бы полной безопасности, поскольку уникальное имя пользователя чат-бота,

пользователям могут передать лица, которые к нему должны иметь доступ по причине учёбы/работы в университете.

Поэтому было бы целесообразно, что чат-бот имел доступ к базе данных образовательной организации, использующей «Moodle», чтобы можно было перед сессией использования бота определить, должен ли пользователь уметь к нему доступ.

Для этого можно при переходе по ссылке, добавлять к ней параметр, который бы содержал бы какую-то комбинацию атрибутов записи пользователя в базе данных. При таком сценарии, пользователю необходимо авторизоваться в «Moodle» и оттуда перейти по ссылке. Такой вариант целесообразно принять за базовый, и дальнейшую интеграцию вести на его основе, поскольку существенно повышает безопасность работы чат-бота. В таком случае целесообразно пожертвовать временем, отведённым на авторизацию пользователей в угоду безопасности всей электронной среды университета.

Для того, чтобы прикрепить к ссылке параметр, необходимо сначала взять информация из базы данных. Для этого необходимо на сервере создать файл, который будет идентифицировать пользователя, брать о нём информацию и формировать значение параметра из комбинации найденных в базе данных атрибутов пользователя (рисунок 20). Если пользователь авторизован и у его учётной записи нет никаких блокирующих атрибутов, то его можно перенаправлять на созданный чат-бот.

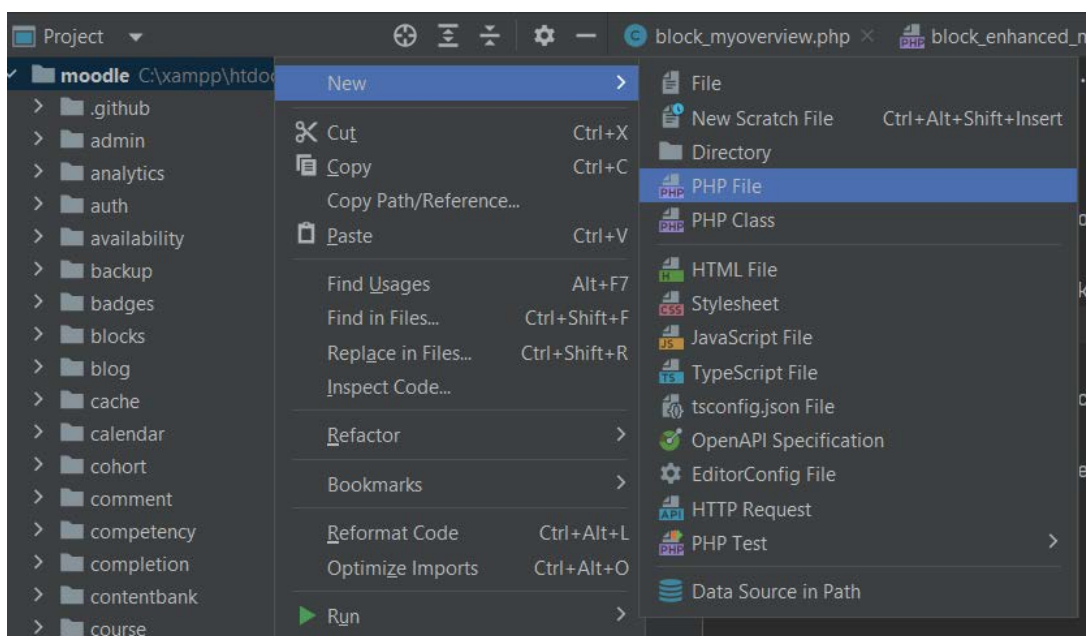


Рисунок 20 – создание скрипта идентификации пользователя

В базовом исполнении установленного плагина присутствует проблема поиска ссылки на созданный чат-бот. Не каждому пользователю захочется исследовать сайт в поисках данного ресурса, поэтому целесообразно разместить ссылку в «шапке» сайта, что она присутствовала на любой странице, которую открывает пользователь. Данная ссылка будет вести не на страницу с ботом, а на страницу обработки информации о пользователе, которая была создана ранее на сервере (рисунок 21).

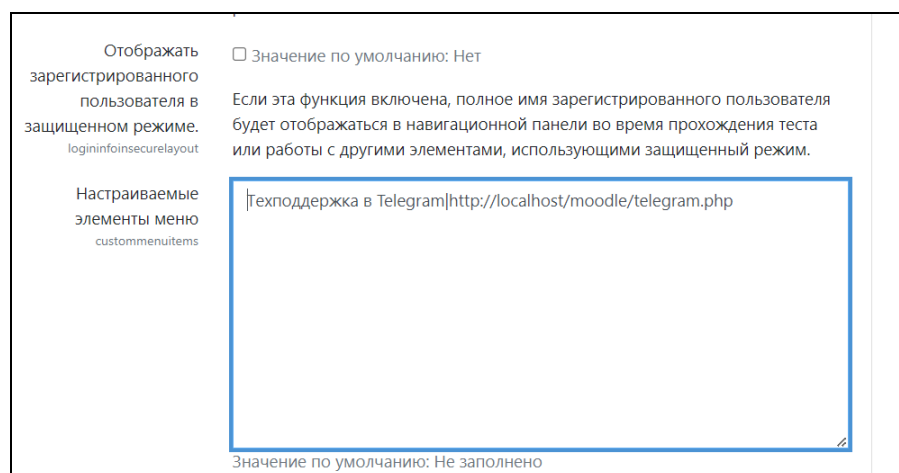


Рисунок 21 – создание ссылки для активации скрипта
Внешний вид созданной ссылки (Рисунок 22).

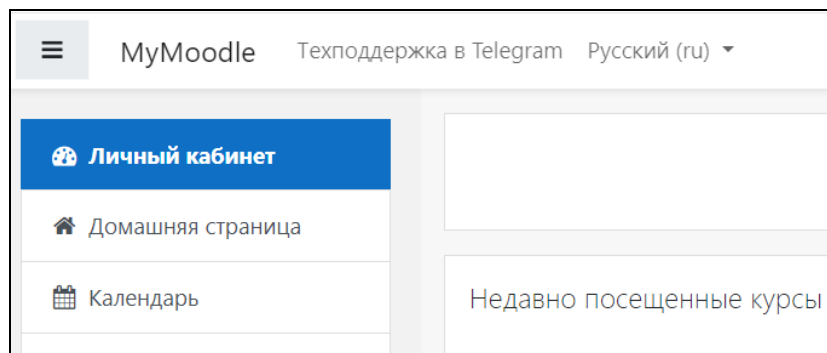


Рисунок 22 – создание ссылки для активации скрипта

Таким образом, была налажена интеграция мессенджера Telegram и LMS Moodle. Для этого были проанализированы существующие плагины по критериям, предусмотренных текущим проектом и выбран один из них, а также добавлена ссылка для перехода в мессенджер на главную страницу сайта.

5.2 Обучение модели нейронной сети

Для удобства формирования данных и их восприятия, для обучения нейронной сети будут использованы файлы JSON-формата. Каждый обучающий объект будет содержать в себе несколько атрибутов, среди которых вопрос пользователя и ответ системы, а также тип контекста, определяющий тематику проблемы, с которой пользователь обращается к модулю. Для удобства обучения, целесообразно разделить данные с разными типами контекста по разным JSON-файлам.

Пользователи будут обращаться к модулю технической поддержки по вопросам, которые касаются работы с несколькими типами объектов, существующих в Электронном деканате ЭИОС ПГУ, например, учебные программы, академические группы, студенты и т.д. В рамках одного вопроса пользователя могут быть упоминания о сразу нескольких сущностях (объектах, типах контекста), с которыми может вестись работа. Например, дисциплины, по которым ведётся обучение, входят в состав учебных программ, поэтому пользователь при задаче вопроса может перечислить в своём сообщении обе эти сущности. В таком случае, упоминание учебной программы не должно сбивать с толку нейронную сеть, и она должна

однозначно определять тип контекста, о котором идёт речь, в данном случае именно о дисциплинах.

Могут произойти ситуации, в которых пользователь будет указывать и более существенное количество объектов. Поэтому в рамках подготовки данных важно правильно расставлять веса при сообщениях такого типа, чтобы пользователь смог получать от нейронной сети ответ, который мог бы удовлетворить его запрос о решении конкретной проблемы, с которой он обратился, а не получал информацию о решении проблем, которые в данный момент перед ним не стоят.

Необходимо также помнить, в данном модуле пользователи имеют свои роли, которые анализируются, прежде чем начать выдавать ответ на вопрос. В связи с этим, при обучении нужно учесть ситуации, когда пользователь может в своём сообщении упомянуть сущности, с которыми он не работает в системе, исходя из своей роли. В таких случаях, нейронная сеть должна правильно определять тип контекста, чтобы не возникло ситуаций, когда пользователи не могут получить ответы на свои запросы из-за того, что система некорректно определяет контекст и блокирует выдачу ответа в связи с тем, что данный пользователь не обладает необходимой ролью для доступа к информации такого типа.

В связи с вышеперечисленными возможными ситуациями, в атрибутах обучающих данных должна быть расположена информация о типе контекста. Для новых поступающих сообщений также будет определяться тип контекста с последующим сохранением в базу данных. Это необходимо в том числе и для последующего переобучения нейронной сети для повышения точности её работы.

Для обучения нейронной сети будет использована библиотека TensorFlow языка Python, а также с помощью её высокоуровневого API Keras. Выбор был сделан в её пользу её гибкости, масштабируемости и поддержке широкого спектра алгоритмов машинного обучения.

Также для каждого вопроса был выставлен тип контекста для последующего обучения модели, которое можно будет проводить с помощью данных, которые были внесены пользователями, благодаря их взаимодействию с разрабатываемым модулем. Пример данных показан в приложении А в конце текущего документа.

Для успешного обучения модели необходимо правильно подготовить данные. В связи с отсутствием необходимых данных было выполнено синтезирование возможных фраз пользователей и сформированы на них подходящие ответы. Формулировки предполагаемых сообщений пользователей были составлены на основе приходящих на электронную почту сотрудников технической поддержки ЭИОС сообщений от пользователей, столкнувшихся с проблемами при использовании Электронного деканата. Правильные ответы формировались на основе опыта работы сотрудников Пензенского государственного университета, отвечающих за администрирование Электронного деканата.

В процессе обучения первоначально выполняются этапы очистки текста, такие как токенизация, удаление стоп-слов, лемматизация, векторизация и другие предобработки. Далее выполняется загрузка данных и их объединение в один объект, поскольку они будут загружаться из разных файлов в зависимости от типа контекста. В связи с тем, что библиотека TensorFlow требует текстов одинакового размера, необходимо выполнить преобразование всех обучающих текстов к одному размеру.

Далее был осуществлён подбор оптимального количества слоёв нейронной сети, эпох обучения и прочих атрибутов. В процессе обучения было многократно выполнено переобучения модели, протестирована её работа, а также измерена точность с помощью соответствующих методов выбранной библиотеки.

В результате выполнения данного этапа работы была получена рекуррентная нейронная сеть со следующим списком слоёв:

- Embedding: преобразует целые числа (индексы слов) в плотные векторы заданной размерности,
- Второй слой — LSTM: рекуррентный слой с определенным количеством внутренних состояний,
- Dense: полносвязанный слой с одним выходом и функцией активации sigmoid, который выводит итоговую оценку.

Параметры регулировались в соответствии с результатами работы разрабатываемого модуля и результатами тестов надёжности модели для достижения точности работы более 90%.

В качестве оборудования для должен использовался ноутбук со следующими характеристиками:

- процессор с 4-мя с тактовой частотой 2,5 ГГц,
- оперативной памятью 16Гб,
- операционной системой Windows 10,
- видеокартой NVIDIA GeForce GTX 1650 Ti (4 Гб).

5.3 Тестирование ПО

Требование ТЗ: «Ресурс должен быть доступен пользователям в любое время суток каждый день без исключений».

Требование ТЗ: «Ресурс должен функционировать без вмешательства технического специалиста».

Был создан полностью автономный телеграмм-бот «Электронный деканат», который принимает и отвечает на сообщения круглосуточно.

Требование ТЗ: «Пользователь всегда должен получать какой-либо ответ от системы, т.е. диалог всегда должен завершаться сообщением от разработанного бота, вне зависимости от содержимого сообщения пользователя».

На рисунке 23 показан пример тестирования бота, на котором видно, что чат-бот, отвечает на любое сообщение пользователя, даже на те, которые не представляют собой связную речь.



Рисунок 23 – Тестирование

Требование ТЗ: «Система должна формировать свой ответ пользователю исходя из анализа ключевых слов в сообщении пользователя».

В примере ниже (рисунок 24) в сообщении от пользователя были разобраны ключевые слова/словосочетания «дисциплина», «оценки за контрольную точку», «не отображаются», на основе чего и был составлен ответ на данный запрос.

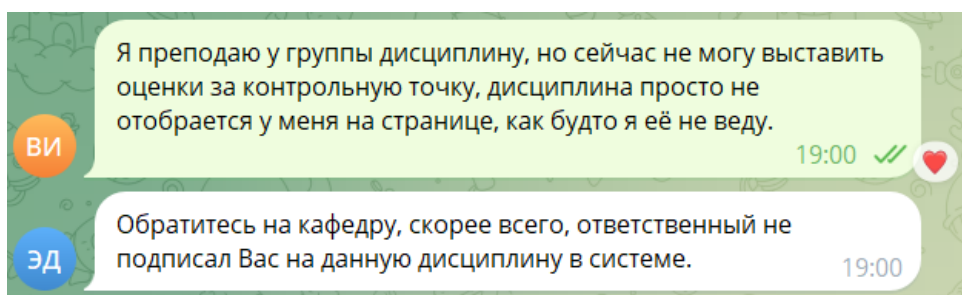


Рисунок 24 – Тестирование

Требование ТЗ: «Ответ пользователю на его запрос должен присылать одним сообщением вне зависимости от объёма контента сообщения от системы».

В примере ниже, несмотря на большой объём ответного сообщения на обращение пользователя, весь ответ присылается одним целым сообщением (рисунок 25).

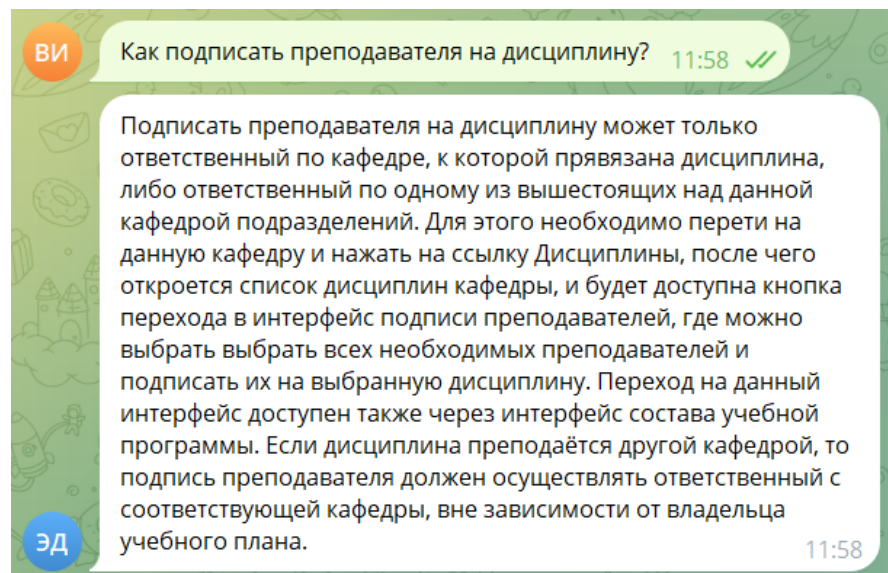


Рисунок 25 – Тестирование

Требование ТЗ: «На одно сообщение пользователя должно отправляться одно ответное сообщение».

В примере ниже (рисунок 26) пользователь отправляет набор не связанных между собой вопросов, в которых ключевые слова относятся к разным тематикам. Система выстраивает диалог так, чтобы решить все вопросы по отдельности и привести общение к формату «один вопрос – один ответ».

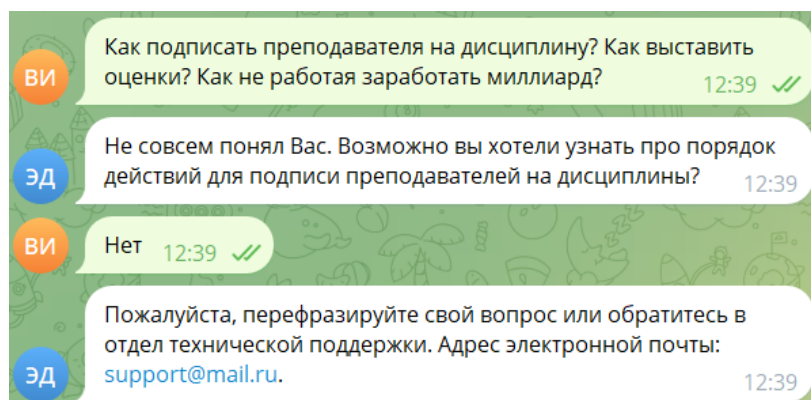


Рисунок 26 – Тестирование

Требование ТЗ: «Ответ на вопрос должен сформулирован так, чтобы он был понятен пользователю «начального уровня»».

В примере ниже (рисунок 27) подробно рассказано о конкретных действиях пользователя для решения его проблемы. Указано по каким

ссылкам надо переходить и что нажимать. Подробное описание действий будет понятно пользователям «начального уровня».

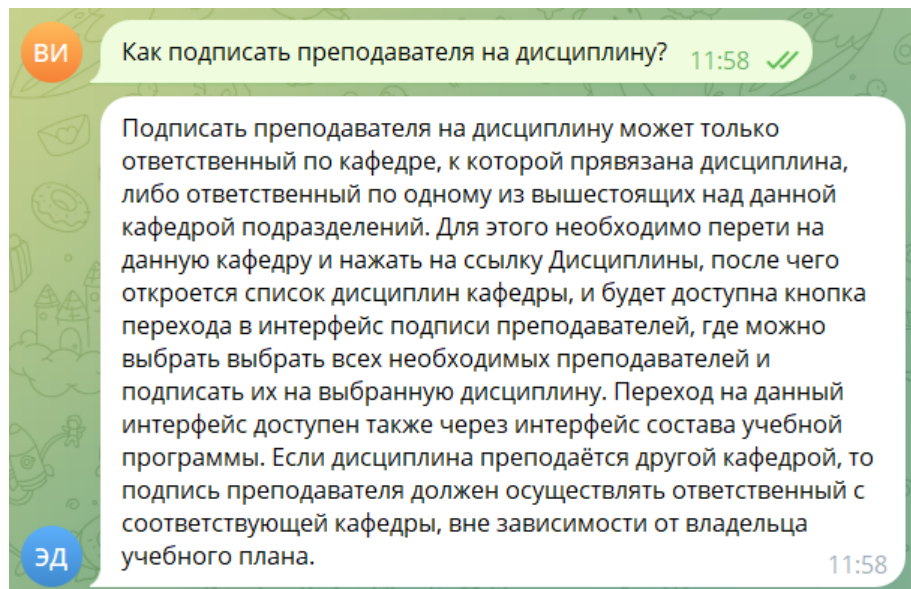


Рисунок 27 – Тестирование

Требование ТЗ: «Если на вопрос пользователя система не может дать соответствующий ответ, то в ответном сообщении должны содержаться контакты технической поддержки ЭИОС ПГУ».

На рисунке 28 показан пример взаимодействия, реализующий данный пункт технического задания. Модуль перенаправляет пользователя к сотрудникам технической поддержки, если не может самостоятельно понять проблему.

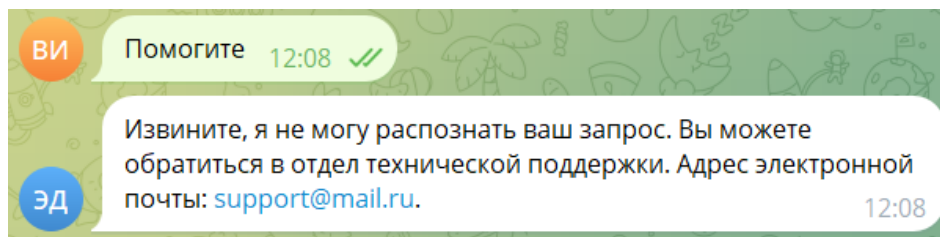


Рисунок 28 – Тестирование

Требование ТЗ: «Бот должен уметь распознавать и отличать прикладные сообщения от общих (например, «Привет», «До свидания», «Как дела?») и давать на них соответствующие ответы».

В примере ниже (рисунок 29) показан пример общения пользователя и чат-бота без непосредственно вопросов, связанных с оказанием услуг технической поддержки при использовании Электронного деканата.

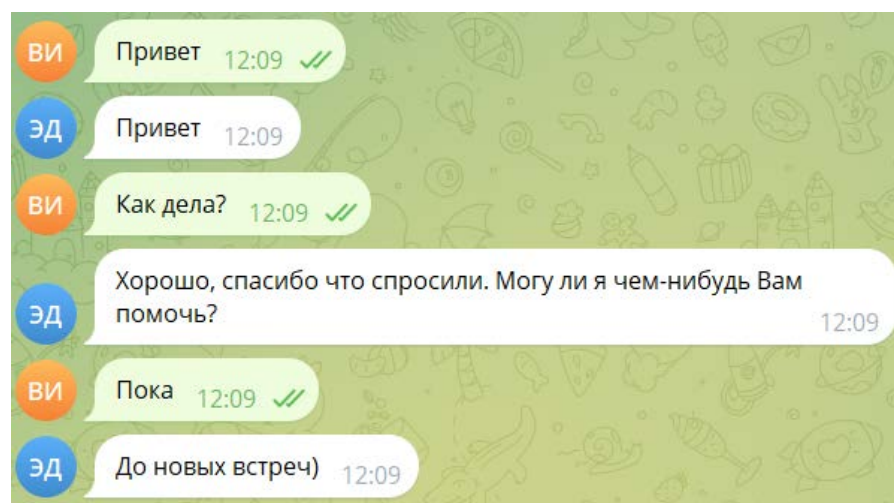


Рисунок 29 – Тестирование

Таким образом, разработанный модуль был протестирован на элементарных примерах для проверки его работоспособности.

ЗАКЛЮЧЕНИЕ

В результате подготовки выпускной квалификационной работы была выполнена разработка программного модуля для консультации пользователей при работе с Электронным деканатом ЭИОС Пензенского государственного университета. Были последовательно выполнены следующие задачи: анализ предметной области и аналогов разрабатываемого модуля, были выбраны необходимые средства разработки, разработаны диаграммы вариантов использования, последовательностей, структура подсистемы, модифицирована структура базы данных подсистемы, а также был создан чат-бот в мессенджере Telegram, и была налажена его интеграция с LMS Moodle. Разработанный модуль выполняет функции оказания технической поддержки путём выполнения следующих действий: идентификация пользователя, приём и анализ сообщения на предмет корректности формулировки и определение типа контекста сообщения, определения роли пользователя, формирование ответа заранее обученной нейронной сетью. Программный модуль готов к внедрению в ЭИОС ПГУ.

Разработка модуля была осуществлена с помощью: GigaIDE, dbForge Studio for MySQL, XAMPP, языков программирования PHP и Python, а также библиотеки TensorFlow для языка Python.

Выпускная квалификационная работа была выполнена в соответствии с поставленным техническим заданием и с соблюдением календарного графика выполнения магистерской работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Игошин И.В., Глотова Т.В., Ширканов А.В. – Разработка компонентов для работы с учебными курсами информационной системы высшего учебного заведения// Информационные технологии в науке и образовании. Проблемы и перспективы. Сборник статей по материалам X Всероссийской научно-практической конференции. – Пенза: Изд-во ПГУ, 2023. – С.
2. И.Г. Кревский, А.В. Антонов, М.В. Деев, Э.А. Зубриянова, И.А. Рензяева, А.В. Ширканов – Управление внедрением информационной системы вуза// Новые информационные технологии и системы. Материалы XV Международной научно-технической конференции, посвященной 75-летию Пензенского государственного университета. Пенза: Изд-во ПГУ, 2018. – С.
3. Национальная исследовательская школа экономики (НИУ ВШЭ): Интернет-ресурс. Ссылка на источник: <https://www.hse.ru> (дата обращения: 27.01.2024).
4. Санкт-Петербургский государственный университет (СПбГУ): Интернет-ресурс. Ссылка на источник: <https://spbu.ru> (дата обращения: 27.01.2024).
5. Московский физико-технический институт (МФТИ): Интернет-ресурс. Ссылка на источник: <https://mipt.ru> (дата обращения: 28.01.2024).
6. Южный федеральный университет (ЮФУ): Интернет-ресурс. Ссылка на источник: <https://sfedu.ru> (дата обращения: 28.01.2024).
7. Уральский федеральный университет имени первого президента России Б.Н. Ельцина (УрФУ): Интернет-ресурс. Ссылка на источник: <https://urfu.ru> (дата обращения: 29.01.2024).
8. Российский университет дружбы народов (РУДН): Интернет-ресурс. Ссылка на источник: <https://rudn.ru> (дата обращения: 29.01.2024).

9. spaCy: Официальный сайт библиотеки spaCy с описанием возможностей и руководствами по использованию. Интернет-ресурс. Ссылка на источник: <https://spacy.io> (дата обращения: 27.05.2024).

10. Transformers (Hugging Face): Документирование и описание библиотеки Transformers от Hugging Face с примером использования и демонстрацией предобученных моделей. Интернет-ресурс. Ссылка на источник: <https://huggingface.co/docs/transformers/index> (дата обращения: 27.05.2024).

11. TensorFlow: Официальный сайт TensorFlow с руководством по использованию модуля обработки естественного языка. Интернет-ресурс. Ссылка на источник: <https://www.tensorflow.org/tutorials/text> (дата обращения: 27.05.2024).

12. NLTK (Natural Language Toolkit): Официальный сайт NLTK с описанием библиотеки и возможностями её применения в обработке естественного языка. Интернет-ресурс. Ссылка на источник: <https://www.nltk.org/> (дата обращения: 27.05.2024).

13. Gensim: Сайт библиотеки Gensim с полным описанием её возможностей и руководства по использованию для обработки естественного языка. Интернет-ресурс. Ссылка на источник: <https://radimrehurek.com/gensim/> (дата обращения: 27.05.2024).

14. FastText: Официальный сайт библиотеки FastText от Facebook с описанием принципов работы и применимости для задач обработки естественного языка. Интернет-ресурс. Ссылка на источник: <https://fasttext.cc/> (дата обращения: 27.05.2024).

15. Python: Официальный сайт Python с описанием языка и его возможностей. Интернет-ресурс. Ссылка на источник: <https://www.python.org> (дата обращения: 27.05.2024).

16. dbForge Studio for MySQL: Руководство по использованию и обзор возможностей dbForge Studio от компании Devart. Интернет-ресурс. Ссылка на источник: [https:// www.devart.com/ dbforge/ mysql/ studio/](https://www.devart.com/dbforge/mysql/studio/)

documentation /index.html (дата обращения: 17.03.2024).

17. Telegram: Информационная статья на ресурсе Хабр о функциях и возможностях мессенджера Telegram.Интернет-ресурс. Ссылка на источник: <https://habr.com/ru/company/telegram/blog/688786/> (дата обращения: 27.03.2024).

18. Giga IDE: Новостная публикация на портале «CNews» о российской среде разработки Giga IDE от компании Сбербанк-Технологии.Интернет-ресурс. Ссылка на источник: https://cnews.ru/news/line/giga_ide_novaya_sreda_razrabotki_dlya_it_specialistov_ot_sberekh (дата обращения: 26.03.2024)

19. ХАМРР: Статья на официальной странице проекта Apache Friends с информацией о пакете ХАМРР и инструкциями по установке.Интернет-ресурс. Ссылка на источник: <https://www.apachefriends.org> (дата обращения: 27.05.2024).

ПРИЛОЖЕНИЕ А

(обязательное)

Программный модуль обработки естественного языка для онлайн-консультаций пользователей электронного деканата ЭИОС ПГУ

Фрагмент обучающей выборки

```
[{
    "type_context": 3,
    "question": "Кто может подписывать преподавателей на дисциплины?",
    "answer": "Подписать преподавателя на дисциплину может только ответственный по кафедре, к которой привязана дисциплина, либо ответственный по одному из вышестоящих над данной кафедрой подразделений. Для этого необходимо перейти на данную кафедру и нажать на ссылку Дисциплины, после чего откроется список дисциплин кафедры, и будет доступна кнопка перехода в интерфейс подписи преподавателей, где можно выбрать всех необходимых преподавателей и подписать их на выбранную дисциплину. Переход на данный интерфейс доступен также через интерфейс состава учебной программы. Если дисциплина преподаётся другой кафедрой, то подпись преподавателя должен осуществлять ответственный с соответствующей кафедры, вне зависимости от владельца учебного плана."
```

```
},{
    "type_context": 3,
    "question": "Что нужно сделать, чтобы прикрепить преподавателя к дисциплине?",
    "answer": "Подписать преподавателя на дисциплину может только ответственный по кафедре, к которой привязана дисциплина, либо ответственный по одному из вышестоящих над данной кафедрой подразделений. Для этого необходимо перейти на данную кафедру и нажать на ссылку Дисциплины, после чего откроется список дисциплин кафедры, и будет доступна кнопка перехода в интерфейс подписи преподавателей, где можно выбрать всех необходимых преподавателей и подписать их на выбранную дисциплину. Переход на данный интерфейс доступен также через интерфейс состава учебной программы. Если дисциплина преподаётся другой кафедрой, то подпись преподавателя должен осуществлять ответственный с соответствующей кафедры, вне зависимости от владельца учебного плана."
```

```
}, {
```

```

        "type_context": 3,
        "question": "Как закрепить преподавателя за
определённой дисциплиной?",
        "answer": "Подписать преподавателя на дисциплину
может только ответственный по кафедре, к которой привязана
дисциплина, либо ответственный по одному из вышестоящих над
данной кафедрой подразделений. Для этого необходимо перейти на
данную кафедру и нажать на ссылку Дисциплины, после чего
откроется список дисциплин кафедры, и будет доступна кнопка
перехода в интерфейс подписи преподавателей, где можно выбрать
выбрать всех необходимых преподавателей и подписать их на
выбранную дисциплину. Переход на данный интерфейс доступен также
через интерфейс состава учебной программы. Если дисциплина
преподаётся другой кафедрой, то подпись преподавателя должен
осуществлять ответственный с соответствующей кафедры, вне
зависимости от владельца учебного плана."
    }, {
        "type_context": 3,
        "question": "Кто уполномочен закреплять
преподавателей за дисциплинами?",
        "answer": "Подписать преподавателя на дисциплину
может только ответственный по кафедре, к которой привязана
дисциплина, либо ответственный по одному из вышестоящих над
данной кафедрой подразделений. Для этого необходимо перейти на
данную кафедру и нажать на ссылку Дисциплины, после чего
откроется список дисциплин кафедры, и будет доступна кнопка
перехода в интерфейс подписи преподавателей, где можно выбрать
выбрать всех необходимых преподавателей и подписать их на
выбранную дисциплину. Переход на данный интерфейс доступен также
через интерфейс состава учебной программы. Если дисциплина
преподаётся другой кафедрой, то подпись преподавателя должен
осуществлять ответственный с соответствующей кафедры, вне
зависимости от владельца учебного плана."
    },
]

```

ПРИЛОЖЕНИЕ Б

(обязательное)

Программный модуль обработки естественного языка для онлайн-консультаций пользователей электронного деканата ЭИОС ПГУ

Презентация

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема: «Программный модуль обработки
естественного языка для онлайн-консультаций
пользователей электронного деканата ЭИОС
ПГУ»

Студент: Игошин И.В.

Руководитель: д.т.н., профессор Бершадский А.М.

Цель работы

Увеличение качества и скорости оказаний услуг технической поддержки для пользователей Электронного деканата ЭИОС ПГУ и повышения удобства работы с данной системой в целом.

Объект разработки: Программный модуль для обработки естественного языка.

Обзор аналогов

- Национальный исследовательский университет «Высшая школа экономики» (НИУ ВШЭ),
- Санкт-Петербургский государственный университет (СПбГУ),
- Московский физико-технический институт (МФТИ),
- Южный федеральный университет (ЮФУ),
- Уральский федеральный университет имени первого президента России Б.Н. Ельцина (УрФУ),
- Российский университет дружбы народов (РУДН).

Диаграмма прецедентов

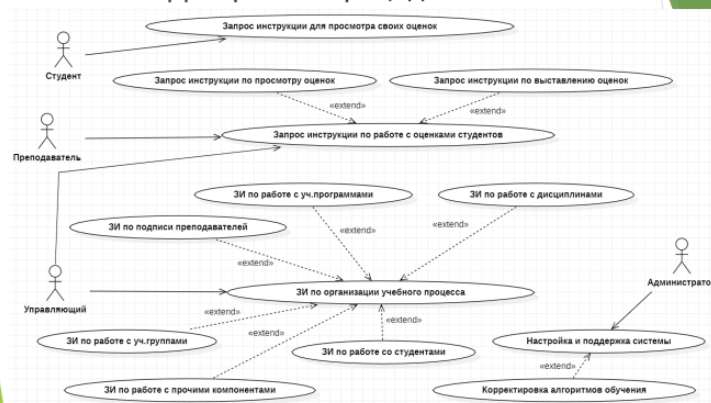
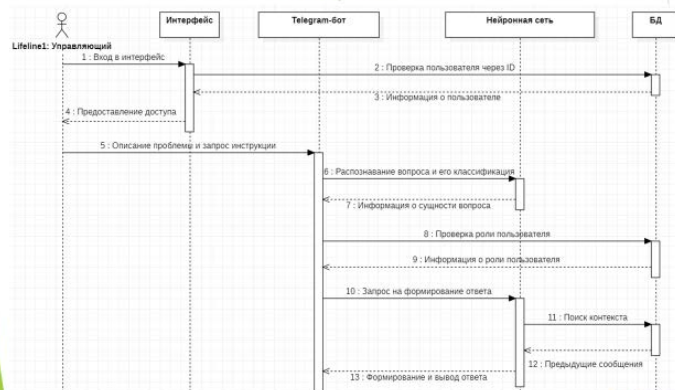


Диаграмма последовательностей работы модуля



5

Средства разработки

- Python 3.7
- TensorFlow
- Telegram
- GigaIDE
- dbForge Studio for MySQL 2020
- XAMPP
- PHP 7.2

6

Обучение модели

Использована рекуррентная нейронная сеть с механизмом долгой краткосрочной памяти, состоящая из трёх основных слоёв:

- Слой 1: Преобразует слова в плотные векторные представления, позволяющие уловить взаимосвязи между ними.
- Слой 2: Анализирует контекст вопроса и учитывает длительные зависимости между словами, позволяя сохранять важные детали и фильтровать несущественную информацию.
- Слой 3: Генерирует финальный ответ путём выбора наиболее подходящего слова на основе распределения вероятностей.

7

Обучение модели

Алгоритм обучения: Используется метод градиентного спуска совместно с обратным распространением ошибки через временные шаги для оптимизации параметров сети.

Входные данные: Набор структурированных JSON-документов, включающий пары вида "вопрос-ответ", используемые для тренировки модели.

Поля каждого обучающего объекта:

- ❖ type_context - тип контекста указывает на компонент Электронного деканата, при работе с которым у пользователя возникли проблемы. Данные типы хранятся в базе данных,
- ❖ question - вопрос пользователя,
- ❖ answer - ответ на вопрос пользователя.

8

Обучение модели

Кто может подписывать преподавателей на дисциплины?
Перефразируй данный вопрос в 10 вариантах

Ответ 🗨

1. Кто обладает правом назначать преподавателей на учебные дисциплины?
2. Кто вправе утверждать участие преподавателей в чтении курсов?
3. Какой сотрудник может закрепить педагогов за предметами?
4. Кто занимается распределением преподавателей по дисциплинам?
5. Кто принимает решение о назначении преподавателей на предметы?
6. Кто отвечает за распределение обязанностей преподавателей по дисциплинам?
7. Кто наделён полномочиями зачислять преподавателей на курсы?
8. Кто распоряжается назначением сотрудников на проведение занятий по дисциплинам?
9. Кто осуществляет прикрепление преподавателей к конкретным предметам?
10. Кто уполномочён оформлять подбор преподавателей для каждой дисциплины?

9

Пример обучающих данных

```
{
  "type_context": 3,
  "question": "Кто может подписывать преподавателей на дисциплины?",
  "answer": "Подписать преподавателя на дисциплину может только ответственный по кафедре, к которой привязана дисциплина, либо ответственный по одному из вышестоящих над данной кафедрой подразделений. Для этого необходимо перейти на данную кафедру и нажать на ссылку Дисциплины, после чего откроется список дисциплин кафедры, и будет доступна кнопка перехода в интерфейс подписи преподавателей, где можно выбрать всех необходимых преподавателей и подписать их на выбранную дисциплину. Переход на данный интерфейс доступен также через интерфейс состава учебной программы. Если дисциплина преподаётся другой кафедрой, то подпись преподавателя должен осуществлять ответственный с соответствующей кафедры, вне зависимости от владельца учебного плана."
}
```

10

Перспективы

- ▶ Приём иных форматов сообщений от пользователей, помимо текста (фото, видео, голосовые сообщения),
- ▶ Генерирование ответа в различных форматах (например, текст можно заменить на видеопояснение для пользователей),
- ▶ Распространить данный модуль не только на Электронный деканат, но и на другие компоненты ЭИОС.

11

Заключение

В результате подготовки выпускной квалификационной работы был разработан программный модуль обработки естественного языка для онлайн-консультаций пользователей Электронного деканата ЭИОС ПГУ.

Разработанный модуль выполняет функции оказания технической поддержки путём выполнения следующих действий: идентификация пользователя, приём и анализ сообщения на предмет корректности формулировки и определение типа контекста сообщения, определения роли пользователя, формирование ответа заранее обученной нейронной сетью.

Выпускная квалификационная работы была выполнена в соответствии с поставленным техническим заданием и с соблюдением календарного графика выполнения магистерской диссертации.

12