Изображение выглядит как графическая вставка, Графика, мультфильм

Автоматически созданное описание

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка программного модуля**

**"Система учета сотрудников в офисе"**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент**  **группы 311ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Д.И. Кузнецов** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Н.И. Кручинкина** |

**Москва 2024**

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ 3**](#_Toc185510696)

[**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5**](#_Toc185510697)

[1.1 Предпроектное обследование 5](#_Toc185510698)

[1.2 Обзор и анализ существующих программных решений 5](#_Toc185510699)

[1.3 Постановка задач к программному продукту 6](#_Toc185510700)

[1.4 Выбор инструментальных средств реализации 8](#_Toc185510701)

[**2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ 10**](#_Toc185510702)

[2.1 Эскизное проектирование 10](#_Toc185510703)

[2.2 Проектирование базы данных 12](#_Toc185510704)

[2.3 Блок схемы основных функций 14](#_Toc185510705)

[2.4 Написание программного кода 16](#_Toc185510706)

[**3 ТЕСТИРОВАНИЕ МОДУЛЯ 19**](#_Toc185510707)

[3.1 Тестирование программного продукта 19](#_Toc185510708)

[**ЗАКЛЮЧЕНИЕ 22**](#_Toc185510709)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23**](#_Toc185510710)

# **ВВЕДЕНИЕ**

Актуальность темы курсового проекта "Учет сотрудников в офисе" обусловлена необходимостью автоматизации процессов управления персоналом в современных организациях. В условиях растущей конкуренции и цифровизации важным фактором успешной работы офиса является эффективное управление кадровыми ресурсами. Учет сотрудников позволяет не только систематизировать информацию о сотрудниках, но и оптимизировать процессы их работы, минимизировать ошибки, связанные с ручным вводом данных, и повысить производительность организации в целом.

Практическая значимость проекта заключается в разработке инструмента, который упростит работу отдела кадров, обеспечит точное хранение данных о сотрудниках и облегчит их последующее использование. Современные информационные технологии позволяют создавать удобные, гибкие и надежные системы учета, что делает тему разработки подобного программного обеспечения актуальной и востребованной.

**Цель работы:**  
Разработка программного модуля для учета сотрудников в офисе.

**Задачи:**

* Проанализировать существующие системы учета сотрудников и выявить их недостатки.
* Разработать структуру базы данных для хранения информации о сотрудниках.
* Создать пользовательский интерфейс для ввода, редактирования и просмотра данных.
* Обеспечить защиту данных сотрудников от несанкционированного доступа.
* Протестировать программный модуль и оценить его эффективность.

**Объект исследования:**  
Деятельность офиса в рамках управления сотрудниками.

**Предмет исследования:**  
Процесс учета информации о сотрудниках офиса.

Проект направлен на решение проблемы автоматизации учета сотрудников, что позволит повысить точность и скорость обработки данных, а также улучшить управляемость кадровыми ресурсами офиса.

**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

1.1 Предпроектное обследование

Предметная область "Учет сотрудников в офисе" охватывает процесс управления информацией о персонале, который включает сбор, обработку, хранение и анализ данных, необходимых для эффективного функционирования организации.

Традиционно учет сотрудников велся вручную с использованием бумажной документации, что приводило к определенным проблемам: избыточность данных, сложности с поиском информации, вероятность ошибок при вводе и обработке данных, а также трудности в хранении и обеспечении безопасности данных. Эти проблемы мотивируют необходимость внедрения автоматизированных систем учета, которые позволяют оптимизировать процессы, повысить их точность и снизить трудозатраты.

Современные подходы к учету сотрудников основываются на использовании информационных технологий для создания единой базы данных, что позволяет эффективно управлять рабочими процессами и улучшать качество обслуживания сотрудников. Внедрение таких систем дает возможность интегрировать данные, автоматизировать процессы и создавать прозрачную информационную среду для всех участников. Это требует разработки как программных решений, так и оптимизации организационных процессов внутри компании.

1.2 Обзор и анализ существующих программных решений

Примеры ИС и их характеристика:

Система управления персоналом "1С:Зарплата и Управление Персоналом":

Эта система позволяет автоматизировать процесс учета всех сотрудников организации, их должностей, начисления заработной платы, а также учета рабочего времени и отпусков. Важной особенностью является интеграция с другими системами 1С для учета финансов и документооборота. Преимущество "1С" заключается в широких возможностях кастомизации под потребности бизнеса, однако для ее успешной работы требуется наличие специализированных знаний у пользователей.

Система учета сотрудников "BambooHR":

"BambooHR" ориентирована на малый и средний бизнес и предлагает удобный интерфейс для учета сотрудников, их данных, рабочего времени, отпуска и других важных аспектов. Особенностью системы является наличие инструментов для анализа и отчетности по HR-метрикам. Несмотря на высокую функциональность, система требует постоянного обновления данных и может не подходить для крупных организаций с более сложными потребностями.

1.3 Постановка задач к программному продукту

Разработка системы учета сотрудников в офисе включает следующие задачи:

Создание базы данных:

* Разработка структуры базы данных для хранения информации о сотрудниках.
* Реализация таблиц для хранения данных о сотрудниках (ФИО, должность, дата начала работы)

Реализация функционала для работы с сотрудниками:

* Добавление сотрудников: создание интерфейса для регистрации новых сотрудников в базе данных.
* Просмотр сотрудников: разработка функционала для отображения всех зарегистрированных сотрудников.
* Редактирование данных сотрудников: возможность изменения информации о сотрудниках (например, должность).
* Удаление сотрудников: создание функционала для удаления сотрудников из системы.
* Поиск сотрудников: разработка функции поиска сотрудников по различным параметрам (например, email, должность, отдел).

Реализация функционала для работы с отпусками:

* Регистрация отпусков: интерфейс для ввода данных о принятии и завершении отпуска.

Реализация функционала для пользователей системы

* Авторизация пользователей: система авторизации для доступа к различным функциям в зависимости от роли пользователя.

Разработка графического интерфейса

* Создание интуитивно понятного интерфейса для работы с системой, позволяющего легко ориентироваться между различными функциями.
* Обеспечение адаптации интерфейса под различные роли пользователей с возможностью ограничений доступа к определенным функциям.

Тестирование и отладка

* Проведение тестирования на наличие ошибок и недостатков.
* Обеспечение корректной работы всех функций и устранение выявленных проблем.
* Проведение отладки системы для гарантии стабильности работы на всех этапах.

1.4 Выбор инструментальных средств реализации

* Язык программирования Python 3.12
* Интерфейс PyQt6
* База данных SQLite
* Тестирование Pytest
* ORM SqlAlchemy
* Создание исполняемого файла PyInstaller
* Асинхронное программирование AsyncIO
* Редактор кода VS Code
* Валидация данных Pydantic

**2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ**

2.1 Эскизное проектирование

Система включает в себя следующие функции:

* Регистрация и авторизация в систему
* Просмотр информации о структуре компании
* Обеспечение безопасности данных через хеширование паролей
* Управление структурой компании, которое включает в себя редактирование пользователей, должностей, отделов, а также создание записей об отпусках

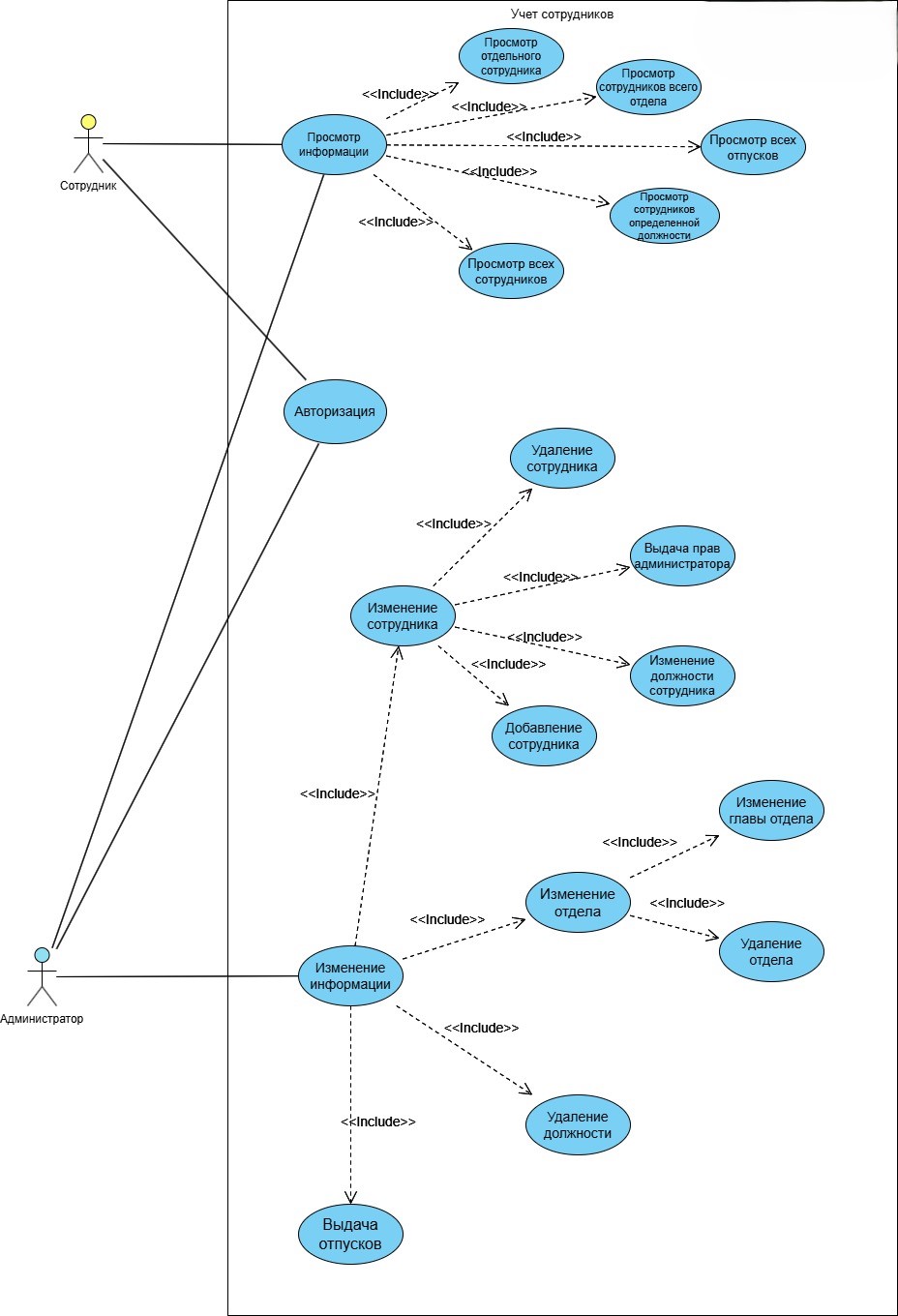


Рисунок 1 – Use-case диаграмма

2.2 Проектирование базы данных

Структура базы данных состоит из 4 таблиц:

* Пользователи (users)
* Должности (positions)
* Отделы (sections)
* Отпуска (vacations)

Таблица 1 - users

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id | Int PK | Уникальный идентификатор пользователя |
| Name | Varchar Not NULL | Имя пользователя |
| Surname | Varchar Not NULL | Фамилия пользователя |
| Position name | Varchar FK positions.name Not Null | Должность пользователя из таблицы positions |
| Email | Varchar Unique Not NULL | Уникальная почта пользователя |
| Hashed password | Varchar Not NULL | Пароль пользователя, хранится в виде хеша |
| Is superuser | Boolean Not NULL | Права администратора пользователя |
| Joined at | Date Not NULL | Дата добавления в систему пользователя |
| Birthday | Date Not NULL | День рождения пользователя |

Таблица 2 – positions

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id | INT PK | Уникальный идентификатор должности |
| Section name | Varchar FK sections.name Not NULL | Название отдела из таблицы section |
| Name | Varchar Unique Not NULL | Название должности, уникальное |

Таблица 3 – sections

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id | INT PK | Уникальный идентификатор отдела |
| Head email | Varchar FK users.email Not NULL | Глава отдела из таблицы users |
| Name | Varchar Unique Not NULL | Название отдела, уникальное |

Таблица 4 – vacations

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип | Описание |
| Id | INT PK | Уникальный идентификатор отпуска/больничного |
| Giver email | Varchar FK users.email | Почтовый адрес пользователя, выдавшего отпуск |
| Receiver email | Varchar FK users.email | Почтовый адрес пользователя, получившего отпуск |
| Start date | Date Not NULL | Начало действия отпуска |
| End date | Date Not NULL | Конец действия отпуска |
| Description | Varchar | Описание, которые может включать в себя причину выдачи отпуска |

Изображение выглядит как текст, диаграмма, число, снимок экрана

Автоматически созданное описание

Рисунок 2 – Er диаграмма

2.3 Блок схемы основных функций

Изображение выглядит как текст, диаграмма, План, зарисовка

Автоматически созданное описание

Рисунок 3 – блок схема регистрации

Изображение выглядит как текст, диаграмма, линия, План

Автоматически созданное описание

Рисунок 4 – блок схема авторизации

2.4 Написание программного кода

Реализация основных функций

Взаимодействие с таблицей users:

1.Регистрация

async def register\_user(data: UserCreate):

    async with get\_session() as session:

        # Хеширование введенного пароля

        data.hashed\_password = hash\_password(data.hashed\_password)

        new\_user = User(

            email=data.email,

            name=data.name,

            surname=data.surname,

            hashed\_password=data.hashed\_password,

            is\_superuser=data.is\_superuser,

            birthday=data.birthday,

            position\_name=data.position\_name,

        )

        session.add(new\_user)

        # Попытка создания нового пользователя

        try:

            await session.commit()

        except IntegrityError:

            logger.error(f"Fail to register user {data.email}")

            return 0

        # При успешном создании пользователя функция вернет 1 и логгер запишет успешное создание

        logger.info(f"Registered user: {data.email}")

        return 1

2.Авторизация

async def login\_user(data: UserLogin):

    async with get\_session() as session:

        # Запрос в базу данных на поиск пользователя с указанным email

        result = await session.execute(select(User).where(User.email == data.email))

        user = result.scalar\_one\_or\_none()

# Если пользователь не найден или хеш пароля не совпал функция возвращает 0 и логгер записывает ошибку

        if user is None or not checkpw(

            data.password.encode("utf-8"), user.hashed\_password.encode("utf-8")

        ):

            logger.error(f"Login failed for {data.email}: Invalid email or password")

            return 0

# При успешном входе в систему функция возвращает 1 и логгер записывает успешный вход в систему

        logger.info(f"Logged in: {data.email}")

        return 1

**3 ТЕСТИРОВАНИЕ МОДУЛЯ**

В ходе тестирования использовался фреймворк Pytest и библиотека faker для создания случайных данных

3.1 Тестирование программного продукта

# Тест входа в аккаунт

@pytest.mark.asyncio

async def test\_login\_user():

    await login\_user(UserLogin(email="root@example.com", password="root"))

# Тест выдачи прав администратора

@pytest.mark.asyncio

async def test\_update\_user():

    await update\_user(fake.get\_email())

# Тест просмотра информации о пользователе

@pytest.mark.asyncio

async def test\_user\_info():

    await get\_user\_by\_email(fake.get\_email())

# Тест удаления пользователя

@pytest.mark.asyncio

async def test\_delete\_user():

    with pytest.raises(AssertionError):

        result = await delete\_user(100)

        assert result == 1

# Тест смены должности пользователя

@pytest.mark.asyncio

async def test\_change\_user\_position():

    await change\_user\_position(

        email="root@example.com", new\_position\_name=fake.it\_job()

    )

# Тест добавления новой записи об отпуске

@pytest.mark.asyncio

async def test\_vacation\_add():

    await add\_vacation(

        VacationCreate(

            giver\_email=fake.get\_email(),

            receiver\_email=fake.get\_email(),

            start\_date=fake.date\_this\_month(before\_today=True, after\_today=False),

            end\_date=fake.date\_this\_month(before\_today=False, after\_today=True),

            description=fake.text(60),

        )

    )

# Тест просмотра всех отпусков

@pytest.mark.asyncio

async def test\_get\_all\_vacations():

    result = await get\_all\_vacations()

# Тест просмотра отпуска

@pytest.mark.asyncio

async def test\_get\_vacation():

    result = await get\_vacation(fake.random\_int(1, 11))

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы была рассмотрена система учета сотрудников в офисе, её значимость для эффективного управления персоналом и оптимизации рабочих процессов. Анализ существующих методов и инструментов учета показал, что внедрение автоматизированных систем учета значительно повышает точность и скорость обработки данных.

Предложенная система учета сотрудников основывается на использовании современных информационных технологий, что позволяет минимизировать ошибки, связанные с человеческим фактором, и автоматизировать рутинные задачи. Разработка и внедрение такой системы помогут компаниям улучшить внутренние процессы, обеспечить прозрачность в учете рабочего времени и повысить эффективность управления персоналом.

В заключение можно отметить, что автоматизация учета сотрудников в офисе не только способствует улучшению качества работы, но и создаёт основу для дальнейшего развития компании, позволяя сосредоточиться на более важных задачах.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Стандарты и законодательные материалы:

* ГОСТ Р ИСО/МЭК 12207-2019
* ГОСТ Р ИСО 25010-2015
* ГОСТ 19.101-77(ЕСПД)

Официальная документация:

* Python: <https://docs.python.org/3/>
* PyQt6: <https://doc.qt.io/qtforpython-6/>
* SQLite: <https://www.sqlite.org/docs.html>
* Pytest: <https://docs.pytest.org/en/stable/>
* SQLAlchemy: <https://docs.sqlalchemy.org/en/14/>
* AsyncIO: <https://docs.micropython.org/en/latest/library/asyncio.html>
* Pydantic: <https://pydantic-docs.helpmanual.io/>

Литература по программированию:

* «Изучаем Python» – Марк Лутц
* «Python – это просто» – Нилаб Нисчал

Дополнительные материалы:

* <https://stackoverflow.com/>
* <https://habr.com/ru/articles/>
* <https://www.reddit.com/>

GitHub репозиторий программного продукта: <https://github.com/wsdm777/deskapp>