**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,**

**СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Санкт-Петербургский колледж телекоммуникаций им. Э.Т. Кренкеля**

**Предметная (цикловая) комиссия** *Информатики и программирования в компьютерных системах*

Допустить к защите

Заместитель директора

по учебной работе

\_\_\_\_\_\_\_\_\_\_ Н.В.Калинина

14 июня 2023 г.

|  |  |  |  |
| --- | --- | --- | --- |
| **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА** | | | |
| Тема | **Разработка автоматизированной информационной** | | |
|  | **системы сервисного ИТ-центра** | | |
|  |  | | |
| Вид выпускной квалификационной работы | | | *Дипломный работа* |
|  | | | *(дипломный проект, дипломная работа)* |
|  | | | |
| Специальность | | *09.02.03 Программирование в компьютерных системах* | |
| Квалификация | | *Техник -программист* | |

|  |  |  |  |
| --- | --- | --- | --- |
| Дипломник | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Самойлов Е.А. |
|  |  | | 10 июня 2023 г. |
| Руководитель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Карелина И.В. |
|  |  | | 10 июня 2023 г. |
| Консультант по экономической части | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Корина М.Н. |
|  |  | | 10 июня 2023 г. |
| Консультант по охране труда | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Ожерельева Н.А. |
|  |  | | 10 июня 2023 г. |

Санкт-Петербург

2023

|  |  |  |
| --- | --- | --- |
| Работа написана мною и не содержит неправомерных заимствований | | |
| 10.06.2023 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *(дата)* | | *(подпись студента)* |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| Текст ВКР размещен в электронно-библиотечной системе университета | | |
| Руководитель отдела комплектования библиотеки | |  |
|  | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(ФИО)* |
| 10.06.2023 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *(дата)* | | *(подпись)* |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
| Коэффициент оригинальности ВКР | | \_\_\_\_\_\_\_ % |
|  | |  |
| Проверил: | Председатель П(Ц)К № 5 Кривоносова Н.В. | |
|  | | *(Должность, ФИО)* |
|  | |  |
| 10.06.2023 | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *(дата)* | | *(подпись)* |

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,**

**СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Санкт-Петербургский колледж телекоммуникаций им. Э.Т. Кренкеля**

**Предметная (цикловая) комиссия** *Информатики и программирования в компьютерных системах*

УТВЕРЖДАЮ

Заместитель директора

по учебной работе

\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Калинина

14 марта 2022 г.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ЗАДАНИЕ**  **на выполнение выпускной квалификационной работы** | | | | | | | | | |
|  | | | | | | | | | |
| Студент | | *Самойлов Евгений Александрович* | | | | | | группа | *К501* |
|  | | *(ФИО)* | | | | | |  |  |
| Руководитель | | | *Кривоносова Наталья Викторовна, преподаватель СПбКТ* | | | | | | |
|  | | | *(фамилия, имя, отчество, должность, уч. степень и звание)* | | | | | | |
| Э  Квалификация | | | *Техник-программист* | | | | | | |
| Вид работы | | | *Дипломная работа* | | | | | | |
|  | | | *(дипломный проект, дипломная работа)* | | | | | | |
| Тема | **Разработка автоматизированной информационной** | | | | | | | | |
| **системы** **сервисного ИТ-центра** | | | | | | | | | |
|  |  | | | | | | | | |
|  | | | | | | | | | |
| утверждена приказом ректора *№ 1473/кс от 14 марта 2022 г.* | | | | | | | | | |
|  | | | | | | | | | |
| Исходные данные: | | | | | *Техническое задание на разработку, документация отдела кадров,* | | | | |
| *Методология управления проектами, ГОСТ 7.32 – 2001* | | | | | | | | | |
|  | | | |  | | | | | |
|  | | | | | | | | | |
| Содержание работы: | | | | | | *обзор аналогов, проектирование, разработка и тестирование* | | | |
|  | | | *(анализ состояния проблемы, проведение исследований, разработка, расчеты параметров, экономическое обоснование и др.)* | | | | | | |
| *программного продукта, технико-экономическое обоснование разработки* | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | |  | | |
| Вид отчетных материалов, представляемых в ГЭК | | | | | | | *пояснительная записка, презентация, исходные коды программного продукта* | | |
| *(пояснительная записка, перечень, графического материала, отчет о НИР, технический проект, образцы и др.)* | | | | | | | | | |

**СОДЕРЖАНИЕ ВКР**

|  |  |  |
| --- | --- | --- |
| Наименование разделов | Объем выполнения в %  от всего задания | Срок выполнения |
| ВВЕДЕНИЕ | 7 | 20.04.2022 |
| 1. ПОСТАНОВКА ЗАДАЧИ | 8 | 30.04.2022 |
| 1. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ | 25 | 05.05.2022 |
| 1. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА | 22 | 17.05.2022 |
| 1. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ | 20 | 22.05.2022 |
| 1. МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ТРУДА | 5 | 28.05.2022 |
| 1. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА | 5 | 02.06.2022 |
| ЗАКЛЮЧЕНИЕ | 8 | 07.06.2022 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель | Карелина Ирина Владимировна | | | | | | | | |
|  | (фамилия имя, отчество) | | | | | | | | |
| Санкт-Петербургский колледж телекоммуникаций им. Э.Т. Кренкеля, тел. 333-18-28 | | | | | | | | | |
| (место работы, телефон) | | | | | | | | | |
| Начало выполнения ВКР | | | 20 апреля 2022 г. | | | | | | |
| Завершение выполнения ВКР | | | 8 июня 2022 г. | | | | | | |
| Представление работы на рецензию | | | 10 июня 2022 г. | | | | | | |
|  | | |  | | | | | | |
| Задание на выполнение ВКР рассмотрено на заседании предметной (цикловой) комиссии | | | | | | | | | |
| 8 февраля 2022 г. Протокол №6 | | | | | | | | | |
|  | | | |  | | | | | |
| Председатель предметной (цикловой) комиссии | | | | | | |  | Н.В. Кривоносова | |
|  | | | | | | |  |  | |
| Задание принял к исполнению | | 4 апреля 2022 г. | | | | | | |
|  | |  | | | |  | | |
|  | |  | | | подпись студента | | | |

**Оглавление**

[ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ: 7](#_Toc135833259)

[ВВЕДЕНИЕ 8](#_Toc135833260)

[1. ПОСТАНОВКА ЗАДАЧИ 9](#_Toc135833261)

[**1.1. Обзор аналогов** 9](#_Toc135833262)

[**1.2 SWOT-анализ** 15](#_Toc135833263)

[**1.3 PEST-анализ** 15](#_Toc135833264)

[**1.4 Формирование требований к программному продукту** 15](#_Toc135833265)

[**1.4.1 Бизнес-требования** 15](#_Toc135833266)

[**1.4.2 Пользовательские требования** 16](#_Toc135833267)

[**1.4.3 Функциональные требования к программному продукту** 17](#_Toc135833268)

[**1.4.4 Нефункциональные требования** 18](#_Toc135833269)

[**1.4.5 Ограничения** 18](#_Toc135833270)

[**1.4.6 Требования к интерфейсам** 19](#_Toc135833271)

[**1.4.7 Требования к данным** 20](#_Toc135833272)

[**1.5 Программные средства разработки** 23](#_Toc135833273)

[**1.6 Аппаратные средства разработки** 25](#_Toc135833274)

[2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ 26](#_Toc135833275)

[**2.1. Архитектура системы** 26](#_Toc135833276)

[**2.2. Моделирование основных сценариев системы** 26](#_Toc135833277)

[**2.3. Проектирование графического интерфейса пользователя** 38](#_Toc135833278)

[**2.4. Проектирование и разработка модели данных** 42](#_Toc135833279)

[3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 50](#_Toc135833280)

[**3. 1. Порядок разработки программных модулей** 50](#_Toc135833281)

[**3. 2. Структурное программирование** 50](#_Toc135833282)

[**3. 3. Реализация серверной части** 53](#_Toc135833283)

[**3. 4. Организация взаимодействия с БД** 54](#_Toc135833284)

[4. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 56](#_Toc135833285)

[**4.1. История изменений** 56](#_Toc135833286)

[**4.2. Терминология** 57](#_Toc135833287)

[**4.3. Стратегия тестирования** 58](#_Toc135833288)

[**4.4. Определение объектов тестирования** 58](#_Toc135833289)

[**4.5. Архитектура тестируемой системы** 60](#_Toc135833290)

[**4.6. Описание процесса тестирования** 61](#_Toc135833291)

[5. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ 63](#_Toc135833292)

[**5.1. Определение защищаемых информационных активов** 63](#_Toc135833293)

[**5.2. Вид и содержание персональных данных** 63](#_Toc135833294)

[**5.3. Модель угроз ПНД** 63](#_Toc135833295)

[**5.4. Возможные сценарии угроз** 63](#_Toc135833296)

[**5.5. Организация защиты данных в проекте** 64](#_Toc135833297)

[**5.6. Защита программного продукта** 64](#_Toc135833298)

[6. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ НА ПРОГРАММНЫЙ ПРОДУКТ 64](#_Toc135833299)

[**6.1. Руководство пользователя** 64](#_Toc135833300)

[**6.2. Руководство администратора** 64](#_Toc135833301)

[**6.3. Определение качества программного продукта по метрикам** 64](#_Toc135833302)

[7. МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ТРУДА 65](#_Toc135833303)

[8. ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА 68](#_Toc135833304)

[ЗАКЛЮЧЕНИЕ 68](#_Toc135833305)

[СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ 69](#_Toc135833306)

[ПРИЛОЖЕНИЕ 1. ЛИСТИНГ ПРОГРАММНЫХ МОДУЛЕЙ 70](#_Toc135833307)

[ПРИЛОЖЕНИЕ 2. ПОЛИТИКА БЕЗОПАСНОСТИ 70](#_Toc135833308)

[ПРИЛОЖЕНИЕ 3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 70](#_Toc135833309)

[ПРИЛОЖЕНИЕ 4. РУКОВОДСТВО АДМИНИСТРАТОРА 75](#_Toc135833310)

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ:**

**SQL — Structured Query Language (язык структурированных запросов) ОЗУ – оперативное запоминающее устройство.**

**UML — Unified Modeling Language (унифицированный язык моделирования).**

**ПК — персональный компьютер.**

**СУБД — система управления базами данных.**

**ПО — программное обеспечение.**

**ERD — диаграмма «сущность-связь».**

**API — интерфейс прикладного программирования.**

**Фреймворк — программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.**

**Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.**

**ЯП — язык программирования.**

**IDE (Integrated development environment) — внутренняя среда разработки.**

**ORM (Object-Relational Mapping) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».**

**АИС (Автоматизированная информационная система) - организационно-техническая система, состоящая из средств автоматизации определенного вида или нескольких видов деятельности людей и персонала, осуществляющего эту деятельность.**

**CRM (Система управления взаимоотношениями с клиентами) - прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками, в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов**

**ВВЕДЕНИЕ**

**Актуальность состоит в том, что в современных условиях работы предприятий возникает потребность автоматизировать процессы.** **В качестве примера: информационные системы позволяют людям, не отходя от рабочего места, узнать все необходимые данные о документации и наличии изделий, хранящихся на складе, и полностью, или частично, избавиться от бумажного документооборота, который имеет ограниченный срок службы, в отличие от электронных средств хранения данных.**

**Целью дипломной работы является разработка АИС, обладающей функционалом с достаточными показателями скорости работы, эффективности и удобства для администрирования и взаимодействия с сотрудниками, а также соответствующего всем действующим на момент написания требованиям к системе.**

* **Задачи, таким образом, подразделяются на:**
* **Формирование требований к программному продукту**
* **Проектирование и моделирование архитектуры, сценариев взаимодействия внутри проекта**
* **Непосредственно разработка и написание кода, разработка базы данных**
* **Тестирование системы**
* **Написание документации**

**1. ПОСТАНОВКА ЗАДАЧИ**

В современном бизнесе нельзя обойтись без технической поддержки в виде программ для ведения бизнеса, учета клиентов, поставщиков и поставок, а также учет продаж. Для удобства ведения коммерческих взаимоотношений, современные технологии предлагают использование онлайн-касс и автоматизированных информационных систем

## **1.1. Обзор аналогов**

Наиболее популярные CRM системы:

1. **AmoCRM** — система управления взаимоотношениями с клиентами, позволяющая автоматизировать продажи и организовать работу отдела маркетинга внутри небольшого предприятия. (Рис 1.1.)

Для менеджеров компании доступны:

* Удобный интерфейс для отслеживания целей и выполнения задач.
* Общение с клиентами непосредственно из CRM-карточки.
* Упрощенное ведение документооборота и запуск рассылок.
* Глобальный мессенджер для общения с сотрудниками внутри компании.

В свою очередь, для руководителя отдела продаж предусмотрены:

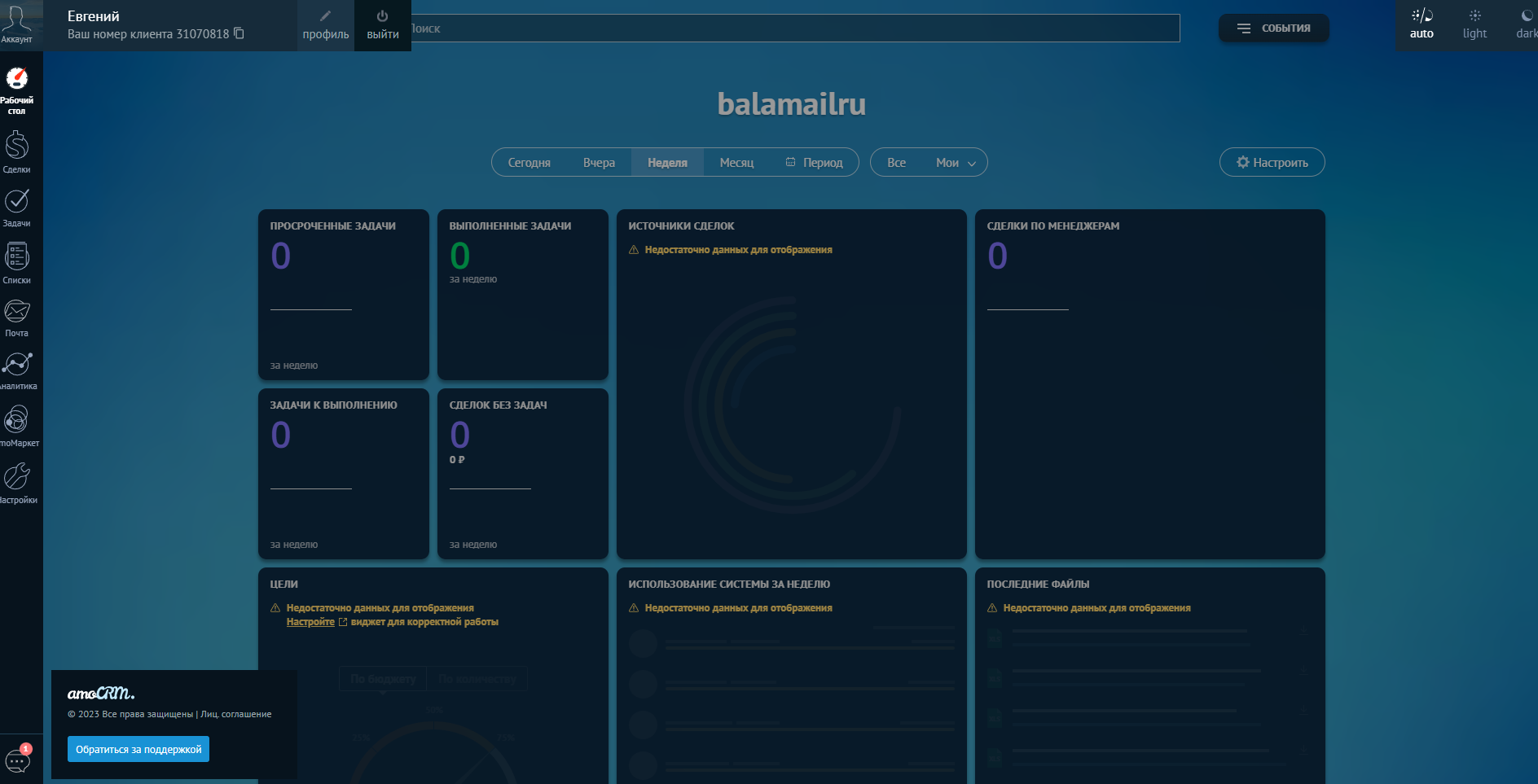
* Наглядная аналитика работы отдела.
* Настройка сложносоставных отчетов под нужды компании.
* Детализированный анализ работы колл-центра.
* Контроль над работой со сделками, в том числе и просроченными.
* Координирование работы менеджеров, распределение нагрузки на персонал.
* Планирование и разработка маркетинговой стратегии и настройка воронок продаж.

Рис. 1.1. AmoCRM

Функциональность программы определяется ее тарифным планом и количеством проведенных интеграций. AmoCRM предусматривает возможность тонкой настройки инструментария под требования компании благодаря открытому API и обширному каталогу приложений во встроенном магазине.

Основные инструменты системы:

* Рабочий стол
* Сделки
* Задачи и цели
* Списки(контакты)
* Почта
* Аналитика
* Настройка

1. **CRM Мегаплан** — мощная система для управления бизнесом с расширенными возможностями в области проектов и заказов. Продукт позволяет автоматизировать внутренние и внешние процессы, наладить взаимоотношение с клиентами, а также управлять и координировать персонал.

Мегаплан — СРМ-система, созданная в первую очередь для управления проектами и обеспечения контроля над сотрудниками. Инструментарий направлен на организацию труда, оптимизацию рутинных обязанностей и автоматизацию бизнес-процессов. (Рис 1.2.)

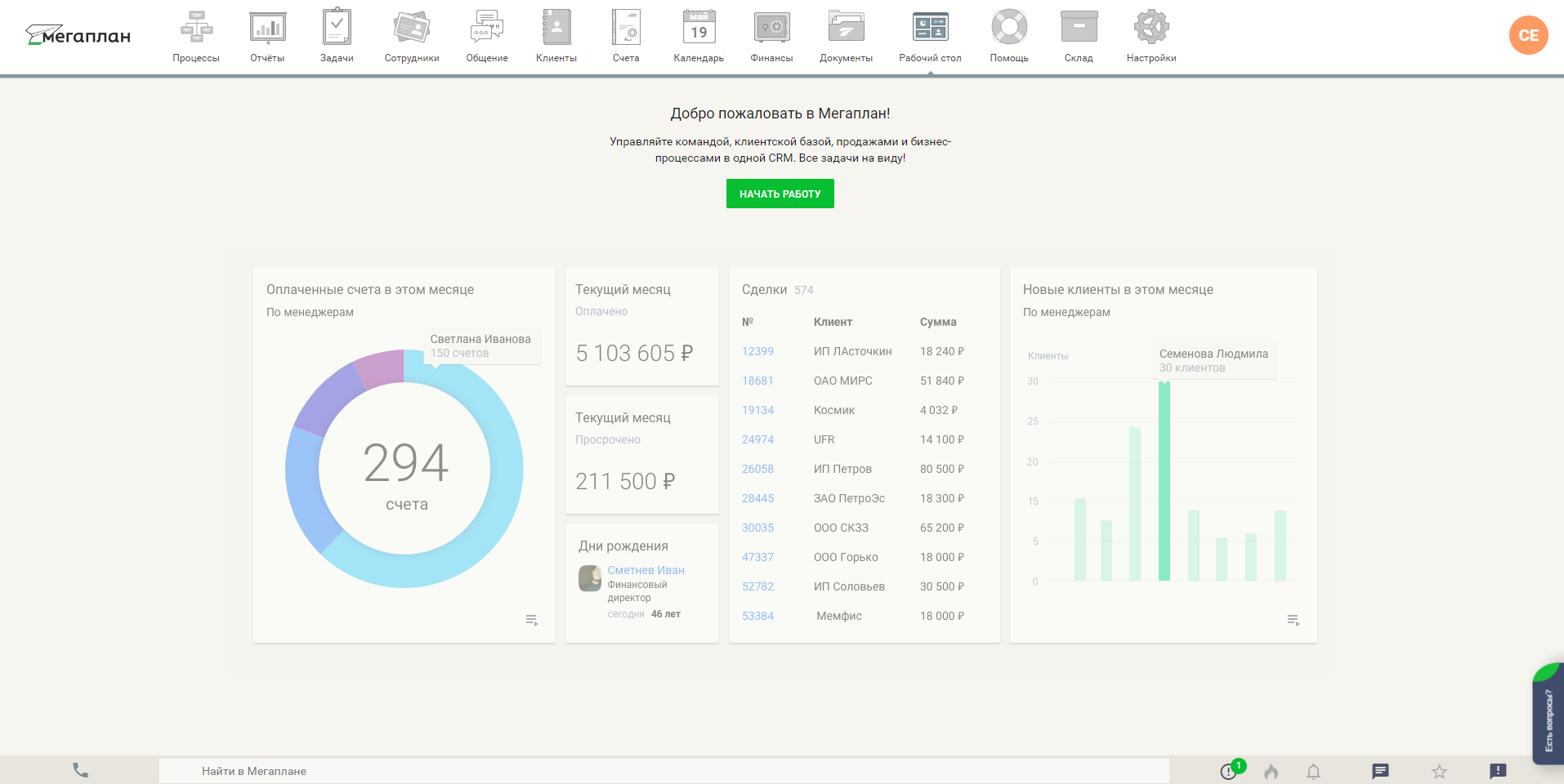


Рис. 1.2. Мегаплан

Задачи предприятия и онлайн-офис

Основной инструментарий направлен на проектную работу и постановку заданий, что позволяет полностью организовать бизнес-процессы предприятия с большим штатом. Система Мегаплан формирует единое рабочее пространство, где объединены:

* Рабочий
* Список заданий.
* Информер-панель
* Облачное хранилище

Департаменты в программе нужны, чтобы организовать все отделы предприятия, а также ограничить доступ и защитить конфиденциальные данные. Кроме того, все версии Мегаплан поддерживают мобильное приложение, через которое удобно контролировать дела, назначать ответственных и координировать подчиненных.

Управление продажами

В старших версиях предусмотрена CRM-система Мегаплан, позволяющая организовать отдел продаж и контакт с покупателями. Софт интегрируется с сайтом или интернет-магазином, предусматривает подключение телефонии и почты, а также все популярные мессенджеры или соцсети.

Основные инструменты:

* Воронка продаж
* База заказчиков и контрагентов
* Обработка счетов
* База сделок
* Учет движения финансов

1. **Salesforce** − это CRM-система, которая предназначена для управления самыми разными видами бизнес-процессов, отношениями с клиентами, аналитикой, маркетингом, продажами и прочее. (Рис 1.3.)

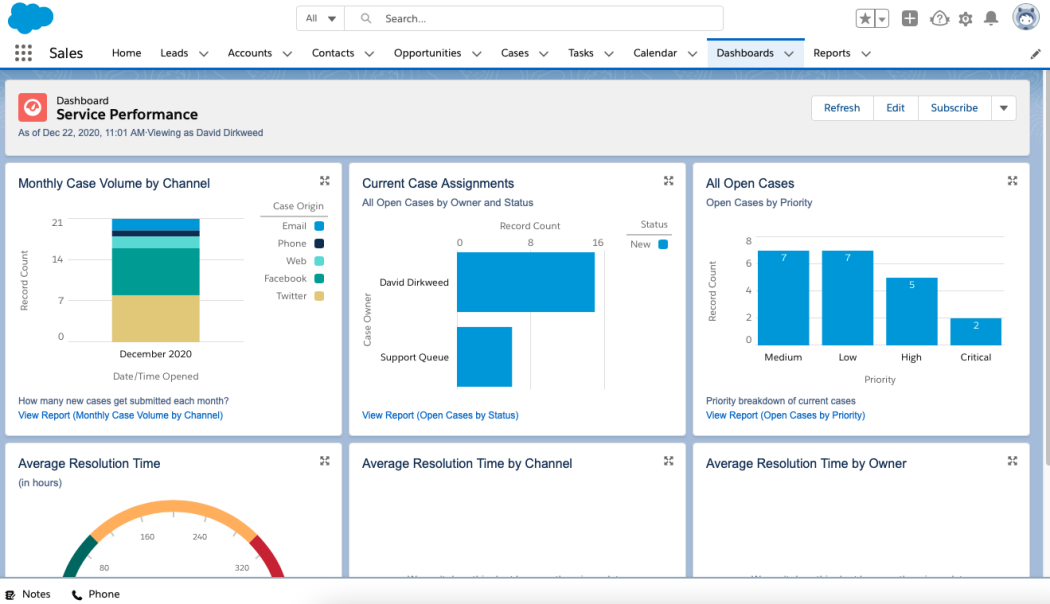


Рис. 1.3. Salesforce

Работает Salesforce CRM по модели SaaS (форме вычислений в облаке). Пользователям CRM-системы предоставляется готовое программное обеспечение, доступ к которому открытый через браузер или мобильное приложение. CRM-система ориентирована на малый, средний и крупный бизнес. Предлагает все необходимое для поиска, удержания клиентской базы, совершения сделки и прочее.

Предоставляет Salesforce в России следующие возможности:

* управление контактами;
* лидогенерация;
* прогнозирование;
* ведение отчетов;
* сотрудничество;
* автоматизация рабочего процесса;
* мобильность доступа.

Управление продажами автоматизировано комплексно, поэтому директор может получить отчет о продажах в любое удобное время, а продавец имеет возможность совершать сделки удаленно, используя смартфон. Система интегрирована с различными решениями по телефонии.

Таблица 1.1. Сравнение аналогов

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | AmoCRM | Мегаплан | Salesforce |
| Управление задачами и проектами | + | + | + |
| Управление сотрудниками | + | + | + |
| Отчеты и аналитика | + | + | + |
| Склад | + | + | + |
| Товары и услуги | + | + | + |
| Цены | Базовый - 499₽  Расширенный - 999₽  Профессиональный - 1499₽ | Базовый - 329₽  Расширенный - 599₽  Профессиональный - 839₽ | Lightning Professional - $75  Lightning Enterprise − $150  Lightning Unlimited − $300 |

Вывод:

Анализируя таблицу полученную таблицу, можно сопоставить результат для реализации АИС, где будут включены все удобства в функциональности и их потребности. Тем самым, система, которая будет разработана, будет иметь множество плюсов для использования в предприятии.

## **1.2 SWOT-анализ**

SWOT - анализ показывает, какие сильные и слабые стороны могут быть у сервиса во внутренней и внешней среде – Таблица 1.2. SWOT-анализ.

Таблица 1.2. SWOT - анализ

|  |  |  |
| --- | --- | --- |
|  | Положительное влияние/Сильные стороны | Отрицательное влияние/Слабые стороны |
| Внутренняя среда | Цены на услуги более низкие чем у конкурентов | Потребность в расходах на «раскрутку», для узнаваемости системы |
| Внешняя среда | Возможность быстрого роста благодаря всем нужным функционалам | Высокая конкуренция, многим пользователем привычнее пользоваться другими системами |

## **1.3 PEST-анализ**

Из PEST-анализа (Таблица 1.3.) понятно, какие риски могут быть у сервиса в политической, экономической, социальной и технологической сферах.

Таблица 1.3. PEST-анализа

|  |  |
| --- | --- |
| Political  Может быть ужесточение ограничений, что приведет к ограничению функционала системы | Economic  Из-за роста курса валют может не хватать средств для поддержания функционирования системы |
| Social  Центры, использующие другие системы могут создать мнение о недостаточности функционала. | Technological  Развитие систем конкурентов может привести к потери актуальности |

## **1.4 Формирование требований к программному продукту**

**1.4.1 Бизнес-требования**

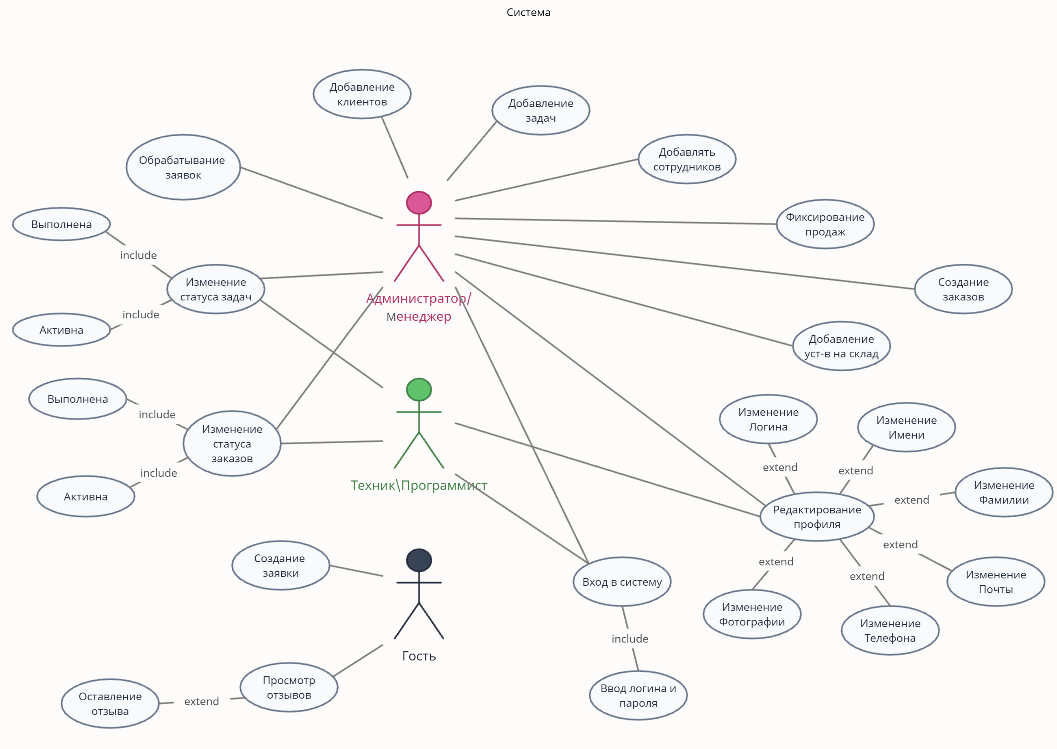
В настоящее время вся информация о принятых заказах хранится в электронных таблицах Microsoft Excel, что существенно затрудняло одновременную работу двум и более пользователям с данными, и тем самым отрицательно влияло на динамику работы сотрудников в условиях дефицита рабочего времени.

Таким образом, большое количество заявок и длительное время обработки может привести к множеству проблем: неправильно введенные данные или их потеря, дублирование информации, невозможность отслеживания хода ремонта, так как невозможно быстро найти нужную информацию, а клиентам сложнее узнать состояние ремонта. Точно так же ручная документация увеличивает вероятность пропуска важных моментов при приеме ремонта. Не исключены случаи подделки документации.

Без централизованного сбора информации труднее получить информацию о конкретном клиенте, а руководству трудно понять, какой объем работы проделал сотрудник за определенный период времени. АИС позволяет решить все эти проблемы.

### **1.4.2 Пользовательские требования**

Диаграмма вариантов использования (use-case) в UML — диаграмма, показывающая отношения между “актерами” и прецедентами, и являющаяся составной частью общей модели прецедентов пользования, позволяющей описать систему на концептуальном уровне. Построенная в рамках проектирования дипломной работы диаграмма для роли всех типов (“Гость”, “Техник” или “Программист”, “Администратор” или “Менеджер”). На рисунке 1.1 изображена диаграмма вариантов использования (диаграмма use-case)

erРис. 1.4. Диаграмма use-case

### **1.4.3 Функциональные требования к программному продукту**

Приложение для учёта заявок строиться на основании нескольких функциональных ролей:

* Администратор\Менеджер – авторизованный сотрудник, наделенный правами администратора, которому доступны все возможности
* Техник\Программист – авторизованный сотрудник, который имеет ограниченный спектр функционала для своей работы
* Гость\Клиент – неавторизованный пользователь, который может оставлять и просматривать отзывы, а также писать заявки

Система должна обладать следующими функциями:

* Авторизация в системе с использованием логина и пароля;
* Регистрация новых сотрудников;
* Авторизация в админ-панели;
* Добавление и обработка заявок;
* Добавление и редактирование задач, заказов, клиентов, сотрудников и продаж;
* Экспорт;
* Управление учетной записью
* Просмотр справки о странице

Перечень функций и use-case диаграмма приведены в пунктах 1.1.3.1 и 1.1.2.

Распределение функций по ролям указано в use-case диаграмме.

### **1.4.4 Нефункциональные требования**

Данные должны передаваться между серверной и клиентской частью приложения по протоколу HTTPS в формате JSON.

Для работы с клиентской частью web-приложения необходимо современный браузер, с поддержкой HTML5, CSS 3 и ECMAScript 2015 (ES6). Например:

* Google Chrome;
* Yandex браузер;
* Opera;
* Safari.

### **1.4.5 Ограничения**

Ограничения функциональности:

* Только администратор может зайти в Админ-панель;
* Гость никак не может попасть в систему;
* Техник или программист имеют ограниченный функционал;

Ограничения на ввод:

Имя пользователя может в себе содержать:

* Латиницу;
* Цифры;
* Знаки ! # $ % & ' \* + — / =? ^ \_ ` { | } ~
* Точку, за исключением первого и последнего знака, которая не может
* повторяться.

Ограничения на ввод текста:

* Длина заявки должна быть не менее 1 и не более 300 символов;
* Длина отзыва должна быть не менее 1 и не более 150 символов;
* Длина описания заказа должна быть не менее 1 и не более 1000 символов;
* Длина описания задачи должна быть не менее 1 и не более 1000 символов;

Ограничения на ввод логина:

* Длина должны быть не менее 3 и не более 40 символов.

Ограничения на ввод пароля:

* Длина должны быть не менее 8 символов;
* Должен содержать хотя бы одну прописную букву;
* Должен содержать хотя бы одну заглавную букву;
* Должен содержать хотя бы одну цифру.

### **1.4.6 Требования к интерфейсам**

Для хранения информации нужен сервер базы данных. Сведения должны выбираться, изменяться, добавляться с помощью API с использованием ЯП Python и фреймворка Django.

Также соблюдаются определенные стилевые установки:

Цветовая гамма программного продукта белый и темно-серо-синий;

Большая часть элементов управления имеет анимацию для визуальной наглядности;

Шрифт: Trebuchet MS;

Цвет шрифта: светло-серый, белый, черный, темно-бордовый, #252930, #eb4f37, #44cc66

Размер шрифта на сайте– от 20 px до 40 px;

### **1.4.7 Требования к данным**

Данные, которые будет вносить пользователь должны соблюдать тип данных, которому он соответствует в базе данных:

1. Для таблицы пользователей:

* id — уникальный идентификатор пользователя.
* first\_name, last\_name — поля, предназначенные для хранения имени и фамилии пользователя.
* username — поле, предназначенное для хранения логина пользователя. Это необходимо для идентификации пользователя во время входа в приложение.
* email — поле почты, предназначенное для хранения электронной почты пользователя.
* password — поле, предназначенное для хранения пароля пользователя. Это необходимо для идентификации пользователя во время входа в приложение.
* user\_phone — поле, предназначенное для хранения телефона пользователя.
* user\_note — поле, предназначенное для хранения примечаний о пользователе.
* user\_time — поле, предназначенное для хранения даты начала работы пользователя.
* user\_photo — поле, предназначенное для хранения Фото профиля пользователя.
* user\_role — поле выбора, предназначенное для хранения должности пользователя. (“Администратор”, “ Техник”, “ Программист”, “ Менеджер”).

1. Для таблицы задач:

* id — уникальный идентификатор задачи.
* task\_name— поле, предназначенное для хранения названия задачи.
* task\_description — поле, предназначенное для хранения описания задачи.
* task\_time— поле времени, предназначенное для хранения крайнего срока.
* task\_active— поле выбора, предназначенное для хранения статуса задачи (“Активна”, “Выполнена”.)

1. Для таблицы клиентов:

* id — уникальный идентификатор клиента.
* client\_type— поле выбора, предназначенное для хранения типа клиента (“Физ. Лицо”, “ Компания”).
* client\_name— поле, предназначенное для хранения имени клиента.
* client\_phone— поле, предназначенное для хранения номера телефона клиента.
* client\_email— поле, предназначенное для хранения почты клиента.
* client\_address— поле времени, предназначенное для адреса клиента.
* client\_found — поле выбора, предназначенное для хранения информации (“Знакомые”, “ Интернет”, “ Реклама”).

1. Для таблицы заказов:

* id — уникальный идентификатор заказа.
* order\_name— поле, предназначенное для хранения названия заказа.
* order\_description — поле, предназначенное для хранения описания заказа.
* order\_time — поле времени, предназначенное для хранения крайнего срока.
* order\_active — поле выбора, предназначенное для хранения статуса задачи (“Активна”, “Выполнена”.)
* order\_price — поле, предназначенное для хранения стоимости заказа.
* order\_client — поле выбора, предназначенное для хранения клиента.

1. Для таблицы склада:

* id — уникальный идентификатор вещи.
* inventory\_type— поле выбора, предназначенное для хранения типа вещи (“Устройство”, “ Запчасти”).
* inventory\_article— поле, предназначенное для хранения артикула.
* inventory\_name — поле, предназначенное для хранения названия вещи.
* inventory\_color — поле, предназначенное для хранения цвета.
* inventory\_condition — поле, предназначенное для хранения состояния.
* inventory\_description — поле, предназначенное для хранения описания.
* inventory\_client — поле, предназначенное для хранения владельца.

1. Для таблицы продаж:

* id — уникальный идентификатор продажи.
* sales\_name— поле, предназначенное для хранения названия продажи.
* sales\_comment — поле, предназначенное для хранения описания продажи.
* sales\_time — поле времени, предназначенное для хранения времени продажи.
* sales\_price — поле, предназначенное для хранения цены.
* sales\_client — поле выбора, предназначенное для хранения клиента.

1. Для таблицы отзывов:

* id — уникальный идентификатор отзыва.
* review\_fio — поле, предназначенное для хранения ФИО.
* review\_gender — поле выбора, предназначенное для хранения пола ('Мужчина', 'Женщина').
* review\_email — поле, предназначенное для хранения почты.
* review\_comment — поле, предназначенное для хранения отзыва.
* review\_grade — поле выбора, предназначенное для хранения оценки ('Отлично!', 'Очень хорошо', 'Хорошо', 'Плохо', 'Очень плохо').

1. Для таблицы заявок:

* id — уникальный идентификатор заявки.
* application\_fio — поле, предназначенное для хранения ФИО.
* application\_phone — поле, предназначенное для хранения телефона
* application\_email — поле почты, предназначенное для хранения электронной почты
* application\_address — поле, предназначенное для хранения адреса
* application\_found — поле выбора, предназначенное для хранения информации (“Знакомые”, “ Интернет”, “ Реклама”).
* application\_comment — поле, предназначенное для хранения заявки
* application\_active — поле выбора, предназначенное для хранения статуса заявки (“Активна”, “Выполнена”.)

### **1.5 Программные средства разработки**

Работа была выполнена с использованием следующих средств разработки программных продуктов:

1. PyCharm — это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для графических отладчиков и работы с кодом. Продукт доступен в двух версиях: PyCharm Community Edition - бесплатная версия, находится под лицензией Apache License, и PyCharm Professional Edition - расширенная версия продукта, обладающая дополнительной функциональностью, является проприетарным ПО.

Возможности:

• Отладка кода при помощи PyDev;

• Рефакторинг кода

• Поддержка Git, SVN, Mercurial и других систем контроля версиями;

• Автодополнение кода

1. Django

В разработке был использован популярный фреймворк Django языка Python. В отличие от подобных фреймворков, обработчики ссылок в Django конфигурируются с помощью регулярных выражений.

Для работы с базой данных Django использует собственный ORM, в котором модель описывается классами Python, и по ней генерируется схема БД.

1. SQLite — компактная (тесно связанная с программой и работающая, не требуя профессионального администрирования) встраиваемая СУБД.

Сама библиотека написана на ЯП “C”; существует большое количество привязок к другим языкам программирования, в том числе “Apple Swift”, “Delphi”, “C++”, “Java”, “C#”, “VB.NET”, “Python”, “Perl”, “PHP”, “PureBasic”, “Tcl”, “Ruby”, “Haskell”, “Scheme”, а также ко многим другим.

1. HTML (HyperText Markup Language) — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Элементами HTML являются строительные блоки на страницах. С помощью HTML разные конструкции, изображения и другие объекты, такие как интерактивная веб-форма, могут быть встроены в отображаемую браузером страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов.
2. CSS (Cascading Style Sheets) — формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). Также может применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

1. JS (JavaScript) — ­­­ мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений.

### **1.6 Аппаратные средства разработки**

В ходе разработки дипломной работы использовались персональные компьютеры на базе архитектуры x64 со следующими характеристиками:

* Стационарный ПК следующей конфигурации:
* Процессор AMD Ryzen 5 2600 Six-Core Processor 3.40 GHz
* Видеоадаптер NVIDIA GeForce GTX 1060 объемом 3GB
* Объем оперативной памяти DDR3 объемом 16 ГБ
* Объем магнитного накопителя (HDD) 500 ГБ
* Операционная система Windows 10.

**Вывод первой главы:**

Проведенное исследование позволило сделать следующее:

* Сформулировать требования к системе;
* Описать назначения программного продукта;
* Определить пользователей системы (их тип, роли, категории);
* Проанализировать входные и выходные сообщения системы.

**2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ**

**2.1. Архитектура системы**

В ходе разработки была выбрана трехуровневая архитектура, состоящая из:

* Уровня представления — представляет из себя интерфейс пользователя, который работает через браузер. На этом уровне располагается простейшая бизнес-логика.
* Уровня приложения — на этом уровне реализуется основная бизнес-логика программного продукта;
* Уровня данных — состоит из базы данных с, непосредственно, данными, находящимися в ней.

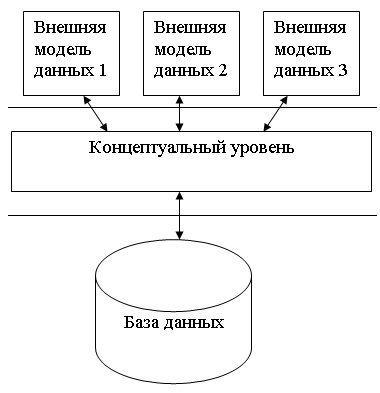


Рис. 2.1. Трехуровневая архитектура системы

**2.2. Моделирование основных сценариев системы**

На рис. 2.2 и 2.3 изображены схемы IDEF0 первого, второго уровня соответственно, отображающие основные функции системы.

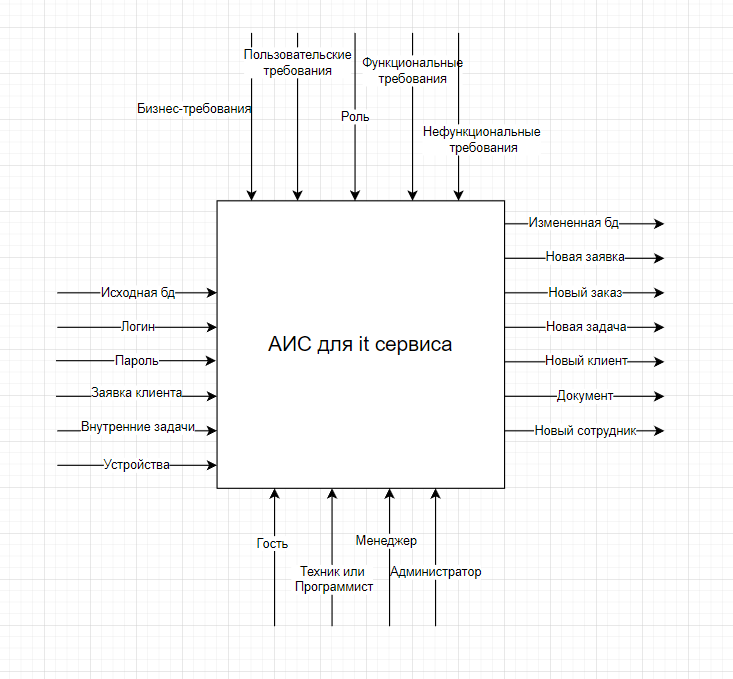


Рис. 2.2. Диаграмма IDEF0 первого уровня для разрабатываемого ПО

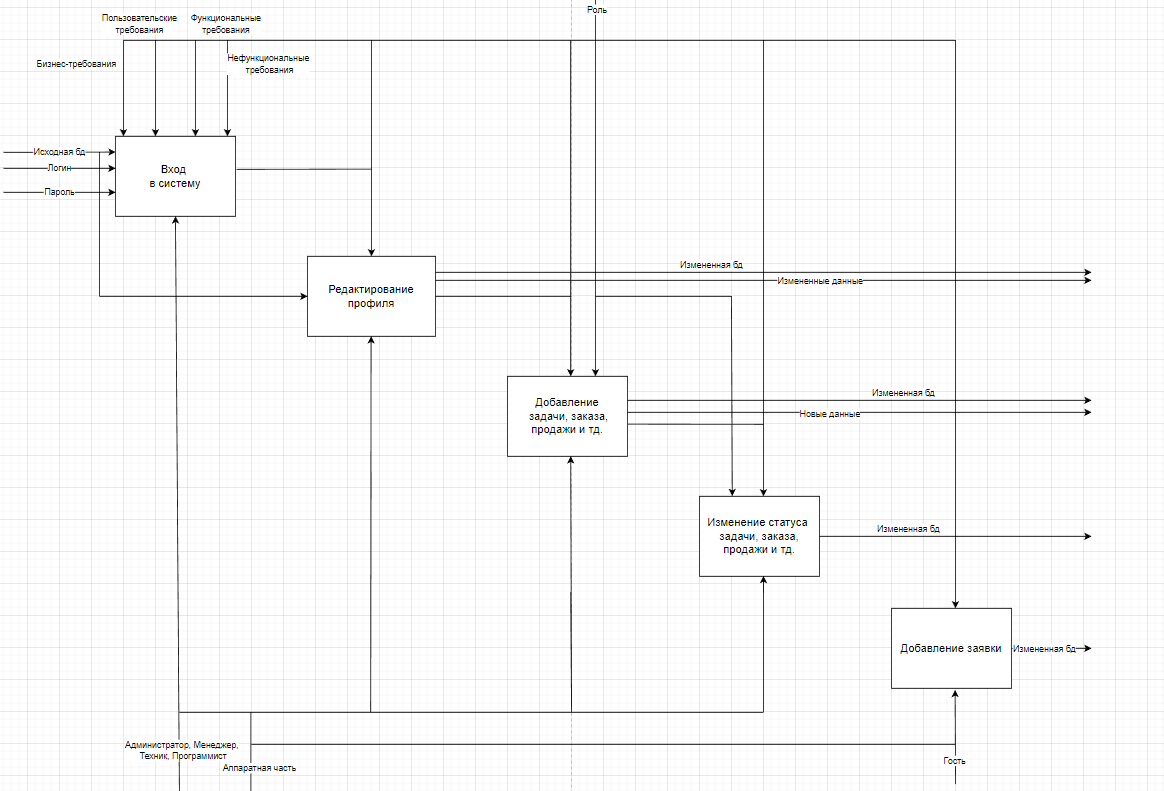


Рис. 2.3. Диаграмма IDEF0 второго уровня для разрабатываемого ПО

Так же была разработана диаграмма третьего уровня для блока “Добавление продажи” изображенная на рис. 2.4.

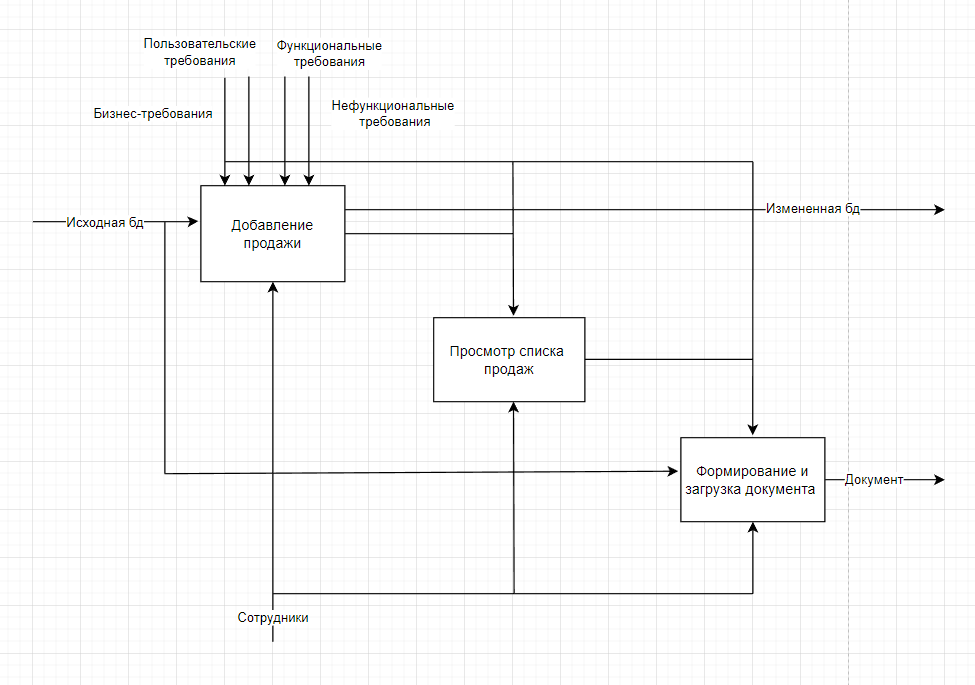


Рис. 2.4. Диаграмма IDEF0 третьего уровня для добавления продажи и формирования отчетного файла Excel

По диаграммам видно, что основной функционал приложения – это инвентаризация оборудования. В свою очередь для реализации данной задачи приложение содержит ряд функций: авторизация, редактирование профиля, добавление и изменения статуса записей.

Глобально ограничениями приложения выступают: бизнес-требования, пользовательские, функциональные и нефункциональные требования.

В системе реализуется 5 ролей: Администратор, Менеджер, Техник, Программист и Гость. Различие между ролями заключается в доступности функционала.

Диаграмма последовательности — это UML-диаграмма, на которой для некоторого набора объектов на временной оси показан жизненный цикл объекта и взаимодействие “актеров” информационной системы в рамках конкретного прецедента.

На рисунке 2.5. изображена диаграмма последовательности для прецедента функции «Добавление нового сотрудника».

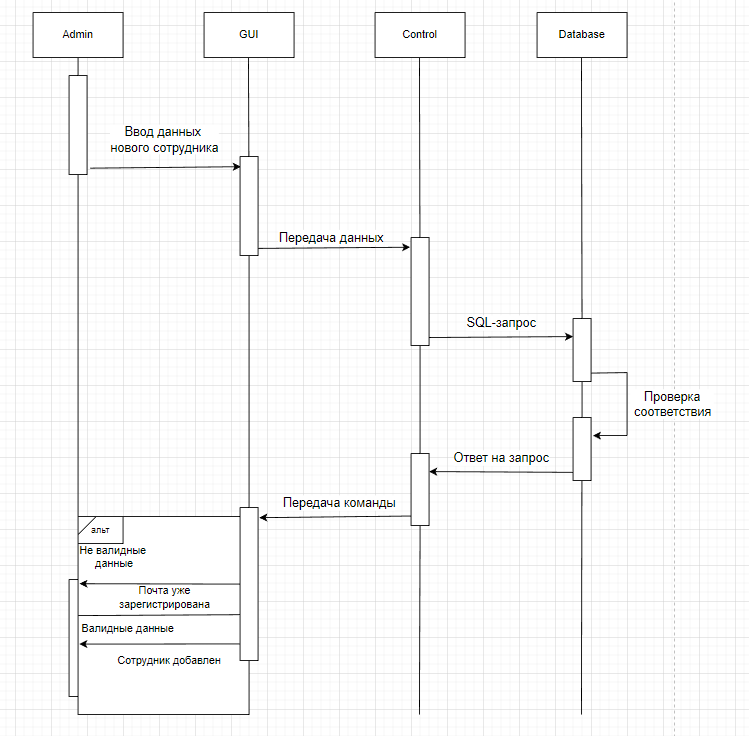


Рис. 2.5. Диаграмма последовательности «Добавление нового сотрудника»

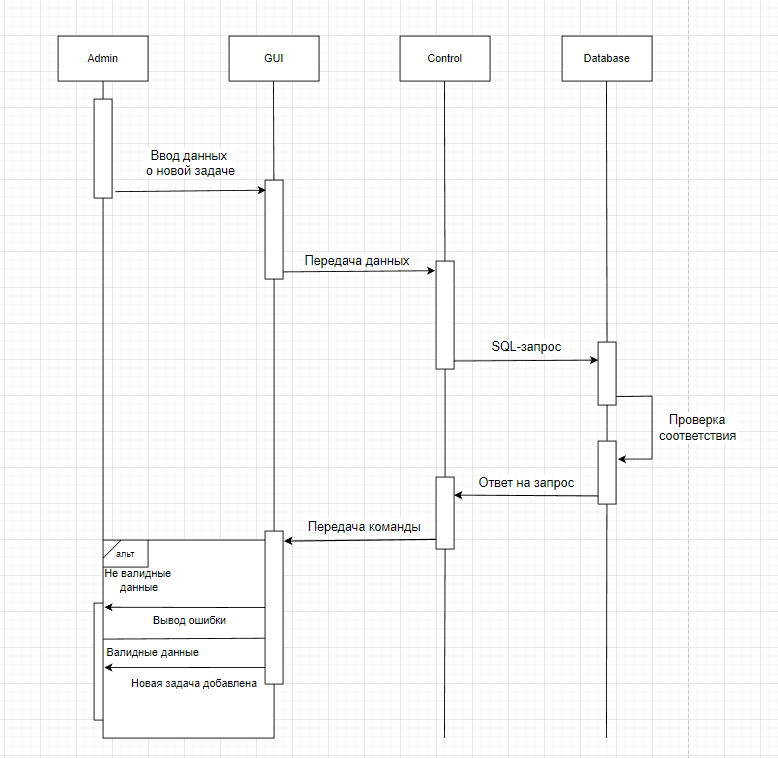
На рисунке 2.6. изображена диаграмма последовательности для прецедента функции «Добавление новой задачи». 

Рис. 2.6. Диаграмма последовательности «Добавление новой задачи»

На рисунке 2.7. изображена диаграмма последовательности для прецедента функции «Добавление нового заказа».

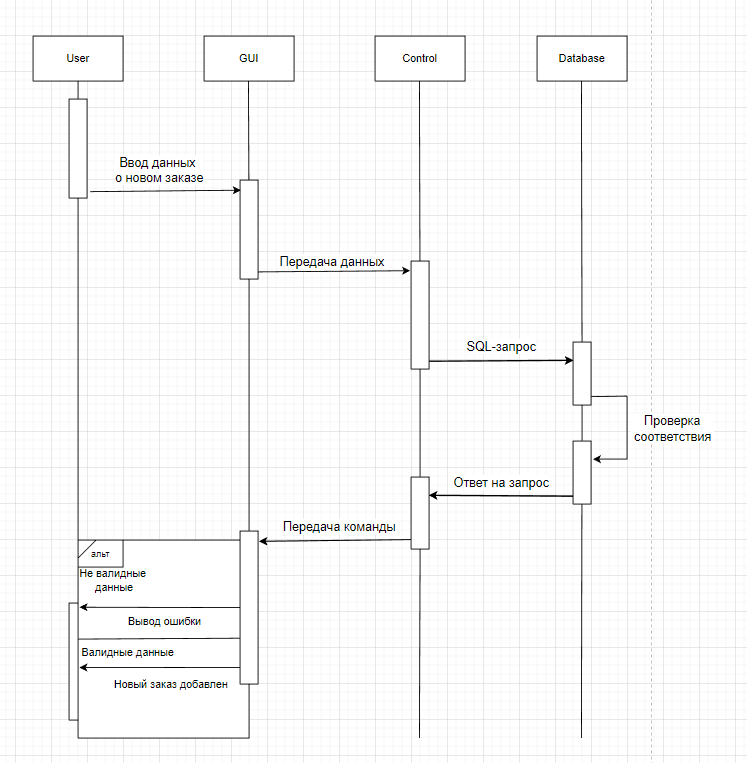


Рис. 2.7. Диаграмма последовательности «Добавление нового заказа»

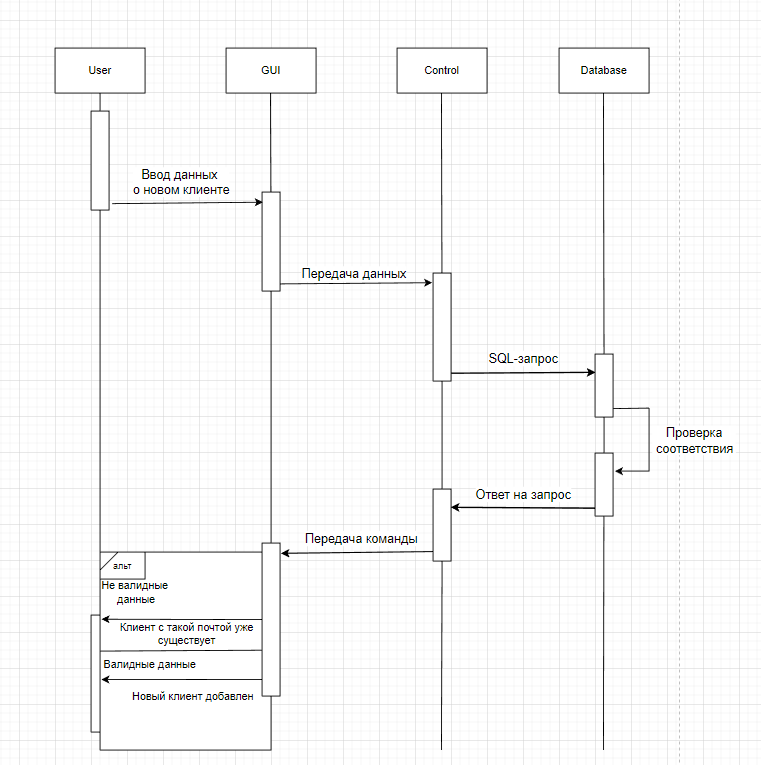
На рисунке 2.8. изображена диаграмма последовательности для прецедента функции «Добавление нового клиента». 

Рис. 2.8. Диаграмма последовательности «Добавление нового клиента»

На рисунке 2.9. изображена диаграмма последовательности для прецедента функции «Изменение личных данных».

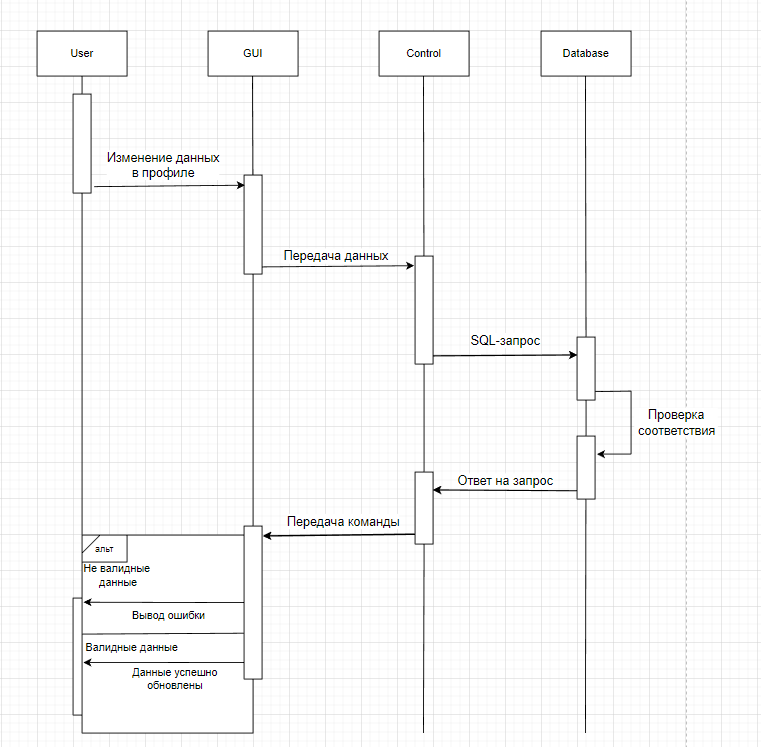


Рис. 2.9. Диаграмма последовательности «Изменение личных данных»

Диаграмма активностей (видов деятельности) — диаграмма, отражающая динамические моменты поведения проектируемой системы. На ней наглядным образом показан процесс перехода потока управления от одного компонента к другому.

На рисунке 2.10 изображена диаграмма активностей для прецедента «Добавление задачи». На диаграмме представлены следующие объекты: Администратор, Система и База данных.

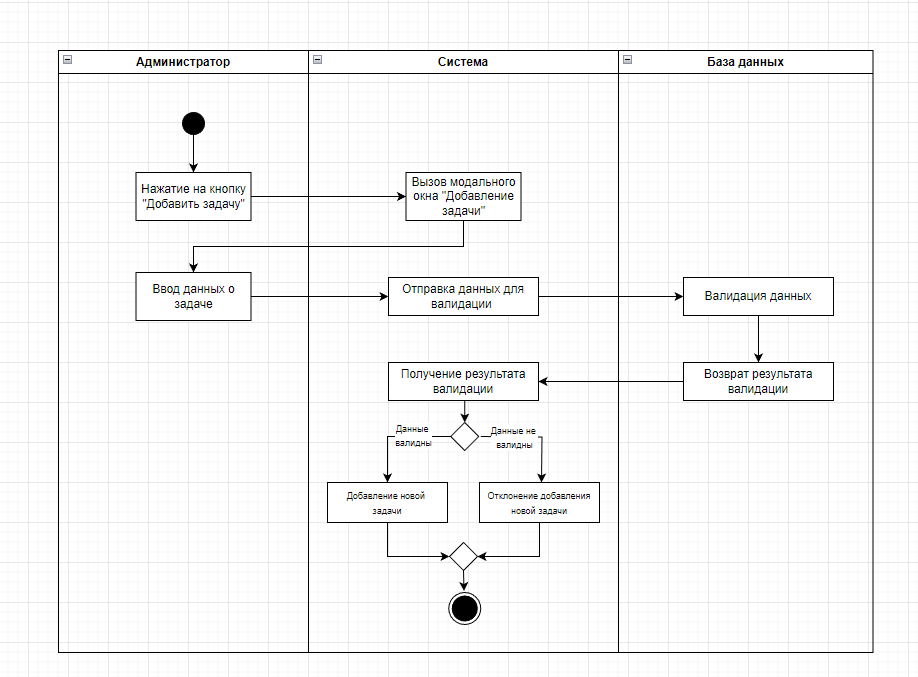


Рис. 2.10. Диаграмма активностей " Добавление задачи ".

На рисунке 2.11 изображена диаграмма активностей для прецедента «Добавление нового сотрудника». На диаграмме представлены следующие объекты: Администратор, Система и База данных.

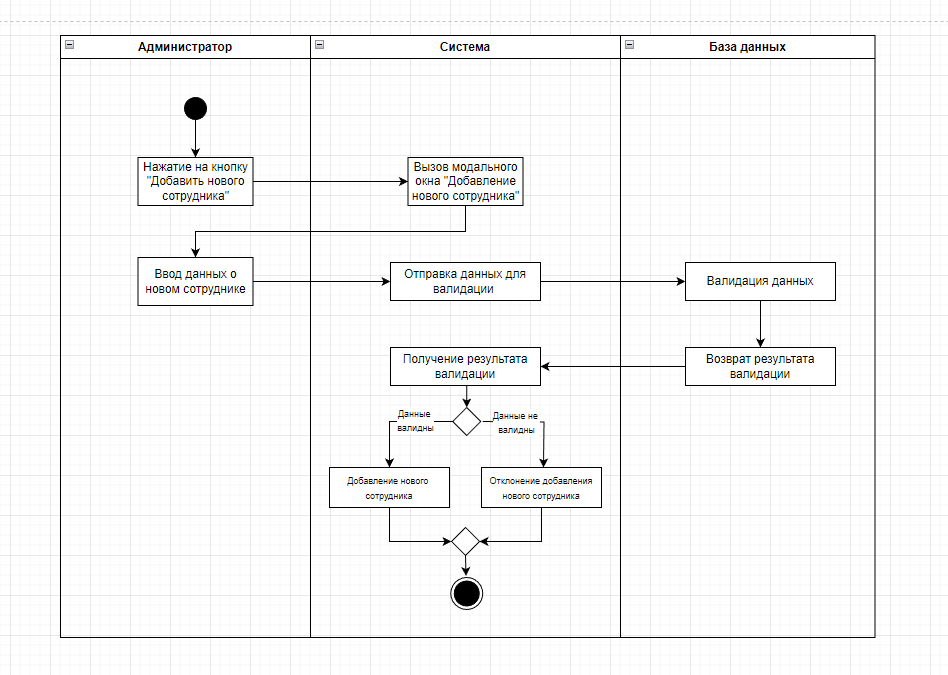


Рис. 2.11. Диаграмма активностей «Добавление нового сотрудника".

На рисунке 2.12 изображена диаграмма активностей для прецедента «Редактирование профиля». На диаграмме представлены следующие объекты: Сотрудник, Система и База данных.

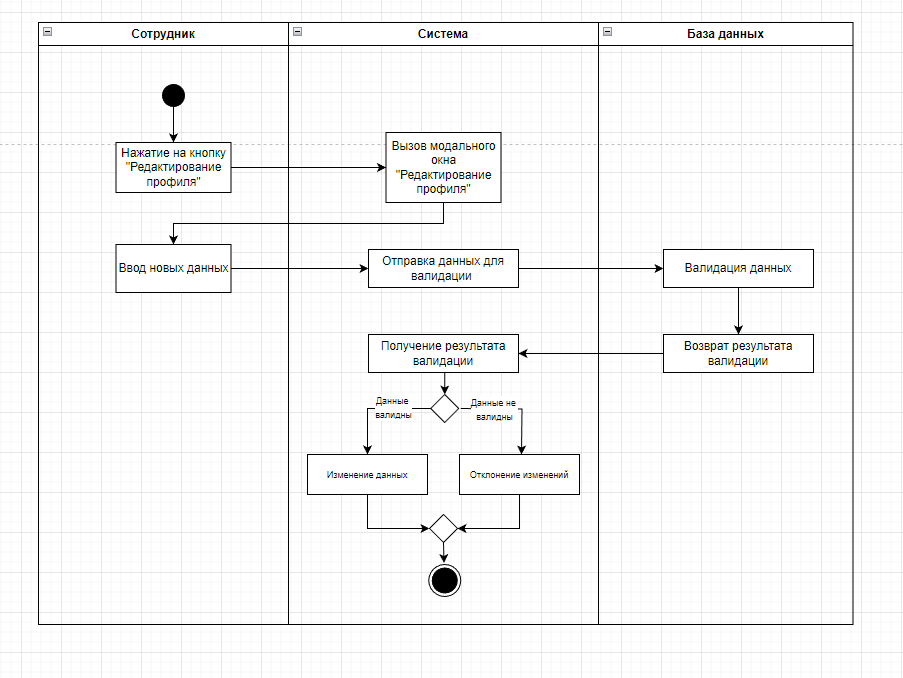


Рис. 2.12. Диаграмма активностей "Редактирование профиля".

На рисунке 2.13 изображена диаграмма активностей для прецедента «Авторизация». На диаграмме представлены следующие объекты: Сотрудник, Система и База данных.

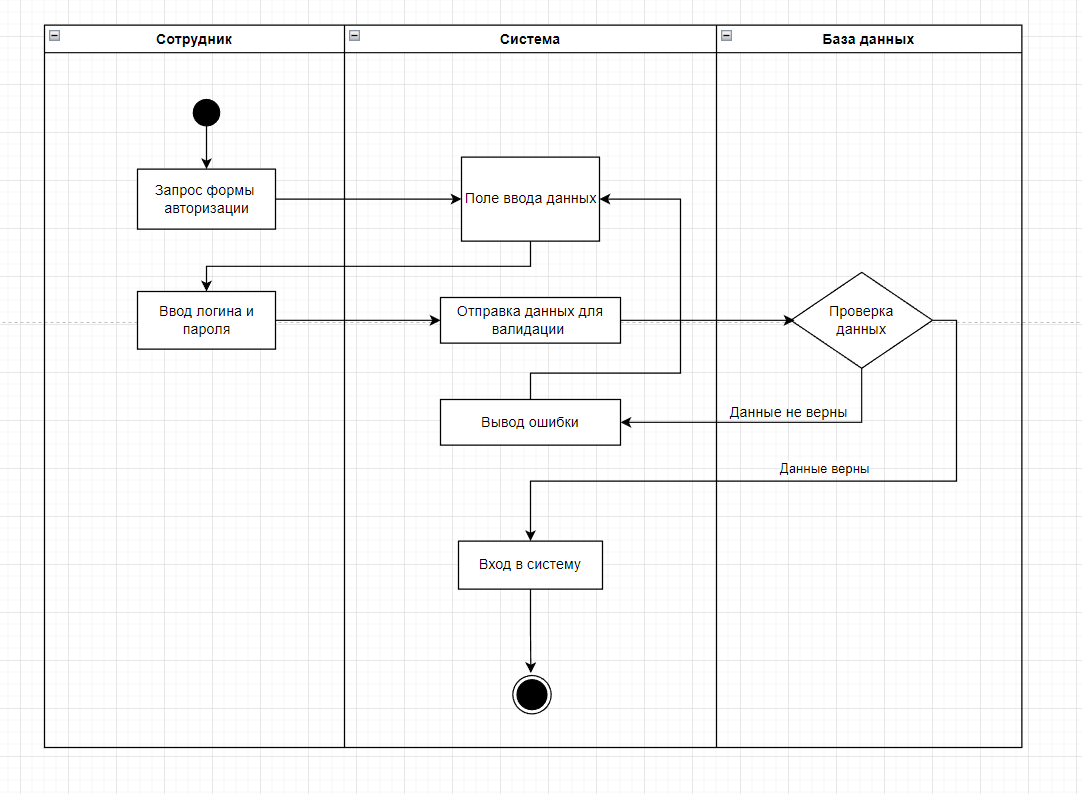


Рис. 2.13. Диаграмма активностей "Авторизация".

На рисунке 2.14 изображена диаграмма активностей для прецедента «Добавление нового отзыва». На диаграмме представлены следующие объекты: Гость, Система.

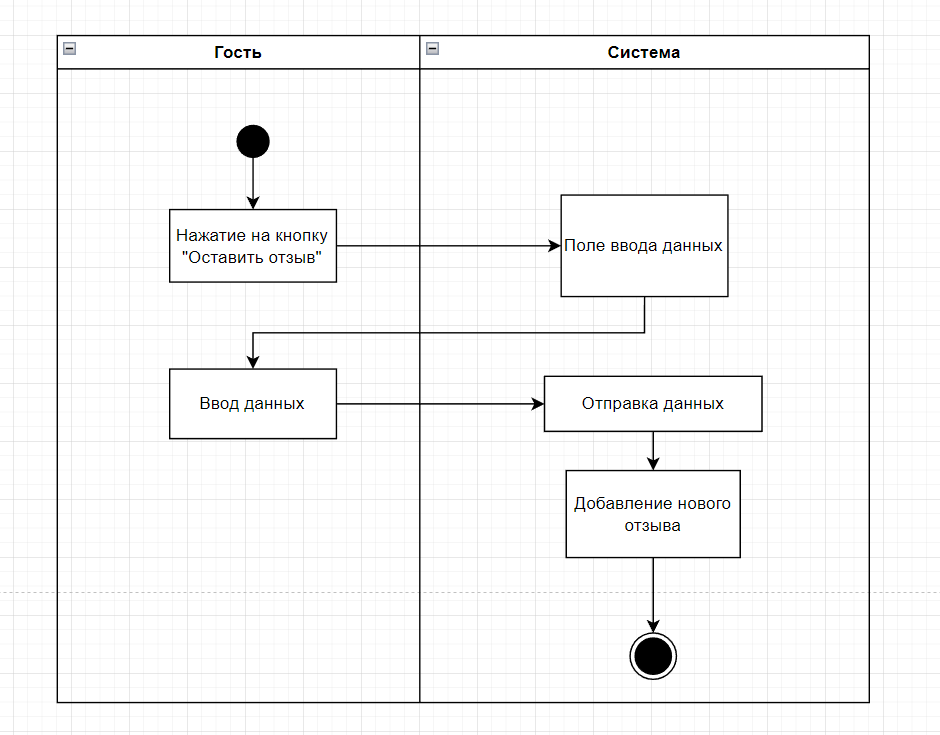


Рис. 2.14. Диаграмма активностей "Добавление нового отзыва ".

**2.3. Проектирование графического интерфейса пользователя**

После согласования всех требований, задач и функционала программного продукта, построения всех диаграмм по стандартам UML и IDEF, разработки базы данных необходимо выполнить разработку самого программного продукта.

Графический интерфейс пользователя должен быть понятен, не запутан и в то же время не перегружен контентом. Это достигается путем использования привычных для пользователя элементов управления.

Разрабатываемый программный продукт для роли «Администратор» или «Менеджер» имеет следующие страницы: «Профиль», «Задачи», «Заказы», «Продажи», «Склад», «Клиенты», «Сотрудники», «Заявки» и «Админ». Все это видно на Рисунке 2.15.

Страница «Профиль» (Рис 2.15.) выводит актуальные данные о сотруднике с фотографией и должностью, так же имеются две кнопки «Редактировать профиль» и «Выйти из системы».

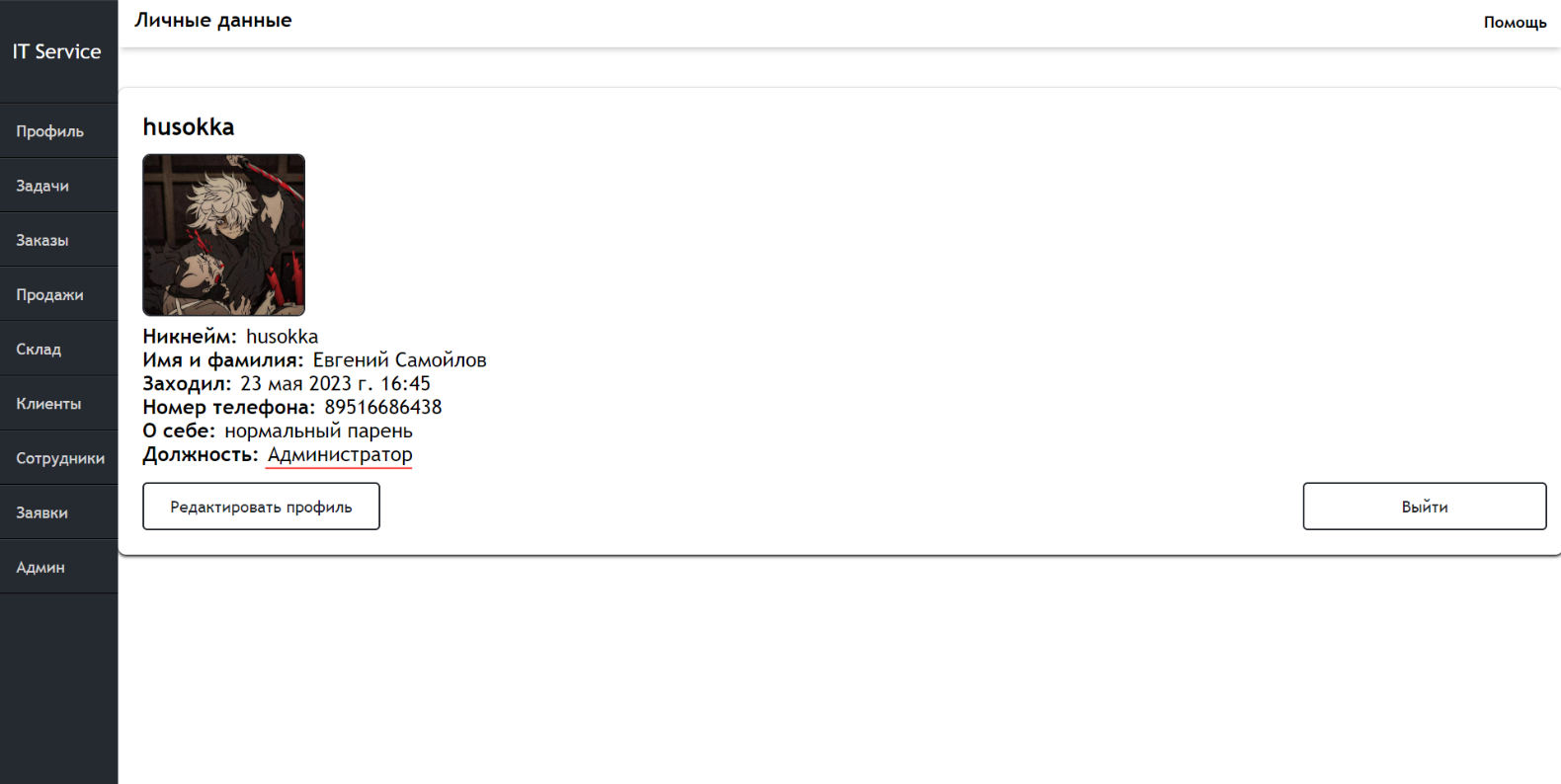


Рис. 2.15. Личный профиль Администратора.

Роли «Техник» и «Программист» имеют следующие страницы: «Профиль», «Задачи», «Заказы», «Продажи», «Склад» и «Клиенты», так как они имеют ограниченный функционал. Все это видно на Рисунке 2.16.

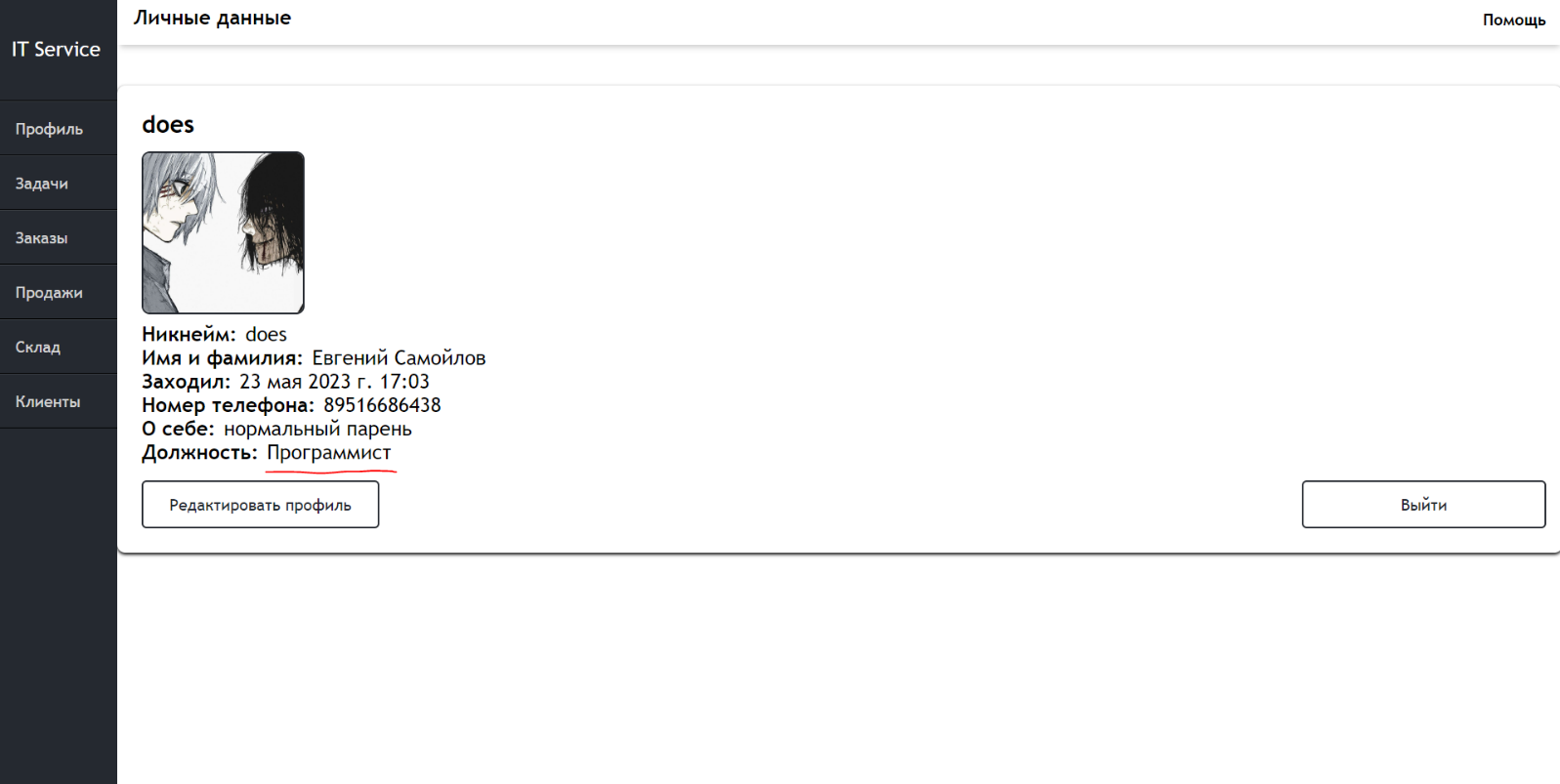


Рис. 2.15. Личный профиль Программиста.

Гость никак не сможет попасть в систему, не зная логина и пароля, так как будет выводится страница с оповещением (Рис. 2.16.)

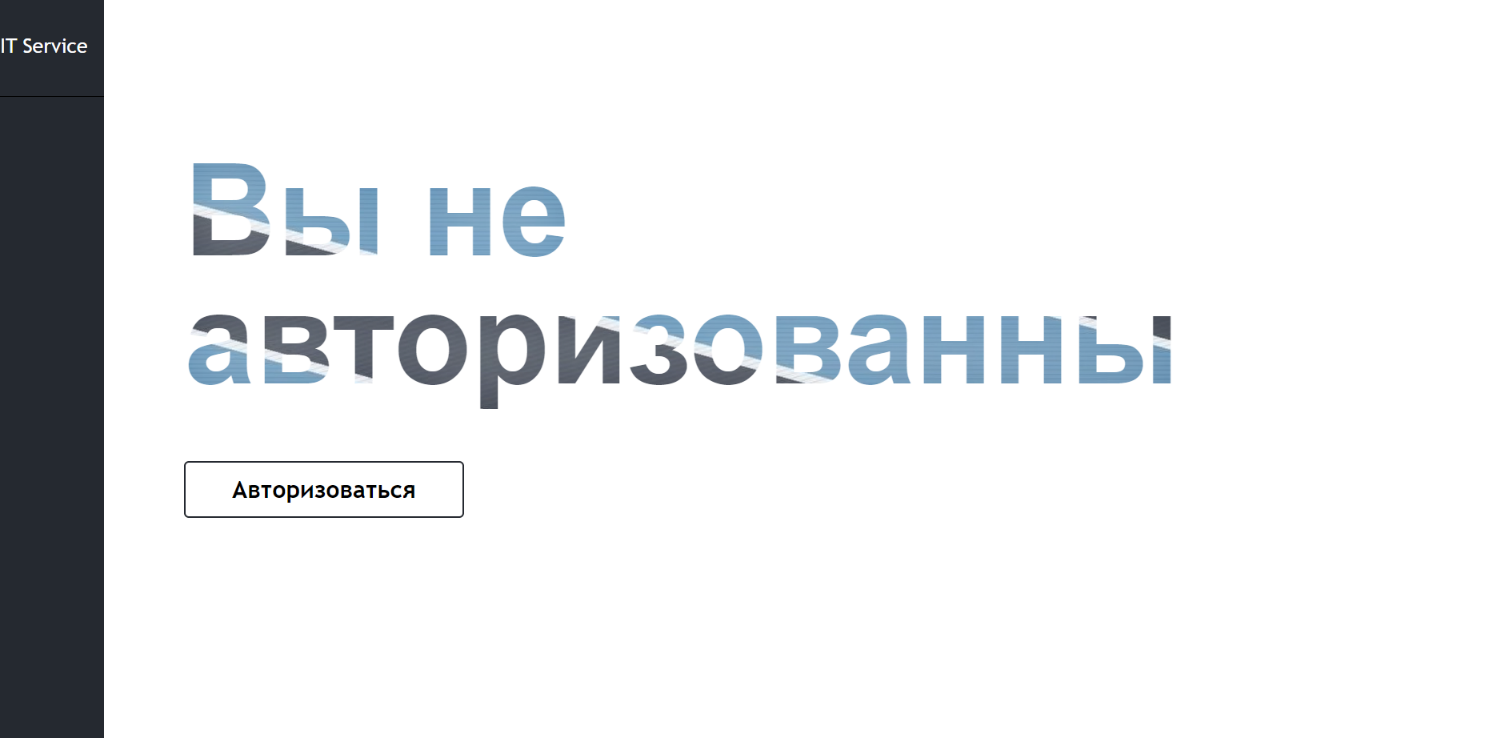


Рис. 2.16. Любая страница системы, когда сотрудник не авторизован.

Гость, заходя на сайт, видит совершенно другую страницу, на которой он может посмотреть или написать отзывы о работе IT Сервиса, так же имеется возможность написать заявку с проблемой и отправить ее на рассмотрение (Рис. 2.17 – 2.20)

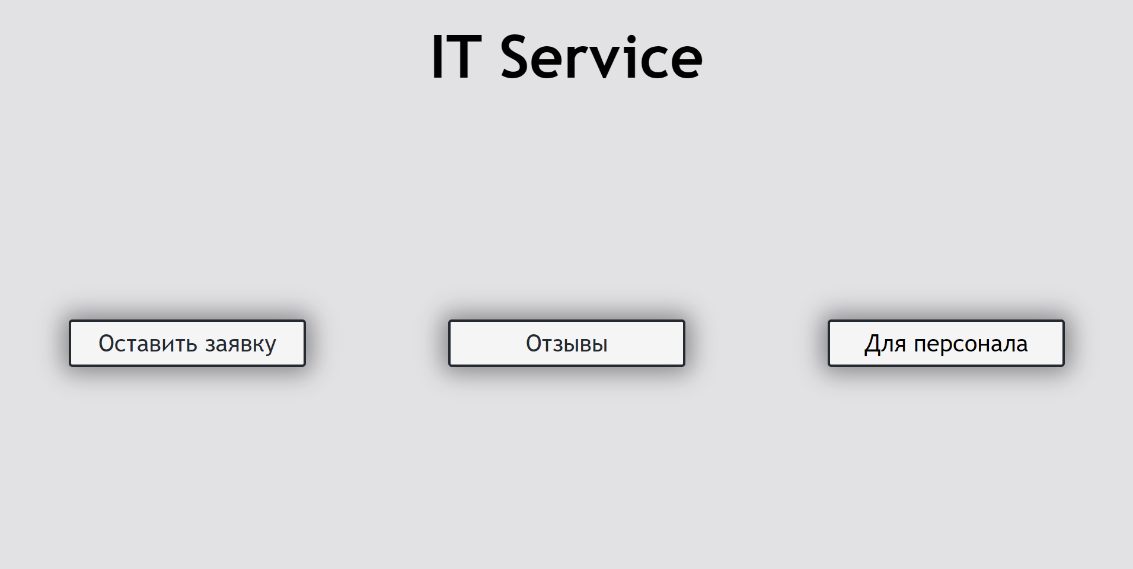


Рис. 2.17. Главная страница для Гостя.

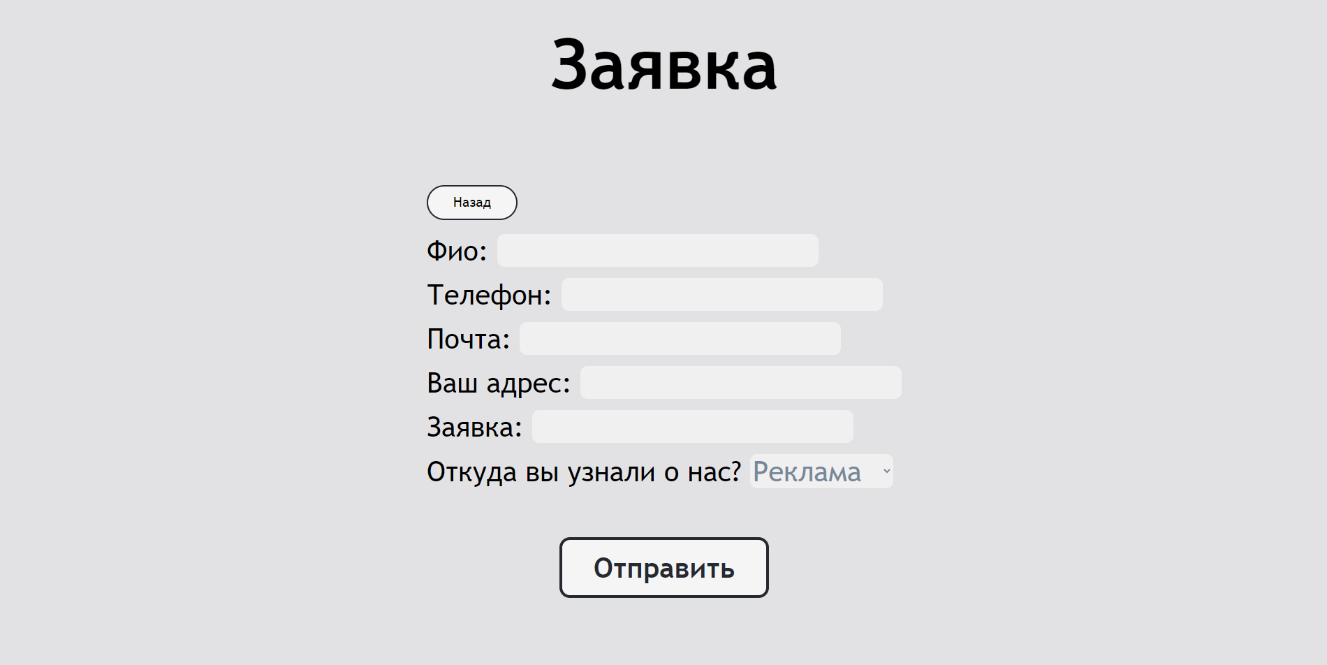


Рис. 2.18. Страница добавления новой заявки.



Рис. 2.19. Страница просмотра отзывов.

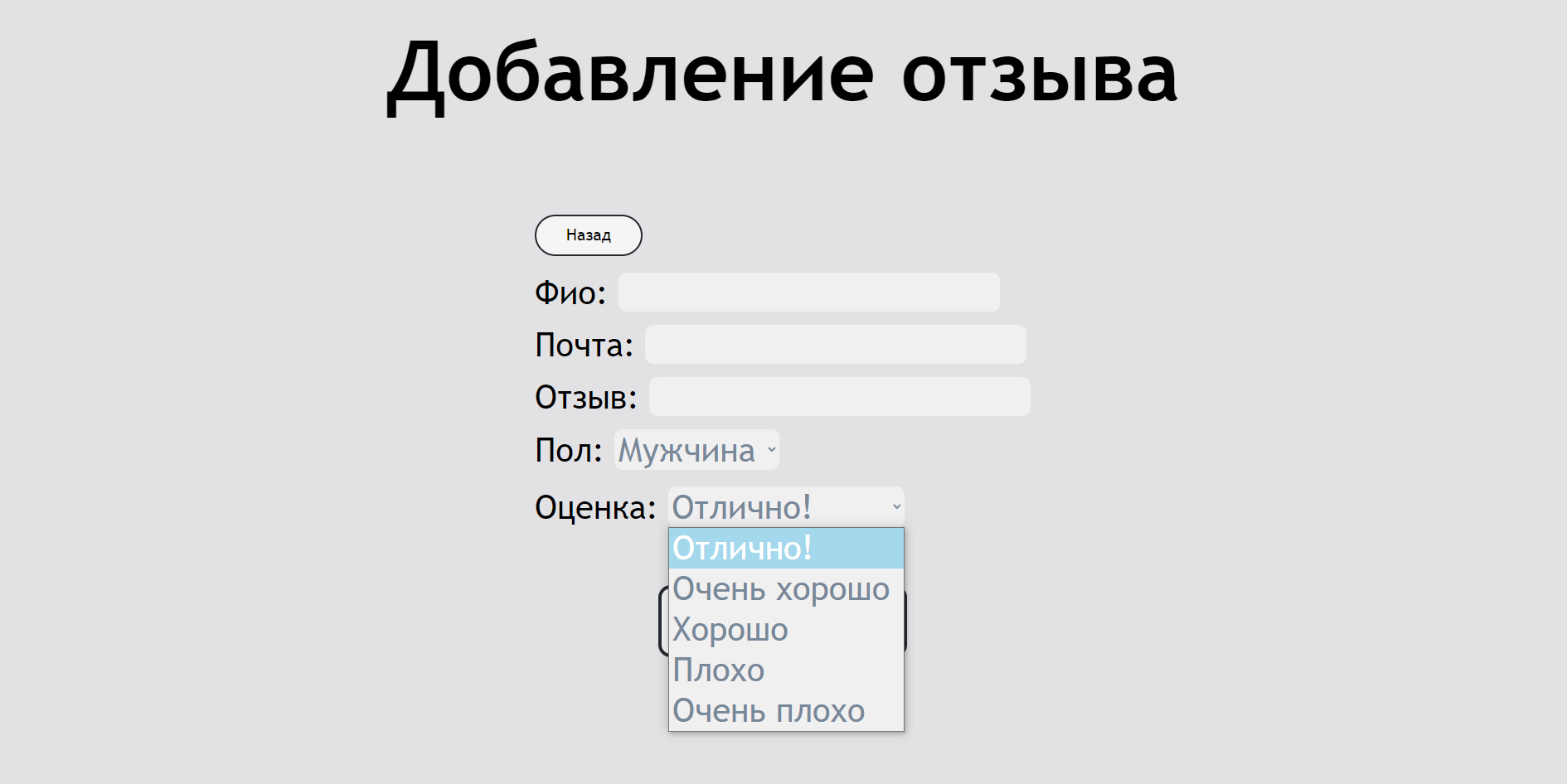


Рис. 2.20 Страница добавления отзыва.

**2.4. Проектирование и разработка модели данных**

При проектировании программного продукта была создана база данных, таблицы и поля, между которыми затем были созданы зависимости и диаграмма “сущность-связь”.

БД содержит следующие таблицы:

1. Таблица “users” – расширенная таблица от AbstractUser, содержит информацию о сотрудниках.

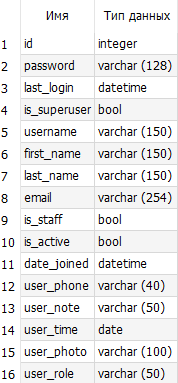


Рис 2.2. Таблица “users”

Описание полей таблицы “ users”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор сотрудника | int |
| password | Пароль сотрудника | varchar(128) |
| last\_login | Дата последней авторизации | datetime |
| is\_superuser | Статус суперпользователя | bool |
| username | Логин сотрудника | varchar(150) |
| first\_name | Настоящее имя сотрудника | varchar(150) |
| last\_name | Настоящая фамилия сотрудника | varchar(150) |
| email | Электронная почта сотрудника | varchar(254) |
| is\_staff | Статус, определяющий, является ли сотрудник администратором | bool |
| is\_active | Статус доступа пользователя (заблокирован или нет) | bool |
| data\_joined | Дата регистрации сотрудника | datetime |
| user\_phone | Номер телефона сотрудника | varchar(40) |
| user\_note | Примечание о сотруднике | varchar(50) |
| user\_time | Дата начала работы сотрудника | date |
| user\_photo | Фото сотрудника | varchar(100) |
| user\_role | Должность сотрудника | varchar(50) |

1. Таблица “ tasks” – содержит информацию о задачах.

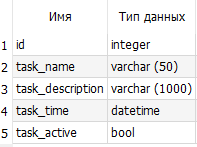


Рис 2.3. Таблица “ tasks”

Описание полей таблицы “ tasks”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор задачи | int |
| task\_name | Название задачи | varchar(50) |
| task\_description | Описание задачи | varchar(1000) |
| task\_time | Крайнего срок задачи | datetime |
| task\_active | Статус | bool |

1. Таблица “ client” – содержит информацию о клиентах.

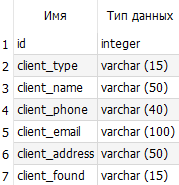


Рис 2.4. Таблица “ client”

Описание полей таблицы “ client”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор клиента | int |
| client\_type | Тип клиента | varchar(15) |
| client\_name | Имя клиента | varchar(50) |
| client\_phone | Телефон клиента | varchar(40) |
| client\_email | Электронная почта клиента | varchar(100) |
| client\_address | Адрес клиента | varchar(50) |
| client\_found | Откуда о нас узнал клиент | varchar(15) |

1. Таблица “ orders” – содержит информацию о клиентах.

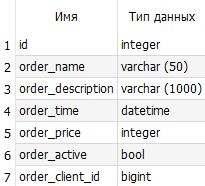


Рис 2.5. Таблица “ orders”

Описание полей таблицы “ orders”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор заказа | int |
| order\_name | Название заказа | varchar(50) |
| order\_description | Описание заказа | varchar(1000) |
| order\_time | Крайний срок заказа | datetime |
| order\_active | Статус заказа | bool |
| order\_price | Цена заказа | int |
| order\_client | Внешний ключ, для клиента | bigint |

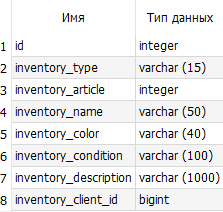
1. Таблица “ inventory” – содержит информацию о клиентах.

Рис 2.6. Таблица “ inventory”

Описание полей таблицы “ inventory”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор вещи | int |
| inventory\_type | Тип вещи | varchar(15) |
| inventory\_article | Артикул вещи | int |
| inventory\_name | Название вещи | varchar(50) |
| inventory\_color | Цвет вещи | varchar(40) |
| inventory\_condition | Состояние вещи | varchar(100) |
| inventory\_description | Для описания | varchar(1000) |
| inventory\_client | Внешний ключ, для клиента | bigint |

1. Таблица “ sales” – содержит информацию о клиентах.

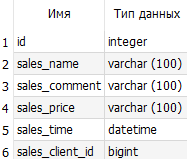


Рис 2.6. Таблица “ sales”

Описание полей таблицы “ sales”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор продажи | int |
| sales\_name | Название продажи | varchar(100) |
| sales\_comment | Описание продажи | varchar(100) |
| sales\_time | Время продажи | datetime |
| sales\_price | Цена продажи | varchar(100) |
| sales\_client | Внешний ключ, для клиента | bigint |

1. Таблица “ review” – содержит информацию о клиентах.

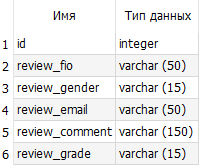


Рис 2.6. Таблица “ review”

Описание полей таблицы “ review”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор отзыва | int |
| review\_fio | Фио | varchar(50) |
| review\_gender | Пол | varchar(15) |
| review\_email | Почта | varchar(50) |
| review\_comment | Отзыв | varchar(150) |
| review\_grade | Оценка | varchar(15) |

1. Таблица “ application” – содержит информацию о клиентах.

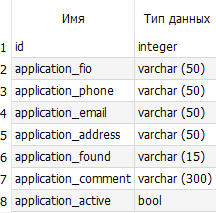


Рис 2.6. Таблица “ application”

Описание полей таблицы “ application”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор заявки | int |
| application\_fio | Фио | varchar(50) |
| application\_phone | Телефон | varchar(50) |
| application\_email | Почта | varchar(50) |
| application\_address | Адрес | varchar(50) |
| application\_found | Откуда нашел? | varchar(15) |
| application\_comment | Заявка | varchar(300) |
| application\_active | Статус | bool |

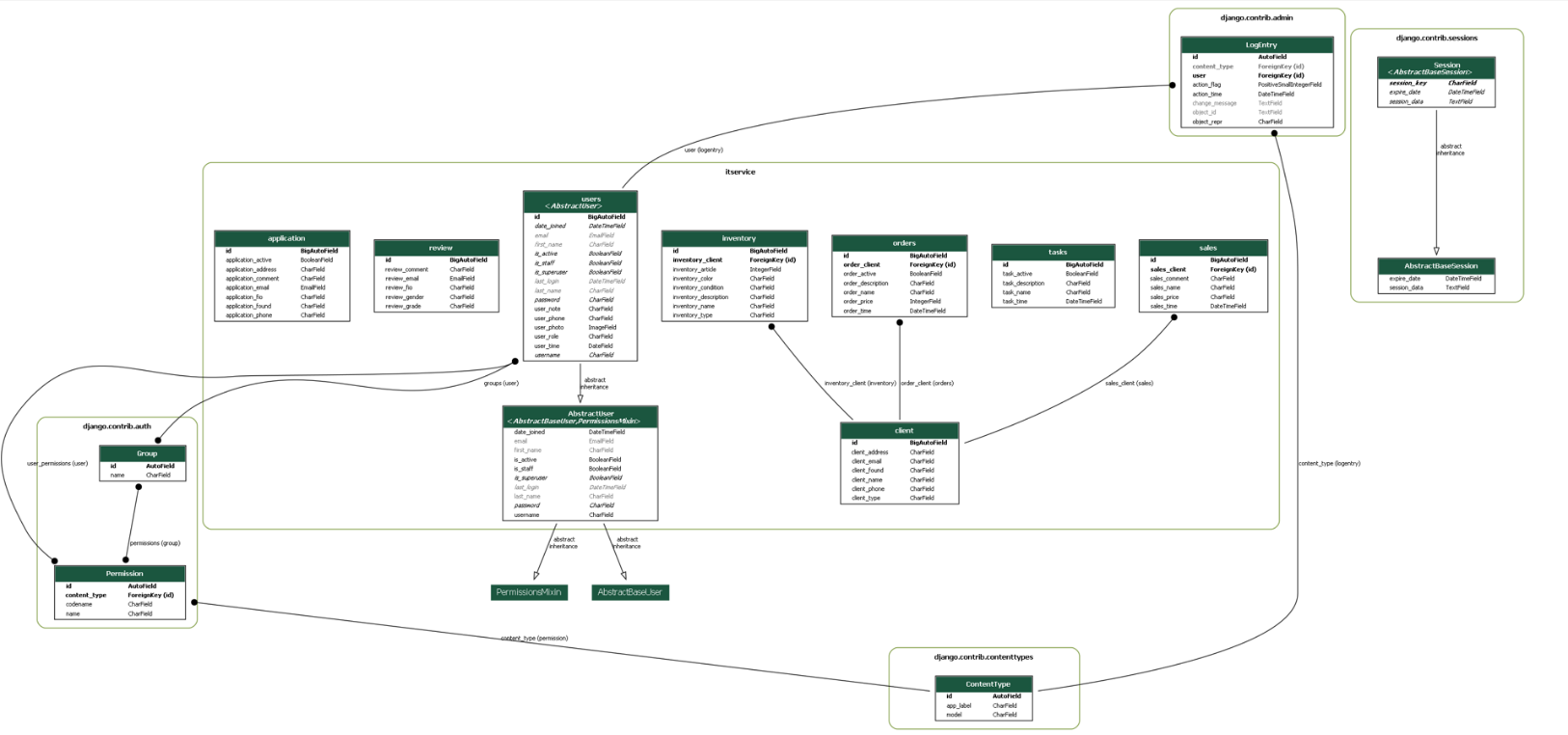


Рис 2.7. ER-диаграмма базы данных программного продукта.

**Выводы второй главы:**

Подводя промежуточные итоги, можно сказать, что разработано следующее:

* представлен вариант использования системы;
* спроектирована архитектура системы;
* разработан графический интерфейс системы;
* реализована база данных программного продукта.

**3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**

## **3. 1. Порядок разработки программных модулей**

При создания программного модуля желательно следовать порядку, представленному ниже:

* анализ классификации модуля, выбор языка программирования;
* разработка алгоритма и структуры данных;
* программирование программного модуля;
* шлифовка текста программного модуля;
* проверка программного модуля;
* компиляция программного модуля.

После согласования всех требований, задач и функционала программного продукта, построения всех диаграмм по стандартам UML и IDEF, разработки базы данных необходимо выполнить разработку самого программного продукта.

Проект выполнен при использовании языка программирования Python с Фреймворком Django в среде разработки PyCharm (Рис. 3.1).

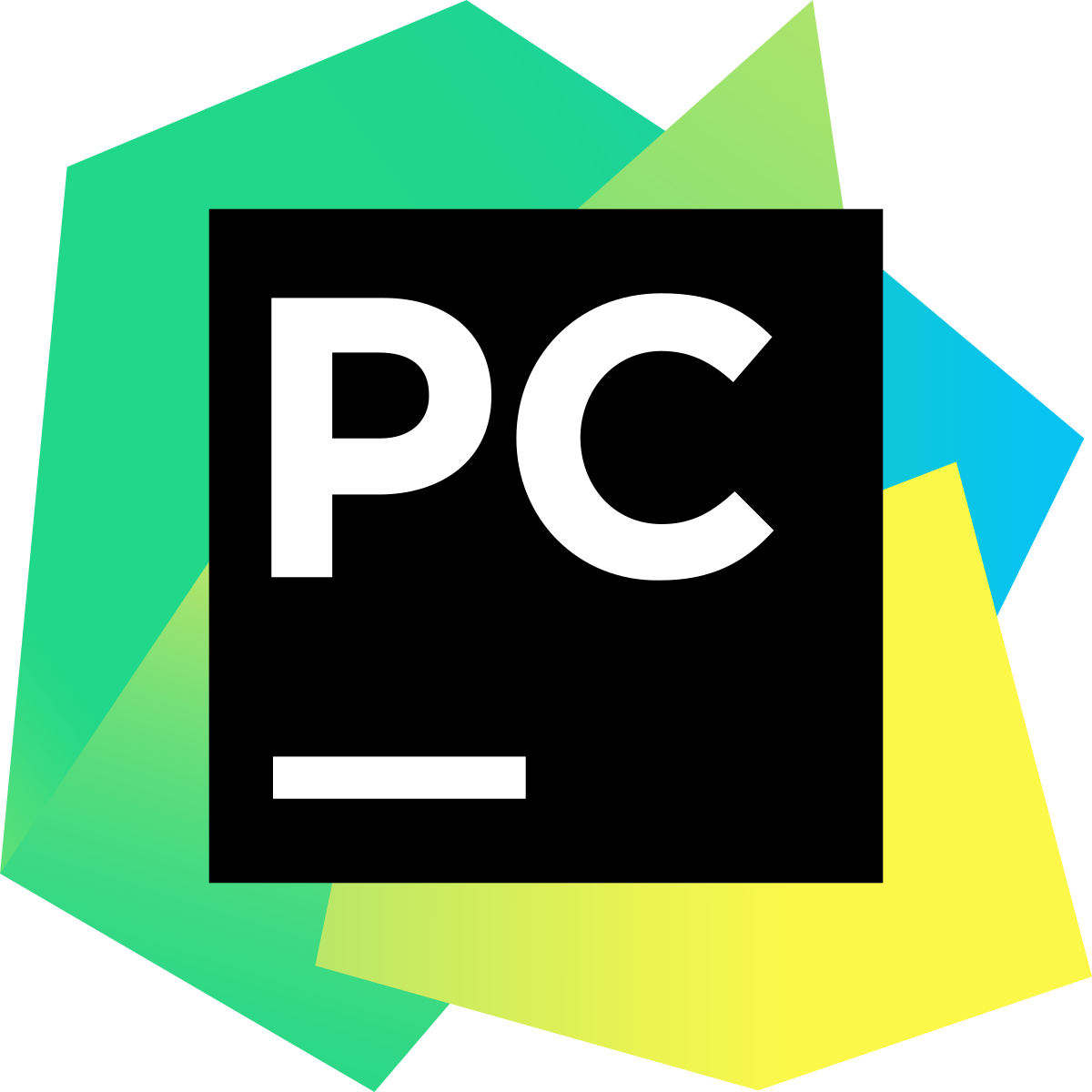


Рис 3.1. Среда разработки PyCharm

## **3. 2. Структурное программирование**

При разработке модуля на языке программирования Python важно помнить, что программа должна быть понятна не только компьютеру, но и людям, которые будут ее использовать и поддерживать. Это включает в себя разработчиков, тестировщиков и системных администраторов, которые будут заниматься отладкой и редактированием модуля.

В современных и новых языках программирования существует множество инструментов, которые могут запутать логику работы модуля и привести к тому, что реализованный модуль будет сложен для понимания и использования. Поэтому при выборе языка программирования и подхода к разработке модуля необходимо учитывать конкретную дисциплину программирования и основываться на мерах, которые обеспечат безопасность и понятность работы модуля.

Структурное программирование - это методология программирования, которая основывается на использовании структурных блоков для организации кода. Она подразумевает разделение программы на небольшие управляющие структуры, такие как последовательность, ветвление и циклы. Эти структуры могут быть объединены в модули, что позволяет создавать более крупные и сложные программы. Главной целью структурного программирования является создание кода, который легко читается, тестируется и сопровождается, а также минимизация количества ошибок в программе.

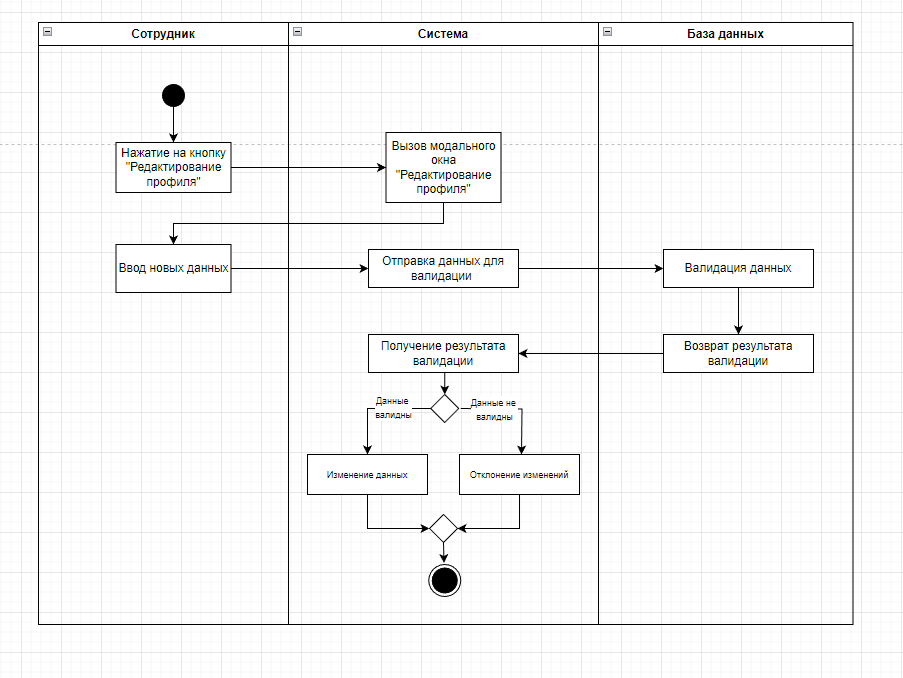


Рис. 3.2. Процесс редактирования данных сотрудника

Сотрудник, который зарегистрирован в системе, изменяет свои данные, например, меняет фотографию профиля, система принимает данные и производит проверку. Если данные не валидны, то все изменения отклоняются и все возвращается к исходным данным, иначе система сохраняет изменения в базе данных.

## **3. 3. Реализация серверной части**

Python - это язык программирования, который используется для написания серверной части системы. PyCharm - это интегрированная среда разработки (IDE), которая поможет создать и отладить код.

Фреймворк Django предоставляет набор инструментов для быстрого и удобного создания веб-приложений. Он позволяет легко работать с базами данных, формами, авторизацией пользователей и другими функциями, которые необходимы для создания полноценного веб-сайта.

Sqlite - это легкая встроенная база данных, которую можно использовать для начала работы с системой. Она не требует установки дополнительного ПО, что делает ее удобной для использования в небольших проектах.

Таким образом, можно использовать Python с помощью PyCharm и фреймворка Django для создания веб-приложений, которые будут хранить свои данные в базе данных Sqlite.

Конечная цель создания моделей в Django - это определение структуры таблиц в базе данных, которые будут хранить данные вашего приложения. Модели в Django - это классы Python, которые наследуются от базового класса «django.db.models.Model». Каждый атрибут класса-модели соответствует полю таблицы в базе данных. В файле models.py создаются модели, которые с помощью миграций перенесутся в базу данных (Рис. 3.3.)

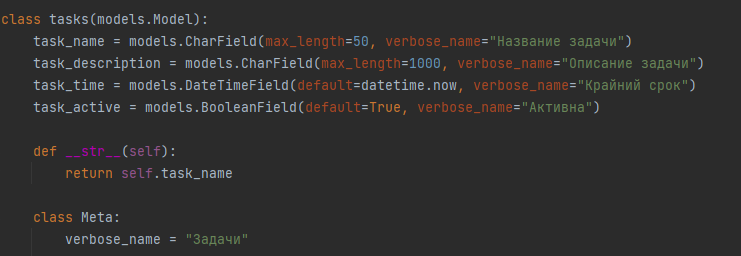


Рис. 3.3. Создание модели на примере класса «tasks»

С помощью «python manage.py makemigrations» мы создаем файл миграции, и он сохраняется в папке проекта «migrations». При успешном создании миграций, с помощью команды «python manage.py migrate» мы переносим наши миграции в базу данных (Рис 3.4.)

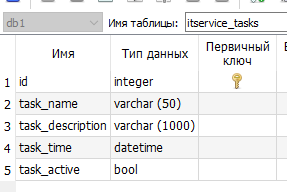


Рис. 3.4. Созданная таблица в бд

## **3. 4. Организация взаимодействия с БД**

Организация взаимодействия моделей базы данных с сайтом на Python в PyCharm с Django заключается в том, что в приложении Django вы можете определить модели базы данных, которые представляют данные, которые нужно хранить. Эти модели могут быть связаны между собой, чтобы создать более сложные структуры данных.

Далее, когда Сотрудник делает запрос на сайт на Python, Django будет использовать модели базы данных для получения необходимых данных из базы данных и передавать их в виде объектов Python веб-страницам. Когда пользователь отправляет данные с веб-страницы, Django использует модели базы данных для сохранения этих данных в базу данных.

Таким образом, организация взаимодействия моделей базы данных с сайтом на Python в PyCharm с Django обеспечивает эффективное хранение данных, быстрый доступ к ним и простое управление данными веб-приложения.

Способ организации взаимодействия моделей базы данных с сайтом на примере страницы «Задачи». В файле views.py создается представление страницы, так же импортируются все модели из models.py, далее создается представление для страницы, в которой мы передаем информацию о конкретной модели (Рис. 3.5.).

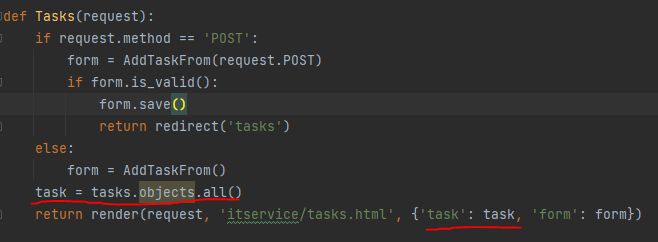


Рис. 3.5. Представление для страницы Задач

После этого, уже в файле tasks.html мы обращаемся к модели по имени как к массиву значений и берем оттуда то, что нам нужно

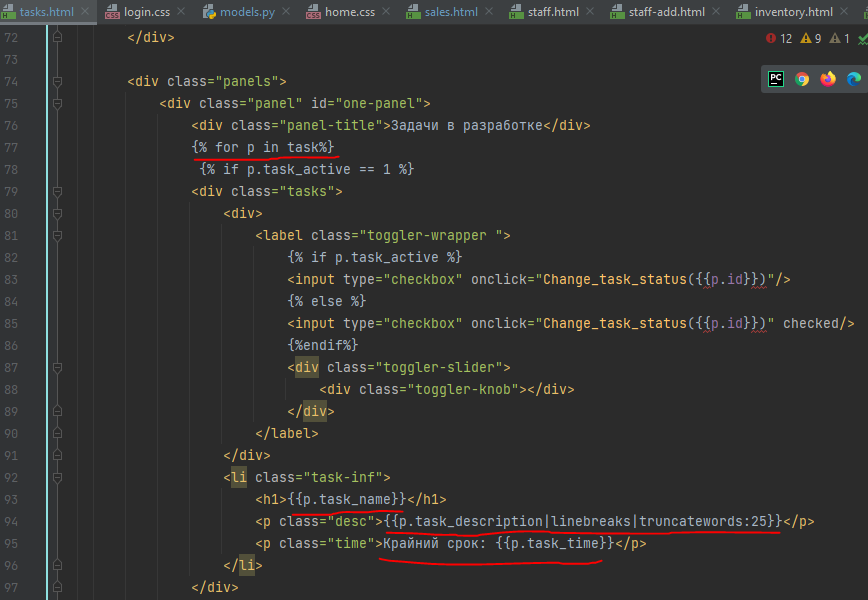


Рис. 3.6. Обращение к модели task

Выводы третьей главы:

Подводя итоги вышесказанному необходимо отметить следующее:

* Описан порядок разработки программных модулей;
* Приведен пример структурного программирования;
* Реализована серверная часть;
* Предоставлена организация взаимодействия с базой данных системы.

**4. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

После разработки программного продукта следует начать его тестировать. В этой главе отображена хронология изменений, исправлений и доработок программного обеспечения, проверена разработанного ПО, определены объекты тестирования, а также описан процесс тестирования.

**4.1. История изменений**

Таблица 4.1. Версионность программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Автор** | **Описание изменений** |
| 21.04.23 | 1.1 | Евгений Самойлов | Создание проекта, добавление виртуального пространства, установка всех необходимых инструментов. |
| 23.04.23 | 1.2-1.3 | Евгений Самойлов | Создание главного шаблона для всех страниц сайта. |
| 27.04.23 | 1.4-1.5 | Евгений Самойлов | Создание моделей базы данных, а так же добавление новых вкладок. |
| 01.05.23 | 1.6 | Евгений Самойлов | Проработка стилей ранее добавленных вкладок. |
| 04.05.23 | 1.7 | Евгений Самойлов | Подключение ко вкладкам базы данных и реализация вывода данных на сайте. |
| 05.05.23 | 1.8 | Евгений Самойлов | Создание модальных окон для добавления новых данных в БД. |
| 10.05.23 | 1.9 | Евгений Самойлов | Добавление стилей и модальных окон на все вкладки. |
| 11.05.23 | 2.0 | Евгений Самойлов | Создание вкладки для редактирования данных пользователя. |
| 14.05.23 | 2.1 | Евгений Самойлов | Проработка стилей, так же добавление возможности изменять статус данных. |
| 16.05.23 | 2.2-2.3 | Евгений Самойлов | Реализация возможности добавления новых сотрудников в систему;  Окно логина в систему;  Проработка стилей. |
| 19.05.23 | 2.4 | Евгений Самойлов | Создание окон для Заявок и Отзывов;  Добавление новых таблиц в бд. |
| 23.05.23 | 2.5 | Евгений Самойлов | Мелкие доработки;  Добавление Фавикона;  Создание ER диаграммы. |

**4.2. Терминология**

Список терминов и определений, используемых далее в документе и в тестировании в целом:

* Производительность - количество выполняемых операций за период времени (N операций за M часов);
* Тестирование стабильности – это проведение тестирования со средней нагрузкой в течение длительного времени. Выявляет проблемы связанные с утечками памяти и некорректными настройками ПО;
* Архитектура “клиент - сервер” – это сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами;
* Нагрузка - совокупное выполнение операций на общем ресурсе (тр./сек, хитов/сек);
* Виртуальный пользователь – программный процесс, циклически выполняющий моделируемые операции;
* Стратегия тестирования — это план проведения работ по тестированию системы или её модуля, учитывающий специфику функциональности и зависимости с другими компонентами системы и платформы;
* Позитивное тестирование – это тестирование с применением сценариев, которые соответствуют нормальному поведению системы;
* Негативное тестирование - это применение сценарий, которые соответствуют внештатному поведению тестируемой системы;
* Адаптивность — это способность сайта “подстраиваться” под различные технические условия (под размеры экрана);
* Тестирование методом “открытого ящика” – это метод тестирования программного продукта, который предполагает, что внутренняя структура, устройство, реализация системы известны тестировщику.

**4.3. Стратегия тестирования**

Стратегия тестирования программного обеспечения заключается в использовании метода тестирования “открытого ящика” для анализа и оценки защиты информационных потоков данных, а также отслежка выходных в входных данных на прецедент ошибки, пустого значения

**4.4. Определение объектов тестирования**

Таблица 4.2. Определение объектов тестирования

| **Тесты** | **Объект тестирования** |
| --- | --- |
| Функциональное тестирование | * Работа системы авторизации; * Работа системы регистрации новых сотрудников; * Работа системы добавления новых задач и заказов; * Работа системы добавления новых продаж; * Работа системы добавления клиентов и устройств; * Работа системы добавления заявок и отзывов; * Работа системы скачивания документов; * Работа системы редактирования профиля; * Работа системы изменения статуса элементов; |
| Тестирование производительности | * Скорость перехода по формам; * Скорость сохранения данных в БД; * Скорость отображения информации из БД |
| Нагрузочное тестирование | * Попытка добавления огромного количества продаж, а далее скачивание сформированного файла Excel; * Изменение фотографии сотрудника на фотографию с очень большим разрешением; |
| Тестирование совместимости | * Корректное отображение программы в разных версиях ОС Windows (8, 10); * Корректное отображение программы на различных соотношениях сторон экрана: 4:3, 16:10, 16:8; * Корректное отображение программы на различных разрешениях экрана: 1024x768, 1280x720, 1600x900, 1920x1080, 2560x1600. |
| Позитивные тесты | * Добавление сотрудника, используя почту, которой еще нет в системе; * Попытка изменения фотографии сотрудника на другую, используя форматы .jpg и .png; * Добавление заявки длинной не более 150 символов. |
| Негативные тесты | * Добавление сотрудника, используя почту, которая уже есть в системе; * Попытка изменения фотографии сотрудника на другую, используя видео, .gif, .txt и т.д.; * Добавление заявки длинной более 150 символов. |
| Модульное тестирование | * Тестирование возможностей Администратора/Менеджера; * Тестирование возможностей Техника/Программиста * Тестирование возможностей Гостя |
| Интеграционное тестирование | * Переход со стартовой страницы на страницу авторизации; * Переход со страницы авторизации на страницу добавления сотрудников; * Переход со страницы добавления сотрудников на страницу редактирования профиля; |
| Системное тестирование | * Попытка обойти средства защиты программы; * Скорость отклика программы; * Способность системы эффективно использовать оперативную память. |

**4.5. Архитектура тестируемой системы**

Правило функционирования двухуровневой архитектуры содействия клиент-сервер заключается в обработке запросов, происходящих на одной станции ЭВМ, без использования иных ресурсов. Данная архитектура выражает серьёзные требования к коэффициент полезного действия сервера, но является весьма устойчивой (рис. 4.1).

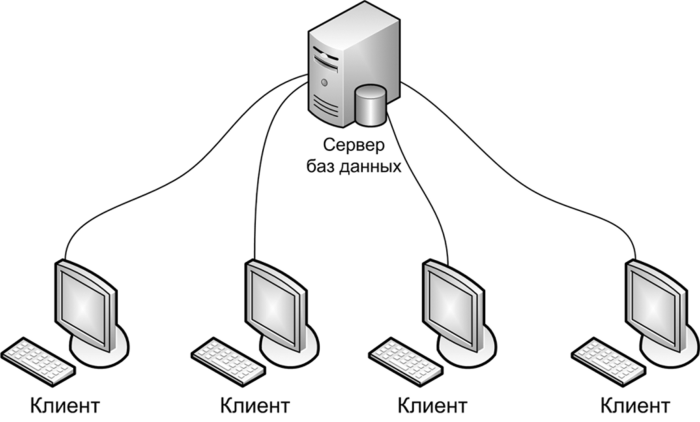


Рис 4.1. Двухзвенная архитектура тестируемой системы

Достоинство реализации модели клиент-сервер является то, что системный код клиентской и серверной программы отделён. Если говорить о локальных компьютерных сетях, то к достоинствам архитектуры клиент-сервер является низкие требования к станциям ЭВМ клиентов, т.к. большая часть вычислительных процессов производиться на сервере, а также архитектура клиент-сервер в меру подвижная и допускает администратору реализовать локальную сеть более безопасной.

**4.6. Описание процесса тестирования**

Таблица 4.3. Тестовые сценарии с указанием конкретных тестовых наборов

| **Название** | **Тестовые данные** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| --- | --- | --- | --- | --- |
| Вход с вводом существующего логина и пароля | Логин: admin  Пароль: static123  Роль сотрудника: Менеджер | Сотрудник успешно вошел в систему | Сотрудник успешно вошел в систему | Пройден |
| Вход с вводом несуществующего логина и пароля | Логин: user  Пароль: user  Роль сотрудника: неизвестна | Система оповещает о неправильном логине и пароле | Сотрудник не вошел в систему, сообщение получено | Пройден |
| Вход с вводом только логина | Логин: admin  Пароль: -  Роль сотрудника: Менеджер | Система оповещает о необходимости ввода пароля | Система оповещает о необходимости ввода пароля | Пройден |
| Вход с вводом только пароля | Логин: -  Пароль: static123  Роль сотрудника: Менеджер | Система оповещает о необходимости ввода логина | Система оповещает о необходимости ввода логина | Пройден |
| Кнопка “Выйти” | Кнопка на странице “Главная” | Выход пользователя из аккаунта | Выход пользователя из аккаунта | Пройден |
| Функционирование личного кабинета | Кнопка «Профиль» на странице | Открытие личного кабинета | Открытие личного кабинета | Пройден |
| Изменение статуса задачи | Переключатель на задаче | Изменение статуса задачи | Изменение статуса задачи | Пройден |
| Функционирование добавления новой задачи | Кнопка «Добавить задачу» на странице | Создание новой задачи | Создание новой задачи | Пройден |
| Удобство пользования. | Графический интерфейс системы. | Удобный и интуитивно понятный интерфейс системы. | Адаптивный программный продукт, единый стиль интерфейса, удобная и понятная навигация по системы. | Пройден |

**Выводы четвертой главы:**

Все вышесказанное доказывает, что:

* описана история изменения системы;
* разработана стратегия тестирования программного обеспечения;
* определены объекты тестирования;
* приведена архитектура тестируемой системы и описание процесса тестирования.

**5. ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ**

**5.1. Определение защищаемых информационных активов**

**5.2. Вид и содержание персональных данных**

**5.3. Модель угроз ПНД**

**5.4. Возможные сценарии угроз**

**5.5. Организация защиты данных в проекте**

**5.6. Защита программного продукта**

**6. ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ НА ПРОГРАММНЫЙ ПРОДУКТ**

**6.1. Руководство пользователя**

Руководство пользователя создавалось по ГОСТ 19 и 34.

* ГОСТ 34 - Автоматизированные системы
* ГОСТ 19 - Единая система программной документации (ЕСПД)

Руководство пользователя расположено в приложении 3.

**6.2. Руководство администратора**

Руководство администратора не имеет собственного стандарта оформления. Выполняется этот документ по РД 50-34.698-90, содержащему общие требования к содержанию документации на автоматизированные системы.

Руководство администратора расположено в приложении 4.

**6.3. Определение качества программного продукта по метрикам**

Таблица 6.1. Расчет метрик стилистики и понятности системы.

|  |  |  |
| --- | --- | --- |
| Количество строк программы | 1328(2301) | |
| Строк в каждом сегменте | 266 | |
| Кол-во сегментов программы | 5 | |
|  | Количество  комментариев |  |
| 1 сегмент | 35 | 1 |
| 2 сегмент | 44 | 1 |
| 3 сегмент | 32 | 1 |
| 4 сегмент | 36 | 1 |
| 5 сегмент | 34 | 1 |
|  |  | 5 |

**Расчет:**

Сегмент 1. F1 = sgn((35 / 266) – 0,1) = 1

Сегмент 2. F2 = sgn((44 / 266) - 0,1) = 1

Сегмент 3. F3 = sgn((32 / 266) – 0,1) = 1

Сегмент 4. F4 = sgn((36 / 266) - 0,1) = 1

Сегмент 5. F5 = sgn((34 / 266) – 0,1) = 1

**Вывод:**

Так как уровень комментированности программы F=5, а число сегментов n=5, то условие нормального уровня комментированности F=n выполняется. При F=n уровень комментированности программы считается нормальным.

**Выводы шестой главы:**

Подводя итоги вышесказанному необходимо отметить следующее:

* Рассчитана метрика стилистики и понятности программы;
* Разработано руководство пользователя и администратора.

**7. МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ТРУДА**

Техника безопасности при работе с ПК на предприятии предусматривает наличие инструкции, в которой описаны обязательные требования к рабочему месту и процессу использования техники. Эти правила общие для всех организаций, их выполнение контролируется соответствующими органами государственного аппарата.

Основными нормативными документами, которые содержат обязательные требования, можно считать:

* Трудовой кодекс;
* Санитарно-гигиенические нормы и правила СанПиН 2.2.2/2.4.1340-03, утвержденные постановлением Главного государственного санврача от 13 июня 2003 года N 118;
* Межгосударственный стандарт ГОСТ 12.0.003-2015, который содержит список основных вредных и опасных производственные факторов и их классифицирующие признаки;
* Типовая инструкция ТОИ Р-45-084-01, в которой содержатся основные правила для работников, использующих компьютер;
* Другие нормативные документы.

Требования перечисленных правовых актов, в которых содержатся общие правила защиты персонала при выполнении трудовых обязанностей и техника безопасности во время работы за компьютером, обязательны к исполнению всеми работниками этой категории.

На специалиста, работающего за компьютером, могут воздействовать вредные факторы, такие как:

* Высокая температура, характерная для отдельных элементов компьютерной техники и создающая общий повышенный температурный фон в рабочем помещении;
* Значительный уровень зрительный нагрузок, испытываемых работником;
* Повышенный уровень напряженности электрического и магнитного полей, генерируемых работающей техникой;
* Неравномерность распределения яркости в поле зрения;
* Другие факторы.

Чтобы донести до работника требования, которые устанавливает техника безопасности при использовании компьютера, работодатель обязан организовать инструктаж. Он проводится на основании специально составленной инструкции, которую обычно разрабатывает специалист по охране труда или, непосредственно, руководитель работника. Проведение инструктажа нужно организовать до начала самостоятельной работы: специалист допускается к работе только после его прохождения. Инструкция-памятка по технике безопасности за компьютером должна быть выдана каждому работнику в бумажном или электронном виде.

Техника безопасности в начале работы на компьютере требует выполнения следующих операций:

1. Проверить исправность элементов электросистемы, обеспечивающей питание компьютера, включая электропроводку, выключатели, вилки и розетки, при помощи которых аппаратура подключается к сети;
2. Проконтролировать заземление компьютера;
3. Проверить его работоспособность.

При выполнении работ нужно соблюдать следующие правила:

1. Запрещается класть на корпус и дисплей компьютера посторонние предметы, прикасаться к элементам аппаратуры мокрыми руками, производить чистку корпуса оборудования, находящегося под напряжением, располагать технику близко к жилищно-коммунальным инженерным системам;
2. В случае обнаружения неисправности компьютера немедленно прекратить работу и сообщить об этом непосредственному руководителю;
3. Избегать частого и необоснованного включения и выключения компьютера во время работы.

Требования к расположению работника за компьютером нацелены на обеспечение его комфорта в течение всей рабочей смены и отсутствие негативных следствий длительной работы. Они действуют для любых рабочих мест и включают следующие правила:

1. Использование компьютерной мебели, отвечающей нормам СанПиН 2.2.2/2.4.1340-03;
2. Отказ от скрещивания конечностей, способного затруднить кровообращение;
3. Соблюдение расстояние до монитора компьютера не меньше 45 сантиметров;
4. Правильная установка освещения, которое не должно светить в глаза и оставлять блики на рабочем мониторе;
5. Клавиатура размещается за 20-30 сантиметров от края стола;
6. Стул стоит таким образом, чтобы спина лишь немного упиралась в его спинку;
7. Локти согнуты под прямым углом, а в кистях рук, лежащих на столе, не чувствуется напряжения;
8. Локти не висят в воздухе, а комфортно располагаются на подлокотниках кресла или столешнице. Их позиция существенно не меняется при передвижении мышки;
9. Ноги упираются в твердую поверхность, распрямлены вперед, а не подогнуты под себя;

Длительность непрерывного труда и отдыха для сотрудников, постоянно работающих за персональной офисной техникой, регламентирована на законодательном уровне. Соответствующие нормы приводятся в СанПиН 2.2.2/2.4.1340-03, который устанавливает обязательную продолжительность перерывов в такой работе. Они зависят от длины рабочей смены сотрудника и определяются так: при 8-часовой рабочей смене в зависимости от тяжести и напряженности труда работника общая продолжительность перерывов за рабочий день должна составлять от 50 до 90 минут; при 12-часовой рабочей смене – от 80 до 140 минут.

**Выводы седьмой главы:**

В данной части выпускной квалификационной работы были рассмотрены мероприятия по технике безопасности и выделены требования охраны труда.

Так же были рассмотрены мероприятия противопожарной защиты, были определены категория и степень пожарной безопасности объекта и приняты решения для уменьшения вероятности пожара и его предотвращение.

**8.** **ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА**

**ЗАКЛЮЧЕНИЕ**

Моя дипломная работа - это автоматизированная информационная система для IT сервиса, разработанная на языке Python с использованием фреймворка Django в среде PyCharm. В результате работы я создал функциональный сайт, который позволяет сотрудникам быстро и удобно получать информацию, клиентам оставлять заявки на обслуживание и писать отзывы, а администраторам контролировать весь процесс работы.

В процессе анализа существующих решений были выявлены преимущества и недостатки конкурирующих систем. Было замечено, что ваш проект отличается более простым и понятным интерфейсом, а также более широким спектром функций.

В ходе разработки были использованы современные технические решения, такие как Django ORM, JavaScript, jSON, Pandas и другие. Также была проведена тестирование приложения и исправлены найденные ошибки.

В результате работы были достигнуты все поставленные цели, и создана полноценная информационная система для IT сервисов. Выводы по результатам работы указывают на то, что проект имеет большой потенциал для дальнейшего развития и улучшения.

**СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ**

* Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. - М.: Мир, 2016.
* Солодушкин, С. И. Web и DHTML : учебное пособие / С. И. Солодушкин, И. Ф. Юманова ; науч. ред. В. Г. Пименов ; М-во образования и науки Рос. Федерации, Урал. федер. ун-т. - Екатеринбург : Изд-во Уральского ун-та, 2018. - 128 с. - ISBN 978-5-7996-2410-1.
* Макфарланд Д. М17 Новая большая книга CSS. — СПб.: Питер, 2021. — 720 с.: ил. — (Серия «Бестселлеры O’Reilly»). ISBN 978-5-4461-1140-4.
* Робсон, Э. Изучаем HTML, XHTML и CSS : практическое руководство / Э. Робсон, Э. Фримен ; перевод В. В. Черника. - 2-е изд. - Санкт-Петербург : Питер, 2021. - 720 с. - (Серия «Head First O’Reilly»). - ISBN 978-5-4461-1247-0.
* Дронов, В. А. Django: практика создания Web-сайтов на Python : практическое руководство / В. А. Дронов. - Санкт-Петербург : БХВ-Петербург, 2016. - 528 с. - (Профессиональное программирование). - ISBN 978-5-9775-0421-8.
* Рощин, С. М. Современные интернет-технологии. Семь главных трендов : научно-популярное издание / С. М. Рощин. - 2-е изд. - Москва : Дашков и К, 2022. - 124 с. - ISBN 978-5-394-04846-3.
* Никитченко, И. И. Основы web-технологий : учебное пособие / И. И. Никитченко, К. Н. Мезенцев, О. В. Зинюк. - Москва : РИО Российской таможенной академии, 2020. - 140 с. - ISBN 978-5-9590-1126-0.
* Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2022. — 216 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5.
* Зачем нам UML? Или как сохранить себе нервы и время. - URL: https://habr.com/ru/post/458680/
* UML. - URL: https://blog.skillfactory.ru/glossary/uml/
* UML для бизнес-моделирования: зачем нужны диаграммы процессов. - URL: https://evergreens.com.ua/ru/articles/uml-diagrams.html
* Краткий путеводитель по семейству нотаций IDEF. - URL: https://www.antonpiskun.pro/kratkij-putevoditel-po-semejstvu-notaczij-idef/
* Руководство по веб-фреймворку Django. - URL: https://metanit.com/python/django/
* Уроки Django. - URL: https://itproger.com/course/django

Эффективный Django. URL: https://habr.com/ru/post/240463/

**ПРИЛОЖЕНИЕ 1. ЛИСТИНГ ПРОГРАММНЫХ МОДУЛЕЙ**

**ПРИЛОЖЕНИЕ 2. ПОЛИТИКА БЕЗОПАСНОСТИ**

**ПРИЛОЖЕНИЕ 3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Документ «Руководство пользователя» относится к пакету эксплуатационной документации. Основная цель руководства пользователя заключается в обеспечении пользователя необходимой информацией для самостоятельной работы с программой или автоматизированной системой. Таким образом, документ Руководство пользователя должен отвечать на следующие вопросы: что это за программа, что она может, что необходимо для обеспечения ее корректного функционирования и что делать в случае отказа системы.

**1. ВВЕДЕНИЕ**

**1.1 Область применения**

Система управления взаимоотношениями с клиентами, позволяющая автоматизировать процесс работы и организовать работу отдела маркетинга внутри небольшого предприятия.

**1.2 Краткое описание возможностей**

Система обладает следующими функциями:

• Авторизация в системе с использованием логина и пароля;

• Регистрация новых сотрудников;

• Авторизация в админ-панели;

• Добавление и обработка заявок;

• Добавление и редактирование задач, заказов, клиентов, сотрудников и продаж;

• Экспорт;

• Управление учетной записью

• Просмотр справки о странице

**1.3 Уровень подготовки пользователя**

Пользователь, работающий в информационной системе, должен обладать следующими знаниями:

–  Иметь опыт работы с операционной системой;

–  Иметь опыт с работой ПК и утилитами.

**1.4 Перечень эксплуатационной документации**

Пользователи должны быть ознакомлены с "Руководством пользователю".

**2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ**

Основная цель АИС - хранение, обеспечение эффективного поиска и передачи информации по соответствующим запросам для наиболее полного удовлетворения информационных запросов большого числа пользователей.

Для работы с клиентской частью web-приложения необходимо современный браузер, с поддержкой HTML5, CSS 3 и ECMAScript 2015 (ES6).

**3. ПОДГОТОВКА К РАБОТЕ**

**3.1 Состав и содержание носителя данных**

Папка программного продукта содержит файлы разрешений HTML, CSS, Python, JavaScript.

**3.2 Порядок загрузки программ и данных**

Систему устанавливает на предприятие Администратор и сам добавляет всех сотрудников в систему.

**3.3 Порядок проверки работоспособности**

Проверка работоспособности автоматизированной информационной системы IT-сервиса обычно включает в себя следующие шаги:

1. Проверка наличия необходимого программного обеспечения и его актуальности. Это включает в себя проверку версий используемых компонентов системы, проверку наличия необходимых библиотек и других зависимостей.

2. Проверка настроек системы. Необходимо убедиться, что конфигурационные файлы и параметры системы настроены правильно.

3. Проверка доступности ресурсов. Необходимо убедиться, что все используемые ресурсы, такие как базы данных, серверы и сетевое оборудование, доступны и работают нормально.

4. Проведение тестирования производительности. Необходимо убедиться, что система может обрабатывать большие объемы данных и нагрузки без существенных задержек или ошибок.

5. Проверка функциональности системы. Необходимо убедиться, что все функциональные возможности системы работают корректно и соответствуют требованиям заказчика.

6. Проверка авторизации. Необходимо убедиться, что система должным образом регистрирует все события и ошибки, чтобы обеспечить возможность быстрой диагностики проблем в случае их возникновения.

**4. ОПИСАНИЕ ОПЕРАЦИЙ**

**1) Редактирование профиля**

Для этого необходимо нажать на кнопку «Редактировать профиль» на странице Профиля (Рис. 4.1.)

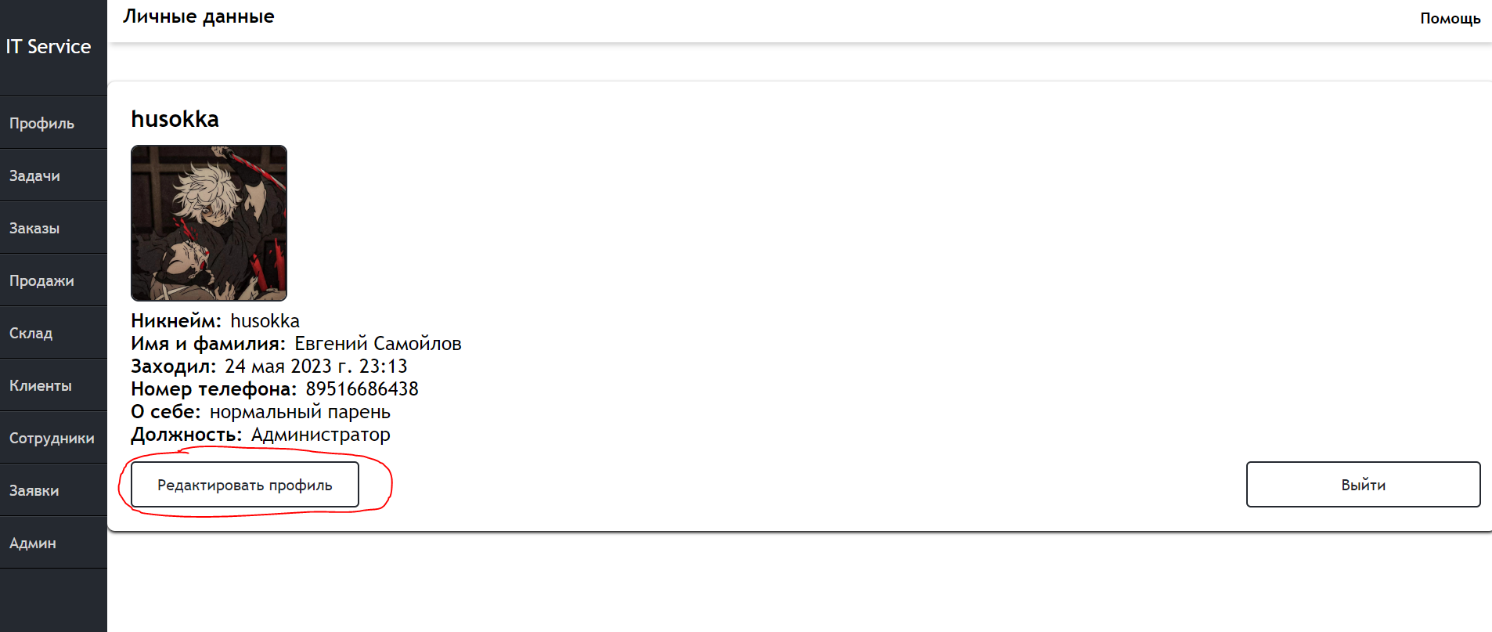


Рис 4.1. Кнопка «Редактировать профиль»

Происходит переход на страницу редактирования профиля (Рис 4.2.), нужно внести нужные изменения и нажать на кнопку подтверждения изменений.

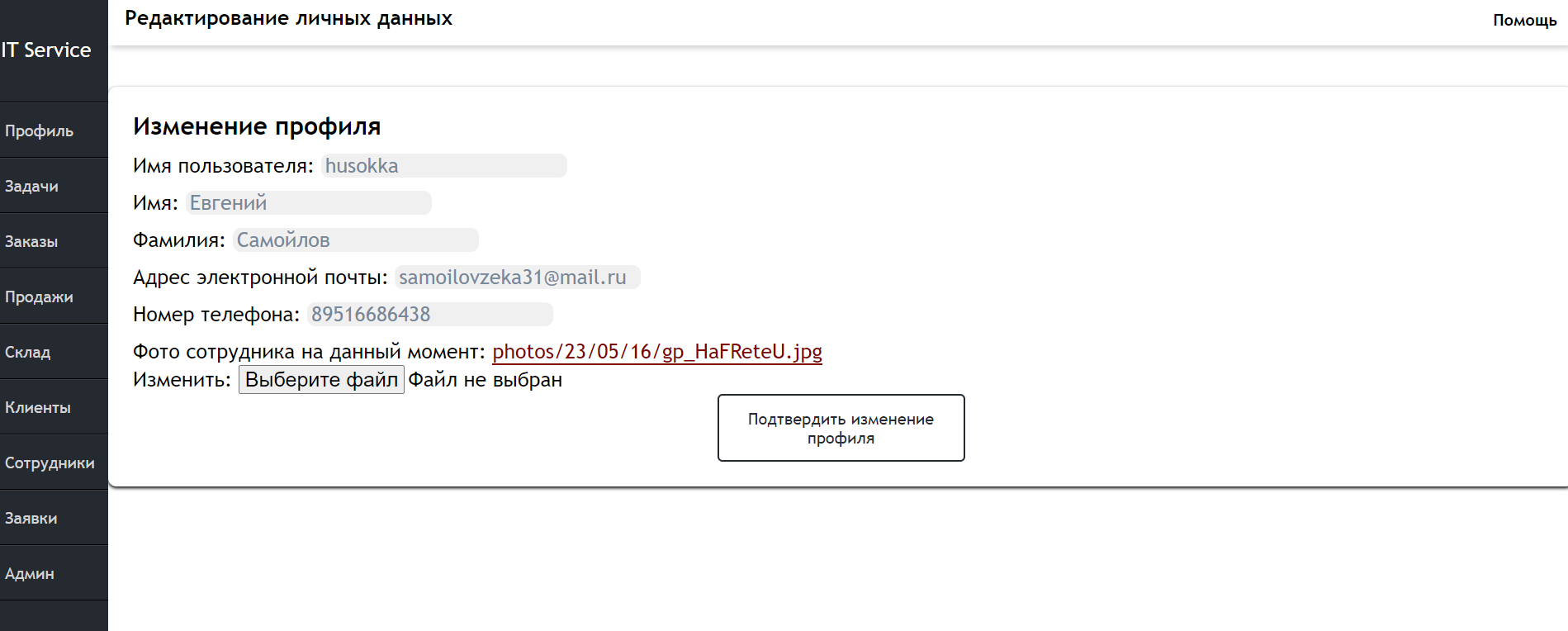


Рис 4.2. Страница редактирования профиля

**2)Добавление задачи**

Нужно перейти на вкладку «задачи» и нажать на кнопку «Добавить задачу» (Рис. 4.3.), после этого откроется модальное окно добавления задачи (Рис. 4.4.)

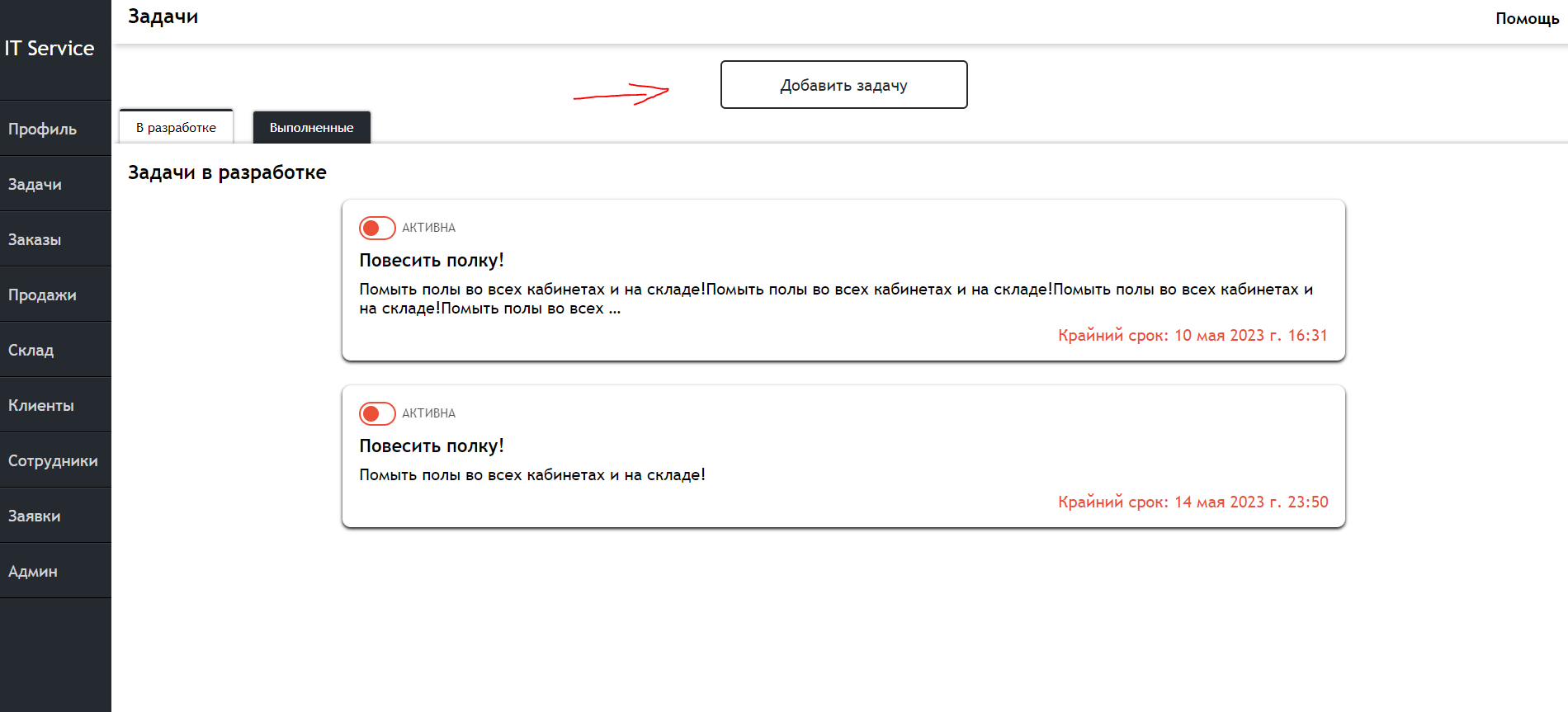


Рис 4.3. Страница Задач

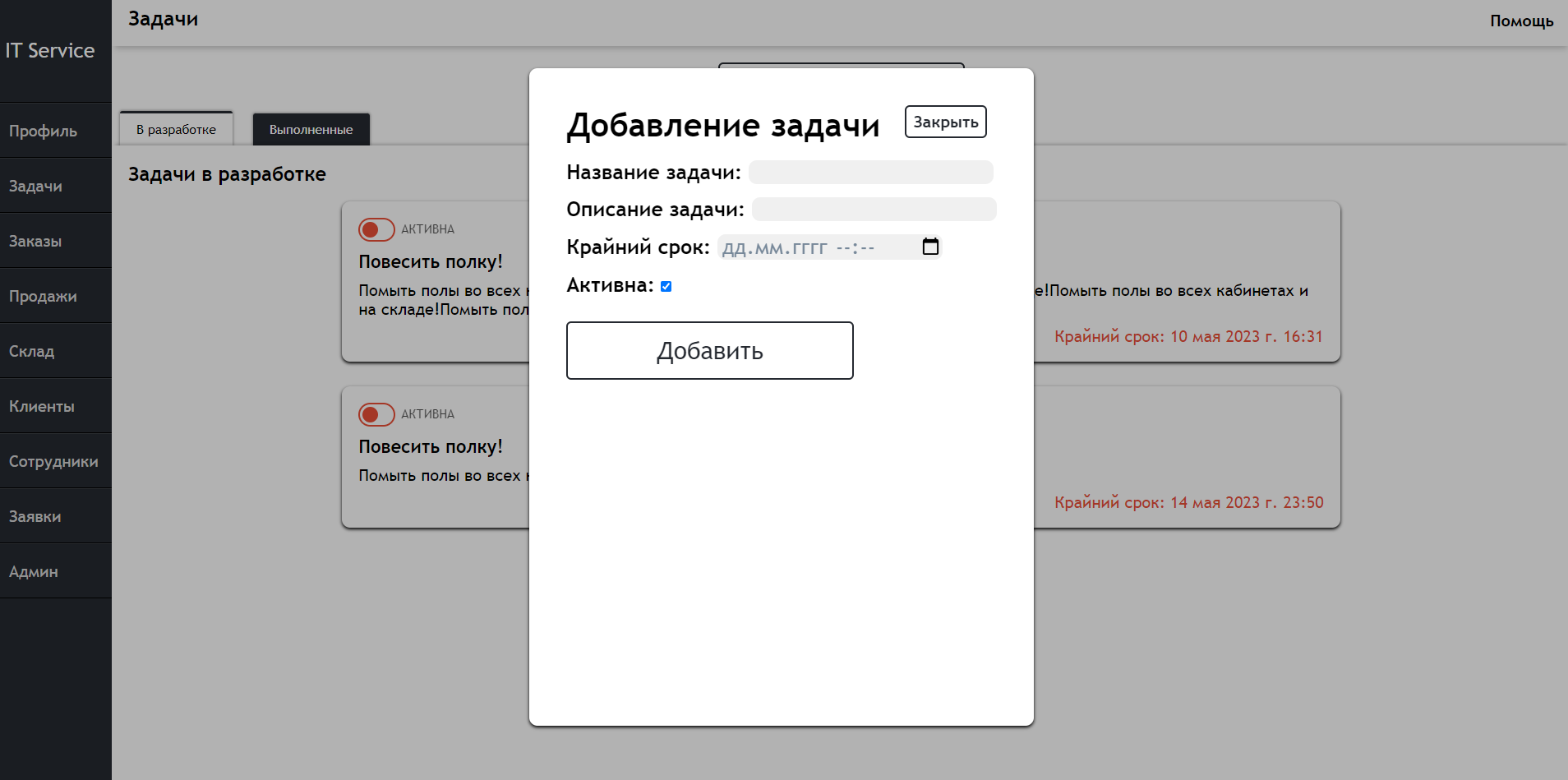


Рис 4.3. Модальное окно добавления задач

**5 АВАРИЙНЫЕ СИТУАЦИИ**

При возникновении аварийных ситуаций сообщите об этом администрации сайта, чтобы оперативно принять необходимые меры для устранения проблемы и предотвратить ее повторное возникновение в будущем.

Если вы обнаружили ошибку на сайте или заметили необычное поведение системы, пожалуйста, свяжитесь с администратором как можно скорее.

**6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ**

Для успешной работы с системой, сотрудник должен:

–  Иметь опыт работы с операционной системой;

–  Иметь опыт с работой ПК и утилитами.

**ПРИЛОЖЕНИЕ 4. РУКОВОДСТВО АДМИНИСТРАТОРА**

Руководство администратора – это составная часть эксплуатационной документации, которая разрабатывается на любую программу или автоматизированную систему. При помощи руководства администратора ответственные пользователи системы получают возможность управлять ее функционированием – выполнять определенные операции по обеспечению порядка работы АСУ, распределять права доступа к ней, редактировать данные и исправлять ошибки.

Руководство администратора адресовано лицу, задача которого — обеспечить определенный порядок функционирования системы. Обычно администратор считается пользователем системы, однако, при этом он наделен как особыми обязанностями, так и необходимыми для их выполнения привилегиями.

**1.НАЗНАЧЕНИЕ ПРОГРАММЫ**

**1.1 Назначение программы**

Цель системы – автоматизация рабочего процесса сервиса, с целью экономии времени

**1.2 Функции программы**

**1.2.1 Авторизация**

Авторизация в системе с использованием логина и пароля.

**1.2.2 Регистрация**

Добавление администратором новых сотрудников в систему.

**1.2.3 Админ-панель**

Авторизация в админ-панели.

**1.2.4 Добавление и изменение информации**

Добавление и редактирование задач, заказов, клиентов, сотрудников и продаж.

**1.2.5 Обработка заявок**

Добавление и обработка заявок.

**1.2.6 Экспорт файлов**

Экспорт информации о продажах

**1.2.7 Управление учетной записью**

Редактирование профиля в личном кабинете

**2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ**

**2.1 Аппаратные требования**

Для работы с клиентской частью web-приложения необходимо современный браузер, с поддержкой HTML5, CSS 3 и ECMAScript 2015 (ES6). Например:

• Google Chrome;

• Yandex браузер;

• Opera;

• Safari.

**2.2 Программные требования**

Для использования системы необходимы следующие системные требования:

ОС: Window 7, 8,10,11

Процессор: 1.66 GHz intel 5i

Оперативная память: 1024 MB ОЗУ

Видеокарта: HD Graphics

DirectX: Версии 10

**3. ВЫПОЛНЕНИЕ ПРОГРАММЫ**

**3.1. Загрузка и запуск программы**

**3.1.1 Загрузка программы**

Система находится на сервере. Добавляется Администратор, далее Администратор добавляет остальных сотрудников в систему

**3.1.2 Запуск программы**

Запуск выполняется системы выполняется администратором.

**3.2 Выполнение программы**

**3.2.1 Главное окно**

На главном экране отображаются 2 кнопки для «Отзывов», «Заявок» и кнопка для персонала «Вход в систему»

**3.2.1 Страница Профиля**

На странице Профиля представлена информация о сотруднике, так же есть кнопка редактирования профиля и кнопка выхода.

**3.2.2 Страница Задачи**

На странице рекомендаций представлены задачи коллектива, они подразделены на «В разработке» и «Выполненные», с возможностью их добавления.

**3.2.3 Страница Заказы**

На странице представлены заказы предприятия, они подразделены на «Активные» и «Выполненные», с возможностью их добавления.

**3.2.4 Страница Продажи**

На странице представлены продажи, с возможностью скачать Excel файл с информацией о продажах, с возможностью их добавления.

**3.2.5 Страница Склад**

На странице представлена информация о складе предприятия, с возможностью их добавления.

**3.2.6 Страница Клиенты**

На странице представлена информация о клиентах предприятия, с возможностью их добавления.

**3.2.7 Страница Сотрудники**

На странице представлена информация о сотрудниках предприятия, с возможностью их добавления Администратором.

**3.2.8 Страница Заявки**

На странице представлены заявки клиентов, они подразделены на «Необработанные» и «Обработанные».

**4.** **СООБЩЕНИЯ АДМИНИСТРАТОРУ**

Сообщения администратору, посредством графического интерфейса пользователя, появляются во всплывающем окне.

Типы сообщений об ошибках:

1) Некорректный пароль при регистрации нового сотрудника

Текст: «Введенные пароль слишком похож на имя пользователя»

Действия: Придумать более сложный пароль.

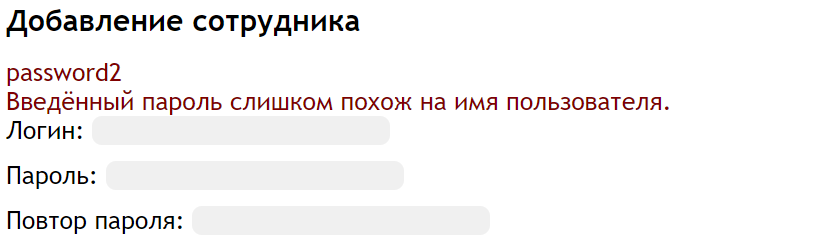


Рисунок 1. Отображение «Защита легкого пароля»

1) Некорректный логин при регистрации нового сотрудника

Текст: «Пользователь с таким именем уже существует»;

Действия: Использовать другое имя пользователя.

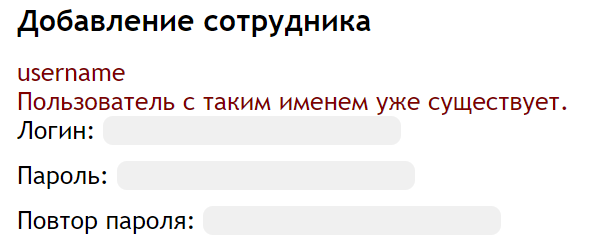


Рисунок 2. Ошибка «Пользователь с таким именем уже существует»

2) Попытка перейти на страницу системы неавторизованным

Текст: «Вы не авторизованы»;

Действия: Перейти на окно логина и авторизоваться.

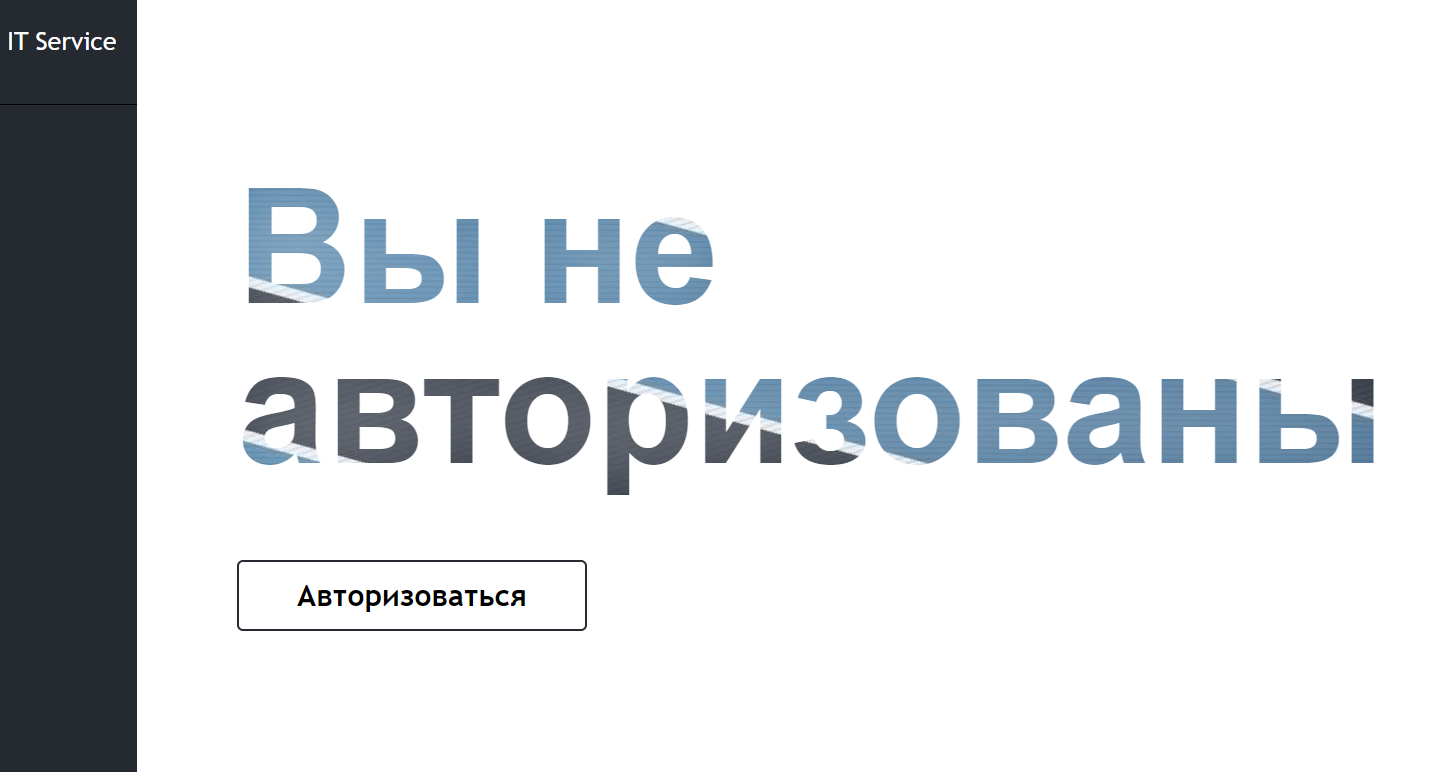


Рисунок 3. «Неавторизованный сотрудник»