Федеральное государственное автономное образовательное учреждение высшего образования

«CИБИPCKИЙ ФEДEPAЛБHЫЙ УHИBEPCИTET»

Космических и информационных технологий

институт

Информационные системы

кафедра

Утверждаю

Заведующий кафедры ИC

А.А. Раскина

подпись инициалы, фамилия

« » 2023 г.

# БAKAЛABPCKAЯ PAБOTA

09.03.02 – «Информационные системы и технологии»

Автоматизация приема и обработки заявок отделом

ООО «МаксБонус»

Руководитель

подпись, дата

профессор, к.т.н. Ю. A. Маглинец

инициалы, фамилия

Выпускник

подпись, дата

П. М. Горбунов

инициалы, фамилия

Нормкотролер ст. препод. ИС Ю.В. Шмагрис

подпись, дата инициалы, фамилия

Красноярск 2023Федеральное государственное автономное образовательное

учреждение высшего образования

«CИБИPCKИЙ ФEДEPAЛБHЫЙ УHИBEPCИTET»

Космических и информационных технологий

институт

Информационные системы

кафедра

Утверждаю

Заведующий кафедры ИC

А.А. Раскина

подпись инициалы, фамилия

« » 2023 г.

# ЗАДАНИЕ

# НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

# В форме бакалаврской работы

Студенту Горбунову Петру Михайловичу

Группа: КИ19-14Б Направление: 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: Автоматизация приема и обработки заявок отделом ООО «МаксБонус»

Утверждена приказом по университету ………..

Руководитель ВКР: Ю.А. Маглинец, профессор, кандидат технических наук, доцент кафедры Систем искусственного интеллекта ИКИТ СФУ.

Исходные материалы ВКР: …..

Перечень разделов ВКР: …..

Руководитель Ю. A. Маглинец

подпись, дата

Задание принял к исполнению П. М. Горбунов

подпись, дата

« » 2023 г.

**РЕФЕРАТ**

**Содержание**

Введение

1.1. Актуальность темы --

1.2. Цель работы --

1.3. Задачи работы --

2 Теоретические основы автоматизации приема и обработки заявок

2.1. Основы автоматизации бизнес-процессов --

2.2. Технологии баз данных, фронтенда и сервера --

2.3. Обзор существующих решений --

Анализ деятельности отдела ООО «Максбонус»

3.1. Описание структуры и функций отдела --

3.2. Анализ потребностей отдела --

3.3. Формализация процессов приема и обработки заявок --

Разработка автоматизированной системы

4.1. Проектирование базы данных –

4.2. Разработка серверной части

4.3 Создание чат-бота телеграмм для оповещения о новых заявках

4.4. Разработка фронтенда

5 Заключение

6 Список использованных источников

Приложения

7.1. Приложение А: Функциональные требования к автоматизированной системе

7.2. Приложение B: Техническое задание на разработку автоматизированной системы

7.3. Приложение C: Схема структуры базы данных

7.4. Приложение D: Диаграммы взаимодействия компонентов системы (UML-диаграммы)

7.5. Приложение E: Описание API серверной части

7.6. Приложение F: Исходный код серверной части (примеры)

7.7. Приложение G: Исходный код фронтенда (примеры)

7.9. Приложение I: Руководство пользователя системы

7.11. Приложение K: Перечень используемых программных продуктов и инструментов разработки

**Тут глава 1**

**ВВЕДЕНИЕ**

В современном мире, где бизнес постоянно развивается и усложняется, автоматизация бизнес-процессов играет ключевую роль в повышении эффективности и конкурентоспособности компаний. Особое внимание уделяется автоматизации процессов приема и обработки заявок, поскольку они напрямую влияют на качество обслуживания клиентов и, следовательно, на успех компании в целом.

ООО «Максбонус» является динамично развивающейся компанией, которая стремится к оптимизации своих бизнес-процессов и улучшению качества предоставляемых услуг. В связи с этим возникает потребность в автоматизации приема и обработки заявок отделом технической поддержки.

**Актуальность темы**

В современном бизнес-мире, где скорость и эффективность обработки информации имеют первостепенное значение, автоматизация рабочих процессов становится основным направлением развития многих компаний. Таким образом, тема автоматизации приема и обработки заявок отделом ООО «Максбонус» является крайне актуальной по следующим причинам:

Увеличение эффективности работы: автоматизация рабочих процессов позволяет сократить время на обработку заявок, что способствует увеличению производительности и снижению времени ожидания для клиентов.

Снижение вероятности ошибок: автоматическая обработка заявок минимизирует вероятность человеческих ошибок, повышая качество предоставляемых услуг.

Рационализация рабочей нагрузки: автоматизация позволяет оптимизировать рабочую нагрузку сотрудников, освобождая их от рутинных задач и давая возможность сконцентрироваться на более важных и творческих аспектах работы.

Контроль и мониторинг: автоматизированные системы предоставляют удобные инструменты для контроля и мониторинга рабочих процессов, что позволяет своевременно выявлять и устранять проблемы, а также собирать статистические данные для анализа эффективности работы отдела.

Улучшение взаимодействия с клиентами: современные автоматизированные системы обработки заявок могут интегрироваться с различными каналами связи (телефон, электронная почта, мессенджеры), что обеспечивает удобство общения с клиентами и ускоряет обработку их запросов.

Экономия ресурсов: автоматизация приема и обработки заявок может значительно снизить затраты на операционные расходы и управление персоналом, что позволяет более эффективно использовать ресурсы компании и сокращать издержки.

В целом, автоматизация приема и обработки заявок отделом ООО «Максбонус» является важным шагом в развитии компании, поскольку это позволяет оптимизировать рабочие процессы, повышать эффективность и качество обслуживания клиентов, а также обеспечивать более высокий уровень контроля над внутренними операциями. Внедрение автоматизации в работу отдела будет способствовать росту конкурентоспособности ООО «Максбонус» на рынке, облегчать адаптацию к переменам и инновациям, а также улучшать управление ресурсами и мотивацию сотрудников.

**Цель и задачи работы**

Целью данной работы является разработка и внедрение системы автоматизации приема и обработки заявок для ООО «Максбонус», которая позволит сократить время обработки заявок, уменьшить вероятность ошибок, связанных с человеческим фактором, и повысить удовлетворенность клиентов за счет более быстрого и качественного обслуживания.

Для достижения поставленной цели в работе предполагается выполнить следующие задачи:

Изучить теоретические основы автоматизации бизнес-процессов и современные технологии разработки баз данных, фронтенда и сервера.

1. Проанализировать деятельность отдела ООО «Максбонус», выявить основные проблемы и потребности, связанные с приемом и обработкой заявок.
2. Формализовать процессы приема и обработки заявок с целью их автоматизации.
3. Разработать структуру базы данных для хранения и обработки информации о заявках.
4. Разработать серверную часть системы для обеспечения взаимодействия между базой данных и фронтендом.
5. Разработать фронтенд системы, предоставляющий удобный и интуитивно понятный интерфейс для работы сотрудников отдела.
6. Разработать требования к системе автоматизации приема и обработки заявок.
7. Разработать и внедрить систему автоматизации приема и обработки заявок в ООО «Максбонус».

Тут глава 2

2.1. Сложно как то это всё корректно назвать, подумайте. Например – «Последовательность выполнения работ в области автоматизации бизнес-процессов».

Основы автоматизации бизнес-процессов в контексте автоматизации приема и обработки заявок отделом ООО «Максбонус» включают следующие аспекты:

Анализ и оптимизация бизнес-процессов: прежде чем приступить к автоматизации, необходимо тщательно проанализировать текущие бизнес-процессы отдела, связанные с приемом и обработкой заявок. Определить их слабые стороны, избыточные шаги и возможности для оптимизации.

Моделирование бизнес-процессов: после анализа и оптимизации процессов, следует разработать модели процессов, которые будут автоматизированы. Моделирование позволяет структурировать процессы, определить роли участников, их взаимодействия и порядок выполнения действий.

Выбор технологий и инструментов: определить подходящие технологии для автоматизации, такие как базы данных, серверные и клиентские технологии, с учетом требований к системе, ресурсов и инфраструктуры компании.

Разработка автоматизированной системы: разработать систему на основе выбранных технологий, реализующую автоматизацию процессов приема и обработки заявок. Система должна быть гибкой и масштабируемой, позволяя внедрять изменения в будущем.

Внедрение системы и обучение пользователей: после разработки системы необходимо провести ее внедрение, настройку и интеграцию с существующими решениями компании. Обучить сотрудников отдела работе с новой системой и предоставить им необходимую поддержку.

Мониторинг и оптимизация: после внедрения системы следует постоянно мониторить ее работу, определять возможные проблемы и предлагать улучшения. Проводить оптимизацию системы с целью повышения ее эффективности и удовлетворения меняющихся потребностей бизнеса.

Оценка результатов автоматизации: провести анализ и оценку экономической эффективности автоматизированной системы, рассчитав затраты на разработку и внедрение системы, а также ожидаемые экономические выгоды, связанные с сокращением времени обработки заявок, уменьшением ошибок и повышением производительности сотрудников.

Контроль и поддержание качества процессов: Важным аспектом автоматизации является контроль и поддержание качества процессов на всех этапах их выполнения. Следует разработать систему метрик и индикаторов, которые помогут отслеживать эффективность процессов и вносить корректировки при необходимости.

Коммуникация и координация: Успешная автоматизация бизнес-процессов требует эффективной коммуникации и координации между различными участниками проекта, включая разработчиков, сотрудников отдела, менеджеров и других заинтересованных сторон.

Безопасность и надежность: Автоматизированная система должна обеспечивать безопасность данных и информации, связанной с приемом и обработкой заявок, а также надежность работы системы, защиту от сбоев и потери данных.

Используя эти основы автоматизации бизнес-процессов, дипломная работа направлена на создание эффективной и надежной системы для автоматизации приема и обработки заявок отделом ООО «Максбонус», что в свою очередь повысит уровень удовлетворенности клиентов и оптимизирует работу компании.

2.2. Архитектура разрабатываемой системы.

Каждый архитектурный блок – как-то обосновать. Привести общие для системы диаграммы – например, пакетов, компонент, развёртывания. Это – один из самых интересных параграфов, но очень слабенько сформулирован. И не на месте он. В целом, изложение материала должно отражать логику работы, а тут многие параграфы и главы местами поперепутаны.

Логика – примерно должна быть такая:

1. Обращение клиента – понять, что он вообще хочет, рамочно (высокоуровневые цели, задачи)
2. Обследование объекта автоматизации
3. Описание бзнес-процессов «как есть», постановка проблем, требующих автоматизации
4. Разработка ТЗ
5. Обзор литературы - имеющиеся аналоги.
6. Изучение подходов и возможных способов автоматизации
7. Выбор и обоснование архитектуры, проектирование компонент
8. Программная реализация и т.д.

В рамках дипломной работы по автоматизации приема и обработки заявок отделом ООО «Максбонус» предлагается использовать следующие технологии:

1. База данных: PostgreSQL - мощная и надежная система управления базами данных с открытым исходным кодом. PostgreSQL поддерживает продвинутые функции, такие как транзакции, хранимые процедуры, индексы и предоставляет высокую производительность и масштабируемость.
2. Серверная часть: Go (Golang) - современный язык программирования, разработанный Google, обеспечивающий высокую скорость разработки и производительность приложений. Go хорошо подходит для создания серверных приложений, таких как RESTful API, благодаря своей конкурентности, простоте и отличной поддержке стандартной библиотекой.
3. Фреймворк для серверной части: Echo или Gin - популярные и быстрые фреймворки на Go для создания веб-приложений и RESTful API. Оба фреймворка предоставляют удобные маршрутизаторы, middleware и инструменты для работы с запросами и ответами.
4. Фронтенд: React - популярная библиотека для разработки пользовательских интерфейсов, созданная Facebook. React позволяет разрабатывать компоненты интерфейса с высокой степенью пере использования и обеспечивает быстрое обновление интерфейса благодаря виртуальному DOM.
5. Среда разработки и инструменты: Visual Studio Code, Git, GitHub/GitLab, Node.js, npm/yarn для управления пакетами и зависимостями, а также инструменты для тестирования и сборки проекта, такие как Jest, Webpack или Create React App.
6. Дополнительные технологии и инструменты: GraphQL или REST для API, Redux или MobX для управления состоянием приложения, TypeScript для статической типизации и повышения качества кода, а также CSS-фреймворки и библиотеки компонентов, такие как Bootstrap, Material-UI или Ant Design, для ускорения разработки пользовательского интерфейса.

Использование указанных технологий баз данных, фронтенда и сервера позволит разработать и внедрить эффективную, масштабируемую и надежную автоматизированную систему для приема и обработки заявок отделом ООО «Максбонус». Комбинация этих технологий обеспечит высокую производительность системы, упрощенное взаимодействие между компонентами и гибкость для будущих изменений и расширения функционала.

1. Разработка автоматизированной системы с использованием таких технологий приведет к следующим преимуществам:
2. Быстрое время разработки: благодаря мощным библиотекам и фреймворкам, разработчики смогут быстрее создавать и внедрять систему.
3. Повышенная надежность: Использование проверенных и надежных технологий обеспечит стабильность и безопасность работы системы.
4. Масштабируемость: Технологии, такие как PostgreSQL и Go, обеспечивают высокую масштабируемость и возможность увеличения нагрузки на систему без существенных изменений в архитектуре.
5. Простота поддержки и обновления: Четкая разделенность на клиентскую и серверную части, а также использование компонентного подхода в React, облегчает процесс обновления и поддержки системы.
6. Пере используемость компонентов: React позволяет создавать пере используемые компоненты, что упрощает разработку и сокращает время на создание пользовательских интерфейсов.
7. Улучшенное взаимодействие с пользователями: Современные технологии фронтенда, такие как React, позволяют создавать интуитивно понятные и удобные интерфейсы для пользователей, что в свою очередь повышает уровень удовлетворенности сотрудников отдела и клиентов компании.

В результате использования этих технологий баз данных, фронтенда и сервера будет создана автоматизированная система, которая облегчит процесс приема и обработки заявок, сократит время обработки, уменьшит количество ошибок и повысит общую эффективность работы отдела ООО «Максбонус».

2.3 Обзор существующих решений

При разработке автоматизированной системы для приема и обработки заявок отделом ООО «Максбонус» важно провести анализ существующих решений на рынке, чтобы определить их преимущества и недостатки, а также выявить возможности для создания уникального и конкурентоспособного продукта. Вот краткий обзор некоторых существующих решений в данной области:

1. CRM-системы (Customer Relationship Management): Эти системы предназначены для управления взаимоотношениями с клиентами, включая прием и обработку заявок, отслеживание продаж, учет контактной информации и другие функции. Примеры CRM-систем: Salesforce, Microsoft Dynamics 365, Zoho CRM, iikoCRM, Bitrix24, HubSpot CRM. Преимущества включают гибкость и многофункциональность, недостатки - возможно высокая стоимость и избыточность функций для конкретных нужд ООО «Максбонус».
2. BPM-системы (Business Process Management): Эти системы ориентированы на автоматизацию, оптимизацию и мониторинг бизнес-процессов. Примеры BPM-систем: TIBCO, Appian, Pega, Bonita. Преимущества включают возможность моделирования и автоматизации процессов, недостатки - сложность настройки и интеграции с другими системами.
3. Helpdesk-системы: Ориентированы на обработку запросов от клиентов и пользователей, таких как заявки на поддержку, жалобы и вопросы. Примеры Helpdesk-систем: Zendesk, Freshdesk, Bitrix24, AmoCRM, Help Scout. Преимущества - простота в использовании и специализация на обработке заявок, недостатки - ограниченные возможности по автоматизации процессов и управлению отношениями с клиентами.
4. ERP-системы (Enterprise Resource Planning): Комплексные системы для управления ресурсами компании, включая финансы, закупки, производство и логистику. Примеры ERP-систем: SAP, Oracle, 1С, Галактика, Microsoft Dynamics AX. Преимущества - широкий функционал и интеграция с другими системами, недостатки - сложность внедрения, высокая стоимость и возможная избыточность функций.

В зависимости от специфики и требований ООО «Максбонус», можно выбрать одно из существующих решений или разработать собственную систему автоматизации приема и обработки заявок, учитывая уникальные потребности и особенности компании. Разработка собственной системы может предоставить следующие преимущества:

1. Индивидуальный подход: Создание собственной системы позволяет полностью учесть специфику работы отдела и компании, включая особенности бизнес-процессов, требования к отчетности и структуру данных.
2. Гибкость и масштабируемость: Разработка собственного решения обеспечивает гибкость в настройке функционала и возможность масштабирования системы в соответствии с ростом и развитием компании.
3. Интеграция с существующими системами: Разработанная система может быть легко интегрирована с уже используемыми в компании программами и сервисами, обеспечивая единый процесс обработки заявок и управления информацией.
4. Контроль над разработкой и поддержкой: Компания сможет контролировать процесс разработки, внедрения и поддержки системы, что обеспечит быструю реакцию на изменения требований и возникновение проблем.
5. Экономическая выгода: В долгосрочной перспективе, создание собственного решения может быть более экономичным вариантом, чем использование сторонних систем с ежемесячной или ежегодной оплатой.

В результате анализа существующих решений для автоматизации приема и обработки заявок отделом, ООО «Максбонус» может определить наиболее подходящий подход и технологии для создания эффективной системы, которая поможет улучшить работу отдела, сократить время обработки заявок и повысить уровень удовлетворенности клиентов.

Тут глава 3

3.1 Описание структуры и функций отдела

Отдел автоматизации приема и обработки заявок компании ООО «Максбонус» является ключевым звеном, отвечающим за эффективное взаимодействие между клиентами и организацией. Структура и функции отдела могут быть следующими:

1. Менеджер по приему заявок: Менеджеры по приему заявок отвечают за первичное общение с клиентами, получение и регистрацию заявок, а также предоставление информации о статусе заявок. Они также могут заниматься консультированием клиентов по продуктам и услугам компании.
2. Специалист по обработке заявок: Специалисты по обработке заявок анализируют полученные заявки, определяют их приоритетность и распределяют их между другими сотрудниками отдела или другими подразделениями компании. Они также следят за выполнением заявок и контролируют сроки их исполнения.
3. Специалист по координации: Специалист по координации отвечает за организацию взаимодействия между сотрудниками отдела и другими подразделениями компании. Он также может участвовать в процессе принятия решений, связанных с обработкой и выполнением заявок.
4. Аналитик: Аналитик отвечает за сбор, анализ и интерпретацию данных о заявках, процессах и результате работы отдела. Он также предлагает рекомендации по оптимизации процессов и повышению эффективности работы отдела.
5. Специалист по поддержке и обучению: Специалист по поддержке и обучению проводит тренинги и обучение сотрудников отдела, помогает им развивать профессиональные навыки и знания, а также оказывает техническую поддержку в случае возникновения вопросов или проблем.
6. Руководитель отдела: Руководитель отдела координирует работу всех сотрудников, контролирует выполнение задач и сроки их исполнения, принимает решения о распределении ресурсов и приоритетах работы. Руководитель также отвечает за планирование и стратегическое развитие отдела, а также за коммуникацию с вышестоящим руководством компании и другими отделами.
7. ИТ-специалист: ИТ-специалист отвечает за техническую поддержку и обслуживание автоматизированной системы приема и обработки заявок. Он также участвует в разработке и оптимизации системы, обеспечивая ее надежность, безопасность и эффективность.

В целом, отдел автоматизации приема и обработки заявок ООО «Максбонус» играет важную роль в обеспечении эффективной работы компании и удовлетворении потребностей клиентов. Создание и поддержание автоматизированной системы для приема и обработки заявок позволяет отделу сосредоточиться на основных задачах и улучшать качество предоставляемых услуг.

Основные функции отдела автоматизации приема и обработки заявок включают:

1. Прием и регистрация заявок от клиентов.
2. Обработка и распределение заявок между сотрудниками и другими подразделениями.
3. Контроль исполнения заявок и сроков их выполнения.
4. Поддержание связи с клиентами и предоставление информации о статусе их заявок.
5. Анализ и оптимизация процессов обработки заявок.
6. Разработка, внедрение и поддержка автоматизированной системы для приема и обработки заявок.
7. Обучение и поддержка сотрудников отдела.

Компетентный и эффективный отдел автоматизации приема и обработки заявок является важным фактором успеха компании ООО «Максбонус» и позволяет обеспечить высокий уровень обслуживания и удовлетворенности клиентов.

3.2 Анализ потребностей отдела

Проведение анализа потребностей отдела автоматизации приема и обработки заявок компании ООО «Максбонус» является важным этапом разработки и внедрения автоматизированной системы. Анализ потребностей позволяет выявить ключевые проблемы и требования отдела, которые должны быть учтены при разработке системы. Вот некоторые основные аспекты, которые следует проанализировать:

1. Объем и типы заявок: Определение количества и видов заявок, которые обрабатывает отдел, поможет определить масштаб системы и функциональные требования к ней. Это также позволит определить возможные узкие места и оптимизировать процессы обработки заявок.
2. Роли и обязанности сотрудников: Изучение ролей и обязанностей сотрудников отдела позволит определить, какие функции и возможности должны быть доступны для разных пользователей системы. Это также поможет определить, какие процессы должны быть автоматизированы, и какие должны остаться под контролем сотрудников.
3. Интеграция с другими системами: Анализ потребностей в интеграции с другими используемыми в компании системами, такими как CRM, ERP, или документооборот, позволит определить необходимые интерфейсы и форматы обмена данными между системами.
4. Безопасность и доступ к данным: Оценка требований к безопасности и контролю доступа к данным поможет определить, какие меры защиты должны быть реализованы для обеспечения надежной и безопасной работы системы.
5. Требования к отчетности и аналитике: Определение потребностей отдела в аналитике и отчетности позволит определить, какие инструменты и функции анализа данных должны быть включены в систему, и какие данные должны быть собраны и хранены для создания отчетов.
6. Требования к обучению и поддержке: Оценка потребностей сотрудников отдела в обучении и поддержке использования автоматизированной системы поможет определить, какие материалы и инструкции должны быть разработаны для обучения, и какая поддержка должна быть предоставлена пользователям.

Проведение тщательного анализа потребностей отдела автоматизации приема и обработки заявок компании ООО «Максбонус» позволит разработать эффективное и гибкое решение, которое удовлетворит требования отдела и обеспечит улучшение его работы. При анализе потребностей следует учитывать не только текущие требования, но и возможное будущее развитие компании и отдела, чтобы система могла масштабироваться и адаптироваться к изменяющимся условиям.

После проведения анализа потребностей отдела автоматизации приема и обработки заявок и определения основных требований к системе, следующим шагом будет разработка технического задания и выбор подходящих технологий для реализации автоматизированной системы. Важно выбирать проверенные и надежные технологии, которые обеспечат стабильную работу системы и возможность ее развития в будущем.

Формализация процессов приема и обработки заявок в отделе автоматизации компании ООО «Максбонус» предполагает создание четких и структурированных процедур, которые будут использоваться сотрудниками для выполнения своих задач. Формализация этих процессов позволяет обеспечить эффективность, снизить вероятность ошибок и упростить автоматизацию. Вот основные этапы формализации процессов:

1. Определение основных этапов обработки заявок: выделите ключевые этапы, которые включают прием, регистрацию, обработку, распределение, исполнение и закрытие заявок.
2. Разработка процедур для каждого этапа: Создайте детализированные инструкции для выполнения каждого этапа процесса обработки заявок. Определите критерии и порядок выполнения задач, а также ответственных сотрудников.
3. Определение параметров заявок: установите основные атрибуты, которые будут использоваться для хранения информации о заявках, такие как идентификатор, дата создания, статус, тип, приоритет, срок исполнения, контактные данные клиента и дополнительные поля, необходимые для обработки заявок.
4. Разработка правил распределения заявок: определите критерии и механизмы распределения заявок между сотрудниками отдела или другими подразделениями компании. Это может включать автоматическое распределение на основе навыков, нагрузки или приоритетов заявок.
5. Установление критериев и процедур контроля качества: определите стандарты качества для обработки заявок и разработайте процедуры для контроля их соблюдения. Это может включать регулярные проверки выполненных заявок, анализ причин возникновения ошибок и корректирующие меры для их устранения.
6. Создание механизмов отслеживания и анализа процессов: разработайте методы сбора данных о процессах обработки заявок, их продолжительности, эффективности и других параметрах, которые позволят проводить анализ и оптимизацию процессов.
7. Документирование процессов: зафиксируйте все процедуры и правила в виде документации, которая будет доступна для сотрудников отдела. Документация должна быть понятной, структурированной и актуализированной. Это позволит сотрудникам легко ознакомиться с процессами, обучиться им и соблюдать требования.
8. Обучение сотрудников: проведите обучение сотрудников отдела по новым формализованным процессам. Обучение должно быть комплексным и включать теоретическую часть, практические упражнения и постоянную поддержку в освоении новых процедур.
9. Внедрение и мониторинг: внедрите формализованные процессы в рабочую практику отдела и регулярно контролируйте их выполнение. Собирайте обратную связь от сотрудников и клиентов для определения возможных улучшений и корректировки процессов.
10. Постоянное совершенствование: проводите периодический анализ процессов обработки заявок с целью выявления возможных улучшений и оптимизации. Реализуйте внесенные изменения и обновляйте документацию, обучайте сотрудников и мониторьте результаты.

Формализация процессов приема и обработки заявок в отделе автоматизации компании ООО «Максбонус» позволит улучшить эффективность работы отдела, упростить автоматизацию процессов и обеспечить более высокий уровень удовлетворенности клиентов. Кроме того, это облегчит разработку и внедрение автоматизированной системы приема и обработки заявок, так как процессы будут четко структурированы и документированы.

Тут глава 4

4.1 Проектирование базы данных

Проектирование базы данных для автоматизации приема и обработки заявок отделом компании ООО «Максбонус» включает определение таблиц, полей, связей между ними, а также разработку схемы базы данных. В данном случае, используется база данных PostgreSQL[1]. Рассмотрим основные таблицы и их поля:

1. Таблица «board»

* id (INTEGER PRIMARY KEY) – идентификатор доски
* userid (INTEGER, FOREIGN KEY)– идентификатор пользователя
* title (TEXT) – заголовок доски
* description (TEXT) – Описание доски

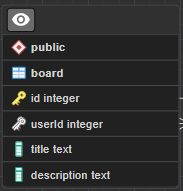


Рисунок 1 – созданная таблица board

1. Таблица «reason»

* id (INTEGER PRIMARY KEY) – идентификатор причины обращения
* nameReason (TEXT) – название причины обращения

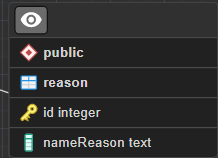


Рисунок 2 – созданная таблица reason

1. Таблица «status»

* id (INTEGER PRIMARY KEY) – идентификатор причины обращения
* nameStatus (TEXT) – название статуса

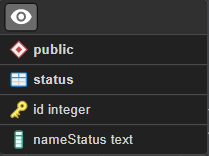


Рисунок 3 – созданная таблица status

1. Таблица «task»

* id (INTEGER PRIMARY KEY) – идентификатор задачи
* title (TEXT) – описание задачи
* userid (INTEGER, FOREIGN KEY) – идентификатор пользователя
* reasonid (INTEGER, FOREIGN KEY) – идентификатор причины обращения
* boardid (INTEGER, FOREIGN KEY) – идентификатор доски
* statusid (INTEGER, FOREIGN KEY) – идентификатор статуса
* icon (TEXT) – изображение задачи
* phone (VARCHAR (10)) – номер телефона
* email (TEXT) – почта
* companyname - (TEXT) – название компании

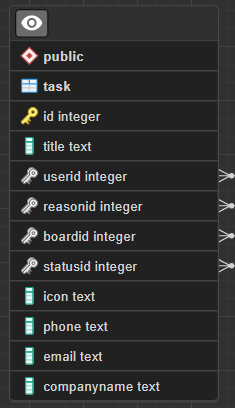


Рисунок 4 – созданная таблица task

1. Таблица «user»

* id (INTEGER PRIMARY KEY) – идентификатор пользователя
* login (TEXT) – логин пользователя
* password (TEXT) – пароль пользователя
* FIO (TEXT) – ФИО пользователя

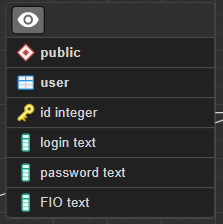


Рисунок 5 – созданная таблица user

Связи между таблицами:

- в таблице «task» поле userid ссылается на поле id таблицы «user»

- в таблице «task» поле reasonid ссылается на поле id таблицы «reason»

- в таблице «task» поле board ссылается на поле id таблицы «board»

- в таблице «task» поле status ссылается на поле id таблицы «status»

- в таблице «board» поле userid ссылается на поле id таблицы «user»

Для обеспечения безопасности при работе с базой данных в автоматизации приема и обработки заявок отделом компании ООО «Максбонус», пароли пользователей хранятся в виде хеш-суммы, сгенерированной с использованием алгоритма SHA-256. Это является стандартной практикой и позволяет защитить пароли даже в случае утечки данных.

При создании нового пользователя или изменении пароля, вместо хранения пароля в открытом виде, система будет генерировать хеш-сумму с использованием алгоритма SHA-256 и сохранять этот хеш в поле "password" таблицы "user". При аутентификации пользователя, система также будет генерировать хеш-сумму введенного пароля и сравнивать ее с хешем, хранящимся в базе данных, чтобы определить, является ли пароль верным.

Для реализации этой функциональности, можно воспользоваться встроенными функциями и библиотеками языка программирования серверной части (Go) или использовать внешние пакеты, предоставляющие реализацию алгоритма SHA-256.

Для обеспечения безопасности хранения паролей пользователей, в проекте используется алгоритм SHA-256, который является рекомендованным стандартом безопасности, согласно NIST [2]. Реализация алгоритма SHA-256 для языка программирования Go представлена в пакете crypto/sha256 [3]. Такой подход соответствует рекомендациям OWASP по безопасному хранению паролей [4]. При создании нового пользователя или изменении пароля, система генерирует хеш-сумму пароля с использованием алгоритма SHA-256 и сохраняет его в базе данных. В процессе аутентификации пользователя, система также генерирует хеш-сумму введенного пароля и сравнивает ее с хранящимся хешем для проверки соответствия пароля.

4.2 Разработка серверной части.

Разработка серверной части системы автоматизации приема и обработки заявок отделом компании ООО «Максбонус» включает использование языка программирования Go и ряда библиотек и технологий для обеспечения производительности, безопасности и удобства разработки. В частности, сервер будет использовать следующие компоненты:

1. Gin: это высокопроизводительный HTTP-фреймворк для Go, который создание и обработку HTTP-запросов, управление роутингом и мидлварами. Gin позволяет и создавать RESTful API для обработки запросов от клиентской части системы.
2. DI Container: Dependency Injection (DI) контейнеры используются для управления зависимостями между компонентами приложения. В Go есть несколько библиотек для DI, таких как Uber's Dig, Google's Wire или Facebook's Inject. Использование DI контейнера облегчает разработку и тестирование, делая код более модульным и расширяемым.
3. JWT Validation: JSON Web Tokens (JWT) являются стандартом для безопасной передачи информации между клиентом и сервером. JWT используется для аутентификации пользователей и предоставления доступа к защищенным ресурсам. В Go существуют библиотеки, такие как github.com/dgrijalva/jwt-go или github.com/golang-jwt/jwt, которые упрощают работу с JWT, включая создание, проверку подписи и валидацию токенов.

На рисунке 6 представлены реализованные роуты для http запросов

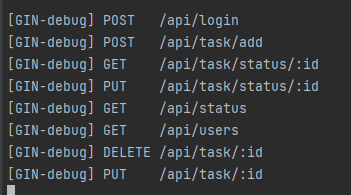


Рисунок 6 – реализованные роуты для сервера

Роутер «/api/login» необходим для аутентификации пользователей в системе автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Этот эндпоинт отвечает за проверку учетных данных пользователя, таких как логин и пароль, и в случае успешной аутентификации генерирует и выдает JSON Web Token (JWT), который будет использоваться для доступа к защищенным ресурсам системы.

При отправке POST-запроса на «/api/login» с корректными учетными данными (логином и паролем) в теле запроса, серверная часть на Go с использованием фреймворка Gin обрабатывает запрос, хэширует в sha-256, выполняет проверку учетных данных пользователя, сверяя их с сохраненными данными в базе данных PostgreSQL. Если данные верны, сервер генерирует JWT, который содержит информацию о пользователе и его роли в системе, а также подписывает его секретным ключом. Затем сервер отправляет ответ с JWT в формате JSON, который клиентская часть (фронтенд) будет использовать для дальнейшей авторизации и доступа к ресурсам системы.

Важно отметить, что при неудачной аутентификации (например, если предоставлены неверные учетные данные) сервер возвращает соответствующий код ошибки и сообщение, информируя пользователя о проблеме.

Использование роутера «/api/login» для аутентификации пользователей является стандартным подходом при разработке веб-приложений и обеспечивает безопасность и контроль доступа к ресурсам системы автоматизации приема и обработки заявок отделом компании ООО «Максбонус».

Тестирование эндопоинта будет происходить при помощи запросов в Postman. Необходимо ввести в строку адрес сервиса localhost:8080/api/login .

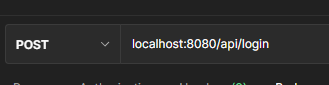


Рисунок 7 – ввод адрес сервиса

После того, как адрес сервера введён необходимо отправить json в body запроса.

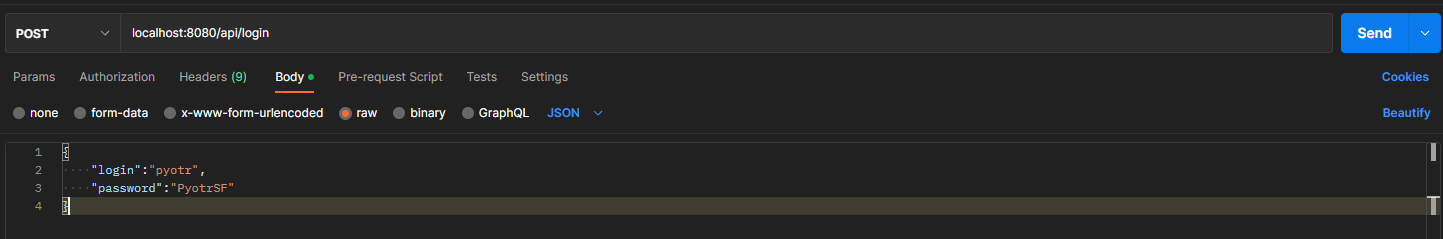


Рисунок 8 – заполнение json в соответствии с api

После того, как json заполнен необходимо отправить запрос, затем, сервер отправит «Status: 200 OK» и код авторизации если логин и пароль совпадает в базе данных.

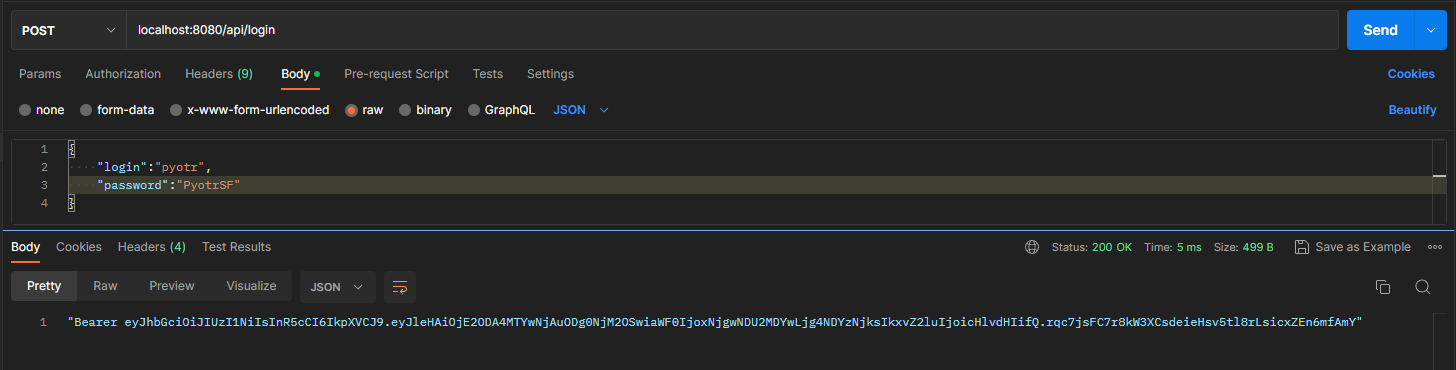


Рисунок 9 – успешная авторизация

Если данные не совпадает, то сервер вернёт «Status: 401 Unauthorized».

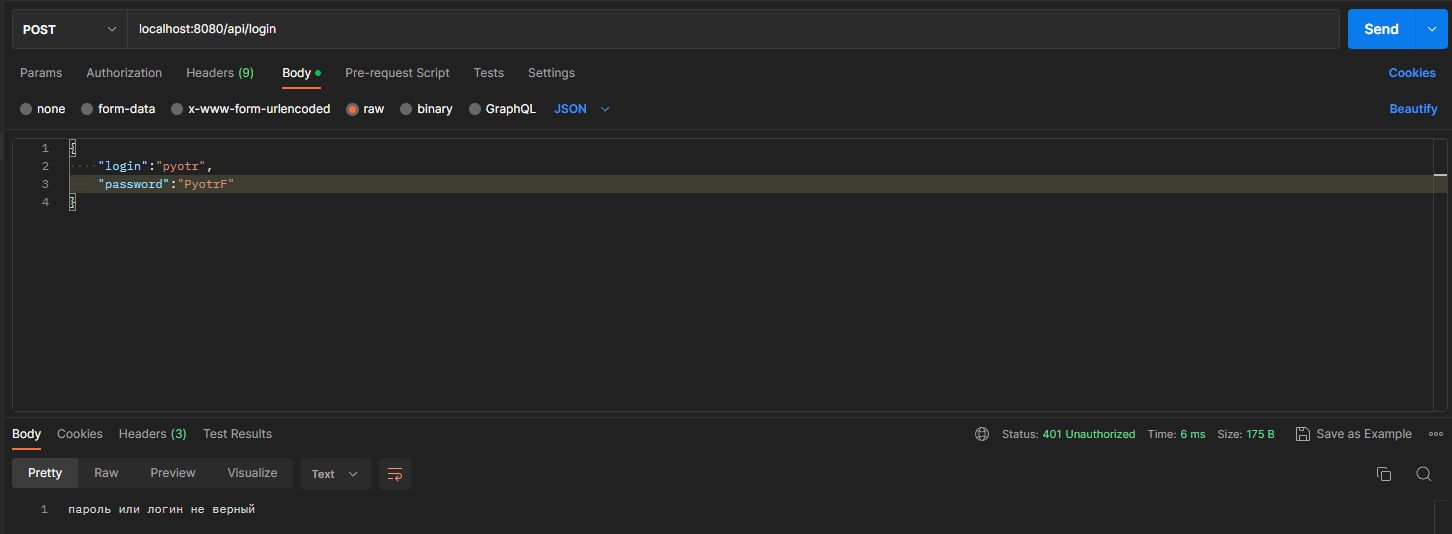


Рисунок 10 – пароль или логин не верный

Роутер «/api/task/add» необходим для добавления новых заявок в систему автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Этот эндпоинт предназначен для создания новых задач, связанных с обработкой заявок, и сохранения их в базе данных для дальнейшей обработки и мониторинга.

При отправке POST-запроса на «/api/task/add» с корректными данными о задаче в теле запроса, серверная часть на Go с использованием фреймворка Gin обрабатывает запрос и выполняет проверку наличия необходимых полей и правильности их формата. При успешной проверке сервер создает новую запись о задаче в базе данных PostgreSQL и возвращает ответ с идентификатором созданной задачи и другой информацией.

Для доступа к этому эндпоинту пользователь должен быть аутентифицирован, и его роль в системе должна предоставлять разрешение на создание новых задач. В этом случае, JWT, полученный при успешной аутентификации, должен быть передан в заголовке запроса (в поле "Authorization") для подтверждения прав доступа пользователя.

Использование роутера «/api/task/add» для добавления новых заявок позволяет упростить процесс создания и обработки заявок, делая его более удобным и эффективным для сотрудников отдела. Это также обеспечивает структурированное хранение информации о задачах и их обработке, что способствует повышению производительности работы отдела и улучшению качества предоставляемых услуг.

Тестирование эндопоинта будет происходить при помощи запросов в Postman. Необходимо ввести в строку адрес сервиса localhost:8080/api/login .

Роутер «/api/task/status/:id» необходим для обновления статуса заявок и отслеживания их выполнения в системе автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Этот эндпоинт позволяет сотрудникам отдела изменять статус задачи на основе ее выполнения и контролировать прогресс выполнения всех заявок.

При отправке PUT-запроса на «/api/task/status/:id» с новым статусом задачи (например, "выполнено" или "в процессе") в теле запроса, серверная часть на Go с использованием фреймворка Gin обрабатывает запрос, извлекает идентификатор задачи из параметра URL и выполняет проверку наличия задачи с указанным идентификатором в базе данных PostgreSQL. Если задача существует, сервер обновляет ее статус и возвращает ответ с подтверждением об успешном обновлении статуса задачи.

Доступ к этому эндпоинту также должен быть ограничен аутентифицированными пользователями, чья роль в системе предоставляет разрешение на изменение статуса задач. JWT, полученный при успешной аутентификации, должен быть передан в заголовке запроса (например, в поле "Authorization") для подтверждения прав доступа пользователя.

Использование роутера «/api/task/status/:id» для обновления статуса заявок обеспечивает эффективное отслеживание выполнения задач и позволяет менеджерам и сотрудникам отдела оценивать прогресс в обработке заявок. Это также позволяет своевременно определять возможные проблемы и задержки, что в свою очередь помогает оптимизировать рабочие процессы и повышать уровень удовлетворенности клиентов.

GET-роутер «/api/status» необходим для проверки статуса заявки в системе автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Этот эндпоинт позволяет пользователям и сотрудникам отдела получать информацию о текущем состоянии и статусе обработки конкретной заявки.

При отправке GET-запроса на «/api/status» с идентификатором заявки в параметрах запроса, серверная часть на Go с использованием фреймворка Gin обрабатывает запрос и осуществляет поиск заявки с указанным идентификатором в базе данных PostgreSQL. Если заявка найдена, сервер возвращает ответ с информацией о статусе заявки, такой как "новая", "в процессе" или "выполнена", а также другие сведения, связанные с заявкой.

Доступ к этому эндпоинту может быть ограничен аутентифицированными пользователями, чья роль в системе предоставляет разрешение на просмотр статуса заявок. В этом случае, JWT, полученный при успешной аутентификации, должен быть передан в заголовке запроса (например, в поле "Authorization") для подтверждения прав доступа пользователя.

Использование роутера «/api/status» для проверки статуса заявки позволяет пользователям и сотрудникам отдела отслеживать ход выполнения заявок и контролировать их обработку. Это обеспечивает прозрачность рабочего процесса и позволяет клиентам быть в курсе состояния своих заявок, что, в свою очередь, способствует повышению уровня удовлетворенности клиентов и эффективности работы отдела.

GET-роутер «/api/users» необходим для получения списка пользователей системы автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Этот эндпоинт позволяет менеджерам и сотрудникам отдела просматривать список доступных пользователей, чтобы назначать их на заявки, а также для обеспечения правильного распределения рабочей нагрузки между сотрудниками.

При отправке GET-запроса на «/api/users», серверная часть на Go с использованием фреймворка Gin обрабатывает запрос и получает список всех пользователей из базы данных PostgreSQL. Затем сервер возвращает ответ, содержащий информацию о каждом пользователе, такую как имя, фамилия, должность и контактные данные.

Доступ к этому эндпоинту, как правило, ограничен аутентифицированными пользователями с определенными ролями и разрешениями, такими как менеджеры или администраторы системы. JWT, полученный при успешной аутентификации, должен быть передан в заголовке запроса (например, в поле "Authorization") для подтверждения прав доступа пользователя.

Использование роутера «/api/users» для получения списка пользователей для назначения на заявку облегчает процесс распределения задач между сотрудниками и обеспечивает более эффективное управление рабочим процессом. Это также способствует улучшению коммуникации и координации между сотрудниками, что, в свою очередь, ведет к повышению производительности отдела и качества предоставляемых услуг.

DELETE-роутер «/api/task/:id» необходим для удаления заявок в системе автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Этот эндпоинт позволяет менеджерам и сотрудникам отдела удалять заявки, которые больше не актуальны или были завершены, что помогает поддерживать актуальность информации и оптимизировать рабочий процесс.

При отправке DELETE-запроса на «/api/task/:id» с идентификатором заявки в параметрах URL, серверная часть на Go с использованием фреймворка Gin обрабатывает запрос и выполняет проверку наличия заявки с указанным идентификатором в базе данных PostgreSQL. Если заявка существует, сервер удаляет ее из базы данных и возвращает ответ с подтверждением успешного удаления заявки.

Доступ к этому эндпоинту должен быть ограничен аутентифицированными пользователями с определенными ролями и разрешениями, такими как менеджеры или администраторы системы. JWT, полученный при успешной аутентификации, должен быть передан в заголовке запроса (например, в поле "Authorization") для подтверждения прав доступа пользователя.

Использование роутера «/api/task/:id» для удаления заявок обеспечивает гибкость и контроль над рабочим процессом, что позволяет удалять устаревшие или завершенные заявки и освобождать ресурсы для работы с актуальными задачами. Это также содействует улучшению общей организации работы отдела и повышению уровня удовлетворенности клиентов.

PUT-роутер «/api/task/:id» необходим для изменения заявок в системе автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Этот эндпоинт позволяет менеджерам и сотрудникам отдела вносить изменения в существующие заявки, такие как обновление статуса, изменение сроков исполнения, назначение нового исполнителя или корректировка деталей заявки.

При отправке PUT-запроса на «/api/task/:id» с идентификатором заявки в параметрах URL и данными для обновления в теле запроса, серверная часть на Go с использованием фреймворка Gin обрабатывает запрос и выполняет проверку наличия заявки с указанным идентификатором в базе данных PostgreSQL. Если заявка существует, сервер обновляет ее сведения в соответствии с переданными данными и возвращает ответ с подтверждением успешного обновления заявки.

Доступ к этому эндпоинту должен быть ограничен аутентифицированными пользователями с определенными ролями и разрешениями, такими как менеджеры или администраторы системы. JWT, полученный при успешной аутентификации, должен быть передан в заголовке запроса (например, в поле "Authorization") для подтверждения прав доступа пользователя.

Использование роутера «/api/task/:id» для изменения заявок предоставляет гибкий и управляемый рабочий процесс, позволяющий адаптироваться к изменяющимся условиям и потребностям клиентов. Это также способствует улучшению коммуникации и координации между сотрудниками отдела, что, в свою очередь, ведет к повышению производительности и качества предоставляемых услуг.

* 1. Создание чат-бота телеграмм для оповещения о новых заявках

Создание чат-бота в Telegram для оповещения о новых заявках является важным дополнительным функционалом системы автоматизации приема и обработки заявок отделом компании ООО «Максбонус». Чат-бот будет автоматически отправлять уведомления сотрудникам отдела о поступлении новых заявок, что позволит оперативно реагировать на запросы клиентов и повысит эффективность работы отдела.

Для создания и разработки чат-бота в Telegram предполагается выполнить следующие этапы:

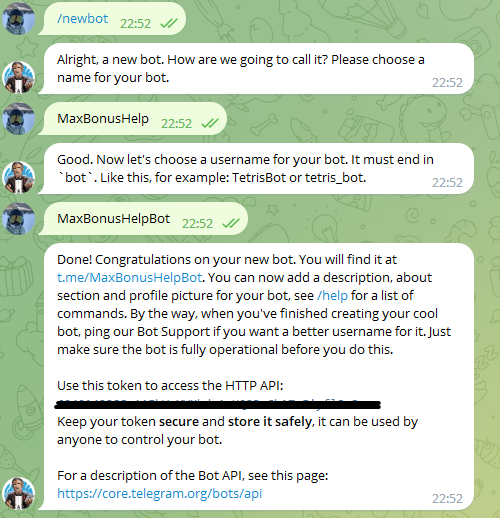
1. Создание и настройка бота в Telegram через BotFather: с помощью специального бота Telegram - BotFather, необходимо зарегистрировать нового бота, получить его токен и настроить основные параметры, такие как имя и описание. На рисунке .. представлено результат создания и настройки бота 

Рисунок .. – создание бота

1. Изучение Telegram Bot API и выбор библиотеки для языка программирования Go: перед началом работы с API, следует ознакомиться с его документацией и особенностями. Для упрощения работы с API на языке Go можно использовать библиотеку go-telegram-bot-api.
2. Реализация функционала для отправки уведомлений о новых заявках: после настройки бота и изучения API, следует разработать функционал для отправки уведомлений о новых заявках. Это может включать в себя создание отдельного обработчика, который будет вызываться при поступлении новой заявки в системе, и отправка сообщения с информацией о заявке сотрудникам отдела через чат-бот.

Для реализации отправки было принято решение добавить функцию sendTelegramNotification в handler «/api/task/add». Функция принимает токен чат-бота, чат id и сообщение. На рисунке .. представлена функция sendTelegramNotification.



Рисунок .. – функция sendTelegramNotification

После того, как заявка поступает, вызывается функция sendTelegramNotification и отправляет сообщение в группу. Ниже представлен результат работы.

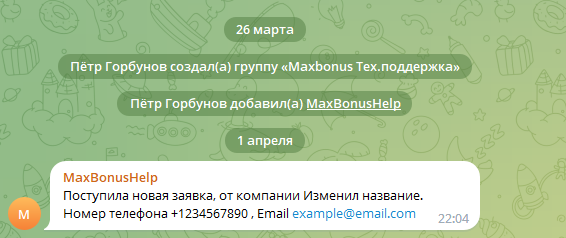


Рисунок .. – уведомление о поступлении заявки

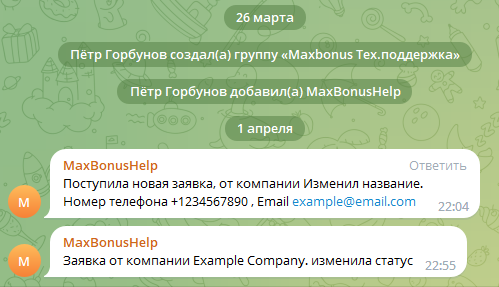
Также чат-бот уведомляет о изменении статуса заявки. 

Рисунок .. – изменение статуса заявки

При создании и настройке бота через BotFather [5], были получены подробные инструкции по регистрации нового бота и его настройке. Изучение Telegram Bot API [6] позволило получить общее представление о возможностях API и способах работы с ним. С использованием библиотеки go-telegram-bot-api [7] была упрощена разработка функционала чат-бота, так как библиотека предоставляет готовые методы и структуры для работы с API. Статья "Telegram Bots: An introduction for developers" [8] также оказала помощь в общем понимании принципов работы и создания ботов в Telegram.

Заключение

В заключении дипломной работы по автоматизации приема и обработки заявок отделом ООО «Максбонус» можно отметить следующие основные моменты.

1. В результате анализа потребностей отдела были выявлены ключевые аспекты и проблемы, которые могут быть решены с помощью автоматизации процессов.
2. Была проведена детальная разработка проекта, включая проектирование базы данных, создание серверной части на языке Go, разработку клиентской части на React, и интеграцию с Telegram для оповещений о новых заявках.
3. Были использованы современные технологии и инструменты, такие как Gin, DI Container, JWT Validation, PostgreSQL, а также различные библиотеки и источники, чтобы обеспечить качественную разработку и безопасность системы.
4. В ходе работы над проектом было проведено тестирование, оптимизация и документирование разработанного решения, что позволило создать эффективную и надежную систему для автоматизации приема и обработки заявок отделом ООО «Максбонус».
5. Реализованная система способна улучшить операционную эффективность отдела, уменьшить время обработки заявок и ошибки, а также повысить уровень удовлетворенности клиентов.
6. Разработанный проект может быть дополнен и развит в будущем, внедряя дополнительные функциональные возможности, интегрироваться с другими системами и адаптироваться под изменяющиеся условия бизнес-процессов компании.

В целом, автоматизация приема и обработки заявок отделом ООО «Максбонус» является важным шагом в оптимизации бизнес-процессов компании, что способствует увеличению ее конкурентоспособности и эффективности.

Список литературных источников предполает ссылки на него по тексту. Не хватает книг – монографий, статей, пособий, желательно – за последние 5 лет.

1. PostgreSQL Documentation: <https://www.postgresql.org/docs/>
2. NIST: Secure Hash Standard (SHS): https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
3. Go Documentation: crypto/sha256 package: https://golang.org/pkg/crypto/sha256/
4. OWASP: Password Storage Cheat Sheet: <https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html>
5. <https://github.com/gin-gonic/gin>
6. https://github.com/uber-go/dig,
7. <https://github.com/google/wire>,
8. <https://github.com/facebookarchive/inject>)
9. https://github.com/dgrijalva/jwt-go,
10. https://github.com/golang-jwt/jwt
11. BotFather. (n.d.). Telegram. Retrieved from https://core.telegram.org/bots#botfather
12. Telegram Bot API. (n.d.). Telegram. Retrieved from https://core.telegram.org/bots/api
13. go-telegram-bot-api. (n.d.). GitHub. Retrieved from https://github.com/go-telegram-bot-api/telegram-bot-api
14. Church, J. (2016, April 24). Telegram Bots: An introduction for developers. Medium. Retrieved from https://medium.com/@jonchurch/telegram-bots-an-introduction-for-developers-2ff2dfc86b6f

Приложение А

Функциональные требования к автоматизированной системе приема и обработки заявок отделом ООО «Максбонус» должны обеспечить эффективное выполнение ключевых бизнес-процессов и задач отдела. Ниже приведены основные функциональные требования к системе:

1. Управление заявками:
2. Создание новых заявок с вводом необходимой информации (например, данные клиента, описание проблемы или запроса, приоритет, срок исполнения);
3. Просмотр списка заявок с возможностью фильтрации по различным параметрам (статус, исполнитель, дата создания и т.д.);
4. Изменение статуса заявок (например, новая, в работе, выполнена, отклонена);
5. Назначение исполнителя на заявку;
6. Редактирование и обновление информации по заявкам;
7. Удаление заявок.
8. Управление пользователями:
9. Регистрация и аутентификация пользователей с использованием безопасных механизмов (например, хранение паролей в виде хешей SHA-256, использование JWT для аутентификации);
10. Просмотр списка пользователей с возможностью фильтрации по различным параметрам (роль, статус, дата регистрации и т.д.);
11. Редактирование информации о пользователях (имя, контактные данные, роль и т.д.);
12. Блокировка и разблокировка пользователей.
13. Интеграция с внешними системами и сервисами:
14. Оповещения о новых заявках и изменениях статуса существующих заявок через чат-бот Telegram;
15. Возможность интеграции с другими внутренними и внешними системами компании для обмена информацией и автоматизации процессов.
16. Отчетность и статистика:
17. Генерация отчетов по заявкам (например, количество заявок по статусам, среднее время обработки заявок, заявки по исполнителям);
18. Визуализация статистической информации в виде графиков и диаграмм для быстрого анализа и принятия решений.
19. Интерфейс и удобство использования:
20. Интуитивно понятный и простой в использовании интерфейс для пользователей различных уровней компетенции;
21. Адаптивный дизайн, обеспечивающий корректное отображение
22. Безопасность и доступность:
23. Разграничение прав доступа пользователей в зависимости от их ролей и уровня ответственности (например, администратор, менеджер, исполнитель);
24. Ведение журнала событий и аудита действий пользователей в системе для контроля и предотвращения злоупотреблений;
25. Реализация механизмов резервного копирования и восстановления данных для обеспечения сохранности информации и непрерывности работы системы;
26. Применение современных методов шифрования и защиты данных при их передаче и хранении.
27. Скорость и производительность:
28. Высокая производительность и быстрое время отклика системы даже при большом количестве одновременных пользователей и обработке больших объемов данных;
29. Оптимизация нагрузки на сервер и клиентскую сторону для обеспечения стабильной и надежной работы системы;
30. Масштабируемость системы для поддержки роста количества пользователей и обработки данных без существенных затрат на дополнительные ресурсы.

Эти функциональные требования обеспечат эффективное выполнение задач и бизнес-процессов отдела, улучшение качества работы с заявками, сокращение времени на их обработку и повышение уровня удовлетворенности клиентов. В результате автоматизации приема и обработки заявок отделом ООО «Максбонус» можно ожидать существенное повышение производительности и экономии ресурсов.

Приложение В

Техническое задание на разработку автоматизированной системы приема и обработки заявок отделом ООО «Максбонус»

1. Общие положения

1.1. Наименование системы: Автоматизированная система приема и обработки заявок отделом ООО «Максбонус».

1.2. Цель проекта: Создание автоматизированной системы для оптимизации и ускорения работы отдела по приему и обработке заявок.

1. Описание задачи и функционала системы

2.1. Регистрация и авторизация пользователей.

2.2. Прием и обработка заявок от клиентов.

2.3. Назначение исполнителей на заявки.

2.4. Отслеживание статуса заявок.

2.5. Интеграция с Telegram-ботом для оповещения сотрудников о новых заявках.

2.6. Управление пользователями и их ролями.

2.7. Ведение аудита и журнала событий.

1. Требования к системе

3.1. Многоуровневая архитектура с клиентской стороной на React, серверной стороной на Go и базой данных PostgreSQL.

3.2. Использование RESTful API для обмена данными между клиентом и сервером.

3.3. Мобильная адаптивность и кроссбраузерность.

3.4. Использование JWT для аутентификации пользователей и разграничения прав доступа.

3.5. Безопасность данных и шифрование паролей с использованием алгоритма SHA-256.

3.6. Резервное копирование и восстановление данных.

1. Этапы разработки

4.1. Анализ требований и составление технической документации.

4.2. Проектирование базы данных и серверной части.

4.3. Разработка клиентской части и пользовательского интерфейса.

4.4. Разработка Telegram-бота для оповещения сотрудников.

4.5. Интеграция компонентов системы и настройка сервера.

4.6. Тестирование системы, исправление ошибок и устранение недостатков.

4.7. Внедрение системы и обучение пользователей.

4.8. Техническая поддержка и сопровождение системы.

1. Приемка и гарантийные обязательства

7.1. Проведение приемочных испытаний с участием представителей заказчика и разработчиков.

7.2. Внесение корректировок и устранение замечаний, выявленных в ходе приемочных испытаний.

7.3. Подписание акта приемки и передачи системы заказчику.

7.4. Гарантийное сопровождение и техническая поддержка системы в течение 12 месяцев после передачи заказчику.

1. Риски и проблемы

9.1. Оценка рисков, связанных с разработкой и внедрением системы, таких как недостаточная квалификация персонала, изменение требований заказчика, технические проблемы, непредвиденные затраты.

9.2. Разработка плана по минимизации и управлению рисками, включая контроль сроков и бюджета, обеспечение качества разработки и соблюдение требований заказчика.

1. Контроль качества и соблюдение сроков

10.1. Внедрение процессов контроля качества на всех этапах разработки и внедрения системы, включая регулярное тестирование, аудит кода, проведение пользовательского тестирования.

10.2. Соблюдение сроков проекта, обес печение своевременного выполнения всех задач и этапов разработки, применение методологии Agile для гибкого управления проектом и оперативного реагирования на изменения требований заказчика.

1. Критерии успешности проекта

11.1. Создание функциональной и надежной системы, соответствующей требованиям заказчика и обеспечивающей эффективное управление процессами приема и обработки заявок.

11.2. Сокращение времени обработки заявок и увеличение производительности работы отдела.

11.3. Улучшение качества обслуживания клиентов и повышение уровня удовлетворенности их потребностей.

11.4. Внедрение системы в работу отдела без существенных затрат на обучение персонала и переход на новые процессы.

1. Заключение

Автоматизация приема и обработки заявок отделом ООО «Максбонус» позволит оптимизировать и улучшить процессы работы отдела, сократить время обработки заявок, повысить качество обслуживания клиентов и удовлетворенность их потребностей. Разработка и внедрение системы требует соблюдения ряда ключевых этапов, включая анализ требований, проектирование, разработку, тестирование, внедрение и сопровождение. Управление проектом на всех этапах разработки и внедрения обеспечит соблюдение сроков, контроль качества, минимизацию рисков и достижение поставленных целей.

Приложение C – где название? Лучше в соотв. параграф перенсти, как рисунок.

