**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 для IT сервиса** | | |
|  |  | | |
| 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 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Кривоносова Н.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)* | | | | | | |
| Квалификация | | | *Техник-программист* | | | | | | |
| Вид работы | | | *Дипломная работа* | | | | | | |
|  | | | *(дипломный проект, дипломная работа)* | | | | | | |
| Тема | **Разработка автоматизированной информационной** | | | | | | | | |
| **системы** **для IT сервиса** | | | | | | | | | |
|  |  | | | | | | | | |
|  | | | | | | | | | |
| 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 | Кривоносова Наталья Викторовна | | | | | | | | |
|  | (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 определений и условных обозначений: 6](#_Toc135490835)

[ВВЕДЕНИЕ 7](#_Toc135490836)

[1. ПОСТАНОВКА ЗАДАЧИ 8](#_Toc135490837)

[**1.1. Обзор аналогов** 8](#_Toc135490838)

[**1.2 SWOT-анализ** 13](#_Toc135490839)

[**1.3 PEST-анализ** 13](#_Toc135490840)

[**1.4 Формирование требований к программному продукту** 14](#_Toc135490841)

[**1.4.1 Бизнес-требования** 14](#_Toc135490842)

[**1.4.2 Пользовательские требования** 15](#_Toc135490843)

[**1.4.3 Функциональные требования к программному продукту** 15](#_Toc135490844)

[**1.4.4 Нефункциональные требования** 16](#_Toc135490845)

[**1.4.5 Ограничения** 17](#_Toc135490846)

[**1.4.6 Требования к интерфейсам** 18](#_Toc135490847)

[**1.4.7 Требования к данным** 18](#_Toc135490848)

[**1.5 Программные средства разработки** 22](#_Toc135490849)

[**1.6 Аппаратные средства разработки** 23](#_Toc135490850)

[2. ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ 24](#_Toc135490851)

[**2.1. Архитектура системы** 24](#_Toc135490852)

[**2.2. Моделирование основных сценариев системы** 25](#_Toc135490853)

[**2.3. Проектирование графического интерфейса пользователя** 25](#_Toc135490854)

[**2.4. Проектирование и разработка модели данных** 25](#_Toc135490855)

[3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА 25](#_Toc135490856)

**Перечень определений и 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:

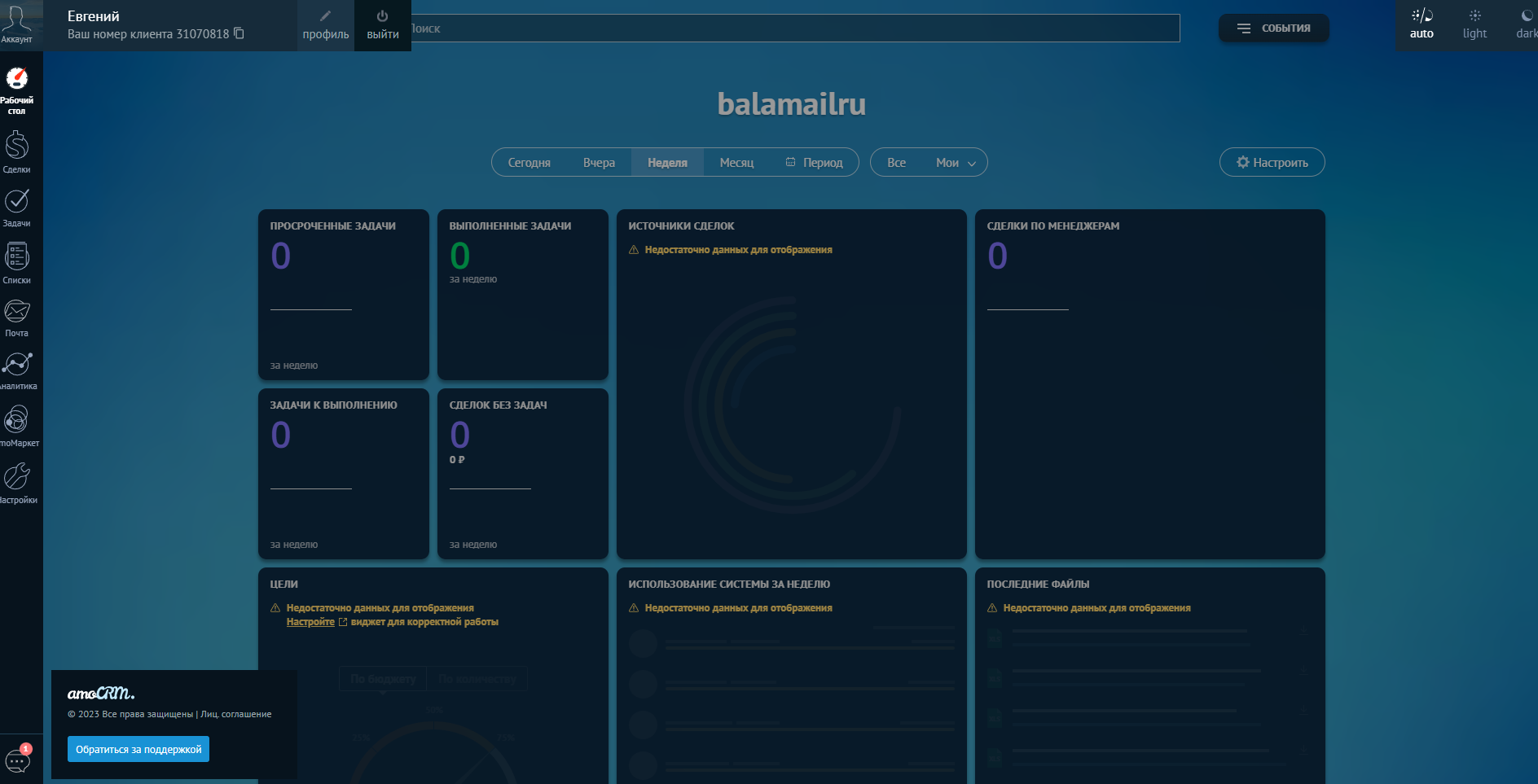
* Наглядная аналитика работы отдела.
* Настройка сложносоставных отчетов под нужды компании.
* Детализированный анализ работы колл-центра.
* Контроль над работой со сделками, в том числе и просроченными.
* Координирование работы менеджеров, распределение нагрузки на персонал.
* Планирование и разработка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

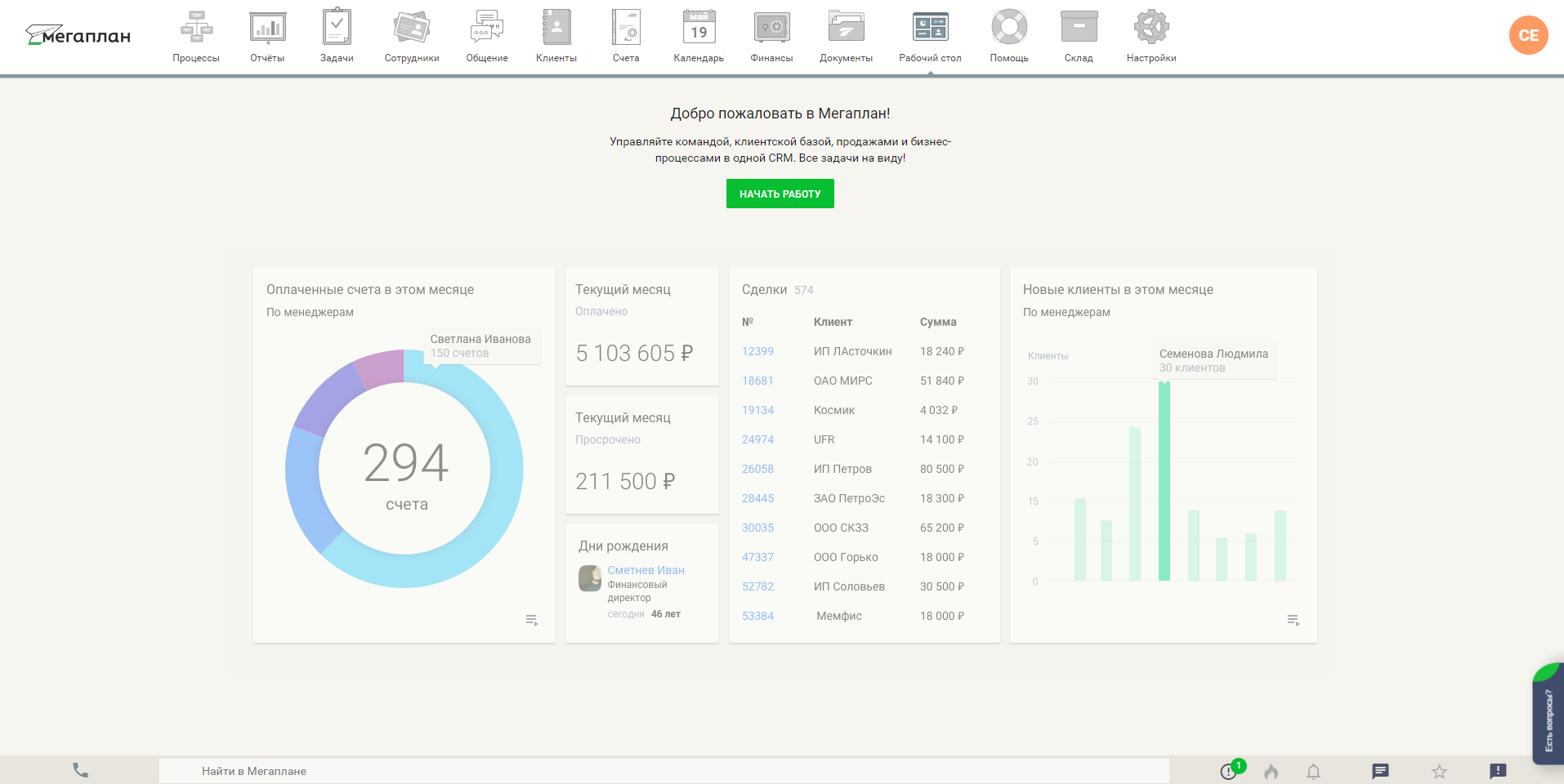


Рис. 1.2. Мегаплан

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.)

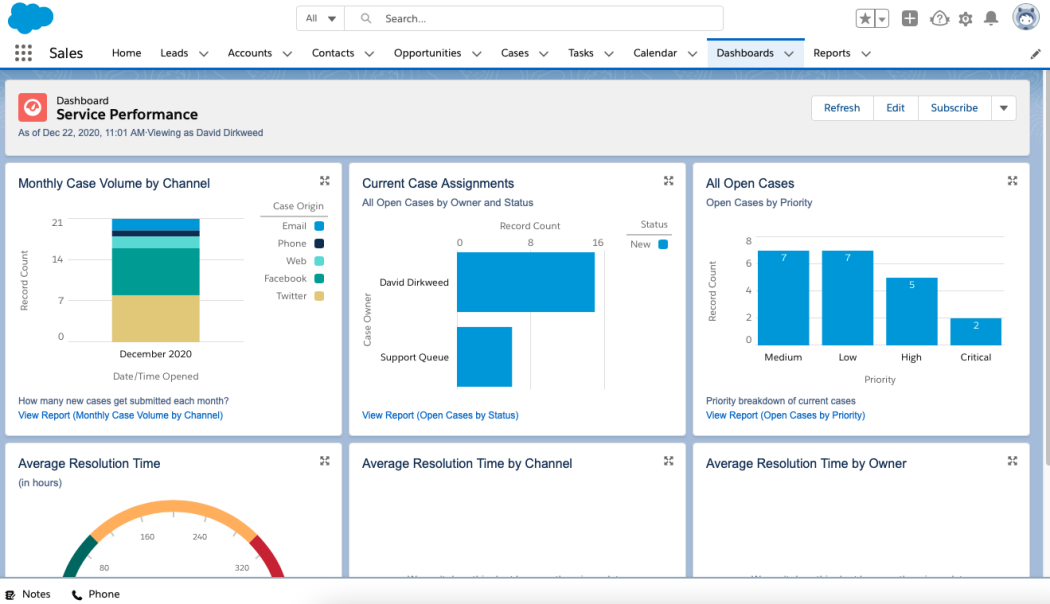


Рис. 1.3. Salesforce

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 корпоративного мессенджера, 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**

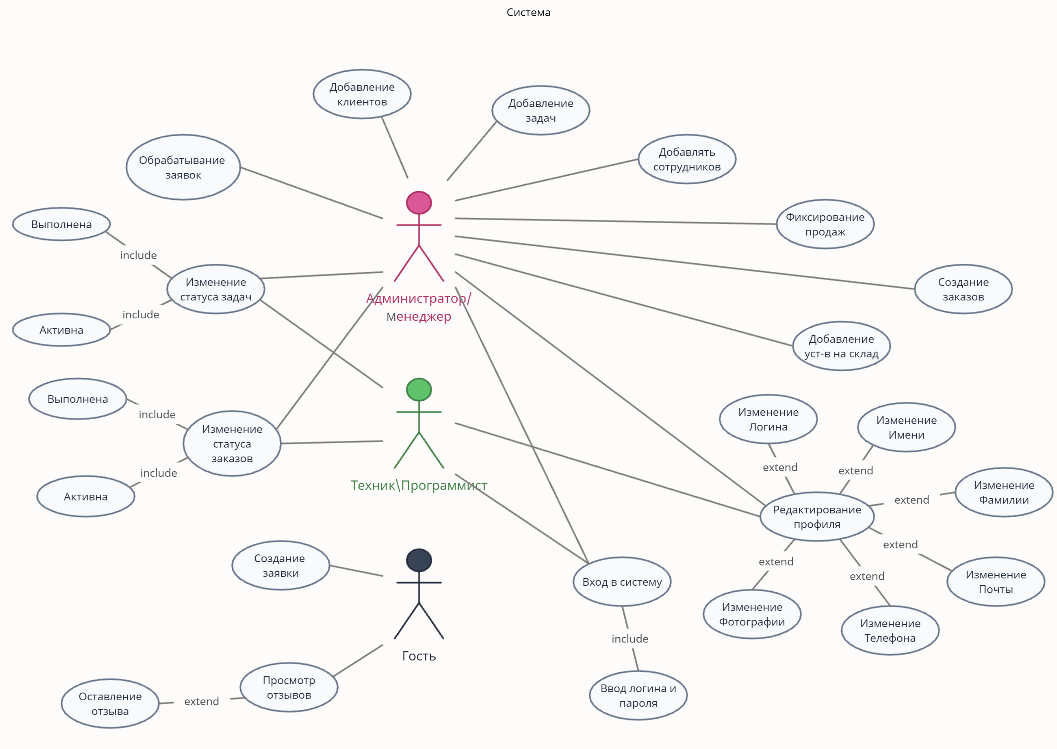
В настоящее время вся информация о принятых заказах хранится в электронных таблицах Microsoft Excel, что существенно затрудняло одновременную работу двум и более пользователям с данными, и тем самым отрицательно влияло на динамику работы сотрудников в условиях дефицита рабочего времени.

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)

Рис. 1.1. 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 сервисе 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 курсового проекта 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.

**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.

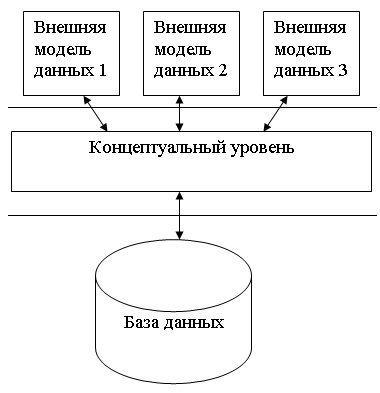


Рис. 2.1. Трехуровневая архитектура системы

**2.2. Моделирование основных сценариев системы**

**2.3. Проектирование графического интерфейса пользователя**

**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.

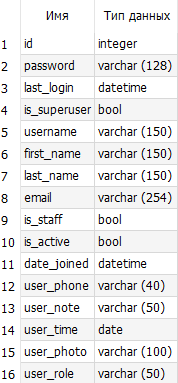


Рис 2.2. Таблица “users”

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.

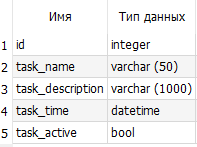


Рис 2.3. Таблица “ tasks ”

Описание полей таблицы “ users”

|  |  |  |
| --- | --- | --- |
| Название столбца | Описание | Тип данных |
| id | Уникальный идентификатор задачи | 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) |

**3. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**