

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

<i>Факультет</i>	<i>Кафедра</i>
вычислительной техники	МОиПЭВМ
Направление	09.03.02 «Информационные системы и технологии»
подготовки	
Направленность	Информационные системы и технологии

БАКАЛАВРСКАЯ РАБОТА
на тему
Разработка чат-бота для иностранных обучающихся

Студент	_____ Пшеницын Владислав Валерьевич
	(подпись, дата) (ФИО полностью)
Руководитель	_____ Попова Н.А.
	(подпись, дата) (фамилия, инициалы)
Нормоконтролёр	_____ Такташкин Д.В.
	(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой	_____ Козлов А.Ю.
	(подпись) (фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК	_____ Попова Н.А.
	(подпись) (фамилия, инициалы)

Пенза, 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
_____ А.Ю. Козлов
«__» _____ 2025 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

1. Студент _____ Пшеницын Владислав Валерьевич _____ гр. 21ВИ1 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы _____ Разработка чат-бота для иностранных обучающихся

Тема утверждена приказом ПГУ № _____ от " _____ " _____ 20 _____

3. Руководитель работы _____ к.т.н., доцент кафедры «МО и ПЭВМ» Попова Н.А.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение: поддержка иностранных обучающихся.

Основные функции системы:

1. Отвечать на вопросы пользователя;

Технология разработки: предметно-ориентированное программирование.

Язык программирования: C#

Среда программирования: Visual Studio 2022.

5. Перечень подлежащих разработке вопросов:

1. анализ предметной области и постановка задачи;

2. проектирование программных средств;

3. реализация программных средств;

4. тестирование программных средств;

5. оформление ВКР.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и постановка задачи	10%	13.04.2025 – 19.04.2025	
2	Проектирование программных средств	20%	20.04.2025 – 03.05.2025	
3	Реализация программных средств	30%	04.05.2025 – 31.05.2025	
4	Тестирование программных средств	20%	01.06.2025 – 07.06.2025	
5	Оформление ВКР	20%	08.06.2025 – 17.06.2025	

Дата выдачи задания "___" _____ 20____

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент _____
(подпись, дата)

Работу к защите допустить
Декан факультета _____
(подпись, дата)

Реферат

Пояснительная записка содержит 57 листов, 16 рисунков, 11 таблиц, 30 использованных источников и 2 приложения.

ЧАТ-БОТ, ИНФОРМАЦИОННАЯ СИСТЕМА, VISUAL STUDIO, UML, IDEF0, IDEF1X.

Предмет работы: разработка информационной системы поддержки предприятий общественного питания.

Объект работы: автоматизация деятельности предприятий общественного питания.

Цель работы: Разработка чат-бота для иностранных обучающихся.

Технология разработки: Visual Studio, SQL Server Management Studio.

Результаты работы: спроектирован и разработан чат-бот в среде Visual Studio 2022 для иностранных обучающихся.

Область применения: вузы с иностранными студентами.

					ПГУ 09.03.02 – 08ВР181.24 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Пшеницын В.В.			Разработка чат-бота для иностранных обучающихся. Пояснительная записка	Лит.	Лист	Листов
Провер.		Попова Н.А.					4	57
Реценз.						группа 21ВИ1		
Н. Контр.		Такташкин Д.В.						
Утверд.		Макарычев П.П.						

Содержание

Введение.....	6
1 Постановка задачи и анализ аналогов разрабатываемой системы	9
1.1 Постановка задачи.....	9
1.2 Анализ информационных систем аналогичного назначения.....	9
1.3 Выбор технологий проектирования и разработки	16
2 Проектирование информационной системы	20
2.1 Анализ бизнес-процессов предприятия	20
2.2 Проектирование логической модели базы данных	23
2.3 Проектирование программного обеспечения	25
2.3.1 Анализ прецедентов.....	26
2.3.2 Диаграмма состояний	29
2.3.3 Диаграмма деятельности	30
3 Планирование и расчет бюджета разрабатываемой системы	32
4 Реализация и тестирование системы	35
4.1 Реализация структур программного средства	35
4.2 Реализация кода информационной системы.....	36
4.3 Тестирование информационной системы	37
4.3.1 Функциональное тестирование.....	37
4.3.2 Метрика Холстеда.....	41
Заключение	45
Список использованных источников	48
Приложение А	52
Приложение Б.....	56

Введение

При разработке чат-бота для взаимодействия с иностранными обучающимися особенно важно учитывать многообразие потребностей, ожиданий и культурных особенностей, характерных для данной категории студентов. Иностранные студенты сталкиваются с рядом уникальных вызовов, связанных не только с учебной программой, но и с адаптацией к новой социальной и культурной среде, языковыми барьерами, организационными вопросами и бытовыми трудностями. В этом контексте чат-боты становятся незаменимым инструментом, позволяющим обеспечить эффективное и своевременное взаимодействие между учебными заведениями и их международными студентами, предоставляя им необходимую поддержку и информацию в удобном и доступном формате.

Данная разработка направлена на создание комплексной информационной системы, способной предоставлять актуальную и релевантную информацию о разнообразных аспектах обучения за границей. Это включает в себя подробные сведения о учебных курсах и программах, процессах поступления и регистрации, требованиях к документам, а также информацию, необходимую для успешной адаптации к новой культурной среде и повседневной жизни в иностранном государстве. Чат-бот также должен обеспечивать поддержку в вопросах проживания, медицинского страхования, социальных мероприятий и других важных сфер, влияющих на комфорт и успешность обучения иностранных студентов.

Современные образовательные учреждения находятся в условиях высокой конкуренции за привлечение и удержание иностранных студентов, что требует постоянного совершенствования качества обслуживания и повышения вовлеченности обучающихся. Стремление к улучшению процессов коммуникации, обеспечения оперативного доступа к информации и создания благоприятных условий для адаптации приводит к необходимости внедрения автоматизированных решений, которые способны значительно

повысить эффективность образовательного процесса и уровень удовлетворенности студентов. В этом плане чат-боты представляют собой инновационный инструмент, позволяющий обеспечить круглосуточный доступ к информации, мгновенно отвечать на вопросы и запросы студентов, а также облегчать их интеграцию в новую образовательную и социальную среду.

Одним из ключевых преимуществ использования чат-ботов является их способность собирать и анализировать данные о взаимодействии пользователей с системой. Это позволяет выявлять наиболее часто возникающие проблемы, понимать потребности и предпочтения студентов, а также адаптировать и улучшать предоставляемые услуги на основе полученной аналитики. Такие возможности способствуют не только повышению качества поддержки иностранных обучающихся, но и снижению уровня стресса и неопределенности, связанных с процессом адаптации к учебе в другом городе или стране. В результате внедрение подобного программного решения способствует созданию более комфортной и поддерживающей образовательной среды, что положительно сказывается на успеваемости и общем опыте студентов.

Объектом исследования в рамках данной выпускной квалификационной работы выступает процесс автоматизации взаимодействия с иностранными обучающимися в условиях образовательных учреждений. Это включает в себя организацию коммуникации, предоставление информационной поддержки и содействие в решении различных учебных и административных вопросов, с которыми сталкиваются зарубежные студенты. Предметом исследования является информационная система, специально разработанная для поддержки и удовлетворения потребностей иностранных студентов, обеспечивающая удобный и эффективный канал коммуникации между студентами и учебным заведением.

Главной целью работы является разработка функционального и удобного в использовании чат-бота, который будет способствовать повышению уровня комфорта, удобства и эффективности взаимодействия иностранных студентов с образовательной системой. Для достижения поставленной цели необходимо решить ряд комплексных задач, включающих в себя глубокое исследование и анализ потребностей целевой аудитории, выявление ключевых сценариев взаимодействия пользователей с системой, проектирование и реализацию необходимого функционала чат-бота, а также проведение тестирования и оценки качества созданного программного продукта. Особое внимание уделяется обеспечению многоязычной поддержки, адаптации интерфейса под различные культурные особенности и интеграции с существующими информационными ресурсами учебного заведения.

Таким образом, реализация данного проекта направлена на создание инновационного цифрового инструмента, который позволит значительно улучшить процесс адаптации и обучения иностранных студентов, повысить качество их взаимодействия с учебным заведением и способствовать развитию конкурентных преимуществ образовательной организации на международном рынке образовательных услуг.

1 Постановка задачи и анализ аналогов разрабатываемой системы

1.1 Постановка задачи

Разработка чат-бота для взаимодействия с иностранными обучающимися становится неотъемлемой и крайне актуальной частью современных образовательных технологий, что отражает глубокое и обширное влияние информационных технологий на различные сферы человеческой деятельности. В настоящее время трудно представить какую-либо область, в которой бы не применялись вычислительные системы и цифровые инструменты, и образование, безусловно, не является исключением из этого правила. В частности, внедрение и активное использование чат-ботов открывает новые инновационные возможности для автоматизации и оптимизации процессов взаимодействия между учебными учреждениями и студентами, особенно теми, кто прибывает из других стран и культурных сред.

Создание и внедрение чат-ботов позволяет эффективно автоматизировать ключевые процессы общения с иностранными обучающимися, предоставляя им своевременный и удобный доступ к необходимой информации в формате, максимально адаптированном под их потребности и особенности восприятия. Чат-боты способны выполнять широкий спектр функций, схожих с задачами фронт-офисных систем: они обеспечивают оперативное и непрерывное взаимодействие, помогают студентам в решении разнообразных вопросов, связанных с учебным процессом, предоставляют актуальные сведения о расписании занятий, оценках, мероприятиях и других аспектах студенческой жизни, которые могут быть особенно важны и интересны для иностранных студентов. Благодаря круглосуточной доступности и возможности функционировать в режиме 24/7, чат-боты обеспечивают постоянную поддержку, что является крайне важным фактором для студентов, находящихся в различных часовых поясах, а также для тех, кто испытывает трудности с личным обращением в учебные отделы из-за языковых барьеров или организационных причин.

Кроме того, современные решения на базе чат-ботов могут включать в себя и бэк-офисные компоненты, которые позволяют собирать, обрабатывать и анализировать аналитические данные о взаимодействии пользователей с системой. Такая возможность предоставляет образовательным учреждениям ценный инструмент для выявления тенденций в поведении студентов, анализа их удовлетворенности предоставляемыми услугами и адаптации прочих образовательных и сервисных процессов в соответствии с реальными потребностями обучающихся. Аналитика взаимодействий с чат-ботом позволяет выявлять наиболее часто возникающие вопросы, проблемы и затруднения, с которыми сталкиваются иностранные студенты, что способствует не только улучшению учебных программ и сервисов поддержки, но и формированию более качественной и ориентированной на пользователя образовательной среды. Помимо этого, чат-боты могут интегрироваться с современными системами управления обучением (Learning Management Systems, LMS), что позволяет автоматизировать и оптимизировать процессы учета успеваемости, контроля посещаемости и взаимодействия между преподавателями и студентами. Такая интеграция способствует созданию единой, скоординированной и информационно насыщенной среды, где вся необходимая информация доступна в режиме реального времени, что значительно повышает качество и эффективность образовательного процесса в целом.

Особое значение внедрение чат-ботов приобретает для небольших и средних учебных заведений, которые зачастую ограничены в ресурсах и не могут позволить себе дорогостоящие и сложные системы управления. В таких случаях использование чат-ботов становится особенно выгодным, поскольку их внедрение и эксплуатация требуют значительно меньших затрат по сравнению с традиционными и крупномасштабными системами. При этом функционал чат-ботов может быть гибко настроен и адаптирован именно под нужды конкретного учебного заведения, что позволяет избежать избыточности и сосредоточиться на наиболее важных аспектах

взаимодействия с иностранными обучающимися. Благодаря модульной архитектуре современные чат-боты легко масштабируются, расширяются и адаптируются под изменяющиеся требования и условия, что обеспечивает долгосрочную эффективность, устойчивость и экономичность внедряемых решений. Кроме того, активное использование искусственного интеллекта, машинного обучения и технологий обработки естественного языка (Natural Language Processing, NLP) позволяет создавать более «человечные», интуитивно понятные и адаптивные интерфейсы, что значительно снижает психологические и коммуникативные барьеры в общении, способствует лучшей интеграции иностранных студентов в образовательное и социальное сообщество, а также повышает уровень их вовлеченности и удовлетворенности. Таким образом, разработка и внедрение чат-бота становится целесообразным и стратегически важным шагом для повышения качества обслуживания иностранных студентов, модернизации образовательного процесса и создания более комфортной и поддерживающей среды обучения. Внедрение подобных цифровых решений способствует не только улучшению коммуникации и доступности информации, но и формированию более гибкой, адаптивной и ориентированной на пользователя образовательной среды, которая способствует успешной учебе, социальной адаптации и личностному развитию иностранных обучающихся. В перспективе развитие технологий чат-ботов будет идти в направлении более глубокой персонализации взаимодействия, расширения возможностей интеграции с различными образовательными и административными системами, а также активного использования аналитики больших данных и искусственного интеллекта для прогнозирования потребностей студентов, оптимизации учебных программ и повышения эффективности образовательных процессов. Это открывает новые горизонты для создания интеллектуальных образовательных экосистем, способных динамично адаптироваться к вызовам современного мира и обеспечивать высокое качество международного образования. региональных образовательных

учреждений, которые зачастую ограничены в ресурсах и не могут обеспечить полноценную круглосуточную поддержку иностранных студентов через традиционные каналы. В таких условиях чат-бот становится незаменимым инструментом, позволяющим расширить возможности коммуникации и повысить уровень сервиса при минимальных затратах. Он способствует снижению нагрузки на административный персонал, освобождая их время для решения более сложных и индивидуальных задач, требующих человеческого участия.

Важным аспектом разработки чат-бота для иностранных обучающихся является учет многообразия культурных и языковых особенностей целевой аудитории. Эффективный чат-бот должен поддерживать несколько языков общения, обеспечивать корректное восприятие и интерпретацию запросов, а также предлагать релевантные и адаптированные ответы с учетом специфики образовательной системы страны пребывания. Это требует тщательной проработки сценариев взаимодействия, использования технологий обработки естественного языка (Natural Language Processing, NLP) и внедрения механизмов машинного обучения, позволяющих системе постоянно совершенствоваться и адаптироваться к меняющимся потребностям пользователей.

Объектом исследования в данном контексте выступает процесс автоматизации взаимодействия с иностранными обучающимися с помощью цифровых технологий, а предметом - информационная система поддержки, реализованная в форме чат-бота. Целью разработки является создание функционального и интуитивно понятного инструмента, способного повысить удобство и эффективность коммуникации студентов с образовательной организацией, облегчить процесс адаптации и интеграции в учебную среду, а также улучшить качество предоставляемых услуг.

Для достижения поставленной цели необходимо провести комплексный анализ потребностей иностранных студентов, выявить ключевые задачи и

проблемы, с которыми они сталкиваются, а также разработать сценарии взаимодействия, обеспечивающие максимально полный и релевантный ответ на их запросы. На основе полученных данных следует спроектировать функционал чат-бота, включающий в себя модули информирования, консультации, поддержки и сбора обратной связи. Важным этапом является тестирование и внедрение системы, а также ее дальнейшее сопровождение и развитие с учетом отзывов пользователей и изменений в образовательной среде.

Итогом работы станет инновационный цифровой инструмент, способствующий улучшению адаптации иностранных студентов, повышению их удовлетворенности учебным процессом и сервисами поддержки, а также укреплению позиций образовательного учреждения на международном образовательном рынке. Таким образом, разработка и внедрение чат-бота для взаимодействия с иностранными обучающимися представляет собой стратегически важное направление развития образовательных технологий, отвечающее современным вызовам и требованиям глобализированного мира.

1.2 Анализ информационных систем аналогичного назначения

Для более глубокого понимания преимуществ и ограничений подобных систем целесообразно рассмотреть дополнительные примеры и особенности их реализации. Например, чат-бот «Mia» демонстрирует высокий уровень интеграции с внутренними информационными системами университета, что позволяет ему предоставлять актуальные и персонализированные ответы. Однако такая тесная интеграция требует значительных усилий по настройке и поддержке, а также высокой степени защиты данных, что может стать вызовом для учебных заведений с ограниченными ресурсами.

Помимо «Mia», на рынке представлены и другие решения, ориентированные на поддержку иностранных студентов. Некоторые из них специализируются на предоставлении информации по визовым вопросам,

адаптации в новой стране, а также культурной интеграции. Например, чат-боты, разработанные в рамках международных программ обмена, включают функции, направленные на помощь студентам в освоении местных правил и традиций, что существенно облегчает процесс адаптации.

При анализе существующих систем важно учитывать такие аспекты, как удобство пользовательского интерфейса, качество обработки естественного языка, возможность масштабирования и адаптации под специфические требования конкретного учебного заведения. Кроме того, значительную роль играет поддержка многоязычности, так как иностранные студенты часто имеют разный уровень владения языком обучения. Интеграция с мобильными платформами и мессенджерами также повышает доступность и удобство использования чат-ботов.

Несмотря на очевидные преимущества, существуют и определённые сложности при внедрении подобных решений. К ним относятся необходимость регулярного обновления базы знаний чат-бота, обеспечение безопасности персональных данных, а также адаптация алгоритмов обработки запросов к разнообразным культурным и языковым особенностям пользователей. Кроме того, важно обеспечить возможность передачи сложных или нестандартных запросов на обработку живым сотрудникам, чтобы не возникало ситуаций, когда пользователь остаётся без поддержки.

Таким образом, анализ существующих систем показывает, что эффективный чат-бот для иностранных студентов должен сочетать в себе высокую степень интеграции с учебными и административными системами, многоязычность, удобство использования и надёжность. Учитывая эти требования и опыт успешных проектов, можно разработать решение, которое не только улучшит качество обслуживания, но и повысит уровень удовлетворённости студентов, способствуя их успешной учёбе и адаптации в новом образовательном пространстве.

Таблица 1 – Сравнение систем-аналогов

Система	Достоинства	Недостатки
---------	-------------	------------

Tandem	Возможность реальных разговоров с носителями языка.культурный обмен.доступность для пользователей разных уровней владения языком.поддержка сообщества.	Непоследовательная обратная связь.сложности с координацией собеседников из разных часовых зон.необходимость привыкания к функциям приложения.опасения пользователей по поводу безопасности при обмене личной информацией
HiNative	Возможность задавать вопросы иностранцам об их языке и культуре.готовые формулировки вопросов.возможность загрузить свою аудиозапись, чтобы носители языка проверили и при необходимости исправили произношение.	На носителей английского языка большой спрос, поэтому не всегда можно получить подробные ответы.невозможность услышать голосовые ответы на вопросы, которые не являются собственными, без платной подписки.
Polyglot Club	Разнообразие ресурсов: учебные материалы, статьи и форумы для обсуждения языковых вопросов.возможность организации встреч и участия в них.настраиваемый процесс обучения: выбор языков для изучения и собеседников в зависимости от уровня владения языком и интересов.	Уровень взаимодействия и отзывчивости пользователей может различаться.отсутствие структурированных курсов и программы обучения, которые могут быть предпочтительнее для некоторых пользователей.потенциальные проблемы с безопасностью при участии в мероприятиях с незнакомыми людьми.
HelloTalk	Возможность перевода сообщений в самом приложении.возможность слышать живую речь на эфирах и звонках.есть люди с полезным материалом на страничках.много людей для общения.можно узнать, как живут люди в другой точке мира.озвучка сообщений.тонкие настройки приватности.хорошая система рекомендаций.	Агрессивная реклама покупки VIP-аккаунта.бесплатно можно одновременно учить только 1 язык.дорогие тарифы.много багов.нельзя отправлять большие фото.сильно ограничены возможности бесплатного аккаунта.сложно найти носителя языка

Согласно характеристикам, представленным в описании систем аналогов, проведено сравнение этих систем. Результаты представлены в таблице 2. Цифровые баллы проявления критериев обозначают следующее: 1 –

низкое проявление критерия, 2 – среднее проявление критерия, 3 – высокое проявление критерия.

Таблица 2 – Критериальная оценка систем-аналогов

Система	Критерий				
	Функциональная сложность организации системы	Качество интерфейса	Анализ данных	Стоимость	Производительность
Tandem	2	3	2	2	3
HiNative	2	3	2	2	3
Polyglot Club	3	2	3	3	3
HelloTalk	3	2	3	3	1

По результатам оценки каждого аналога по выделенным критериям можно сделать вывод, что существующие информационные системы поддержки предприятий общественного питания являются достаточно дорогими и имеют перегруженный функционал, что не подходит для малых предприятий. Вследствие чего возникает необходимость в разработке программного обеспечения, которое будет включать в себя только наиболее важные функции для небольших заведений. Такое решение будет оптимальным, так как сократит затраты на приобретение программного решения, а также уменьшит время обучения сотрудников предприятия ввиду отсутствия избыточного функционала.

1.3 Выбор технологий проектирования и разработки

Процесс проектирования информационной системы поддержки предприятий общественного питания представляет собой комплексный и многоэтапный цикл, включающий в себя детальное моделирование бизнес-процессов, функциональных требований и структуры данных. Для достижения высокой степени формализации и обеспечения качества проектирования в данной работе применяется сочетание различных нотаций и инструментальных средств, что позволяет охватить все ключевые аспекты системы — от процессов и функционала до организации данных. В частности, основными методами визуализации и спецификации выступают UML-

диаграммы, функциональная диаграмма в нотации IDEF0 и модель данных в нотации IDEF1X.

UML (Unified Modeling Language) является универсальным языком моделирования, который широко используется для описания архитектуры, поведения и структуры программных систем. Диаграммы UML позволяют разработчикам и заказчикам получить общее представление о системе, выявить основные сценарии взаимодействия пользователей с системой, определить классы, их атрибуты и связи. В рамках проектирования информационной системы поддержки предприятий общественного питания применение UML-диаграмм обеспечивает формализацию требований и способствует выявлению всех ключевых функций и ролей, что особенно важно для создания комплексных и масштабируемых решений.

Функциональная диаграмма в нотации IDEF0 служит для моделирования и анализа бизнес-процессов, отображая функции системы, управляющие и информационные потоки между ними. Данный метод позволяет структурировать процессы, выявить взаимосвязи и определить входы, выходы, механизмы и управляющие воздействия на каждый этап работы системы. Использование IDEF0 способствует оптимизации бизнес-процессов, выявлению узких мест и повышению эффективности работы предприятий общественного питания за счет четкого понимания и документирования всех этапов деятельности.

Модель данных в нотации IDEF1X обеспечивает формальное описание структуры данных, их атрибутов, связей и ограничений. Это особенно важно для построения надежной и масштабируемой базы данных, которая станет основой хранения и обработки информации в системе. IDEF1X позволяет не только визуализировать структуру данных, но и подготовить логическую модель, которая впоследствии трансформируется в физическую модель, учитывающую особенности конкретной системы управления базами данных (СУБД). Такой подход гарантирует целостность данных, оптимизацию запросов и удобство сопровождения информационной системы.

Для реализации вышеописанных моделей и диаграмм были выбраны специализированные CASE-средства, обладающие необходимым функционалом и удобством использования. В частности, ERwin Process Modeler является мощным инструментом, позволяющим создавать функциональные модели в нотации IDEF0, а также моделировать потоки данных с помощью DFD и технологические процессы в нотации IDEF3. Это обеспечивает комплексный подход к анализу и оптимизации бизнес-процессов, что критично для предприятий общественного питания, где важна координация множества операций и ресурсов.

ERwin Data Modeler 7.3 специализируется на построении логических и физических моделей данных, предоставляя широкие возможности для проектирования баз данных и генерации кода для различных СУБД. Это позволяет эффективно создавать и поддерживать структуру данных, необходимую для хранения информации о меню, заказах, клиентах, поставщиках и других ключевых объектах системы. Использование данного инструмента обеспечивает высокое качество моделей данных и упрощает интеграцию с другими компонентами системы.

ArgoUML выступает как удобное и бесплатное средство для построения UML-диаграмм, что делает его привлекательным выбором для разработки моделей взаимодействия пользователей и компонентов системы. Этот инструмент поддерживает основные типы диаграмм UML, включая диаграммы прецедентов, классов, последовательностей и состояний, что позволяет всесторонне описать поведение и структуру системы.

Visual Paradigm представляет собой современный и многофункциональный инструмент для UML-моделирования, поддерживающий не только UML 2.0, но и SysML, а также нотации моделирования бизнес-процессов. Помимо создания диаграмм, Visual Paradigm предлагает возможности генерации отчетов, автоматической разработки кода и интеграции с популярными средами разработки, что

значительно ускоряет процесс проектирования и реализации системы. Наличие бесплатных версий и интуитивно понятного интерфейса делает его удобным для использования как опытными инженерами, так и начинающими разработчиками.

Выбор именно этих CASE-средств обусловлен сочетанием нескольких факторов: наличием полного набора необходимых функций для моделирования процессов, данных и функционала; простотой освоения и использования; удобством пользовательского интерфейса; а также доступностью бесплатных версий, что снижает затраты на разработку и позволяет гибко управлять ресурсами проекта. Такое комплексное применение инструментов способствует созданию качественной, хорошо структурированной и документированной информационной системы, отвечающей современным требованиям и стандартам.

Для непосредственной реализации разработанной системы была выбрана платформа Visual Studio — интегрированная среда разработки (IDE), которая предоставляет разработчикам все необходимые инструменты для полного цикла создания программного продукта. Visual Studio отличается высокой производительностью, поддержкой множества языков программирования, таких как C#, C++, JavaScript, TypeScript, Python и других, что позволяет реализовывать разнообразные компоненты системы — от серверной логики и баз данных до пользовательских интерфейсов и веб-приложений.

Visual Studio объединяет в себе редактор кода с интеллектуальным автодополнением и подсветкой синтаксиса, мощные отладочные средства, инструменты для управления версиями и интеграцию с системами контроля исходного кода, что облегчает совместную работу над проектом в команде. Кроме того, среда поддерживает создание кроссплатформенных приложений, что позволяет разрабатывать решения, работающие на различных операционных системах и устройствах, включая мобильные платформы.

Особое внимание в Visual Studio уделено поддержке современных технологий и инструментов искусственного интеллекта, которые улучшают эффективность разработки за счет автоматической генерации кода, анализа ошибок и предложений по оптимизации. Это значительно сокращает время разработки и повышает качество конечного продукта.

Установщик Visual Studio построен на основе концепции рабочих нагрузок, что позволяет разработчику выбирать и устанавливать только те компоненты и инструменты, которые необходимы для конкретного проекта, избегая лишних затрат места и ресурсов. Такой подход обеспечивает гибкость и адаптивность среды под индивидуальные потребности команды.

В совокупности, использование Visual Studio в качестве платформы разработки обеспечивает полный цикл — от написания и редактирования кода до его отладки, тестирования и развертывания готового приложения. Благодаря этому достигается высокая производительность и качество разработки, а также возможность эффективного управления проектом на всех этапах жизненного цикла программного обеспечения. Это делает Visual Studio оптимальным выбором для реализации информационной системы поддержки предприятий общественного питания, учитывая требования к функциональности, надежности и удобству использования конечного продукта

2 Проектирование информационной системы

Проектирование информационной системы (ИС) - процесс создания проекта-прототипа системы, который включает разработку её структуры, программ, интерфейсов и других элементов. Цель проектирования — обеспечить требуемую функциональность системы, её адаптивность к изменяющимся условиям, безотказную работу и безопасность.

Некоторые этапы проектирования ИС:

- Определение требований к системе. Формулируются цели и задачи проекта, формируются требования заказчиков.

- Моделирование бизнес-процессов. Создаётся модель организации, которая описывает процессы и функции, позволяющие сформулировать основные требования к ИС.
- Формирование моделей данных. Построение логической и физической моделей данных, которые преобразуются из информационной модели, разработанной на этапе анализа.
- Проектирование процессов. Отображение функций, полученных на этапе анализа, в модули ИС.
- Разработка архитектуры ИС. Выбор платформы и операционной системы, определение характеристик архитектуры (например, «файл-сервер» или «клиент-сервер»).

2.1 Анализ бизнес-процессов предприятия

Перед разработкой информационной системы необходимо провести анализ бизнес-процессов организации, деятельность которой предполагается автоматизировать. Для того чтобы проанализировать бизнес-процессы организации, необходимо построить функциональную модель, соответствующую предметной области [13]. Построение функциональной модели осуществляется в нотации IDEF0, которая предназначена для формализации и описания бизнес-процессов [14].

Для построения диаграммы бизнес-процессов было выбрано CASE-средство ERwin Process Modeler.

Контекстная диаграмма бизнес-процессов в нотации IDEF0 для предметной области предприятие общественного питания представлена на рисунке 1.

Контекстная диаграмма состоит из единственного функционального блока, который представляет собой общее описание системы и ее взаимодействие с внешней средой.

Основная функция, выделенная на контекстной диаграмме, была разделена (декомпозирована) на составные части путем создания дочерней

диаграммы следующего, более низкого уровня. Диаграмма декомпозиции представлена на рисунке 2.

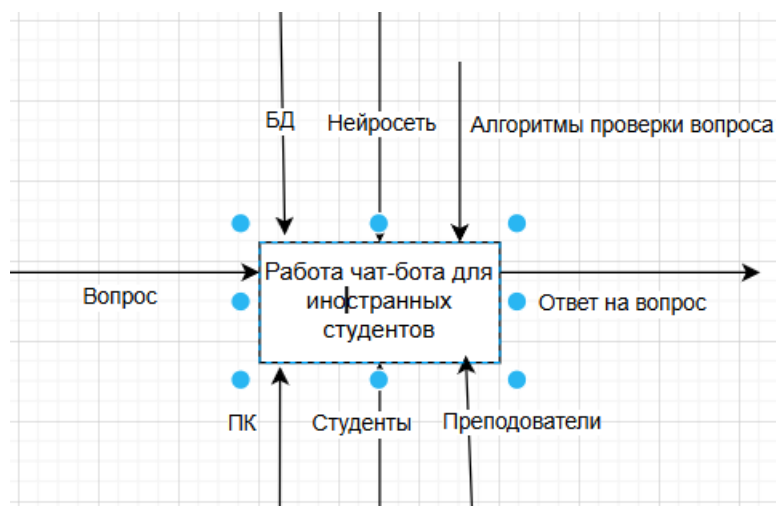


Рисунок 1 - Контекстная диаграмма

В результате декомпозиции основного процесса «Работа чат-бота для иностранных студентов» было выделено 2 дочерних процесса: «Получение вопроса» и «Генерация ответа».

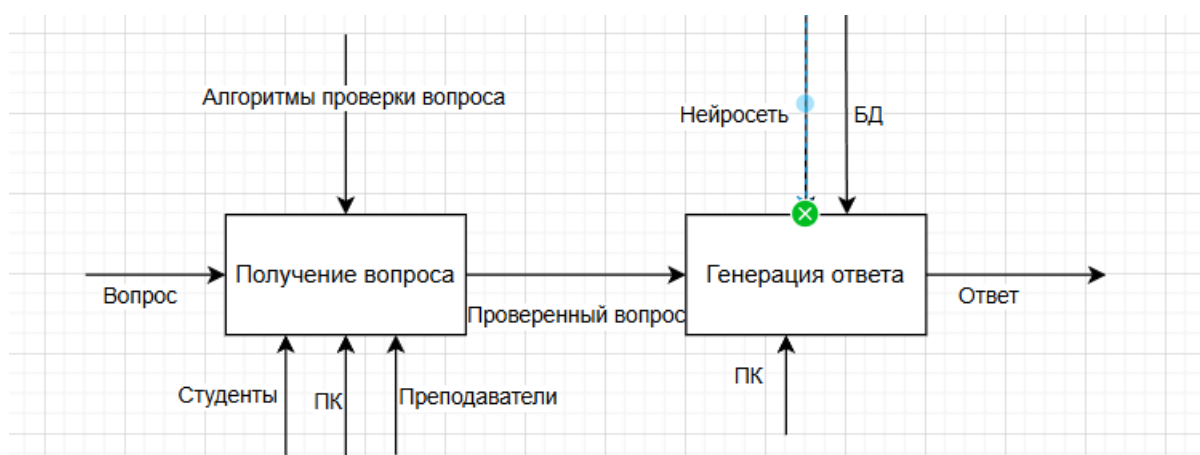


Рисунок 2 - Детализирующая диаграмма процесса «Работа чат-бота для иностранных студентов» в нотации IDEF0

Процесс «Получение вопроса» предполагает проверку вопроса на корректную длину и символы.

Процесс «Генерация ответа» предполагает поиск ответа в БД или его генерацию от нейросети в случае если ответа нет.

Декомпозиция процесса «Генерация ответа» в нотации IDEF0 представлена на рисунке 3.

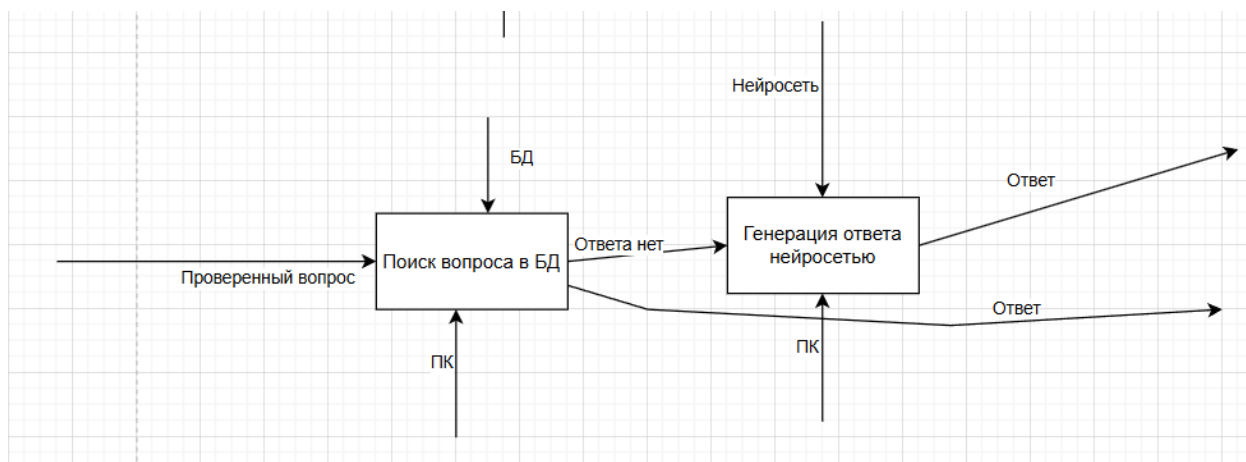


Рисунок 3 - Детализирующая диаграмма процесса «Генерация ответа»

Генерация ответа начинается поиска ответа в базе данных и ,если его там нет, запрашивает ответ у нейросети .

2.2 Проектирование логической модели базы данных

На этапе инфологического проектирования базы данных должна быть построена модель предметной области, не привязанная к конкретной СУБД, понятная не только разработчикам информационной системы, но и экономистам, менеджерам и другим специалистам. В то же время модель предметной области должна максимально точно отражать семантику предметной области, выявлять бизнес-правила и позволять легко перейти к модели данных конкретной СУБД.

Таковыми моделями являются модели «сущность-связь». Известно несколько методологий построения моделей «сущность-связь». Наибольшее распространение получила методология IDEF1X. Построение логической

модели «сущность-связь» осуществлялось средствами продукта ERwin Data Modeler.

Логическая модель данных относится к инфологическому этапу проектирования и не привязана к конкретной СУБД. Модели логического уровня оперируют с понятиями сущностей, атрибутов и связей, которые на этом уровне именуются на естественном языке (в нашем случае – на русском) так, как они называются в реальном мире [9].

Спроектированная логическая модель базы данных для информационной системы поддержки предприятий общественного питания представлена на рисунке 5. Данная модель содержит 26 сущностей, связанных между собой логическими связями.

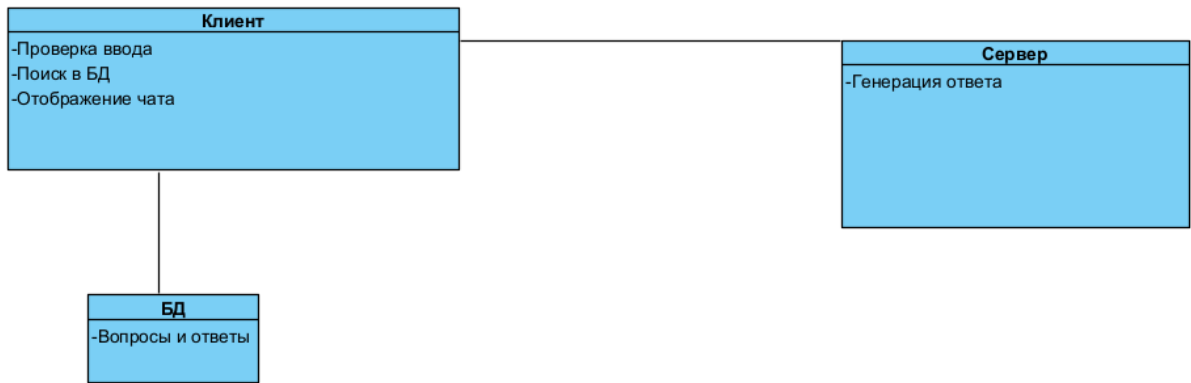


Рисунок 4 - Логическая модель данных

Описание сущностей и их атрибутов логической модели представлены в таблице 3.

Таблица 3 – Описание сущностей и атрибутов логической модели (доделать переносы таблицы)

Сущность	Атрибут	Описание атрибута
Клиент	Проверка ввода	Проверяет вопрос на корректность
	Поиск в БД	Ищет ответ в БД
	Отображение чата	Выводит вопрос и ответ
Сервер	Генерация ответа	Генерирует ответ

БД	Вопросы и ответы	Хранит вопросы и ответы к ним

После построения логической модели данных необходимо провести ее нормализацию, то есть осуществить организацию данных таким образом, чтобы обеспечить защиту данных и сделать БД более гибкой, без избыточности и несогласованности [15]. Цели нормализации заключаются в исключении дублирования информации в таблицах, обеспечении возможности изменений в структуре таблиц и уменьшении влияния структурных изменений БД на работу программы [16].

Так как сущности спроектированной логической модели базы данных содержат простые атрибуты, а также каждый не ключевой атрибут зависит от первичного ключа и не зависит от всех остальных атрибутов, то следует вывод, что данная модель является нормализованной и находится в третьей нормальной форме.

2.3 Проектирование программного обеспечения

Под проектированием программного обеспечения (ПО) понимается процесс создания проекта ПО. Целью проектирования является определение внутренних свойств системы и детализации её внешних (видимых) свойств на основе требований к ПО [17].

Процесс проектирования программного обеспечения заключается в построении диаграмм, отражающих структуру программы. Для их создания может быть использован язык UML – унифицированный язык моделирования, предназначенный для описания программных систем. Язык UML позволяет описать систему с различных точек зрения и разные аспекты поведения системы, а диаграммы UML являются сравнительно простыми для чтения [18].

Проектирование системы осуществлялось с использованием UML-диаграмм с помощью CASE-средств Visual Paradigm и ArgoUML.

2.3.1 Анализ прецедентов

Представление функциональности разрабатываемой системы осуществляется с помощью диаграммы прецедентов, которая является одним из ключевых инструментов моделирования в языке UML (Unified Modeling Language). Диаграмма прецедентов, также известная как диаграмма вариантов использования, предназначена для визуализации того, какой именно функционал доступен различным группам пользователей (акторам) системы. Она позволяет наглядно представить взаимодействие пользователей с системой, выявить основные сценарии использования и определить границы ответственности системы.

Прецедент — это конкретная возможность или функция моделируемой системы, которая предоставляет пользователю определённый результат, важный и полезный для его задач. Каждый прецедент соответствует отдельному сервису или функциональному блоку системы, описывает один из вариантов её использования и демонстрирует типичный способ взаимодействия пользователя с системой. Это позволяет не только структурировать требования, но и обеспечить их полноту и непротиворечивость. Варианты использования служат основой для спецификации внешних требований к системе, что особенно важно на этапах анализа и проектирования программного обеспечения.

Цель построения диаграммы прецедентов заключается в детальном описании функциональности разрабатываемой системы и её поведения с точки зрения конечных пользователей. Диаграмма помогает выявить все основные функции, которые система должна выполнять, а также определить, какие пользователи и каким образом будут с ними взаимодействовать. Это способствует более точному пониманию требований, облегчает коммуникацию между заказчиком и разработчиками, а также служит основой для последующего проектирования и реализации системы.

Диаграмма прецедентов обычно включает в себя следующих основных элементов: акторов — внешних пользователей или систем,

взаимодействующих с разрабатываемой системой; прецеденты — отдельные функции или варианты использования; а также связи между ними, отражающие взаимодействие и участие акторов в тех или иных функциях системы. Такой подход позволяет структурировать функциональность, выявить основные сценарии использования и обеспечить прозрачность архитектуры системы.

Построенная диаграмма вариантов использования, представленная на рисунке 6, демонстрирует основные функции чат-бота для взаимодействия с иностранными обучающимися и распределение этих функций между различными группами пользователей. На диаграмме четко прослеживаются ключевые прецеденты, такие как получение информации об учебном расписании, консультации по административным вопросам, поддержка в адаптации, а также сбор обратной связи и аналитика взаимодействий. Для каждой группы пользователей — будь то сами студенты, сотрудники учебного отдела или администраторы системы — определены соответствующие варианты использования, что обеспечивает ясное понимание ролей и возможностей в рамках функционирования чат-бота.

Таким образом, диаграмма прецедентов служит неотъемлемым инструментом в процессе разработки информационной системы, позволяя формализовать и структурировать требования, обеспечить прозрачность функционала и создать основу для дальнейшего детального проектирования.

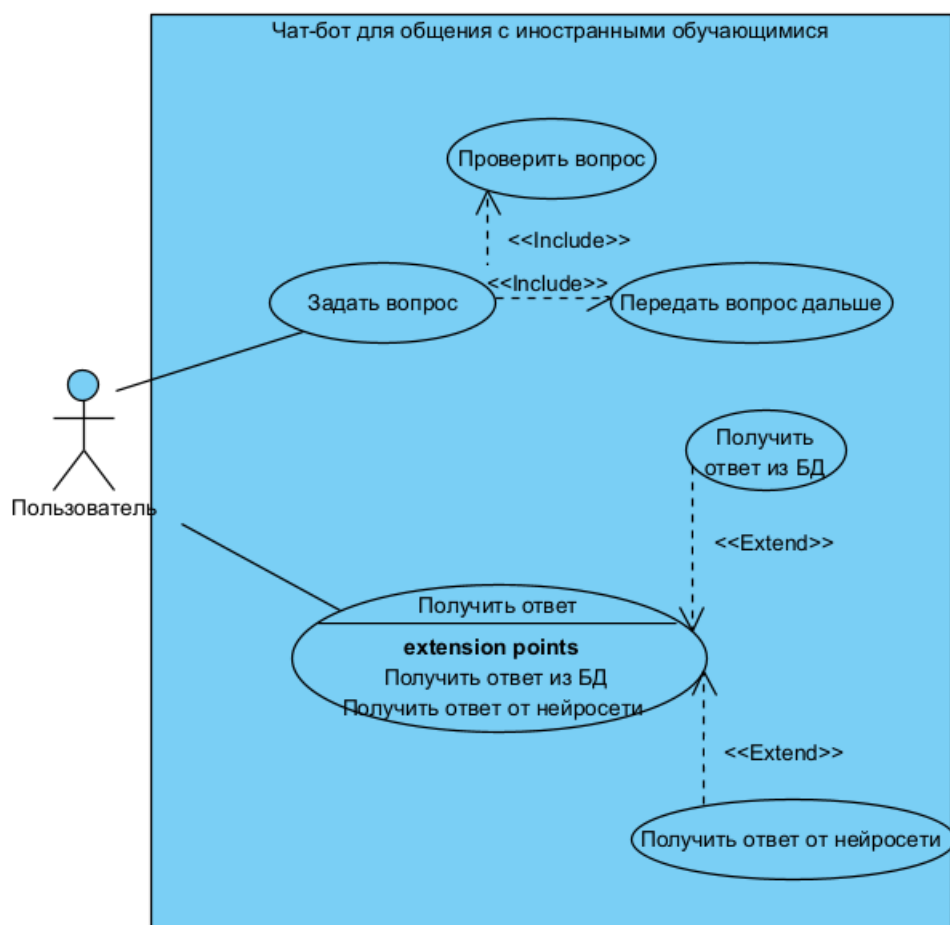


Рисунок 5 - Диаграмма вариантов использования

Описание актеров диаграммы вариантов использования представлено в таблице 4.

Таблица 4 – Описание действующих лиц

Название	Профиль, подготовка и навыки
Пользователь	Регулярный пользователь информационной системы .

Краткое описание прецедентов представлено в таблице 5.

Таблица 5 – Описание прецедентов

Действующее лицо	Цель	Краткое описание
------------------	------	------------------

Пользователь	Задать вопрос	Пользователь вводит вопрос и,если он корректный,ожидает ответа.
	Получить ответ	Программа ищет ответ в БД и,если находит его, выдает его пользователю. В противном случае ответ генерируется нейросетью.

По результатам проведенного анализа были определены и описаны прецеденты и действующие лица системы, а также построена диаграмма вариантов использования.

2.3.2 Диаграмма состояний

Диаграмма состояний предназначена для описания поведения отдельно взятого объекта и показывает, каким образом объект переходит из одного состояния в другое. Данная диаграмма полезна при моделировании жизненного цикла объекта [20].

Для отображения смены состояний объекта «Приходная накладная», была разработана диаграмма состояний, представленная на рисунке 7.

При разработке были выделены начальное и конечное состояния, а также промежуточные состояния и переходы между ними.

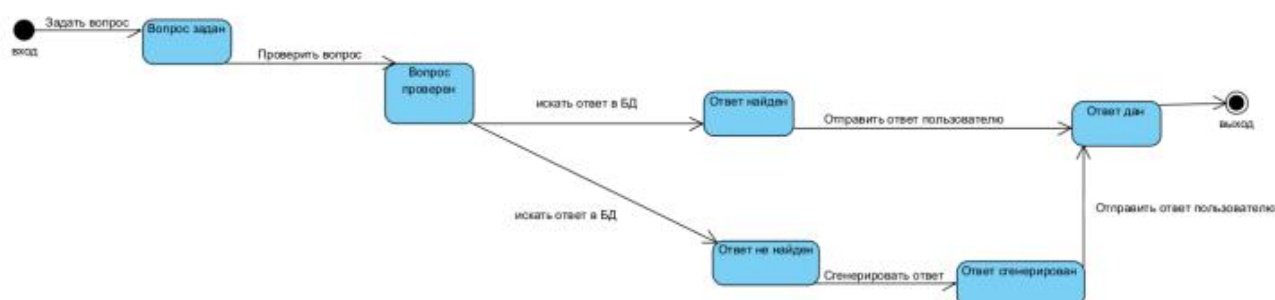


Рисунок 6 – Диаграмма состояний программы чат-бота

Спецификации переходов объекта из одного состояния в другое определены в таблице 6.

Таблица 6 – Спецификации переходов объекта из состояния в состояние

Состояние 1	Событие	Состояние 2	Описание перехода
Начальное состояние	Задать вопрос	Вопрос задан	Программа переходит из начального состояния в состояние «Вопрос задан»
Вопрос задан	Проверить вопрос	Вопрос проверен	При проверке вопроса программа переходит из состояния «Вопрос задан» в состояние «Вопрос проверен»
Вопрос проверен	Искать ответ в БД	Ответ найден	При поиске ответа в БД программа переходит из состояния «Вопрос проверен» в состояние «Ответ найден», если ответ найден
Вопрос проверен	Искать ответ в БД	Ответ не найден	При поиске ответа в БД программа переходит из состояния «Вопрос проверен» в состояние «Ответ не найден», если ответ не найден
Ответ не найден	Сгенерировать ответ	Ответ сгенерирован	Программа передает вопрос нейросети которая генерирует ответ и переходит из состояния «Ответ не найден» в состояние «Ответ сгенерирован»
Ответ найден	Отправить ответ пользователю	Ответ дан	При отправке ответа выводит ответ на экран и переходит из состояния «Ответ найден» в состояние «Ответ дан»
Ответ сгенерирован	Отправить ответ пользователю	Ответ дан	При отправке ответа выводит ответ на экран и переходит из состояния «Ответ сгенерирован» в состояние «Ответ дан»

Таким образом, была разработана диаграмма состояний, которая показала все возможные состояния объекта «Чат-бот для иностранных обучающихся».

2.3.3 Диаграмма деятельности

Диаграммы деятельности позволяют моделировать процесс как деятельность, которая состоит из коллекции соединенных ребрами узлов.

Диаграмма деятельности в нотации UML приведена на рисунке 8.

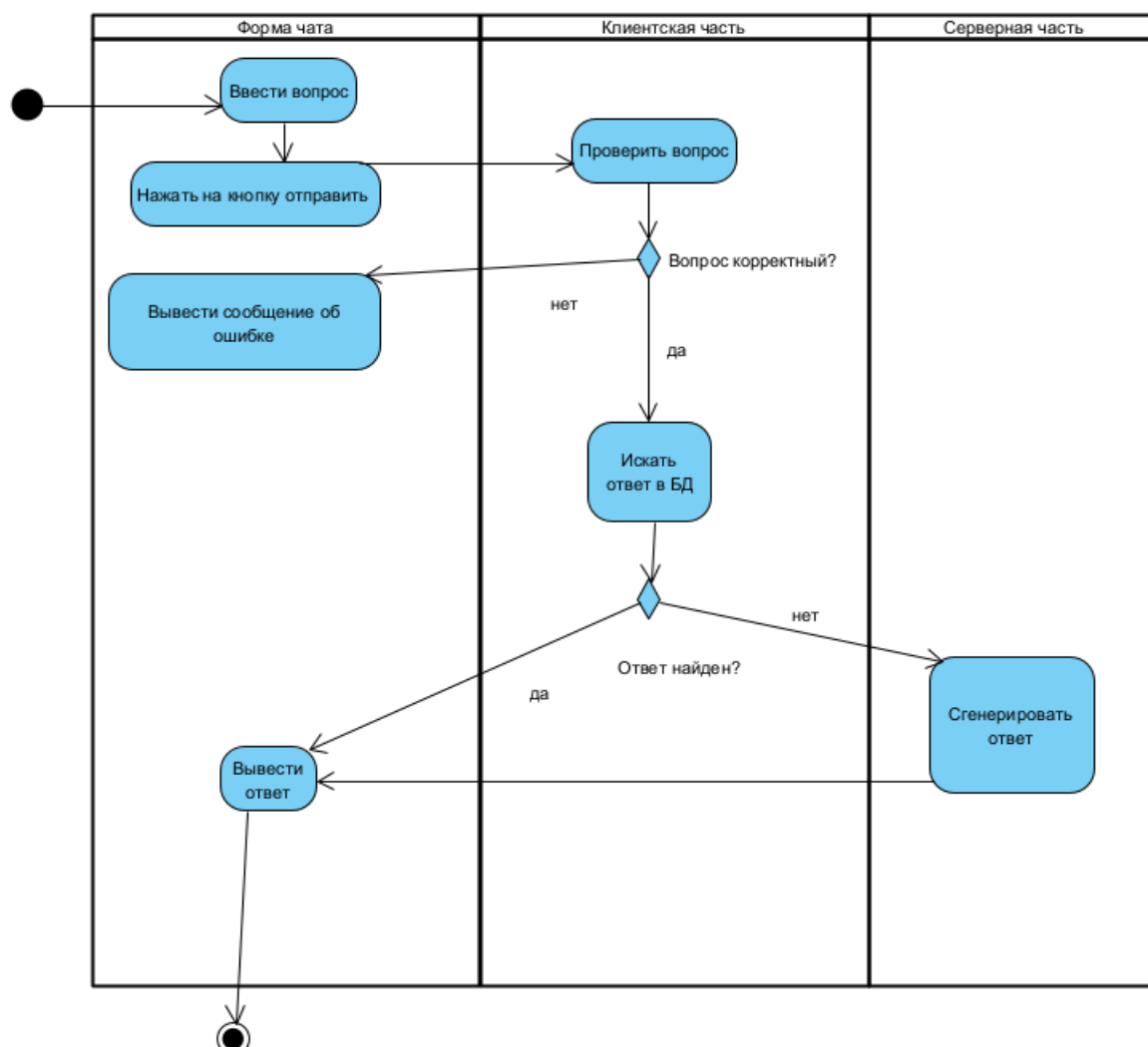


Рисунок 7 – Диаграмма деятельности

Для работы чат-бота для иностранных обучающихся необходимо написать в текстовое поле вопрос и нажать на кнопку «Отправить». После этого чат-бот проверит вопрос на корректность и, если все в порядке, выдаст ответ. Таким образом, была построена диаграмма деятельности в нотации UML.

3 Планирование и расчет бюджета разрабатываемой системы

Для упрощения управлением задачами и ресурсами проекта необходимо наличие системы управления проектом. Управление проектом представляет собой процесс планирования, организации и контроля над задачами и ресурсами проекта, который направлен на своевременное достижение цели проекта.

Организация проектирования ИС отличается значительной сложностью ввиду следующих причин:

- масштабности и длительности разработки ИС;
- взаимосвязи в рамках системы различных по своей природе объектов (информационные, программные, технические средства, математические модели, методы и средства проектирования и др.);
- различий в жизненном цикле элементов системы;
- индивидуальности проекта, обусловленной спецификой объекта проектирования;
- необходимости коллективного характера работы над проектом специалистов разной специализации и квалификации.

Таким образом, проектирование информационных систем предполагает использование проектного управления.

Диаграмма Ганта – это горизонтальная линейная диаграмма, на которой работы проекта представлены отрезками времени выполнения, характеризующимися сроками начала и окончания, вероятными задержками и другими временными параметрами [24].

Построение диаграммы Ганта для осуществления планирования и расчета бюджета разрабатываемой системы выполнялось в программе для управления проектами Microsoft Office Project.

В ходе планирования проекта было выделено 20 задач, описание которых представлено в таблице 7.

Таблица 7 – Задачи проекта

№	Название задачи	Длительность	Начало	Окончание	Предшественники
1	Начало проекта	0 дней	14.04.25	14.04.25	
2	Анализ предметной области и систем аналогов	5 дней	14.04.25	18.04.25	1
3	Постановка задачи	1 день	14.04.25	14.04.25	
4	Поиск и изучение систем аналогов	3 дней	15.04.25	17.04.25	3
5	Выбор технологий проектирования и разработки	1 день	18.04.25	18.04.25	4
6	Проектирование ИС	10 дней	19.04.25	28.04.25	2
7	Анализ требований	1 день	19.04.25	19.04.25	7
8	Проектирование логической модели БД	2 дней	21.04.25	22.04.25	8
9	Проектирование ПО	4 дней	23.04.25	28.04.25	9
10	Реализация ИС	16 дней	29.04.25	20.05.25	6
11	Кодирование	12 дней	29.04.25	14.05.25	
12	Реализация пользовательского интерфейса	4 дней	15.05.25	20.05.25	12
13	Тестирование ИС	4 дней	21.05.25	26.05.25	11
14	Тестирование метрикой Холстеда	2 дней	21.05.25	22.05.25	
15	Функциональное тестирование	2 дней	23.05.25	26.05.25	15
16	Подготовка документации	2 дней	27.05.25	28.05.25	14
17	Написание документации	1 день	27.05.25	27.05.25	
18	Оформление и печать документации	1 день	27.05.25	28.05.25	18
19	Конец проекта	0 дней	29.05.25	29.05.25	17

Также были определены необходимые ресурсы. Список ресурсов представлен в таблице 8.

Таблица 8 – Ресурсы проекта

Название ресурса	Тип	Стандартная ставка	Начисление
Аналитик	Трудовой	150,00 Р/ч	Пропорциональное
Проектировщик	Трудовой	130,00 Р/ч	Пропорциональное
Программист	Трудовой	170,00 Р/ч	Пропорциональное
Тестировщик	Трудовой	150,00 Р/ч	Пропорциональное
Компьютер	Материальный	15 000,00 Р	В начале

Принтер	Материальный	2 000,00 Р	В начале
Бумага	Материальный	300,00 Р	В начале
Интернет	Затраты		В начале
Электроэнергия	Затраты		По окончании

Выделенные задачи и ресурсы были занесены в созданный проект в Microsoft Office Project. Результаты представлены на рисунке 8.

		Название	Продолжит...	Начало	Окончание	Предшествующие	Название ресурса	Стоимость	% завершения
1		Начало проекта	0 дней	14.04.2025, 08:00	14.04.2025, 08:00		Интернет;Бумага;Принтер;...	18300,00 Р	0 %
2	✓	Анализ предметной области	5 дней	14.04.2025, 08:00	18.04.2025, 17:00	1	Аналитик	6000,00 Р	100 %
3	✓	Постановка задачи	1 день	14.04.2025, 08:00	14.04.2025, 17:00		Аналитик	1200,00 Р	100 %
4	✓	Поиск и изучение систем ана...	3 дней	15.04.2025, 08:00	17.04.2025, 17:00	3	Аналитик	3600,00 Р	100 %
5	✓	Выбор технологий проекта	1 день	18.04.2025, 08:00	18.04.2025, 17:00	4	Аналитик	1200,00 Р	100 %
6	✓	Проектирование ИС	7 дней	19.04.2025, 08:00	28.04.2025, 17:00	2	Проектировщик	7280,00 Р	100 %
7	✓	Анализ требований	1 день	19.04.2025, 08:00	19.04.2025, 17:00		Проектировщик	1040,00 Р	100 %
8	✓	Проектирование логической...	2 дней	21.04.2025, 08:00	22.04.2025, 17:00	7	Проектировщик	2080,00 Р	100 %
9	✓	Проектирование ПО	4 дней	23.04.2025, 08:00	28.04.2025, 17:00	8	Проектировщик	4160,00 Р	100 %
10	✓	Реализация ИС	16 дней	29.04.2025, 08:00	20.05.2025, 17:00	6	Программист	21760,00 Р	100 %
11	✓	Кодирование	12 дней	29.04.2025, 08:00	14.05.2025, 17:00		Программист	16320,00 Р	100 %
12	✓	Реализация пользовательск...	4 дней	15.05.2025, 08:00	20.05.2025, 17:00	11	Программист	5440,00 Р	100 %
13	✓	Тестирование ИС	4 дней	21.05.2025, 08:00	26.05.2025, 17:00	10	Тестировщик	4800,00 Р	100 %
14	✓	Тестирование метрикой Хо...	2 дней	21.05.2025, 08:00	22.05.2025, 17:00		Тестировщик	2400,00 Р	100 %
15	✓	Функциональное тестирова...	2 дней	23.05.2025, 08:00	26.05.2025, 17:00	14	Тестировщик	2400,00 Р	100 %
16	✓	Подготовка документации	2 дней	27.05.2025, 08:00	28.05.2025, 17:00	13	Аналитик	2400,00 Р	100 %
17	✓	Написание документации	1 день	27.05.2025, 08:00	27.05.2025, 17:00		Аналитик	1200,00 Р	100 %
18	✓	Оформление и печать доку...	1 день	28.05.2025, 08:00	28.05.2025, 17:00	17	Аналитик	1200,00 Р	100 %
19	✓	Окончание проекта	0 дней	29.05.2025, 08:00	29.05.2025, 08:00	16		0,00 Р	100 %

Рисунок 8 – Задачи проекта с распределенными по ним ресурсами в Microsoft Office Project

В соответствии с занесенными задачами и ресурсами была построена диаграмма Ганта, представленная на рисунке 9.

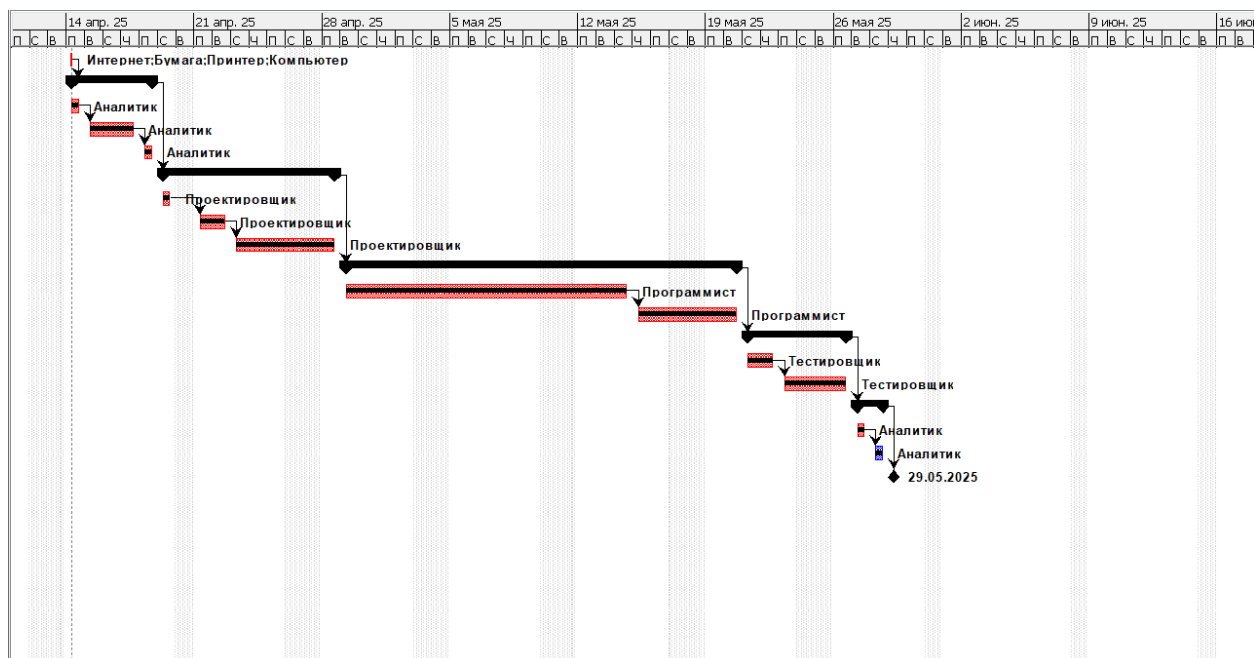


Рисунок 9 – Диаграмма Ганта

В соответствии с построенной диаграммой был сформирован отчет по затратам на разработку, приведенный на рисунке 10.

BKP

Dates			
Start	14.04.2025, 08:00	Finish	29.05.2025, 08:00
Baseline Start		Baseline Finish	
Actual Start	14.04.2025, 08:00	Actual Finish	29.05.2025, 08:00

Duration			
Scheduled	34 дней	Remaining	0 дней
Baseline	0 дней	Actual	34 дней
		Percent Complete	100 %

Work			
Scheduled	272 часов	Remaining	0 часов
Baseline	0 часов	Actual	272 часов

Costs			
Scheduled	42240,00 P	Remaining	0,00 P
Baseline	0,00 P	Actual	42240,00 P
		Variance	-42240,00 P

Рисунок 4 – Отчет по затратам проекта

Таким образом, по результатам построения диаграммы Ганта видно, что срок разработки проекта составил 34 дня включая выходные, бюджет проекта – 42240 рублей.

4 Реализация и тестирование системы

4.1 Реализация структур программного средства

Работа в среде Visual Studio начинается с создания объектов формы – средств, предназначенных для взаимодействия с пользователем. Для демонстрации компонентов разрабатываемой информационной системы была построена диаграмма компонентов в нотации UML.

Диаграммы компонентов используются для визуализации организации компонентов системы и зависимостей между ними. Они позволяют получить высокоуровневое представление о компонентах системы [25].

Разработанная диаграмма компонентов представлена в приложении А.

Все выделенные подкомпоненты распределены между 6 основными подсистемами. Подсистема «Главное» является специальной и содержит главный экран ИС, который отображается при запуске программы.

Главный экран ИС представлен на рисунке 11.



Рисунок 5 – Расположение подсистем и главный экран программы

Рассмотрим подробнее структуру и особенности основных объектов формы:

Текстбокс «textBoxQest» предназначен для ввода вопроса.

Листбокс «listBoxChat» предназначен для вывода ответа.

Кнопка «Отправить» для отправки вопроса.

4.2 Реализация кода информационной системы

Программный код пишется в различных модулях, которые взаимодействуют по REST API. Приведем пример программного кода серверной части приложения. Данная часть необходима для генерации ответа на вопрос. Код данной части представлен в листинге 1.

Листинг 1 – Программный код серверной части

```

from flask import Flask, request, jsonify
import nltk
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import torch

app = Flask(__name__)

# Загрузка модели и токенизатора GPT-2
model_name = 'gpt2'
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)
model.eval()

def generate_response(prompt):
    inputs = tokenizer.encode(prompt, return_tensors='pt')
    with torch.no_grad():
        outputs = model.generate(inputs, max_length=100,
                                num_return_sequences=1, no_repeat_ngram_size=2)
    response = tokenizer.decode(outputs[0],
                                skip_special_tokens=True)
    return response

@app.route('/ask', methods=['POST'])
def ask():
    data = request.get_json()
    user_input = data.get('question', '')
    response = generate_response(user_input)
    return jsonify({'response': response})

if __name__ == '__main__':
    app.run(port=5000)

```

Программный код клиентской части представлен в приложении Б.

4.3 Тестирование информационной системы

Тестирование программного обеспечения является одним из наиболее важных и эффективных средств определения и обеспечения качества создаваемых программных продуктов. Под тестированием программного обеспечения понимается комплексный процесс тщательного анализа, проверки и эксплуатации программных компонентов с целью выявления различных дефектов, ошибок, несоответствий и потенциальных уязвимостей. Этот процесс направлен на то, чтобы обеспечить максимально полное соответствие программного продукта предъявляемым к нему требованиям, а

также гарантировать его надежную и корректную работу в заданных условиях эксплуатации.

В ходе тестирования осуществляется наблюдение и контроль за функционированием программного обеспечения в специально организованных и контролируемых условиях, которые максимально приближены к реальным сценариям использования. Это позволяет определить степень соответствия программного продукта техническим, функциональным и эксплуатационным требованиям, установленным на этапе проектирования и разработки [3]. Кроме того, тестирование способствует выявлению скрытых дефектов, которые могут негативно повлиять на стабильность, производительность и безопасность системы при ее эксплуатации.

Тестирование включает в себя разнообразные методы и подходы, начиная от модульного тестирования отдельных компонентов и заканчивая комплексным системным тестированием, что позволяет обеспечить всестороннюю проверку программного обеспечения. Важной составляющей процесса является разработка и выполнение тестовых сценариев, которые отражают реальные пользовательские ситуации и бизнес-процессы. Благодаря этому можно выявить не только явные ошибки, но и скрытые проблемы, возникающие при взаимодействии различных модулей системы.

Таким образом, тестирование играет ключевую роль в цикле разработки программного обеспечения, обеспечивая выявление и устранение дефектов на ранних этапах, что значительно снижает риски возникновения сбоев и повышает качество конечного продукта. Регулярное и систематическое проведение тестирования способствует повышению удовлетворённости пользователей, снижению затрат на поддержку и развитие программного обеспечения, а также обеспечивает его соответствие высоким стандартам и требованиям современного рынка.

4.3.1 Функциональное тестирование

Одним из основных видов тестирования программного обеспечения является функциональное тестирование, которое играет ключевую роль в процессе обеспечения качества и надежности создаваемой системы. Главная задача функционального тестирования состоит в установлении соответствия между разработанным программным обеспечением и исходными функциональными требованиями, определёнными на этапе проектирования. Это позволяет убедиться в том, что все заявленные функции реализованы корректно и система способна выполнять необходимые пользователям задачи в соответствии с техническим заданием и бизнес-процессами.

Проведение функционального тестирования включает в себя комплекс мероприятий по проверке способности информационной системы работать в определённых условиях эксплуатации, а также обеспечивать выполнение конкретных операций и процедур, востребованных конечными пользователями. Такой подход позволяет выявить несоответствия, ошибки и недочёты на ранних этапах внедрения, что существенно снижает риски возникновения проблем в дальнейшем использовании системы [3].

В рамках данного проекта по разработке информационной системы поддержки предприятий общественного питания было организовано и проведено функциональное тестирование, результаты которого представлены в таблице 9. Основное внимание при тестировании уделялось проверке ключевых функциональных блоков системы, включая создание и проведение различных документов, формирование печатных форм и отчетов, а также корректную работу специализированных обработок и модулей автоматизации. Каждая из этих составляющих была протестирована на соответствие требованиям, надёжность выполнения операций и удобство взаимодействия с пользователем.

Особое значение имела проверка корректности формирования печатных форм и отчетов, поскольку эти функции напрямую влияют на качество документооборота и аналитическую поддержку принятия управленческих решений в сфере общественного питания. Тестирование включало в себя

проверку отображения данных, правильности расчётов, а также соответствия форм установленным стандартам и нормативам.

Результаты функционального тестирования были тщательно задокументированы и систематизированы, что позволило выявить как положительные стороны разработанной системы, так и области, требующие доработки и оптимизации. Подробные данные и выводы по итогам проведённого тестирования представлены в приложении В, где отражены конкретные сценарии тестирования, обнаруженные ошибки, а также рекомендации по их устранению

Таким образом, проведённое функциональное тестирование подтвердило работоспособность ключевых модулей информационной системы, обеспечило проверку соответствия её функционала заявленным требованиям и заложило основу для дальнейшего этапа внедрения и эксплуатации системы на предприятиях общественного питания. Такой комплексный подход к тестированию способствует повышению качества программного продукта и удовлетворённости конечных пользователей.

Таблица 9 – Функциональное тестирование работы системы

№ п/п	Название теста	Предполагаемый результат	Полученный результат	Результат тестирования
1	Некорректный вопрос	Программа выведет соответствующее сообщение	Соответствует ожидаемым результатам	Тест пройден (рисунок В.1)
2	Слишком короткий вопрос	Программа выведет соответствующее сообщение	Соответствует ожидаемым результатам	Тест пройден (рисунок В.2)
3	Слишком длинный вопрос	Программа выведет соответствующее сообщение	Соответствует ожидаемым результатам	Тест пройден (рисунок В.3)
4	Вопрос из БД	Программа выведет ответ из БД	Соответствует ожидаемым результатам	Тест пройден (рисунок В.4)
5	Вопрос для нейросети	Программа выведет сгенерированный ответ	Соответствует ожидаемым результатам	Тест пройден (рисунок В.5)

Таким образом, для разработанной системы было выполнено функциональное тестирование, по результатам которого можно сделать

вывод, что система функционирует корректно и не имеет явных ошибок и сбоев в работе.

4.3.2 Метрика Холстеда

Для получения численного значения какого-либо свойства ПО или его спецификации используются различные метрики программного обеспечения [34]. Одной из основных и наиболее популярных ввиду своей простоты метрик является метрика Холстеда. Суть данной метрики основана на анализе числа строк и синтаксических единиц кода программы.

Основу метрики Холстеда составляют четыре измеряемые характеристики программы:

- n_1 – количество уникальных операторов;
- n_2 – количество уникальных операндов;
- N_1 – общее число всех операторов, появляющихся в программе;
- N_2 – общее число всех операндов, появляющихся в программе.

Используя данные характеристики можно рассчитать следующие показатели:

- длина фрагмента программы, которая рассчитывается по формуле:

$$N = N_1 + N_2;$$

- словарь фрагмента программы, который рассчитывается по формуле:

$$n = n_1 + n_2;$$

- объем фрагмента программы, который рассчитывается по формуле:

$$V = N \times \log_2 n;$$

- сложность фрагмента программы, которая рассчитывается по формуле:

$$HDiff = (n_1 \div 2) \times (N_2 \div n_2).$$

Для серверной части программы был произведен расчет количественных показателей с использованием метрики Холстеда. Код процедуры приведен в листинге 2.

Листинг 2 – Код серверной процедуры

```
from flask import Flask, request, jsonify
import nltk
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import torch

app = Flask(__name__)

# Загрузка модели и токенизатора GPT-2
model_name = 'gpt2'
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)
model.eval()

def generate_response(prompt):
    inputs = tokenizer.encode(prompt, return_tensors='pt')
    with torch.no_grad():
        outputs = model.generate(inputs, max_length=200,
                                num_return_sequences=1, no_repeat_ngram_size=2)
        response = tokenizer.decode(outputs[0],
                                    skip_special_tokens=True)
    return response

@app.route('/ask', methods=['POST'])
def ask():
    data = request.get_json()
    user_input = data.get('question', '')
    response = generate_response(user_input)
    return jsonify({'response': response})

if __name__ == '__main__':
    app.run(port=5000)
```

Все операторы, используемые в серверной части, представлены в таблице 10.

Таблица 10 – Операторы процедуры

Название	Количество
=	7
.	15
()	19
def	2
with	1
return	2
:	2

[]	1
,	3
{}	1
$n_l = 11$	$N_l = 53$

Все операнды, используемые в серверной части, представлены в таблице 11.

Таблица 11 – Операнды процедуры

Название	Количество
model_name	3
'gpt2'	1
tokenizer	3
GPT2Tokenizer	2
from_pretrained	2
model	3
GPT2LMHeadModel	2
eval	1
generate_response	2
prompt	2
inputs	2
encode	1
return_tensors	1
'pt'	1
torch	1
no_grad	1
outputs	2
generate	1
max_length	1
200	1
num_return_sequences	1
1	1
no_repeat_ngram_size	1
2	1
response	2
decode	1
skip_special_tokens	1
ask	1
app	3
route	1
ask'	1
'POST'	1
data	2
get_json	1
user_input	2
get	1
'question'	1
' '	1

jsonify	1
$n_2 = 38$	$N_2 = 57$

После подсчета количества операторов и операндов были рассчитаны значения перечисленных выше показателей:

- длина фрагмента программы (N): 110;
- словарь фрагмента программы (n): 49;
- объем фрагмента программы (V): 505,3;
- сложность фрагмента программы ($HDiff$): 4.34.

Таким образом, был выполнен расчет количественных показателей серверной части с использованием метрики Холстеда.

Заключение

В процессе выполнения выпускной квалификационной работы была тщательно проанализирована актуальность разработки информационной системы, которая заключается в необходимости повышения конкурентоспособности образовательных учреждений и улучшения качества их деятельности. Одним из ключевых факторов достижения этих целей является внедрение современных цифровых технологий, в частности, разработка и использование чат-бота, предназначенного специально для поддержки иностранных обучающихся. Такой инструмент позволяет значительно облегчить процесс адаптации студентов из других стран, обеспечивая оперативную и качественную помощь в решении различных учебных и организационных вопросов.

По результатам проведения глубокого анализа существующих на рынке систем аналогичного назначения были выявлены их сильные и слабые стороны, а также определены основные функциональные и технические требования к разрабатываемому программному обеспечению. Этот этап позволил сформировать чёткое представление о том, каким должен быть будущий продукт, чтобы максимально эффективно удовлетворять потребности целевой аудитории и соответствовать современным стандартам качества.

В дальнейшем, в рамках выпускной квалификационной работы был осуществлён детальный анализ бизнес-процессов предметной области, связанной с поддержкой иностранных студентов в образовательном процессе. Для этого была построена функциональная модель с использованием нотации IDEF0, что позволило наглядно представить и структурировать основные процессы, определить их взаимосвязи и выделить ключевые функции системы. Такой подход обеспечил системное понимание предметной области и стал основой для последующего проектирования программного обеспечения.

Проектирование информационной системы проводилось с применением методологии UML, которая является одним из наиболее распространённых и эффективных средств моделирования программных систем. В ходе работы были построены различные диаграммы UML, включая диаграммы вариантов использования, классов, последовательностей и состояний, что позволило детально описать структуру, поведение и взаимодействие компонентов системы. Каждая диаграмма сопровождалась подробным описанием, обеспечивающим полное понимание архитектуры и функционала разрабатываемого программного продукта.

Кроме того, в рамках работы было выполнено планирование и расчет бюджета информационной системы поддержки предприятий общественного питания, что позволило оценить необходимые ресурсы, сроки реализации и финансовые затраты проекта. Такой комплексный подход к планированию обеспечил реалистичность и выполнимость поставленных задач, а также способствовал оптимальному распределению ресурсов.

В результате выполнения всех этапов разработки, с использованием средств платформы Visual Studio, был создан полнофункциональный чат-бот, предназначенный для иностранных студентов. Данный чат-бот обеспечивает эффективную поддержку и сопровождение обучающихся в процессе их учебной деятельности. Он предоставляет пользователям возможность получать оперативные ответы на часто задаваемые вопросы, связанные с расписанием занятий, доступом к учебным материалам, а также с административными процедурами и правилами учебного заведения. Кроме того, чат-бот фиксирует обращения пользователей, что позволяет проводить последующий анализ и выявлять наиболее актуальные проблемы и запросы студентов.

Особое внимание было уделено созданию системы аналитики, которая позволяет отслеживать и анализировать взаимодействие студентов с чат-ботом. Благодаря наличию различных аналитических отчетов,

образовательное учреждение получает ценную информацию о потребностях и поведении иностранных студентов, что способствует постоянному улучшению качества поддержки и адаптации обучающихся. Таким образом, разработанный чат-бот не только облегчает процесс обучения и интеграции иностранных студентов, но и становится важным инструментом повышения эффективности образовательного процесса в целом.

Список использованных источников

1. Кучина, Е.Д. Инструменты автоматизации предприятий общественного питания / Е.Д. Кучина, Н.Э. Бунина // Новая наука: опыт, традиции, инновации. Материалы международной научно–практической конференции. – Стерлитамак: РИО АМИ, 2015. – С. 116.
2. R-Keeper StoreHouse // Центр КТ [Электронный ресурс]. URL: https://shtrih-center.ru/po_oborud/r-keeper_storehouse.html (дата обращения 21.04.2025).
3. Горбаченко В. И. Проектирование информационных систем с СА ERwin Modeling Suite 7.3: учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154 с.
4. ArgoUML // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/ArgoUML> (дата обращения 16.05.2025).
5. Visual Paradigm // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Visual_Paradigm (дата обращения 16.05.2025).
6. Мазур, И. И. Управление проектами: учебное пособие / И. И. Мазур, В. Д. Шапиро, Н. Г. Ольдерогге. – М. : Издательство «Омега-Л», 2007. – 664 с.
7. Методология и нотация IDEF0 // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/IDEF0> (дата обращения 16.05.2025).
8. Описание основных приемов нормализации базы данных // Microsoft [Электронный ресурс]. URL: <https://docs.microsoft.com/ru-ru/office/troubleshoot/access/database-normalization-description> (дата обращения 18.05.2025).
9. Нормализация данных // StudFiles [Электронный ресурс]. URL: <https://studfile.net/preview/385775/> (дата обращения 18.05.2025).

10. Проектирование программного обеспечения // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Проектирование_программного_обеспечения (дата обращения 19.05.2025).
11. UML // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/UML> (дата обращения 19.05.2025).
12. Диаграмма прецедентов // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Диаграмма_прецедентов (дата обращения: 19.05.2025).
13. Диаграмма состояний // ИНТУИТ. Национальный открытый университет [Электронный ресурс]. URL: https://intuit.ru/studies/professional_skill_improvements/1364/courses/229/lecture/5954?page=4 (дата обращения: 19.05.2025).
14. Диаграмма последовательности // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Диаграмма_последовательности (дата обращения: 24.05.2025).
15. Диаграмма последовательности // ИНТУИТ. Национальный открытый университет [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/1007/229/lecture/5954?page=3> (дата обращения: 24.05.2025).
16. Диаграмма коммуникации // Картаслов.ру – карта слов и выражений русского языка [Электронный ресурс]. URL: <https://kartaslov.ru/карта-знаний/Диаграмма+коммуникации> (дата обращения: 24.05.2025).
17. Солонин, Е.Б. Управление проектами при разработке информационных систем / Е.Б. Солонин. – Екатеринбург: УрФУ, 2010. – 40 с.

18. Простое руководство по диаграммам компонентов // The Creately Blog [Электронный ресурс]. URL: <https://creately.com/blog/ru/uncategorized-ru/учебное-пособие-по-компонентной-диаг/> (дата обращения: 24.05.2025).

19. Основы тестирования и отладки приложений // ИНТУИТ. Национальный открытый университет [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/12786/1219/lecture/22487> (дата обращения: 13.06.2025).

20. Криспин, Лайза Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд / Лайза Криспин, Джанет Грегори; пер. с англ. – М.: ООО «И.Д. Вильямс», 2010. – 464 с.

21. Метрика программного обеспечения // Википедия. Свободная энциклопедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Метрика_программного_обеспечения (дата обращения: 13.06.2025).

22. Что такое Visual Studio? [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022> (дата обращения: 13.06.2025).

23. Добро пожаловать в интегрированную среду разработки Visual Studio [Электронный ресурс]. URL: <https://web.archive.org/web/20210110233636/https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2017> (дата обращения: 13.06.2025).

24. Знакомство с Visual Studio [Электронный ресурс]. URL: <https://csharp.webdelphi.ru/znakomstvo-s-visual-studio/> (дата обращения: 13.06.2025).

25. Выбор лучших: интегрированная среда для веб-разработки – Visual Studio [Электронный ресурс]. URL: <https://webformymself.com/sreda-dlya-veb-razrabotki-visual-studio/> (дата обращения: 13.06.2025).

26. Что такое чат-бот [Электронный ресурс]. URL: <https://sberbusiness.live/publications/что-такое-chat-bot> (дата обращения: 13.06.2025).

27. Что такое чат-бот: виды, способы создания [Электронный ресурс]. URL: <https://gb.ru/blog/что-такое-chat-bot/> (дата обращения: 13.06.2025).

28. Что такое Visual Studio? [Электронный ресурс]. URL: <https://learn.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022> (дата обращения: 13.06.2025).

29. Чат-бот [Электронный ресурс]. URL: <https://blog.skillfactory.ru/glossary/chat-bot-v-internete/> (дата обращения: 13.06.2025).

30. Всё о чат-ботах: виды, преимущества, успешные примеры [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/что-такое-chat-bot/> (дата обращения: 13.06.2025).

31. REST API: что это такое и как работает [Электронный ресурс]. URL: <https://practicum.yandex.ru/blog/что-такое-rest-api-i-kak-rabotaet/> (дата обращения: 13.06.2025).

Приложение А
ЛИСТИНГ ПРОГРАММЫ
(рекомендуемое)

Листинг Б.1 – Код клиентской части

```

using System;
using System.Data.SqlClient;
using System.Linq;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Newtonsoft.Json;

namespace Chat
{
    public partial class Form1 : Form
    {
        // Объявление HttpClient как статического поля класса
        private static readonly HttpClient client = new HttpClient();

        public Form1()
        {
            InitializeComponent();
        }

        private async void buttonSubmit_Click(object sender, EventArgs e)
        {
            string question = textBoxQest.Text?.Trim();

            if (string.IsNullOrEmpty(question))
            {
                MessageBox.Show("Please enter a question.", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            if (question.Length < 3)
            {
                MessageBox.Show("The question is too short.", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            if (question.Length > 500)
            {
                MessageBox.Show("The question is too long. Maximum 500
                characters.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            if (!IsValidQuestion(question))
            {
                MessageBox.Show("The question contains invalid characters.
                Only Latin letters, digits, spaces and basic punctuation are allowed.",
                "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }

            string answer;
            try
            {
                answer = GetAnswerFromDatabase(question);
            }
            catch (Exception)
            {

```

```

        MessageBox.Show("Error accessing the database.", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    if (answer == "Ответ не найден.")
    {
        try
        {
            answer = await GetAnswerFromChatGPT(question);
        }
        catch (Exception)
        {
            MessageBox.Show("Error contacting the chatbot.", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }
    }

    listBoxChat.Items.Add($"Question: {question}");
    listBoxChat.Items.Add($"Answer: {answer}");
    listBoxChat.Items.Add("");
}

// Метод для проверки допустимых символов в вопросе
private bool IsValidQuestion(string question)
{
    foreach (char c in question)
    {
        if (!(IsLatinLetter(c) || char.IsDigit(c) ||
        char.IsWhiteSpace(c) || ",.?!-()\"'\".Contains(c)))
        {
            return false;
        }
    }
    return true;
}

private bool IsLatinLetter(char c)
{
    return (c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z');
}

private string GetAnswerFromDatabase(string question)
{
    string answer = "Ответ не найден.";
    string connectionString = "Server=DESKTOP-
2F09F29\\SQLEXPRESS;Database=Chat;Integrated Security=True;";

    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        conn.Open();
        string sql = "SELECT answer FROM faq WHERE question =
@question;";
        using (SqlCommand cmd = new SqlCommand(sql, conn))
        {
            cmd.Parameters.AddWithValue("@question", question);
            using (SqlDataReader reader = cmd.ExecuteReader())
            {
                if (reader.Read())
                {
                    answer = reader["answer"].ToString();
                }
            }
        }
    }
}

```

```

    }
    }
    return answer;
}

private async Task<string> GetAnswerFromChatGPT(string question)
{
    string apiUrl = "http://localhost:5000/ask"; // URL вашего Flask
API
    var json = JsonConvert.SerializeObject(new { question = question
});
    var content = new StringContent(json, Encoding.UTF8,
"application/json");

    HttpResponseMessage response = await client.PostAsync(apiUrl,
content);
    if (response.IsSuccessStatusCode)
    {
        var jsonResponse = await
response.Content.ReadAsStringAsync();
        dynamic result = JsonConvert.DeserializeObject(jsonResponse);
        return result.response;
    }
    return "Не удалось получить ответ от чат-бота.";
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
}
}
}

```

Листинг Б.2 – Код серверной части

```

from flask import Flask, request, jsonify
import nltk
from transformers import GPT2LMHeadModel, GPT2Tokenizer
import torch

app = Flask(__name__)

# Загрузка модели и токенизатора GPT-2
model_name = 'gpt2'
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)
model.eval()

def generate_response(prompt):
    inputs = tokenizer.encode(prompt, return_tensors='pt')
    with torch.no_grad():
        outputs = model.generate(inputs, max_length=200,
num_return_sequences=1, no_repeat_ngram_size=2)
        response = tokenizer.decode(outputs[0], skip_special_tokens=True)
    return response

@app.route('/ask', methods=['POST'])
def ask():
    data = request.get_json()
    user_input = data.get('question', '')
    response = generate_response(user_input)
    return jsonify({'response': response})

if __name__ == '__main__':
    app.run(port=5000)

```

Приложение Б
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
(рекомендуемое)

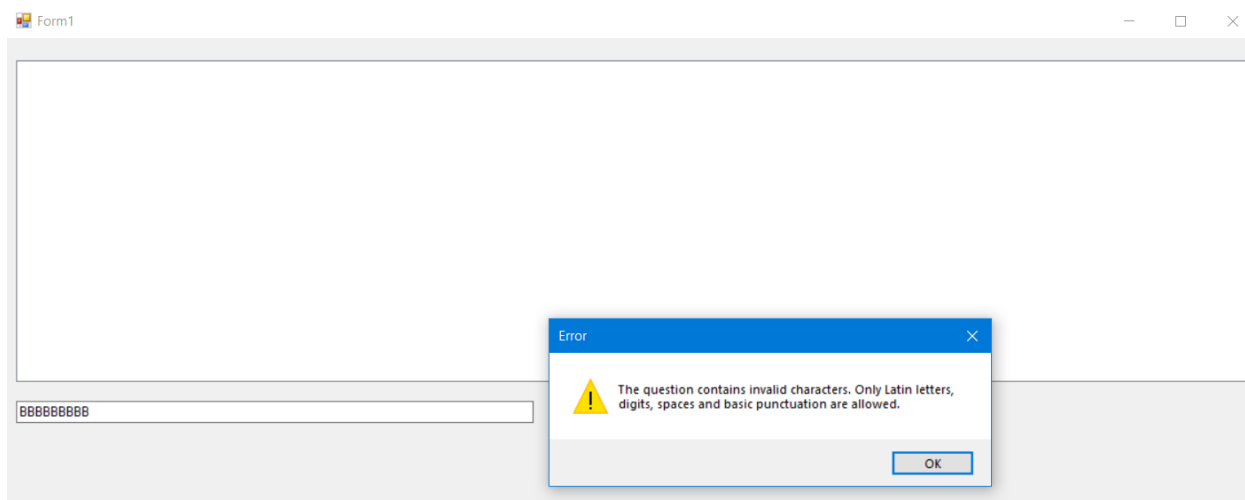


Рисунок В.1 - Некорректный вопрос

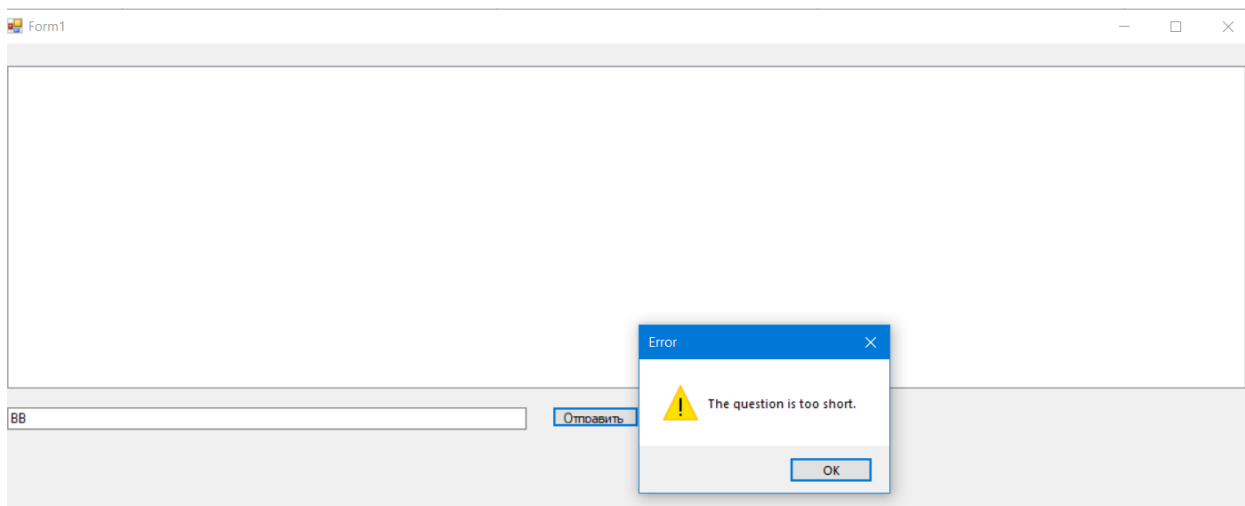


Рисунок В.2 - Слишком короткий вопрос

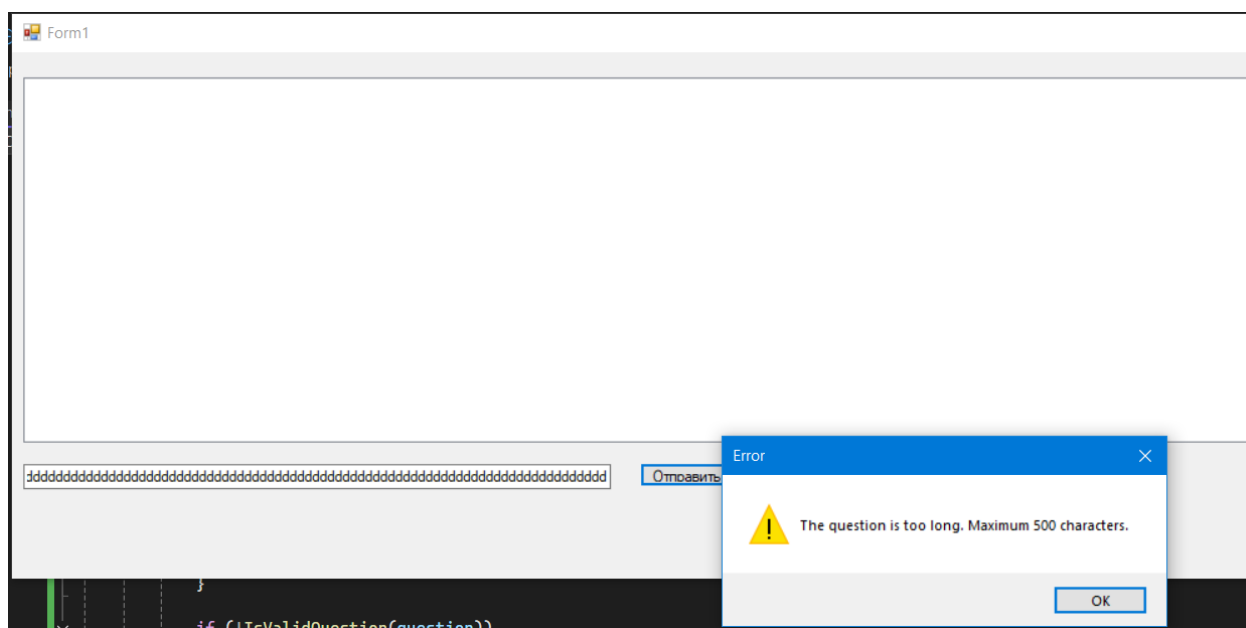


Рисунок В.3 - Слишком длинный вопрос

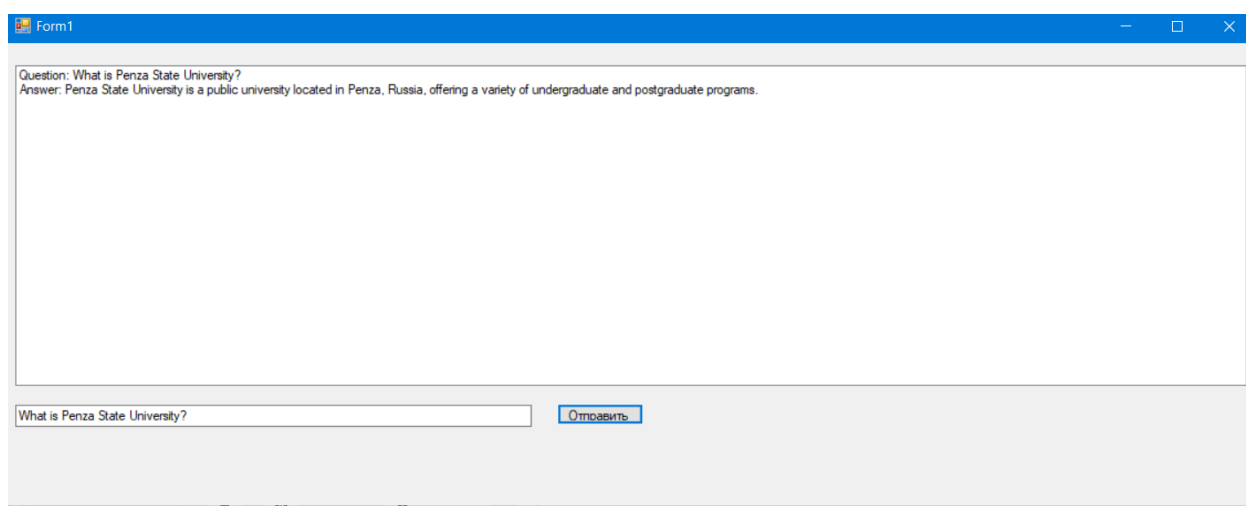
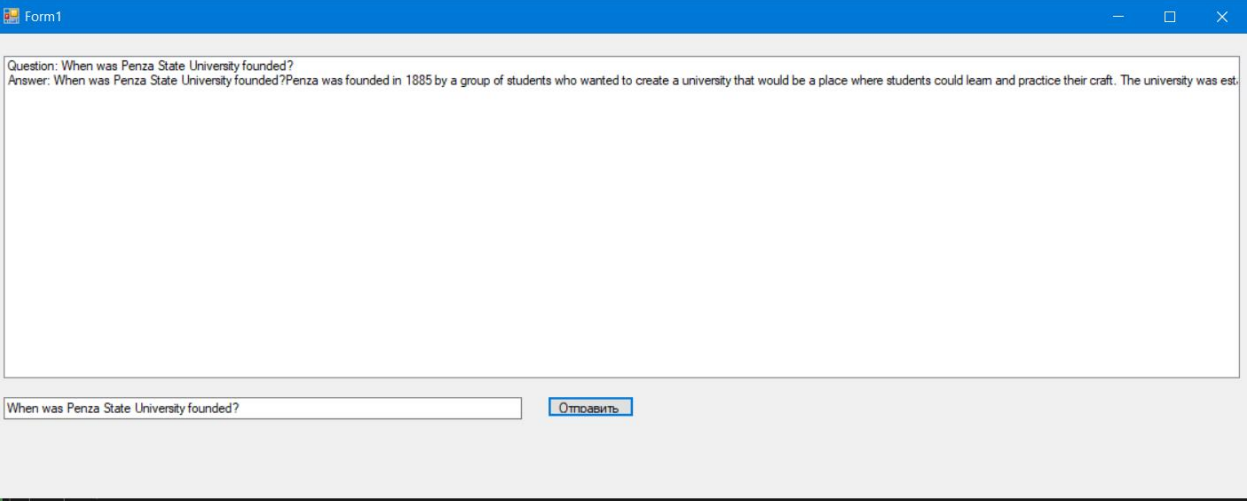


Рисунок В.4 - Вопрос из БД



The image shows a screenshot of a Windows application window titled "Form1". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area is white and contains the following text:

Question: When was Penza State University founded?
Answer: When was Penza State University founded?Penza was founded in 1885 by a group of students who wanted to create a university that would be a place where students could learn and practice their craft. The university was est.

Below the text area, there is a text input field containing the question "When was Penza State University founded?". To the right of the input field is a button labeled "Отправить" (Send).

Рисунок В.5 - Вопрос для нейросети