

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки)

Бизнес-информатика
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Автоматизация процесса управления заявками в техническую поддержку
образовательного учреждения

Обучающийся А.А. Бронникова
(Инициалы Фамилия) (личная подпись)

Руководитель к.п.н., доцент Е.А. Ерофеева
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема: «Автоматизация процесса управления заявками в техническую поддержку образовательного учреждения»

Выпускную квалификационную работу выполнила студентка Тольяттинского государственного университета, обучающаяся в институте математики, физики и информационных технологий, Бронникова Анастасия Александровна.

Целью данной работы является создание системы управления заявками в техническую поддержку образовательного учреждения.

Объектом исследования является процесс обработки и управления заявками в техническую поддержку, поступающими от сотрудников образовательного учреждения.

Предметом исследования является разработка программного обеспечения, которое позволит эффективно управлять заявками, автоматизировать их обработку, а также обеспечить удобный доступ к информации для сотрудников технической поддержки.

Дано описание учебного учреждения, выполнен анализ бизнес-процесса. Построена диаграмма «как есть» и «как должно быть». Произведен и обоснован выбор системы управления базами данных и языка программирования. Разработана система приема и обработки заявок в техническую поддержку. Также проведен анализ экономической эффективности данного проекта.

Оглавление

Введение.....	4
Глава 1 Анализ текущего процесса управления заявками в техническую поддержку мбоу сургутского естественно-научного лица	7
1.1 Характеристика деятельности мбоу сургутского естественно-научного лица	7
1.2 Концептуальное моделирование системы автоматизации процесса управления заявками в техническую поддержку	11
1.3 Анализ существующих программных решений для процесса управления заявками.....	20
Глава 2 Логическое проектирование автоматизированной системы управления заявками.....	23
2.1 Проектирование структуры и функционала автоматизированной системы	23
2.2 Разработка логической и физической модели данных и её описание	25
2.3 Подготовка к внедрению системы: настройка программного обеспечения, обучение персонала.....	30
Глава 3 Физическое проектирование автоматизированной системы приема и обработки заявок в техническую поддержку	33
3.1 Выбор средств реализации автоматизированной системы.....	33
3.2 Разработка и тестирование системы управления заявками в техническую поддержку.....	38
3.3 Оценка экономической эффективности системы управления заявками в техническую поддержку	49
Заключение	53
Список используемой литературы	54

Введение

В современном образовательном учреждении техническая поддержка играет важную роль в обеспечении бесперебойной работы информационных систем и технического оборудования. Школьники, преподаватели, учителя и студенты, сотрудники активно пользуются компьютерами, сетевыми ресурсами и другими техническими средствами для выполнения своих задач. Возникающие проблемы и неполадки в работе оборудования или программного обеспечения требуют оперативного решения, чтобы минимизировать простои и обеспечить продолжительность и эффективность образовательного процесса.

Разработка программного обеспечения для управления заявками в техническую поддержку образовательного учреждения является актуальной темой, так как она позволяет автоматизировать процесс управления заявками и улучшить качество предоставляемых услуг. В данной работе будут рассмотрены основные этапы разработки такого программного обеспечения, начиная от анализа существующих систем и заканчивая внедрением и эксплуатацией разработанного решения [1].

Объектом исследования в работе «Автоматизация процесса управления заявками в техническую поддержку образовательного учреждения» является процесс обработки и управления заявками в техническую поддержку, поступающими от сотрудников образовательного учреждения.

Предметом исследования является разработка программного обеспечения, которое позволит эффективно управлять заявками, автоматизировать их обработку, а также обеспечить удобный доступ к информации для сотрудников технической поддержки.

Цель: создание системы управления заявками в техническую поддержку образовательного учреждения

Задачи:

- Исследовать существующие системы управления заявками в образовательных учреждениях и провести анализ их преимуществ и недостатков.
- Выявить потребности пользователей (администраторов технической поддержки, пользователей) в системе управления заявками.
- Разработать техническое задание на разработку программного обеспечения для управления заявками в техническую поддержку образовательного учреждения.
- Спроектировать базу данных для хранения заявок, информации о пользователе и других необходимых данных.
- Написать программный код для реализации основных функций системы, таких как создание заявки, назначение исполнителя, отслеживание статуса заявки.
- Протестировать разработанное программное обеспечение на соответствие функциональным и техническим требованиям.

Глава 1. Анализ текущего процесса управления заявками в техническую поддержку МБОУ Сургутского естественно-научного лицея

1.1. Характеристика деятельности МБОУ Сургутского естественно-научного лицея

Муниципальное бюджетное общеобразовательное учреждение Сургутский естественно-научный лицей было основано 22 декабря 1974 года и носила название средняя школа №9. Позже школа была переименована в лицей №2. В 2014 году началась реорганизация образовательного учреждения. Благодаря объединению лицея №2 с Гимназией №4 было образовано уникальное учреждение, которое и по сей день носит название «Сургутский естественно-научный лицей».

Основная цель Сургутского естественнонаучного лицея – формирование общей личностной культуры учащихся на основе усвоения необходимого минимального содержания общеобразовательной программы, адаптации к общественной жизни, осознанного выбора и закладки основы овладения последующим профессиональным образовательными программами, а так же развитие таких качеств как гражданственность, трудолюбие, уважение прав и свобод человека, любовь к окружающей природе, отечеству, семье, формирование здорового образа жизни [2].

Целями Сургутского естественнонаучного лицея являются:

- Создание благоприятных условий для всестороннего развития личности, чтобы граждане Российской Федерации могли реализовать гарантированное государством право на бесплатное общее образование всех уровней, если гражданин получает этот уровень образования впервые;
- Формирование и развитие базовых компетенций, обучающихся;

- Построение учебной программы на основе психологической, медицинской и образовательной поддержки;
- Дифференциация и индивидуализация образовательного процесса как средство достижения эффективности обучения;
- Формирование устойчивой мотивации обучения;
- Формировать у учащихся потребность в самообразовании, самообразовании, самореализации и повышении их творческого потенциала;
- Поддерживать и укреплять физическое и психическое здоровье учащихся, воспитывать стремление к здоровому образу жизни
- Осуществлять раннюю профориентацию студентов и создавать им условия для адаптации к общественной жизни;
- Социализация личности студента, формирование культуры, основанной на общечеловеческих ценностях;
- Организация досуговой деятельности, способствующей развитию нравственных качеств личности.

Система управления образовательными учреждениями осуществляется в рамках действующего законодательства Российской Федерации и с учетом социально-экономических, материально-технических и внешних условий [13].

В его основе лежит пятиуровневая структура управления.

- Первый уровень структуры – уровень директора (по содержанию – уровень стратегического управления).
- На втором уровне структуры (по содержанию это также уровень стратегического управления). Кадровое собрание, Комитет по образованию, Комитет по управлению, Родительский комитет лица и Профсоюзный комитет.
- Третий уровень структуры управления – уровень заместителя директора. Этот уровень представлен Советом управления, Методическим советом и Аттестационным комитетом.

- Четвертый уровень структуры управления – структурное подразделение педагогического уровня, функциональной службы (уровня оперативного управления по содержанию) и лица.
- Пятый уровень организационной структуры – ученический уровень, представленный Советом средней школы.

На рисунке 1 представлена структура образовательного учреждения, являющегося объектом исследования [14]. Отдел технической находится под руководством заместителя директора по административно-хозяйственной работе, который в свою очередь находится под непосредственным руководством директора данного учреждения.

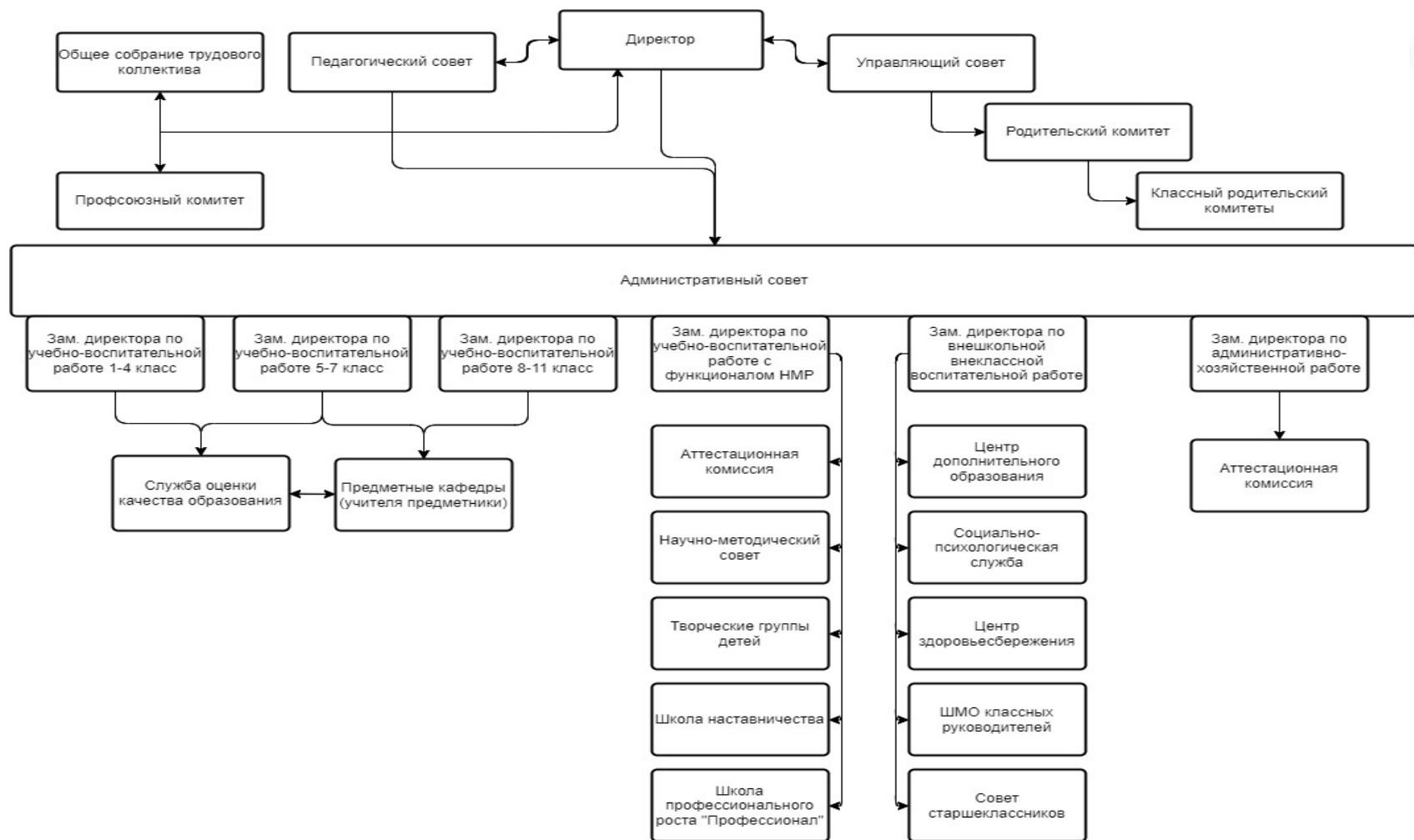


Рисунок 1 - Структура МБОУ Сургутского естественно-научного лицея

1.2. Концептуальное моделирование системы автоматизации процесса управления заявками в техническую поддержку

Описание текущей системы приема и обработки заявок, анализ проблем и недостатков. Существующая система формирования и обработки заявок на техническую помощь внутри образовательного учреждения существенно устарела и неэффективна. Когда сотрудники сталкиваются с техническими проблемами, традиционные протоколы требуют, чтобы они физически посещали определенные места и вручную регистрировали проблемы. Этот процесс включает внесение соответствующих данных, таких как имя, номер кабинета и подробное описание технической проблемы, в физический журнал. Этот журнал обычно находится на вахте лица. Техник, находящийся на рабочем месте, несет ответственность за регулярную проверку журнала, проверка выполняется примерно раз в несколько часов. Это означает, что между моментом сообщения о проблеме и моментом ее обнаружения техническим специалистом может пройти значительная задержка. В течение этих интервалов может накапливаться множество проблем, и их приоритетный статус может быть не сразу ясен, что приводит к потенциальной неэффективности и увеличению времени простоя.

В сценариях, где технические проблемы срочны и требуют немедленного внимания, процесс отличается от стандартных процедур регистрации [15]. Сотрудники, которым нужна срочная помощь, часто вызывают технического специалиста непосредственно со своего личного телефона. Эта прямая линия связи работает быстрее, но полностью обходит формальный процесс регистрации, в результате эти неотложные проблемы могут вообще не регистрироваться в центральном журнале, или технические специалисты могут попросить персонал задним числом внести данные в журнал после того, как проблема будет решена. Альтернативно, техник может вести личный журнал, записывая эти проблемы, но такая практика приводит к фрагментации записей и путанице. Из-за того, что большинство работников

прибегают к методу подачи заявок через смартфон, возникает несколько проблем: во-первых, это препятствует систематической регистрации заявок, затрудняя ведение полного и точного учета всех запросов и решений технической поддержки [15]. Отсутствие документации мешает не только отслеживать повторяющиеся проблемы, но и усложняет проверку работы техника вышестоящим начальством. Во-вторых, использование частных каналов связи, таких как личный телефон технического специалиста, может привести к несоответствию времени ответа и приоритетам [3].

В целом, существующая система сильно страдает от неэффективности. Такой подход не только задерживает решение технических проблем, но и препятствует способности вести последовательный и прозрачный учет всей деятельности IT-отдела [16]. Внедрение более эффективной автоматизированной системы управления запросами на техническую помощь значительно повысит оперативность, подотчетность и общую эффективность процесса технической помощи в любом образовательном учреждении.

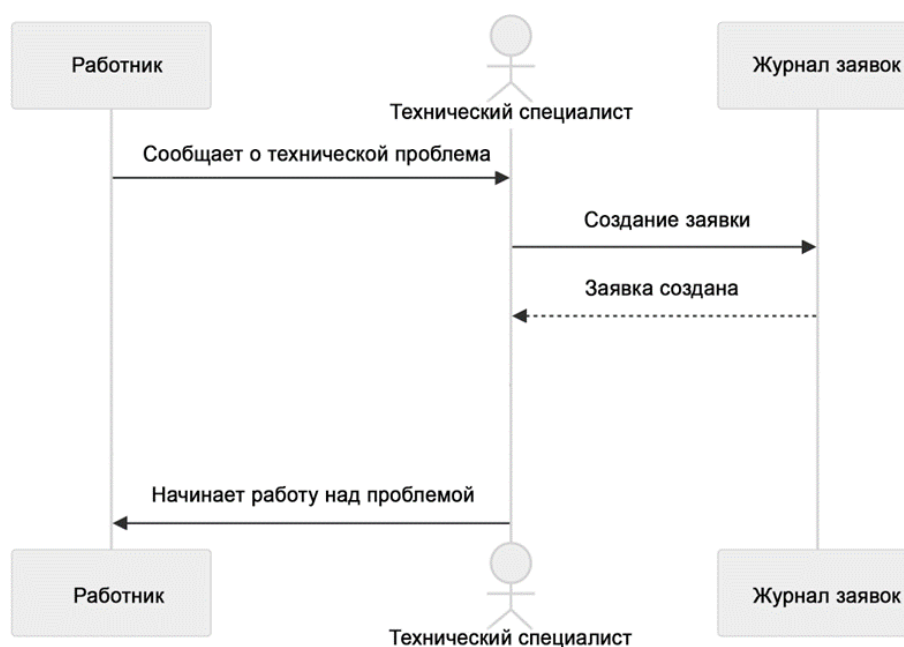


Рисунок 2 - Диаграмма последовательности текущей системы подачи заявок.

Разработка и анализ бизнес-процесса. Проведем классификацию бизнес-процессов внутри общеобразовательной организации. Операционным процессом будет выступать преподавание различных дисциплин и методическая организация. В рамках этого блока возможно выделить определенный тип процессов: управление преподавателями и обучающимися, контроль деятельности учителей и результатов учащихся, а также организацию учебного процесса [17].

Методы реинжиниринга разделяются на кризисный и реинжиниринг развития. Кроме того, существуют и уровни изменений: операционный, процессный и системный. Так как в нашем случае мы собираемся затронуть только один процесс, прием заявок в технический отдел, уровень изменений будет считаться «процессным».

При рассмотрении бизнес-модели в образовательном учреждении важно учитывать отраслевые особенности:

- Количество внутренних и внешних связей большое;
- Государство играет большую роль в стандартизации процессов;
- Муниципальная бюджетная общеобразовательная организация существует за счет государственных субсидий, грантов;
- Основной бизнес-процесс — это передача, накопление и создание знаний;
- Достаточно низкий уровень цифровизации.

Несмотря на приведенную выше специфику всё ещё возможно использовать стандартные методики и методологии для моделирования бизнес-процесса [18]. Как в IT язык – это различные языки программирования, так для моделирования бизнес-процессов язык — это нотации. На момент исследования наиболее популярными нотациями являются: IDEF0, EPC и BPMN. EPC для создания автоматизированной системы подачи заявок в техническую поддержку не совсем подходит, изначально данная нотация создавалась для высокоуровневых описаний бизнес-процессов. И не смотря на то что BPMN в основном создавалась для автоматизируемых бизнес-

процессов для реализации данного проекта было решено использовать нотацию IDEF0 как доступную, наглядную и исключающую возможности неправильной трактовки схемы [19]. Данная нотация обладает следующими преимуществами:

- Высокая степень детализации;
- Взаимодействия процессов показаны без лишних подробностей, в общем виде;
- Основной упор данной нотации на иерархическое отображение элементов.

Основным назначением IDEF0 является проектирование систем автоматизации, и данная нотация отлично подходит в том случае, когда преобразуются и физические и информационные потоки [20].

Рассмотрим возможные решения данного проекта на основе других исследований и практических задач. В большинстве случаев по причине того, что состав технического отдела состоит менее чем из 4 человек в государственных общеобразовательных учреждениях отсутствует система приема заявок [21]. Редко встречающийся способ оптимизации данного процесса является система на базе 1С, самостоятельно разработанная IT-отделом школы, либо простая система, состоящая из программы и сервера, аналогично разработанная техническим отделом самостоятельно и не распространяющаяся за пределами учреждения.

В качестве технической поддержки работников школы выступает техник и системный администратор. В зависимости от количества ставок на здание может быть, как достаточный штат IT-сотрудников (4-5 работника), так и небольшое количество работников (1-2 работника). В МБОУ Сургутском естественно-научном лицее на ставке техника работает один человек и обслуживает два здания (младший и старший корпус лицея). В связи с отсутствием автоматизированной системы заявки на выполнение работ приходят технику в нескольких мессенджерах одновременно, что затрудняет не только сортировку заявок по срочности выполнения, но и по корпусам.

Техническому специалисту приходится вести письменный журнал для фиксирования заявок. В качестве решения по оптимизации процесса директором учебного заведения было принято решения ввести автоматизированную информационную систему.

В данной выпускной квалификационной работе рассматривается бизнес-процесс формирования заявок в техническую поддержку учебного учреждения как объект автоматизации. Анализ этого бизнес-процесса предполагает тщательное изучение функционирования технической поддержки и других работников учреждения, определение основных выполняемых ими задач, а также выявление возникающих при этом трудностей [22]. Основная актуальность выбора данного бизнес-процесса заключается в его сложности и многозадачности, что обусловлено участием в нем большого числа работников.

Сложность данного процесса увеличивает вероятность технических неполадок, особенно учитывая количество технических устройств необходимых для организации учебного процесса. Такие неполадки требуют немедленного исправления для предотвращения ухудшения качества образовательного процесса, а также.

На рисунке 3 представлена главная контекстная диаграмма «Как есть» бизнес-процесса обработки заявок технического специалиста.

С учетом определенных требований к разрабатываемому программному обеспечению возможно разработать модель «Как должно быть» (Рисунок 4), улучшенную версию приведенной ранее диаграммы «Как есть», которая станет основополагающим элементом концепции разрабатываемой системы. Как можно увидеть по диаграмме при приеме заявок будет включена автоматизированная система [23]. Разрабатываемая система автоматически будет принимать заявки от сотрудников учреждения и направлять всю необходимую информацию техническим специалистам.

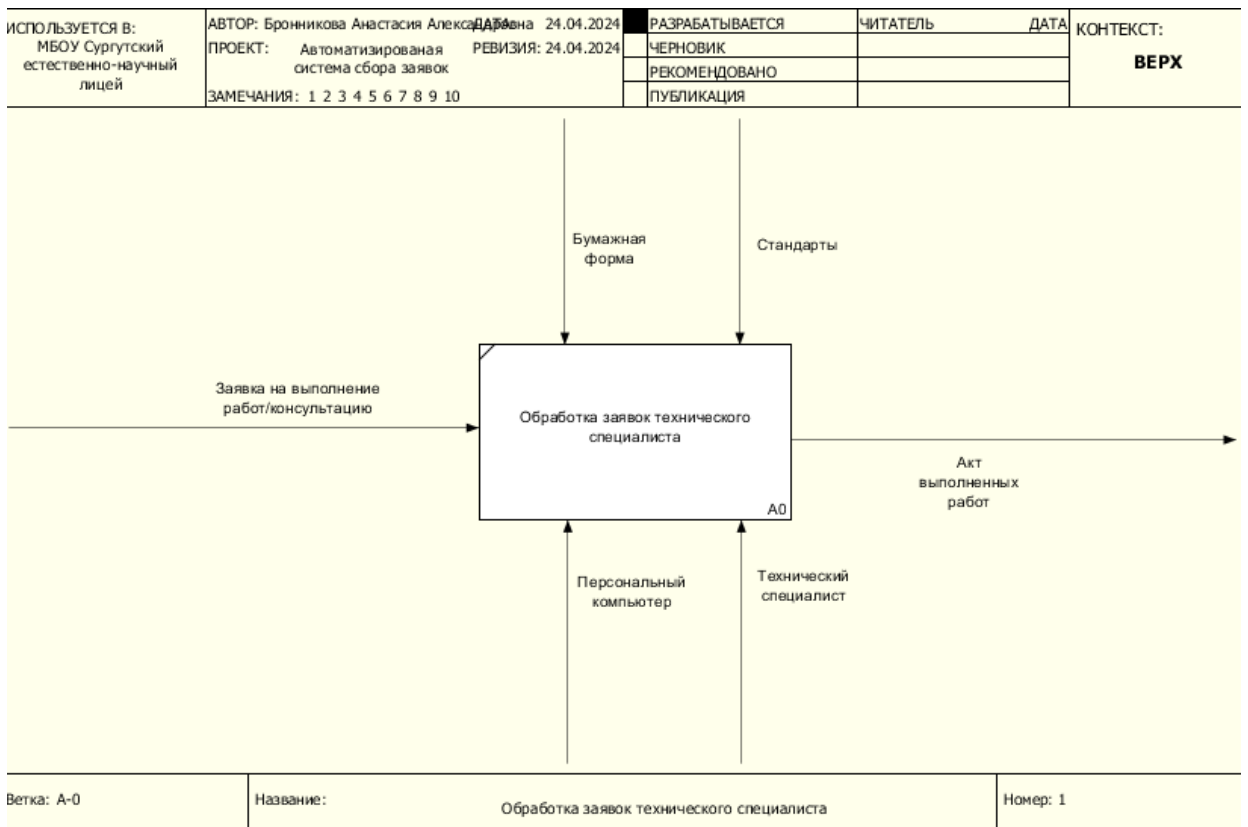


Рисунок 3 - Диаграмма «Как есть».

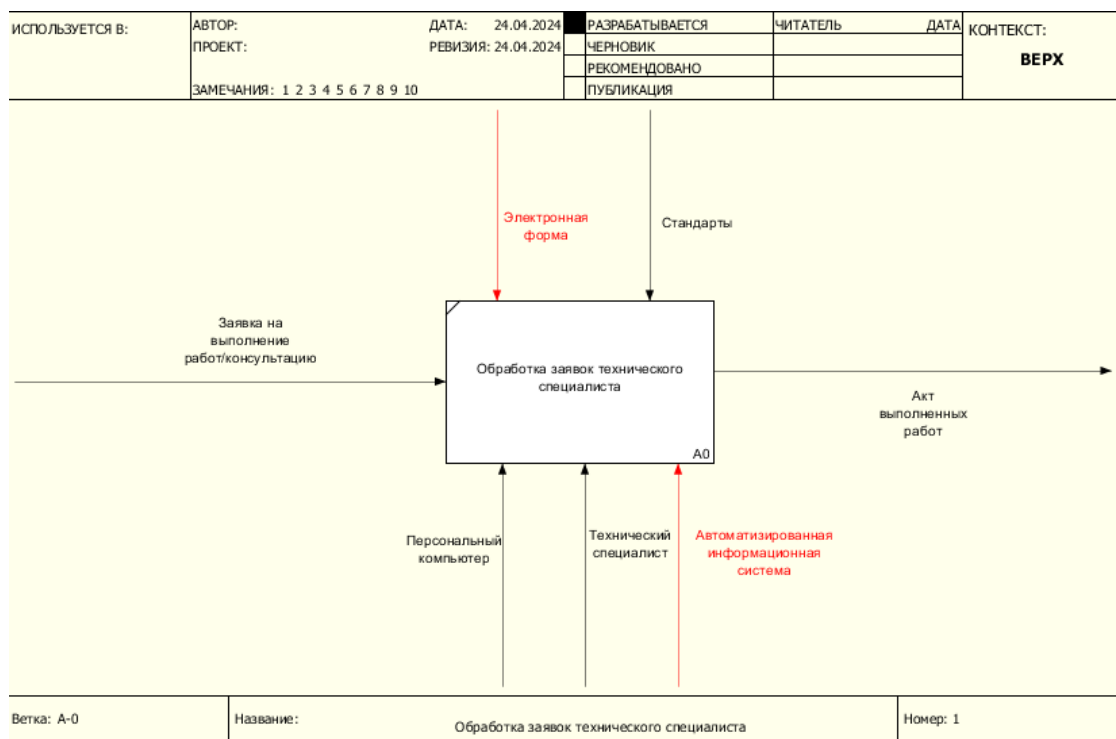


Рисунок 4 - «Как должно быть».

Определение требований к автоматизированной системе управления заявками. Главной бизнес-целью данного проекта является оптимизирование работы технического специалиста для более качественной и быстрой поддержки сотрудников общеобразовательного учреждения.

Немаловажным при написании любого проекта является определение и согласование всех требований к проекту. В среде IT для проектов такого плана используются функциональные требования [24]. Функциональные требования – это описание того, как должна работать система или же программный продукт, какие функции должны быть в неё включены, какие действия она должна выполнять или быть доступны для конечного пользователя. Требования необходимы для того, чтобы описать конкретное поведение системы в момент, когда пользователь вводит какие-то входные данные или производит действие.

Для того чтобы приступить к написанию функциональных требований, необходимо проанализировать все потребности пользователей, а также четко и ясно сформулировать цель проекта. После формирования цели можно приступить к освещению ключевых функций, которые необходимо реализовать для удовлетворения этих самых требований. В первую очередь необходимо решить, что должна делать система, а не как она будет реализована технически. Функциональные требования необходимо представить четкими и конкретными, не лишним будет написать их и измеримыми и, что немаловажно, достижимыми [25]. Они должны включать описание каждой функции, ее цели, предполагаемые входные данные, ожидаемые результатов и условия, при которых проект должен выполняться. Требования также могут включать описание интерфейсов, взаимодействия с другими системами или ограничения использования.

Так же очень важно, при написании функциональных требований, учитывать потребности всех заинтересованных сторон и обеспечивать соответствие требованиям бизнес-процессов. В ходе разработки проекта

функциональные требования могут быть изменены и дополнены на основе выявленных потребностей и отзывов пользователей.

С точки зрения функциональных возможностей программное обеспечение должно включать в себя реализацию следующих требований:

- Должен предоставить пользователям возможность следить за состоянием заявки.
- Должен фиксировать поступающие заявки путем присвоения уникального номера и ведения базы данных.
- Должен осуществлять автоматическую классификацию заявок по корпусу и срочности.
- Должен предоставлять возможность подать заявление как с мобильного устройства, так и с компьютера.

Составим функциональные требования для данного проекта:

- Система должна поддерживать удаленную подачу заявлений на выполнение работ/консультацию;
- Система должна автоматически формировать заявки по стандартам школы;
- Система должна формировать шаблон акта выполненных работ с возможностью комментирования исполнителем;
- Система должна автоматически разделять заявки по типу и срочности;
- Система должна хранить электронные акты выполненных работ не менее полугодом.

Пользовательские требования:

- Прием заявок производится не через социальные сети;
- Легкая в заполнение форма подачи заявки
- Автоматический ответ, на подачу заявки (присвоение уникального номера) и на закрытие заявки.

Теперь, когда мы смогли сформулировать функциональные требования, мы можем лучше понимать то, как должен выглядеть конечный продукт и какие функции он должен выполнять [26]. Это также формирует ожидания всех заинтересованных сторон и становится, практически, основой всего

процесса разработки. В целом, создание функциональных требований — это важный шаг, способный помочь структурировать процесс разработки, обеспечить понимание требований заказчика и улучшить качество конечного продукта [27]. В соответствии с этими требованиями теперь важно подобрать подходящее программное обеспечение, которое будет использоваться для реализации данного проекта, оптимизирующего процесс регистрации обращений в техподдержку.

Успешно сформулировав функциональные требования, мы имеем более четкое представление о том, как будет выглядеть и функционировать разрабатываемый проект. Эта ясность позволяет нам устанавливать точные ожидания для всех заинтересованных сторон и обеспечивает прочную основу для всего процесса разработки. Определив эти требования, мы можем более эффективно организовать разработку, глубже понять требования заказчика и улучшить общее качество конечного продукта. Формулирование функциональных требований является ключевым шагом в жизненном цикле разработки. Он не только организует и направляет процесс разработки, но и согласовывает усилия команды с ожиданиями клиентов [28]. Такое согласование гарантирует, что конечный продукт будет не только функциональным, но и ориентированным на пользователя, отвечающим реальным потребностям и решающим реальные проблемы пользователей.

Теперь, когда эти требования четко обозначены, следующим важным шагом является выбор подходящих программных инструментов и технологий, которые будут использоваться для реализации этого проекта. Этот процесс выбора важен для оптимизации различных аспектов проекта, включая эффективность, масштабируемость и ремонтпригодность.

1.3. Анализ существующих программных решений для процесса управления заявками

Перед созданием собственного продукта важно изучить существующие разработки, поскольку это позволяет понять текущее состояние рынка, выявить сильные и слабые стороны конкурентов, а также определить пробелы или неудовлетворенные потребности пользователей. Изучение существующих продуктов помогает сформировать ясное видение о том, какой уникальный вклад может внести ваш продукт, какие функции он должен иметь, и как он может быть успешным на рынке. Этот анализ также помогает избежать повторения ошибок конкурентов и использовать лучшие практики в своем проекте, что способствует его успешной реализации и конкурентоспособности [12].

Проанализируем существующие разработки для решения обозначенной задачи. Сравнительную характеристику представим в виде таблицы (Таблица 1) с учетом определенных критериев анализа. Исходя из составленных функциональных требований, мы можем выделить критерии для оценивания готовых программных продуктов. Так как в нашем образовательном учреждении большая часть сотрудников относится к пользователям новичкам была бы нежелательно регистрация на отдельном сайте. Автоматическое распределение заявок – основополагающее требование к системе, так же, как и работа с мобильными устройствами, ведь в случае выхода из строя у учителя персонального компьютера у него должна быть возможность подать заявку другим способом. И гибкая настройка приложения необходима для дальнейшего развития проекта, в связи с тем, что технический специалист довольно часто сменяется должна быть возможность настроить приложения так как это будет необходимо новому сотруднику или при смене требований директора [4].

Исходя из полученных результатов можно сформировать следующие выводы:

- Практически все системы обеспечивают возможность работы с мобильного устройства, что означает в реализуемой системе, данное требование должно соблюдаться.
- Многие системы ориентированы на использования без регистрации. Необязательная регистрация повысит удобство пользователя.

Таблица 1 - Существующие разработки.

Наименование аналогичного решения	Требуется регистрация пользователя	Автоматическое распределение заявок	Работа с мобильного устройства	Гибкая настройка приложения
Класс365	Да	Нет	Сайт	Нет
Мультичат	Нет	Нет	Да	Да
Окдеск	Нет	Да	Да	Да
ITSM 365	Нет	Нет	Да	Нет
Alloy Navigator 2021	Да	Нет	Нет	Да
Freshworks	Да	Да	Да	Нет

Окдеск доступная и удобная система для автоматизации технической поддержки. Несмотря на то, что система подходит под большинство параметров оптимизации обладает одним главным минусом. Система работает по принципу «подписки» на что школа не может выделить средства из бюджета.

На рынке присутствует еще достаточное множество различных программ автоматизации технической поддержки, но не один из продуктов не устроил заказчика. Большинство готовых решений предназначены не для сбора заявок в техническую поддержку, а для формирования заказа, что в корне отличается от необходимого функционала.

Создание своего программного продукта при реализации данного проекта будет более предпочтительным вариантом, чем готовый продукт. В первую очередь готовый продукт не будет соответствовать всем необходимым потребностям. При разработке собственного программного продукта можно точно адаптировать его под уникальные потребности и требования. Это означает, что мы можем включить только те функции, которые действительно необходимы, избежав излишних возможностей или ограничений готовых программ [29]. Так как собственный продукт будет разработан на базе бесплатных сервисов (MySQL и Telegram) возможны гибкие расширения и адаптации в будущем. Возможно с легкостью добавлять временные функции (например, на период ОГЭ) или интегрировать программный продукт с другими системами, если это будет необходимо.

В долгосрочной перспективе также можно выделить и экономическую перспективу и, хотя создание собственного программного продукта может потребовать больших начальных инвестиций времени и ресурсов, в долгосрочной перспективе это может быть экономически эффективным, поскольку мы будем иметь полный контроль над расходами и не зависеть от сторонних поставщиков программного обеспечения.

Однако стоит учитывать, что создание собственного программного продукта также связано с определенными рисками, такими как возможные задержки в разработке, нехватка опыта или ресурсов, а также необходимость обеспечения поддержки и обновлений продукта в будущем самостоятельно.

Вывод по главе. В процессе работы была построена диаграмма «Как есть» и «Как должно быть», рассмотрена текущий способ приема заявок в образовательном учреждении, а также проанализированы существующие программные продукты, которые было бы возможно использовать в качестве готового решения для данного проекта.

Глава 2 Логическое проектирование автоматизированной системы управления заявками

2.1. Проектирование структуры и функционала автоматизированной системы

Выявив требования к проекту, мы можем приступить к проектированию структуры и функционала нашей системы [5].

Структура проекта:

- Telegram Bot API: интерфейс, который позволяет чат-боту общаться с пользователями Telegram. Он будет обрабатывать отправку и получение сообщений.
- Внутренний сервер: это ядро приложения. Он обрабатывает запросы от чат-бота, подключается к базе данных MySQL и выполняет необходимую логику для обработки запросов в техническую поддержку.
- База данных MySQL: реляционная база данных, в которой хранятся все данные, относящиеся к пользователям, запросам на поддержку и персоналу службы поддержки. У него будет несколько таблиц для эффективной организации этих данных.
- Webhook: механизм получения обновлений в реальном времени от Telegram. Когда пользователь взаимодействует с чат-ботом, Telegram отправляет обновление на Webhook, которое обрабатывает внутренний сервер.

Распишем основной функционал системы:

- Регистрация и аутентификация пользователя:

Новые пользователи должны зарегистрироваться в системе не только для безопасности и быстрого поиска по сотрудникам учреждения, но и для того, чтобы пользователь мог быстро подавать заявку без ввода повторяющейся информации. Регистрация производится через отправку имени и фамилии, а также номера кабинета и корпуса.

– Создание запроса на поддержку:

Пользователи могут создать новый запрос на техническую поддержку, выбрав срочность выполнения заявки и предоставив описание проблемы. Бот подтверждает создание запроса и присваивает уникальный идентификатор запроса.

– Отслеживание и обновление запросов:

Сотрудники могут проверить статус своих запросов в поддержку в любое время. Дополнительно возможно дополнить свою заявку. В этом случае бот оставит идентификационный номер и только обновит информацию самой заявки.

– Система оповещения:

Пользователи получают уведомления, когда статус их запроса на поддержку меняется (например, с «открыто» на «в обработке» и на «закрыто»).

Сотрудники службы поддержки получают уведомления, когда им назначается новый запрос.

– Обработка команд:

Общие команды, такие как /start для инициирования взаимодействия, /register для регистрации, /newrequest для создания запроса, /status <request_id> для проверки статуса запроса и /help для вывода списка доступных команд.

Таким образом, данная работа предполагает создание чат-бота Telegram, который взаимодействует с пользователями для получения и обработки заявок на техническую поддержку [30]. Внутренний сервер обрабатывает эти взаимодействия, сохраняет данные в базе данных MySQL и обеспечивает бесперебойную связь между пользователями и службой технической поддержки. Предоставляемые функции включают регистрацию пользователей, создание и отслеживание запросов на поддержку и систему уведомлений, позволяющую информировать все стороны о статусе запросов на поддержку.

2.2. Разработка логической и физической модели данных и её описание

Переходя к разработке моделей данных и их описаний, начнем работу с создания концептуальной модели. Концептуальная модель — это абстрактное представление или структура, которая отражает фундаментальные аспекты системы, процесса или явления. Цель составления данных моделей состоит в том, чтобы упростить сложные концепции и сделать их более понятными.

При разработке программного обеспечения концептуальные модели помогают разработчикам понять требования и конструкцию программной системы перед ее созданием и внедрением. Модели и схемы служат основой для разработки и помогают обеспечить визуальное представление архитектуры системы, компонентов и их взаимодействий. В бизнес-анализе концептуальные модели помогают заинтересованным сторонам визуализировать и передавать сложные бизнес-процессы, организационные структуры и взаимосвязи данных. Они облегчают принятие решений, решение проблем и стратегическое планирование, обеспечивая четкое понимание того, как связаны между собой различные элементы бизнеса.

Для построения концептуальной модели данного проекта воспользуемся ER-моделью. (Рисунок 5).

Причины использования ER-модели:

- Представляет собой графическую диаграмму, которая легко читается и понимается.
- Позволяет ясно определить сущности (таблицы) и их отношения (связи), что упрощает процесс проектирования базы данных.
- Позволяет визуализировать связи между сущностями, что помогает легче воспринимать структуру данных и их взаимосвязи.
- Поддерживает нормализацию данных, что поможет устранить избыточность и повысить эффективность работы с базой данных.
- Позволяет легко создать физическую структуру базы данных, так как она показывает ясное представление о таблицах, их атрибутах и связях.

- Позволяет легко воспринимать и анализировать данные

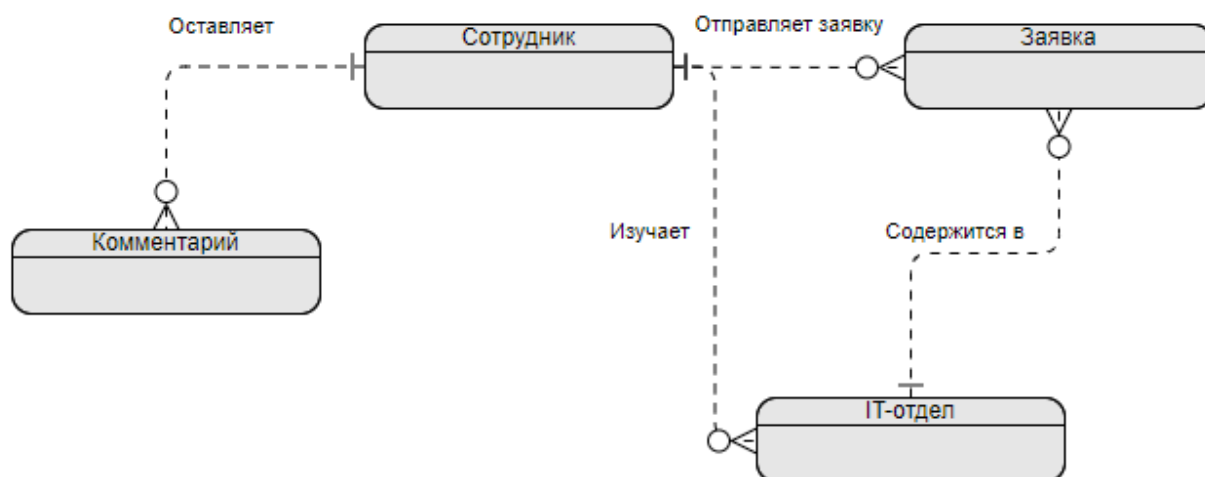


Рисунок 5 - ER-модель

Переходя к следующему этапу, разработки программного обеспечения необходимо составить прототип базы данных [6]. Создание прототипа базы данных для нашего продукта имеет несколько целей:

- Прототип базы данных обеспечит помощь разработчику лучше понять требования к системе. Прототип предоставит визуализированную структуру данных и отношения между ними (данными), что позволит уточнить и доработать требования.
- С помощью прототипа возможно проверить, насколько хорошо задуманная концепция работает на практике. Мы сможем быстро создать и изменить структуру данных, чтобы выбрать оптимальное решение.
- Благодаря прототипу мы сможем провести тестирование и отладку перед полным запуском проекта. Это обеспечит выявление и исправление разработчиком ошибок и проблем, которые не были замечены ранее.
- По своей сути прототип является эффективным инструментом для коммуникации между разработчиком и заказчиком. Он позволяет визуализировать и обсудить структуру данных.

- Прототип базы данных может также помочь в проведении экспериментов с различными методами хранения и организации данных, для определения наиболее эффективного способа работы с базой данных. Это оптимизирует производительность системы и улучшить ее работу.

В качестве прототипу будущей базы данных для программы приёма заявок в техническую поддержку представим структуру данных с полной атрибутикой. (Рисунок 6)

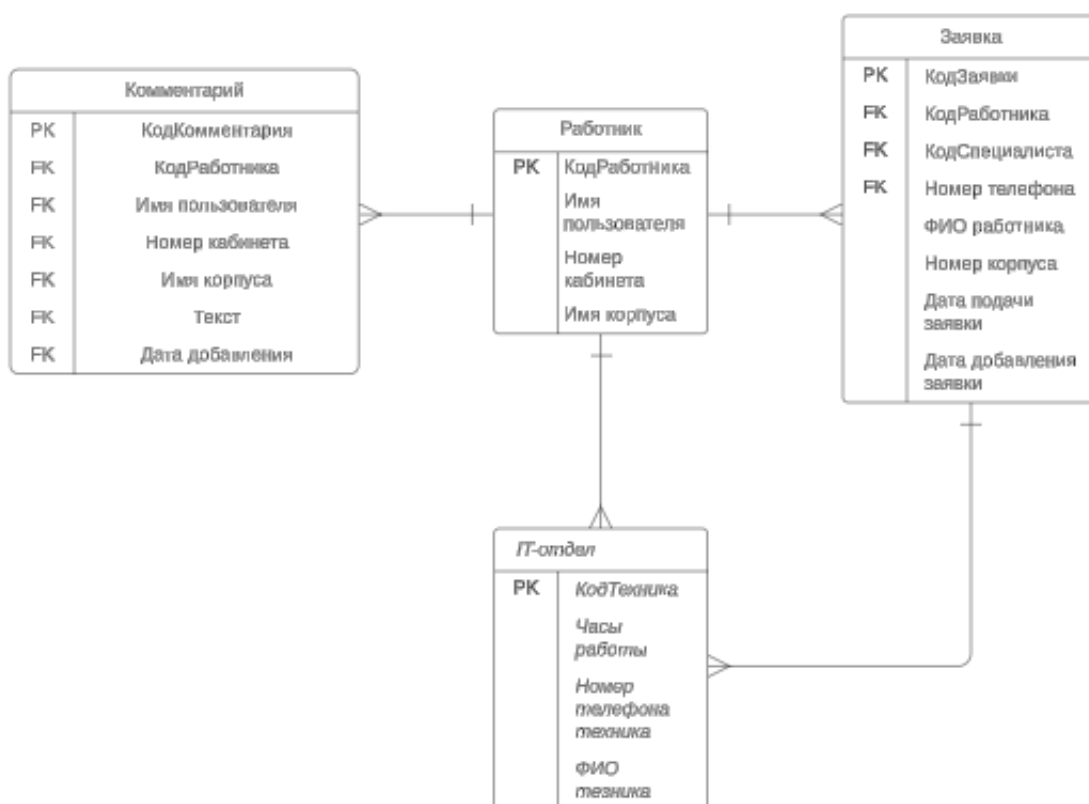


Рисунок 6 - FA-диаграмма.

База данных нашего проекта будет состоять из четырех таблиц. Таблицы «Работник», куда будут записываться данные с регистрации пользователя, таблицы «IT-отдел», в которой содержится состав технической поддержки образовательного учреждения со всей необходимой информацией, которую бот сможет предоставить пользователю, таблицы «Заявки», куда будет записываться вся самая нужная информация, которую в последующем можно

будет выгрузить для быстрого оформления акта об выполненной работе и таблицы «Комментарий», которая необходима либо для дополнения заявки, либо для указания специфичной информации от пользователя. Определим тип данных для физической модели. При создании базы данных желательно отсутствие связи «многие ко многим». Причина этому плохая масштабируемость и сложность. Рассмотрим данный случай поподробнее.

Отношения «многие-ко-многим» в базе данных — это тип отношений, при котором несколько записей из одной таблицы могут быть связаны с несколькими записями из другой таблицы. Например, рассмотрим связь между «Работник» и «Заявка» в контексте системы технической поддержки образовательного учреждения.

- Работник: один пользователь может отправить несколько запросов на поддержку.
- Заявка: один запрос на поддержку может касаться нескольких пользователей (например, несколько пользователей сталкиваются с одной и той же проблемой и связаны с одним и тем же запросом).

Для обработки связей «многие ко многим» используются «соединительные таблицы» (также известные как «соединяемые таблицы» или «ассоциативные таблицы»). Эта таблица содержит внешние ключи, которые ссылаются на первичные ключи двух связанных таблиц.

Если отношения в таблицы не относятся к «многие-ко-многим» считается что данные нормализованы и приведены к третьей нормальной форме. Нормализация – это процесс организации данных в базе данных для уменьшения избыточности и повышения целостности данных. Основными целями нормализации являются:

- Удаление повторяющихся данных
- Данные хранятся логически, для избежание ошибок, «аномалий» при вставке, обновлении или удалении данных.

Нормализация обычно включает в себя разделение большой таблицы на более мелкие связанные таблицы и определение связей между таблицами.

Этапы нормализации называются «нормальной формой», и наиболее часто используются первые три шага.

- Первая нормальная форма: гарантирует, что каждый столбец содержит только атомарные (неделимые) значения и что каждый элемент в столбце имеет один и тот же тип.
- Вторая нормальная форма: все неключевые атрибуты полностью функциональны в соответствии с первичным ключом.
- Третья нормальная форма: все свойства зависят только от первичного ключа.

В нашей модели нет связей «многие-ко-многим, по этой причине данные нормализованы и приведены к третьей нормальной форме. Составим структура таблиц нашей будущей базы данных в табличной форме (Таблица 2, таблица 3, таблица 4, таблица 5)

Таблица 2 – Структура таблицы «Работник»

Идентификатор	Тип	Примечание
КодРаботника	Счетчик	Автозаполнение, ключевое
Имя пользователя	Текстовый	-
Номер кабинета	Числовой	-
Имя корпуса	Текстовый	-

Таблица 3 – Структура таблицы «Комментарий»

Идентификатор	Тип	Примечание
КодКомментария	Счетчик	Автозаполнение, ключевое
КодРаботника	Текстовый	-
Имя пользователя	Текстовый	-
Номер кабинета	Числовой	-
Имя корпуса	Текстовый	-
Текст	Текстовый	-
Дата добавления	Дата/Время	-

Таблица 4 – Структура таблицы «Заявка»

Идентификатор	Тип	Примечание
КодЗаявки	Счетчик	Автозаполнение, ключевое
КодРаботника	Текстовый	-
КодСпециалиста	Текстовый	-
Номер телефона	Числовой	-
ФИО работника	Текстовый	-
Дата подачи заявления	Дата/Время	-
Дата добавления заявления	Дата/Время	-

Таблица 5 – Структура таблицы «IT-отдел»

Идентификатор	Тип	Примечание
КодТехника	Счетчик	Автозаполнение, ключевое
Часы работы	Дата/Время	-
Номер телефона техника	Числовой	-
ФИО техника	Текстовый	-

2.3. Подготовка к внедрению системы: настройка программного обеспечения, обучение персонала

Наш чат бот может запускаться как с помощью браузера, так и с помощью приложения на мобильном телефоне или компьютере. Проведя опрос устный опрос среди работников и перепроверив ноутбуки и моноблоки, находящиеся в учреждении Сургутского естественно-научного лицея, было выявлено, что в школе отсутствует техника, не проходящая под минимальные требования бота, так же личные мобильные телефоны учителей тоже проходят под минимальные требования. Дополнительная настройка техники для программы данного типа не требуется. Но по просьбе учителей и некоторых

работников лица, которые являются начинающими пользователями персональных компьютеров и смартфонов была оказана помощь в установке программы или вывода «значка» Telegram на рабочий стол. В большинстве случаев работники уже были зарегистрированы в самой системе телеграмм, так как многие его используют для личного общения или рабочих чатов [7].

Немаловажным, несмотря на то что более 80% сотрудников уверенно используют Telegram в личных целях, является составление инструкции для пользователя. Инструкция для конечного пользователя поможет последнему найти Телеграмм-бота, а также будет содержать немало подсказок-сценариев для пользователей плохо знакомых с мессенджером.

Пример инструкции конечного пользователя:

«Чат-бот «Техник СЕНЛ» - это программа-робот, которая поможет Вам заполнить заявку и сможет ответить на некоторые несложные вопросы.

Найдем нашего чат-бота в Telegram. Это можно сделать, нажав на поисковую строку в приложении Telegram и введя название чат-бота @TechnicSENL_bot

- Нажмите на найденный чат-бот, чтобы перейти к его странице. Вы легко сможете его определить по картинке с гербом лица.
- Нажмите кнопку "Старт" или "начать", чтобы запустить чат-бота.
- Следуйте инструкциям и вводите необходимую информацию, просите о помощи или объявите свой запрос. Чат-бот будет отвечать на ваши вопросы и выполнять задачи в соответствии с его возможностями.
- Вам могут потребоваться команды или ключевые слова для взаимодействия с чат-ботом. Они указаны на странице чат-бота или в справочной информации.
- Если у вас возникли проблемы с использованием чат-бота или необходима срочная консультация технического специалиста нажмите кнопку «Вызвать технического специалиста».

Это основная инструкция по использованию чат-бота в Telegram.»

Для сотрудников IT-отдела разработана отдельная инструкция с пояснениями возможных ошибок и полным логом написания программы и связи её с базой данных. Так как сотрудники в IT-отделе школы могут довольно часто сменяться необходимо подготовить инструкции, которые позволять новому работнику сразу разобраться в тонкостях работы системы и смело использовать данную разработку.

Вывод по главе 2. В данной главе были расписаны структура и функционал программы, разработана логическая и физическая модель базы данных. Также была подготовлена инструкция для будущих пользователей.

Глава 3 Физическое проектирование автоматизированной системы приема и обработки заявок в техническую поддержку

3.1. Выбор средств реализации автоматизированной системы

В своей работе любая система использует базу данных. Для решения нашей задачи по оптимизации будет необходимо создать два программных продукта непосредственно чат-бота и базу данных [8]. На момент исследования в среде разработчиков чаще всего используется продукт SQLite или же MySQL. Для выбора одной из систем сравним оба продукта по нескольким критериям:

Масштабируемость

MySQL:

- MySQL предназначен для обработки больших приложений и может управлять большим количеством одновременных транзакций и соединений.
- Он поддерживает многопользовательскую среду и может быть развернут на выделенных серверах для обработки больших баз данных и большого трафика пользователей.

SQLite:

- SQLite — это легкая файловая база данных, не предназначенная для многопользовательских приложений с высоким трафиком.
- Он лучше подходит для небольших проектов, сред разработки и тестирования.

Параллелизм

MySQL:

- MySQL поддерживает мощные механизмы управления параллелизмом, позволяющие нескольким пользователям одновременно читать и писать без конфликтов.

- Эффективно управляет одновременным доступом с помощью сложных механизмов блокировки.

SQLite:

- SQLite поддерживает только ограниченный параллелизм. Он позволяет использовать несколько устройств чтения, но одновременно только одного устройства записи, что может стать узким местом в приложениях с частыми операциями записи.
- Эти ограничения могут привести к проблемам с производительностью и задержкам в загруженных средах чат-ботов, где несколько пользователей могут одновременно отправлять сообщения и создавать запросы на поддержку.

Надежность и целостность данных

MySQL:

- MySQL предоставляет такие функции, как соответствие ACID (атомарность, согласованность, изоляция, долговечность) и поддерживает сложные транзакции для обеспечения целостности и надежности данных.
- Он предлагает широкие возможности восстановления и резервного копирования, что делает его идеальным для производственных сред, где потеря данных имеет решающее значение.

SQLite:

- SQLite совместим с ACID, но менее эффективен для обработки сложных транзакций и операций с большими данными.
- Хотя он надежен для небольших приложений, ему не хватает расширенных функций, таких как восстановление на определенный момент времени и репликация, которые необходимы для поддержания высокой целостности данных в больших системах.

Расширенные функции

MySQL:

- MySQL предоставляет широкий спектр расширенных функций, включая репликацию, кластеризацию и секционирование, которые помогут вам масштабировать ваши приложения по горизонтали и вертикали.
- Обширная поддержка хранимых процедур, триггеров и возможностей сложных запросов.
- SQLite:
- SQLite более ограничен с точки зрения расширенных функций. Ему не хватает поддержки некоторых функций, таких как определяемые пользователем функции, полнотекстовый поиск (так же, как его поддерживает MySQL) и расширенное индексирование.
- Он эффективен для простых и легких приложений, но не так богат, как MySQL.

Безопасность

MySQL:

- MySQL предоставляет мощные функции безопасности, включая аутентификацию пользователей, роли, разрешения и поддержку SSL для шифрования данных при передаче.
- Его можно настроить в соответствии с различными стандартами безопасности и передовыми практиками, что делает его идеальным для обработки конфиденциальных данных.

SQLite:

- SQLite имеет более ограниченные функции безопасности. Он не имеет встроенного управления пользователями и использует разрешения файловой системы для ограничения доступа.
- Он не подходит для приложений, требующих строгих мер безопасности и соблюдения нормативных требований.

MySQL — лучший выбор для создания чат-ботов Telegram, особенно в средах, где важны масштабируемость, параллелизм, целостность данных и расширенный функционал. Обеспечивает надежность и производительность,

необходимые для эффективной работы с многопользовательскими приложениями с высоким трафиком. SQLite отлично подходит для разработки, прототипирования и облегченных приложений, но может не соответствовать потребностям полностью развернутого чат-бота, поддерживающего большую базу пользователей.

Обоснование выбора технологий разработки.

Для создания Telegram-ботов возможно использовать множество различных языков программирования. Система не является требовательной к какому-то определенному языку, а разработчики уже создали все необходимые библиотеки для комфортной работы практически в каждом популярном языке программирования. Разберем какой из языков программирования можно выбрать.

Для разработки программного обеспечения технической поддержки будет использоваться высокоуровневый язык программирования Python как один из самых востребованных и легких на момент написания проекта.

Таблица 6 - Языки программирования

Язык программирования	Краткая характеристика	Необходимая библиотека
Python	Python — популярный язык программирования, известный своей простотой и универсальностью. Python часто используется для веб-разработки, анализа данных, искусственного интеллекта, автоматизации и многого другого. Его обширные библиотеки и фреймворки предоставляют решения для широкого круга задач.	python-telegram-bot

Язык программирования	Краткая характеристика	Необходимая библиотека
JavaScript	JavaScript в основном используется для создания динамического и интерактивного контента на веб-сайтах. Он работает в веб-браузерах и позволяет разрабатывать такие функции, как анимация, интерактивные формы и т.д. JavaScript универсален и может использоваться как для Front, так и для Backend веб-разработки. Он известен своей гибкостью и широко используется наряду с HTML и CSS для создания современных веб-приложений.	Telegraf.js или другие фреймворки, такие как Node.js.
Java	Java известен своей портативностью и надежностью. Разработан так, чтобы быть независимым от платформы, то есть программы Java могут запускаться на любом устройстве с виртуальной машиной Java (JVM). Java часто используется для создания настольных, веб- и мобильных приложений, а также для разработки на уровне предприятия. Характеризуется строгой типизацией, объектно-ориентированным подходом и обширными библиотеками, что делает его популярным выбором для крупномасштабных проектов разработки программного обеспечения.	Существуют различные библиотеки, такие как TelegramBots и TelegramBotsJava API, которые облегчают работу с Telegram API.
Ruby	Ruby — динамический объектно-ориентированный язык программирования, известный своей простотой и производительностью. Он часто используется для веб-разработки, особенно с платформой Ruby on Rails. Ruby довольно популярен среди программистов благодаря своему элегантному синтаксису и упору на удобство для разработчиков.	telegram-bot-ruby,

Язык программирования	Краткая характеристика	Необходимая библиотека
PHP	PHP обычно используется для веб-разработки. Он предназначен для создания динамических и интерактивных веб-сайтов. PHP встроен непосредственно в HTML-код и запускается на сервере, создавая динамические веб-страницы, которые могут взаимодействовать с базами данных и обрабатывать вводимые пользователем данные. Он известен своей простотой использования и гибкостью.	BotMan,
C#	C# — мощный язык программирования, разработанный Microsoft. Является частью платформы .NET и в основном используется для создания приложений под Windows, включая настольное программное обеспечение, веб-приложения и игры. C# широко используется в индустрии программного обеспечения для разработки широкого спектра приложений на разных платформах.	Telegram.Bot

Благодаря использованию связки Telegram-бота и базы данных MySQL конечный продукт будет отличаться лёгкостью и удобством использования как конечным пользователем, так и IT-отделом образовательного учреждения.

3.2. Разработка и тестирование системы управления заявками в техническую поддержку

Процесс написания программного кода для чат-бота технической поддержки на Python с использованием базы данных MySQL - это процесс который включает в себя несколько ключевых шагов, начиная с настройки окружения разработки и заканчивая тестированием и развертыванием бота.

Во-первых, необходимо установить инструменты разработки, такие как Python, MySQL и библиотеки Python, необходимые для использования

Telegram API (А именно Telebot) и MySQL (pymysql). Создадим новый проект Python и виртуальную среду.

- Создадим базу данных MySQL для хранения информации о обращениях в техническую поддержку. Определим структуру таблицы для хранения данных приложения.
- Используем библиотеку pymysql для Python, чтобы установить соединение с базой данных MySQL и выполнить запрос. Мы напишем функции для добавления новых данных, обновления статуса и получения данных.
- Создадим базовый класс для чат-бота на Python. Инициализируем бота и настроим обработчики команд и сообщений. Напишем функцию для обработки команды.
- Бот должен взаимодействовать с базой данных MySQL для хранения и получения информации. Мы будем использовать ранее написанные функции для выполнения соответствующих операций с базой данных в ответ на команды пользователя.
- При взаимодействии с базой данных и API Telegram важно обеспечить обработку всех возможных ошибок и исключений. Это поможет предотвратить сбои бота и обеспечит удобство работы с пользователем.
- После написания кода нам следует протестировать все его функции, чтобы убедиться в безошибочной работе системы. Мы используем тестовые данные для проверки работы всех команд и функций.

В связи с тем, что основная программа (Telegram-бот) будет создаваться на базе приложения Telegram интерфейс пользователя не будет отличаться от обычного интерфейса программы. Главной задачей является доступно и удобно организовать опрос пользователя о его проблемы и причины подачи заявки.

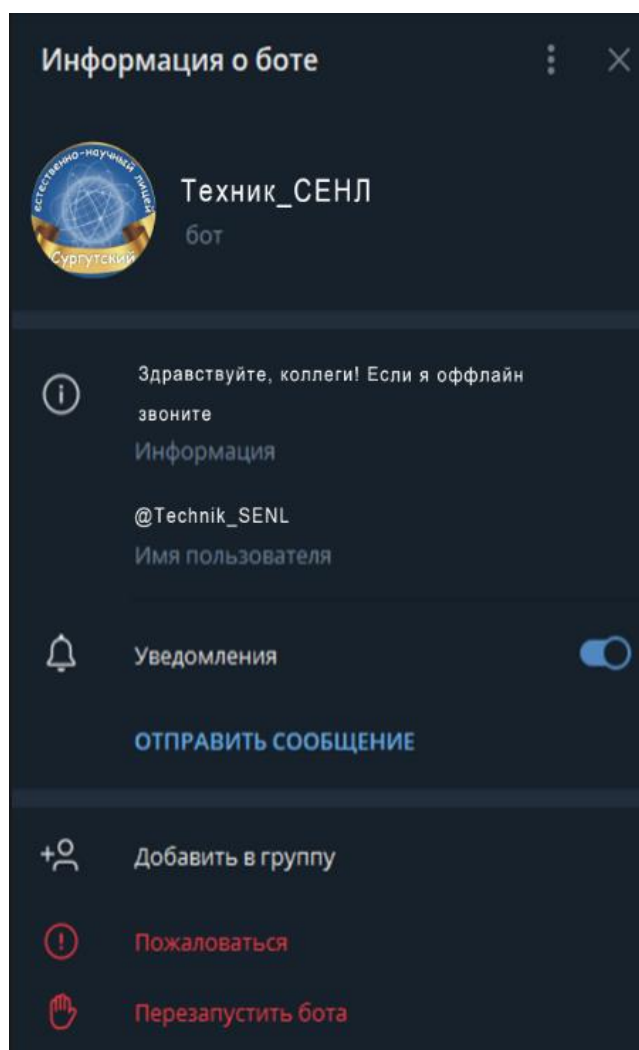


Рисунок 7 - Информационный экран бота.

Разработка автоматизированной системы

Приступим к написанию кода. Для создания собственного программного продукта нам потребуется использовать редактор кода. Редакторы кода могут оказать огромное влияние на производительность и процесс разработки программного обеспечения. Редакторы могут проверять код на наличие ошибок, отображать подсказки или автоматически вносить изменения во время написания кода. Для создания нашего продукта мы будем использовать довольно популярную, среди самостоятельных разработчиков, программу. Sublime Text — это удобный и мощный редактор, позволяющий быстро и эффективно писать код. Он поддерживает несколько языков

программирования и имеет множество функций, таких как завершение кода, быстрый поиск и замена, разделение экрана на несколько панелей, поддержка плагинов и т.д. Sublime Text известен своей скоростью и простотой настройки, это делает его отличным выбором для нашего проекта по разработки системы заявок. Также будет использоваться бесплатно распространяемая программа PyCharm Community Edition.

Перед тем как приступить к написанию функций и команд чат-бота необходимо обратиться к @BotFather – главному сервису в Телеграмме, отвечающему за регистрацию пользовательских ботов.

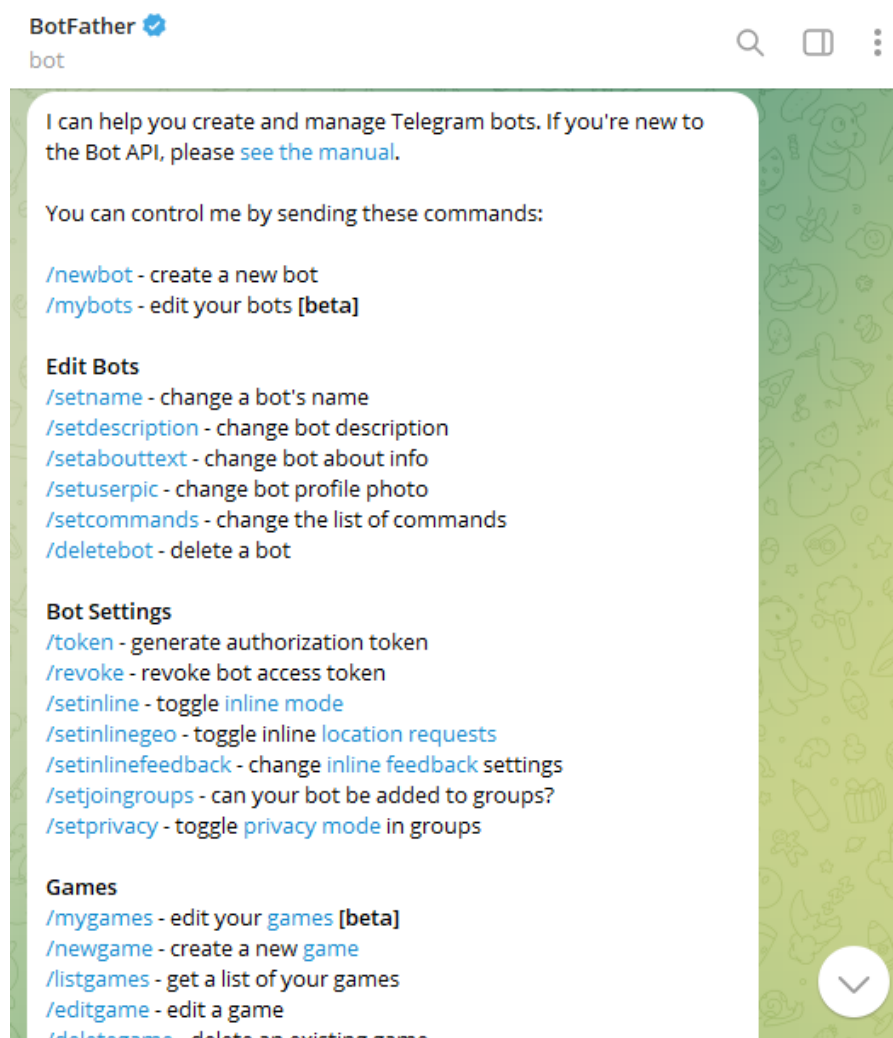


Рисунок 8 - Основные команды BotFather.

Для создания нового бота воспользуемся командой /newbot. BotFather автоматически предложит дать название ново созданному боту, а также его уникальный адрес. После всех необходимых вопросов к боту прикрепляется HTTP API, токен, что является уникальным для каждого бота.

Что же такое HTTP API , он же токен? Токен-бота Telegram похож на секретный ключ, который идентифицирует бота и позволяет ему взаимодействовать с API Telegram. Это длинная строка символов, автоматически генерируемая Telegram, а конкретно BotFather-ом. Этот токен используется разработчиками для аутентификации своего бота при отправке запросов на серверы Telegram. Если упростить можно сказать что это специальный пароль, который дает боту разрешение на безопасное общение с серверами Telegram. Важно обращаться с токеном осторожно и не раскрывать его третьим лицам, чтобы избежать возможных угроз безопасности и злоупотреблений именно по этой причине в целях безопасности на Рисунке 9 был убран уникальный токен для бота @TechnicSENL_bot.

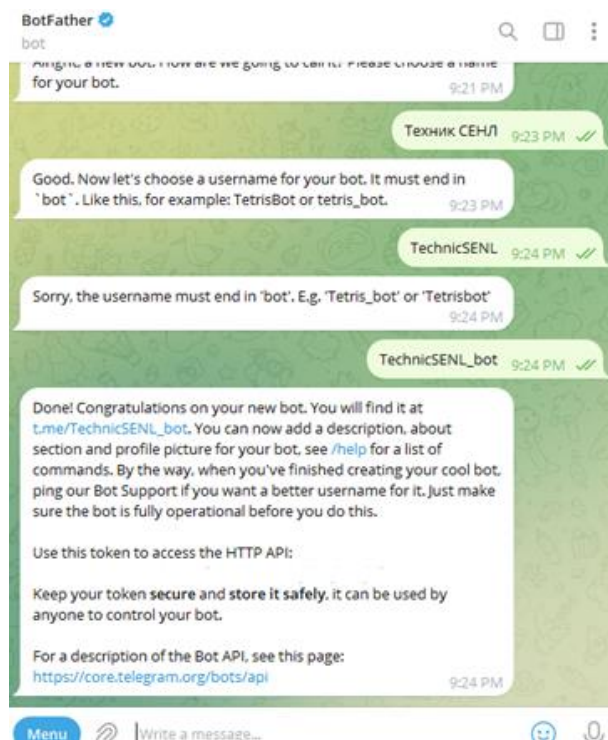
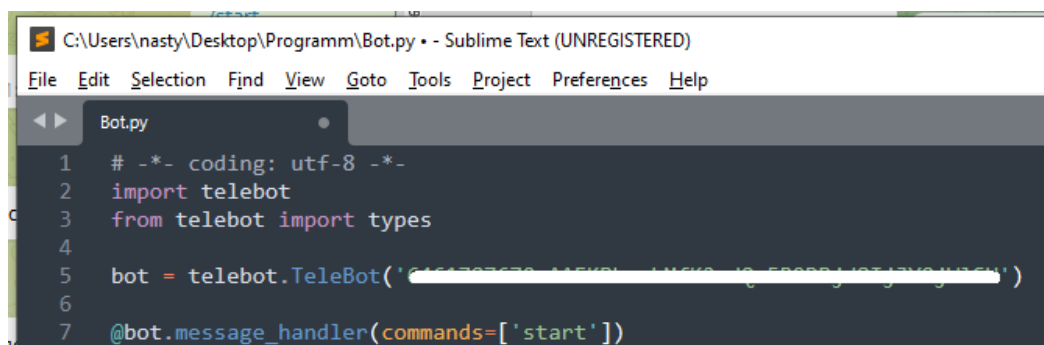


Рисунок 9 - Создания чат-бота.

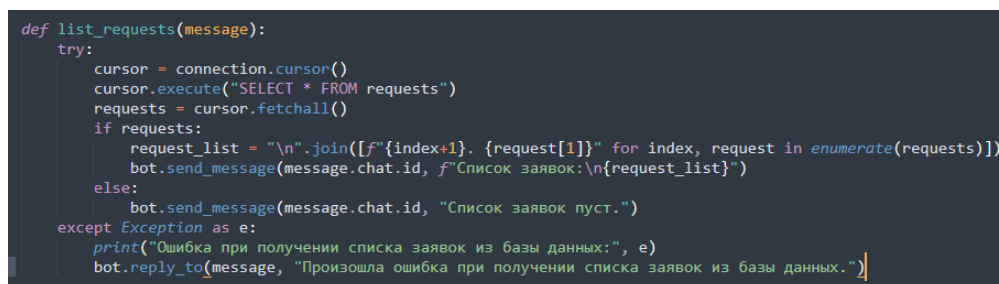
Так как за основу взять язык программирования Python приступим к написанию кода непосредственно с объявления используемых библиотек. Для создания Telegram-бота в библиотеках Python присутствует специально разработанная библиотека.



```
C:\Users\nasty\Desktop\Programm\Bot.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Bot.py
1 # -*- coding: utf-8 -*-
2 import telebot
3 from telebot import types
4
5 bot = telebot.TeleBot('151783673:AAEKP...')
6
7 @bot.message_handler(commands=['start'])
```

Рисунок 10 - Команда старта бота и передача его уникального токена.

Для дальнейшей работы нам необходимо связать нашу базу данных MySQL и Telegram-бота, который будет направлять собранную от пользователей информацию для записи её в базе данных. Для этого создадим отдельный метод.



```
def list_requests(message):
    try:
        cursor = connection.cursor()
        cursor.execute("SELECT * FROM requests")
        requests = cursor.fetchall()
        if requests:
            request_list = "\n".join([f"{index+1}. {request[1]}" for index, request in enumerate(requests)])
            bot.send_message(message.chat.id, f"Список заявок:\n{request_list}")
        else:
            bot.send_message(message.chat.id, "Список заявок пуст.")
    except Exception as e:
        print("Ошибка при получении списка заявок из базы данных:", e)
        bot.reply_to(message, "Произошла ошибка при получении списка заявок из базы данных.")
```

Рисунок 11 - Метод для связки с MySQL

Несмотря на то, что бот будет автоматически запрашивать всю необходимую информацию и главной целью создания бота является сбор заявок, не исключено что сотруднику может понадобится задать короткий вопрос без желания создавать заявку. Либо же у пользователя могут возникнуть трудности

при работе с ботом. Для этого создадим кнопку связи с техническим специалистом, благодаря которой технический специалист увидит запрос и вручную ответит независимо из чата бота.

Для этого нам понадобится создать пару функций и непосредственно кнопку связи с техническим специалистом.

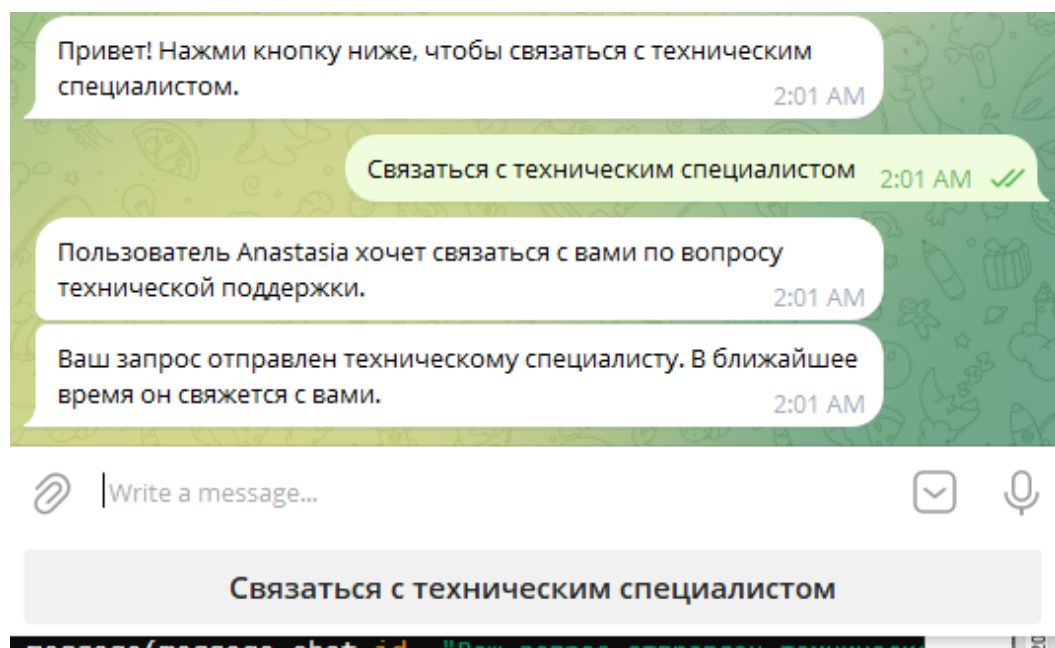


Рисунок 12 - Кнопка связи и вывод текста.

Для работы данной системы нам необходимо в первую очередь создать саму кнопку.

```
def start_message(message):  
    # Создание клавиатуры с кнопкой "Связаться с техническим специалистом"  
    keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)  
    button_contact_support = types.KeyboardButton("Связаться с техническим специалистом")  
    keyboard.add(button_contact_support)
```

Рисунок 13 - Создание кнопки связи.

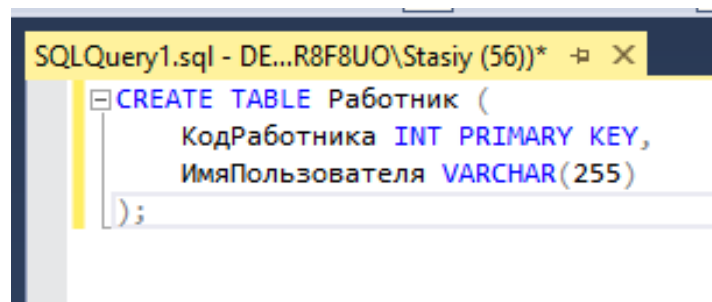
И непосредственно обработчик события нажатия кнопки «Связаться со специалистом».

```
# Обработчик нажатия на кнопку "Связаться с техническим специалистом"
@bot.message_handler(func=lambda message: message.text == "Связаться с техническим специалистом")
def contact_support(message):
    # Отправка сообщения с запросом в личные сообщения создателю бота
    bot.send_message('1848678417', f"Пользователь {message.from_user.first_name} хочет связаться с вами по вопросу технической поддержки.")
    bot.send_message(message.chat.id, "Ваш запрос отправлен техническому специалисту. В ближайшее время он свяжется с вами.")
```

Рисунок 14 - Обработчик события.

Разработка СУБД

Разработка базы данных начинается с создания таблиц, так как выше уже была полностью рассмотрена структура и представлена табличная форма (таблица 1.1-1.4 наше базы данных приступим к непосредственному написанию кода. Для начала, создадим таблицы, определив их структуру с помощью команды CREATE TABLE, где указываются названия столбцов и их типы данных (например, INT, VARCHAR, DATE и т. д.).



```
SQLQuery1.sql - DE...R8F8UO\Stasiy (56)*
CREATE TABLE Работник (
    КодРаботника INT PRIMARY KEY,
    ИмяПользователя VARCHAR(255)
);
```

Рисунок 15 - Создание таблицы «Работник»

КодРаботника стандартно остается числовым значением и является первичным ключом, а вот ИмяПользователя ограничивать не будем, ведь у нас в стране могут встречаться довольно длинное сочетания имени фамилии и отчества.

Аналогичным образом перейдем к созданию оставшихся трех таблиц.

```

CREATE TABLE Комментарий (
    КодКомментария INT PRIMARY KEY,
    КодРаботника INT,
    ИмяПользователя VARCHAR(255),
    НомерКабинета INT,
    ИмяКорпуса VARCHAR(255),
    Текст TEXT,
    ДатаДобавления DATETIME,
    FOREIGN KEY (КодРаботника) REFERENCES Работник(КодРаботника)
);

CREATE TABLE Заявка (
    КодЗаявки INT PRIMARY KEY,
    КодРаботника INT,
    КодСпециалиста INT,
    НомерТелефона VARCHAR(15),
    ФИОРаботника VARCHAR(255),
    ДатаПодачиЗаявления DATETIME,
    ДатаДобавленияЗаявления DATETIME,
    FOREIGN KEY (КодРаботника) REFERENCES Работник(КодРаботника)
);

CREATE TABLE IT_отдел (
    КодТехника INT PRIMARY KEY,
    ЧасыРаботы DATETIME,
    НомерТелефонаТехника VARCHAR(15),
    ФИОТехника VARCHAR(255)
);

```

Рисунок 16 - Создание таблиц «Комментарий», «Заявка» и «IT-отдел»

Уточним, что в основном у нас в таблице используются следующие типы данных:

- VARCHAR используется для хранения строковых данных, таких как имена и номера телефонов.
- INT используется для уникальных идентификаторов и числовых данных.
- DATETIME используется для хранения даты и времени.

Как уже было упомянуто выше важно соблюдать целостность данных и учитывать все связи. Для того чтобы учесть связь с другой таблицей был использован внешний ключ (FOREIGN KEY). В нашей таблице присутствуют две связи: таблица «Комментарий» ссылается на таблицу «Работник2», связывая комментарии с конкретными работниками и таблица «Заявка» также ссылается на таблицу «Работник» для связи заявок с их авторами.

В процессе общения чат бот будет автоматически задавать вопросы для формирования качественной заявки. Основные данные, которые необходимо будет ввести пользователю для формирования заявки отображены в ГА-диаграмме (Рисунок 1). В состав обязательных данных будет входить: ФИО сотрудника, подающего заявку, территориальное расположение - корпус и номер кабинета. При дальнейшем обращении чат-бот будет автоматически определять сотрудника и повторный ввод данных не потребует. В качестве дополнительной информации для качественного формирования заявки в технический отдел необходимо указать причину обращения [31]. Пример: поломка принтера, замятие бумаги, проблемы с ноутбуком, не работает интернет и т.д. После того как все необходимые процедуры будут пройдены заявке присуждается уникальный номер, и вся информация передаётся в базу данных.

Для тестирования данного продукта необходимо будет завести тестовую учетную запись. Создадим нового учителя, укажем его кабинет и корпус, в котором расположен кабинет. Главная задача теста чтобы данные синхронизировались с нашей БД.

Основные функциональные требования:

- Бот должен принимать текстовые сообщения от пользователей
- Бот должен запрашивать у пользователя контактную информацию (ФИО, телефон, корпус, кабинет)
- Бот должен запрашивать у пользователя описание проблемы
- Бот должен отправлять полученные данные в систему технической поддержки
- Бот должен отправлять пользователю подтверждение о принятии заявки

Чтобы обеспечить надежность, функциональность и качество программы, необходимо написать сценарии для её тестирования. По своей сути это модуляция реальных способов использования программы в различных условиях. Автоматизируя процесс тестирования, сценарии позволяют разработчикам быстро выявлять ошибки и проблемы с

производительностью, чтобы их можно было своевременно исправлять и улучшать. Сценарии тестирования также обеспечивают согласованность процедур тестирования, облегчают регрессионное тестирование, чтобы гарантировать, что новые изменения не нарушают существующую функциональность.

Приведем пару сценариев тестирования для нашей системы сбора заявок:

Тест 1: Отправка заявки в техническую поддержку

- Пользователь отправляет боту текстовое сообщение
- Бот запрашивает контактные данные пользователя (ФИО, корпус, номер кабинета)
- Пользователь вводит свои контактные данные
- Бот запрашивает описание проблемы
- Пользователь вводит описание проблемы
- Бот отправляет полученные данные в систему технической поддержки
- Бот отправляет пользователю подтверждение о принятии заявки
- Проверить, что заявка успешно передана в систему технической поддержки

Тест 2: Отмена отправки заявки

- Пользователь отправляет боту текстовое сообщение
- Бот запрашивает контактные данные пользователя
- Пользователь вводит только имя и отказывается вводить остальные данные
- Бот должен предложить пользователю завершить оформление заявки или отменить её
- Пользователь выбирает отмену
- Бот должен подтвердить отмену заявки
- Проверить, что заявка не была передана в систему технической поддержки

Тест 3: Ошибочный ввод данных

- Пользователь отправляет боту текстовое сообщение
- Бот запрашивает контактные данные пользователя

- Пользователь вводит некорректный корпус (например, 1 или 2, когда в системе они записаны как младший и старший)
- Бот должен запросить повторный ввод корпуса
- Пользователь вводит корректный корпус
- Бот должен запросить описание проблемы
- Пользователь вводит описание проблемы
- Бот должен отправить полученные данные в систему технической поддержки
- Бот должен отправить пользователю подтверждение о принятии заявки
- Проверить, что заявка успешно передана в систему технической поддержки

3.3. Оценка экономической эффективности системы управления заявками в техническую поддержку

Расчеты экономической эффективности важны для принятия обоснованных решений о распределении ресурсов. Оценивая взаимосвязь между понесенными затратами и достигнутыми результатами, организация может определить наиболее эффективный и экономичный способ достижения своих целей. Также проводя расчёты, можно оценить эффективность различных вариантов или стратегий и расставить приоритеты среди тех, которые обеспечивают наилучшее соотношение цены и качества. Также это поможет улучшить подотчетность и прозрачность в управлении ресурсами. Количественно оценивая затраты и выгоды, связанные с различными направлениями действий, можно обосновать свои решения. В целом, расчет экономической эффективности необходим для оптимизации использования ресурсов, снижения финансовых рисков и обеспечения подотчетности при принятии решений [10].

Проведем расчет экономического эффекта, получаемый за счет экономии рабочего времени сотрудника. Так как времени на работу будет

уходить меньше работник будет выполнять значительно большее количество работы.

Для расчетов следует ввести следующие обозначения:

T1 – время, затрачиваемое на операции до внедрения программного продукта, ч;

T2 – время, затрачиваемое на операции после внедрения программного продукта, ч.

До внедрения программы большая сотрудник в среднем тратил на одну операцию – 15 минут, необходимо было дойти до физического журнала, находящегося на вахте, выписать все необходимые заявки отдельно себе в блокнот, рассортировать заявки и приступить к выполнению

$T1 = 15$ минут или $0,25$ часа.

После внедрения программы время сократилось и составляет 3 минут:

$T2 = 3$ минут или $0,05$ часа

$\Delta t = 15 - 3 = 12$ минут или $0,2$ часа.

Рассчитаем стоимость выполнения работ сотрудниками притом, что в месяц он работает $8 \text{ часов} * 24 \text{ дня} = 192$ часа. Должностной оклад сотрудника составляет 25000 рублей. Следовательно, стоимость одного часа работы:

$25000/192 = 130,2$ руб.

Страховые взносы, начисляемые на з/п:

$15000 * 0,34 / 192 = 44,27$ руб.

Определим суммарные затраты работника с учетом страховых взносов по формуле:

$$P_{\text{сум}} = P_{\text{час}} + B_{\text{СТР}} \quad (2)$$

где $P_{\text{сум}}$ – стоимость одного часа работы, руб;

$B_{\text{стр}}$ – страховые взносы, руб.

Из этого следует, что

$P_{\text{сум}} = 130,2 + 44,27 = 174,47$ руб.

Рассчитаем годовую экономию времени и срок окупаемости.

Для расчёта годовой экономии времени воспользуемся формулой:

$$\Delta T = \Delta t * N_{\text{опер}} * N_{\text{дней}} * N_{\text{месяц}} \quad (3)$$

где Δt – экономия времени, ч;

$N_{\text{опер}}$ – количество операций;

$N_{\text{дней}}$ – количество рабочих дней в месяце;

$N_{\text{месяц}}$ – количество рабочих месяцев.

Из этого следует, что

$$\Delta T = 0,2 * 5 * 23 * 12 = 276 \text{ ч.}$$

Рассчитаем размер годовой экономии времени от внедрения программного продукта по формуле

$$\Delta T_{\text{пп}} = \Delta T * R_{\text{сум}} \quad (4)$$

где ΔT – годовая экономия времени, ч;

$R_{\text{сум}}$ – стоимость одного часа работы, руб.

Из этого следует, что

$$\Delta T_{\text{пп}} = 276 * 174,47 = 48153,72 \text{ ч.}$$

Так как данная информационная система будет предоставлена работником образовательной организации, следующие расчеты будут вестись от оклада сотрудника [11].

Экономический эффект составляет величину, равную годовой экономии средств за вычетом затрат на разработку и реализацию программного продукта определяется по формуле

$$E = \Delta T_{\text{пп}} - Z / \text{Посн} \quad (5)$$

где $\Delta T_{\text{пп}}$ – экономия времени от внедрения программного продукта, ч;

$Z / \text{Посн}$ – заработная плата разработчика программного продукта, руб.

Из этого следует, что

$$E = 48153,72 - 25000 = 23153 \text{ руб.}$$

Расчет срока окупаемости производится по формуле

$$PP = Z / \text{Посн} / E \quad (6)$$

где $Z / \text{Посн}$ – заработная плата разработчика программного продукта, руб;

E – экономический эффект, руб.

Из этого следует, что

$$25000/23153 = 1 \text{ год.}$$

Таким образом, экономическая часть выпускной квалификационной работы позволила определить затраты на разработку автоматизированной системы подачи заявок в МБОУ Сургутском естественно-научном лицее и рассчитать срок окупаемости затрат.

Вывод по главе 3.

В данной главе был обоснован выбор средств разработки программного обеспечения, описаны основные этапы разработки системы и расписано обоснование экономической эффективности данного проекта.

Заключение

Итогом бакалаврской работы является разработанная система приема и обработки заявок в техническую поддержку образовательного учреждения, которое позволит повысить скорость обработки и выполнение заявок, а также позволит упростить учет выполненной работы IT-отдела.

Для этого были выполнены следующие пункты:

- Исследованы существующие системы управления заявками в образовательных учреждениях и проведен анализ их преимуществ и недостатков.
- Выявлены потребности пользователей (администраторов технической поддержки, пользователей) в системе управления заявками.
- Разработано техническое задание на разработку программного обеспечения для управления заявками в техническую поддержку образовательного учреждения.
- Спроектирована база данных для хранения заявок, информации о пользователе и других необходимых данных.
- Написан программный код для реализации основных функций системы, таких как создание заявки, назначение исполнителя, отслеживание статуса заявки.
- Протестировано разработанное программное обеспечение на соответствие функциональным и техническим требованиям.

Список используемой литературы

1. Аверченков В., Лозбинев Ф., Тищенко А. Информационные системы в производстве и экономике: учебное пособие. – Litres, 2015.
2. Аветисян Татьяна Владимировна, Львович Яков Евсеевич, Преображенский Андрей Петрович, Преображенский Юрий Петрович МЕТОДИКА УПРАВЛЕНИЯ РЕСУРСАМИ ДЛЯ IT-КОМПАНИИ, ЗАНИМАЮЩЕЙСЯ СЕРВИСНЫМ ОБСЛУЖИВАНИЕМ // НК. 2023. №1.
3. Анянова Е. В. Разработка информационной системы заявок на ремонт компьютерной техники. – 2023. URL: <https://elar.usfeu.ru/handle/123456789/12716>
4. Афонин Денис Сергеевич, Астахова Татьяна Николаевна, Романова Анна Александровна РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ И ПРИЛОЖЕНИЯ ПО ПРИЕМУ И СОПРОВОЖДЕНИЮ ЗАЯВОК СЛУЖБЫ ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ АДМИНИСТРАЦИИ // International Journal of Open Information Technologies. 2021. №5. URL: <https://cyberleninka.ru/article/n/razrabotka-informatsionnoy-sistemy-i-prilozheniya-po-priemu-i-soprovozhdeniyu-zayavok-sluzhby-tehnicheskoy-podderzhki>
5. Безвесильная А. А., Федин Ф. О., Чискидов С. В. Разработка информационной системы учета заявок на выделение вычислительных ресурсов в корпоративной сети образовательной организации // Научные и образовательные проблемы гражданской защиты. – 2017. – №. 1 (32). – С. 29-40. URL: <https://cyberleninka.ru/article/n/razrabotka-informatsionnoy-sistemy-ucheta-zayavok-na-vydelenie-vychislitelnyh-resursov-v-korporativnoy-seti-obrazovatelnoy>
6. Волк В. К., Симахин В. А. Базы данных Ч. 1. Проектирование и программирование. – 2018.

7. Глушенко С.А., Долженко А.И., Малеев Д.В. Разработка системы Helpdesk для отдела сопровождения ООО «Электронная Медицина» // Вестник РГЭУ РИНХ. 2016. №2 (54).
8. Задорин М. А. Разработка и создание программного обеспечения для IT службы организации: дис. – 2017.
9. Зайцева Т. В. Анализ и обоснование выбора инструментальных средств автоматизированной разработки прикладного программного обеспечения корпоративных информационных систем. – 2018.
10. Зацаринный А. А., Шабанов А. П. Системные аспекты технологии управления научными и образовательными сервисами // Открытое образование. – 2017. – Т. 21. – №. 2. – С. 88-96.
11. Истратова Е. Е., Черний Ю. С. СРЕДСТВО АВТОМАТИЗАЦИИ РАБОТЫ ОТДЕЛА ТЕХНИЧЕСКОЙ ПОДДЕРЖКИ ДЛЯ ПРЕДПРИЯТИЙ МАЛОГО И СРЕДНЕГО БИЗНЕСА // Научно-редакционная коллегия: ОВ Асканова, д. э. н., зам. директора по научному и инновационному. – 2021. – С. 295.
12. Казарин О., Шубинский И. Основы информационной безопасности: надежность и безопасность программного обеспечения. Учебное пособие для СПО. – Litres, 2020.
13. ГОСТ Р ИСО/МЭК 15504-1-2009 Информационные технологии. Оценка процессов. Часть 1. Концепция и словарь
14. ГОСТ Р ИСО/МЭК 15504-2-2009 Информационная технология. Оценка процесса. Часть 2. Проведение оценки
15. ГОСТ Р ИСО/МЭК 15504-3-2009 Информационная технология. Оценка процесса. Часть 3. Руководство по проведению оценки
16. ГОСТ Р 53798-2010 Стандартное руководство по лабораторным информационным менеджмент-системам (ЛИМС)
17. ГОСТ Р 54360-2011 Лабораторные информационные менеджмент-системы (ЛИМС). Стандартное руководство по валидации (ЛИМС)
18. ГОСТ Р 54593-2011 Информационные технологии. Свободное

- программное обеспечение. Общие положения
- 19.ГОСТ Р 55711-2013 Комплекс технических средств автоматизированной адаптивной ВЧ (КВ) дуплексной радиосвязи. Алгоритмы работы
 - 20.ГОСТ Р ИСО/МЭК ТО 9294-93. Информационная технология. Руководство по управлению документированием программного обеспечения
 - 21.ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программного обеспечения
 - 22.ГОСТ Р 50739-95. Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования
 - 23.ГОСТ Р 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов. Типовое руководство.
 - 24.ГОСТ 24.104-85. ЕССАСУ. Автоматизированные системы управления. Общие требования
 - 25.ГОСТ 24.701-86. ЕССАСУ. Надежность автоматизированных систем управления. Основные положения
 - 26.ГОСТ 19.102-77. ЕСПД. Стадии разработки
 - 27.ГОСТ 19.201-78. ЕСПД. Техническое задание. Требование к созданию и оформлению
 - 28.ГОСТ 34.003 – 90 Информационные технологии(ИТ). Комплекс стандартов на автоматизированные системы. Термины и определения
 - 29.ГОСТ 34.201 – 89 Информационные технологии(ИТ). Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем
 - 30.ГОСТ 34.601 – 90 Информационные технологии(ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания автоматизированных систем
 - 31.ГОСТ 34.602 – 89 Информационные технологии(ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание АС