

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ
КУРСОВОЙ ПРОЕКТ
ЗАЩИЩЕН С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

| | | |
|---|---------------|----------------------------------|
| канд. техн. наук, доцент должность, уч. степень, звание | подпись, дата | Т. И. Белая инициалы, фамилия |
|---|---------------|----------------------------------|

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ

Разработка приложения для контроля доступа в организацию

по дисциплине: ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ СИСТЕМ

РАБОТУ ВЫПОЛНИЛ

| | | | |
|---------------|-------|---------------|------------------------------------|
| СТУДЕНТ ГР. № | 4134К | подпись, дата | Столяров Н.С. инициалы, фамилия |
|---------------|-------|---------------|------------------------------------|

Санкт-Петербург 2024

РЕФЕРАТ

ОТЧЕТ с 107., 25 рис., 3 прил.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ КОНТРОЛЕМ ПРОПУСКНОГО ПУНКТА

Управление пропускным пунктом является важным аспектом обеспечения безопасности и контроля доступа в компании. Эффективная система контроля позволяет отслеживать, кто входит и выходит, что способствует повышению уровня безопасности и упрощению учета рабочего времени.

Система контроля доступа помогает минимизировать риски несанкционированного входа, обеспечивая соблюдение установленных правил и процедур. Это включает в себя идентификацию пользователей и регистрацию их действий.

Анализ данных о проходах позволяет выявлять закономерности и аномалии, что может быть полезно для оптимизации работы пропускного пункта и повышения его эффективности.

Интеграция современных технологий в систему управления пропускным пунктом позволяет автоматизировать процессы, улучшить учет и обеспечить удаленный доступ к информации о проходах.

Этика и социальная ответственность в управлении доступом важны для соблюдения прав и свобод граждан, а также учета интересов всех заинтересованных сторон, что способствует созданию безопасной и комфортной среды для работы и посещения.

СОДЕРЖАНИЕ

| | |
|--|----|
| РЕФЕРАТ..... | 2 |
| ВВЕДЕНИЕ..... | 5 |
| 1. ЦЕЛИ И НАЗНАЧЕНИЕ СОЗДАНИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ..... | 6 |
| 1.1 Назначение программы..... | 6 |
| 1.2 Бизнес-функции, для автоматизации которых предназначена системы..... | 6 |
| 2. ПРОЕКТИРОВАНИЕ И КОМПЛЕКСНОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ..... | 8 |
| 2.1 Характеристика и структура программы..... | 8 |
| 2.2 Описание основных особенностей программы..... | 9 |
| 2.3 Основные пользователи системы..... | 10 |

| | |
|---|-----------|
| 2.4 Функциональное моделирование в методике IDEF0..... | 10 |
| 2.4.1 Контекстная диаграмма..... | 10 |
| 2.4.3 Декомпозиция контекстной диаграммы..... | 11 |
| 2.4.3 Декомпозиция задачи A1..... | 12 |
| 2.4.4 Декомпозиция задачи A2..... | 13 |
| 2.4.5 Декомпозиция задачи A3..... | 14 |
| 2.4.6 Декомпозиция задачи A4..... | 15 |
| 2.4.7 Дерево узлов..... | 16 |
| 2.5 Анализ результатов функционального моделирования в методике IDEF..... | 16 |
| 2.6 Объектное моделирование в методике UML..... | 17 |
| 2.6.1 Диаграмма вариантов использования..... | 17 |
| 2.6.2 Диаграмма компонентов..... | 18 |
| 2.6.3 Диаграмма пакетов..... | 19 |
| 2.6.4 Диаграмма размещения..... | 20 |
| 3. ТРЕБОВАНИЯ К ИНФОРМАЦИОННОЙ СИСТЕМЕ..... | 21 |
| 3.2 Функциональные требования к системе..... | 21 |
| 3.2.1 Загрузка и запуск программы..... | 21 |
| 3.2.2 Страница учета рабочего времени..... | 23 |
| 4. РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ..... | 24 |
| 4.1 Описание архитектуры приложения..... | 24 |
| 4.2 Методы разработки программного обеспечения, использованные в проекте..... | 24 |
| 4.3 Используемые технологии и паттерны проектирования..... | 24 |
| 4.4 Особенности реализации на Fastapi..... | 25 |
| 4.5 Интеграция с базами данных..... | 26 |
| 4.6 Описание ключевых классов и методов..... | 27 |
| 4.7 Описание интерфейса и его удобства для пользователя..... | 31 |
| 4.10 Скриншоты интерфейса и его удобства для пользователя..... | 32 |
| 4.10.1 Основные навигационные кнопки..... | 32 |
| 4.10.2 Вход в систему..... | 33 |
| 4.10.3 Главное меню..... | 33 |
| 5 ТЕСТИРОВАНИЕ ПРОГРАММЫ..... | 43 |
| 5.1 Методы тестирования, использованные для проверки функциональности и надежности программы..... | 43 |
| 5.2 Результаты тестирования и обработка ошибок..... | 43 |
| ЗАКЛЮЧЕНИЕ..... | 44 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 45 |
| ПРИЛОЖЕНИЕ А..... | 46 |
| ПРИЛОЖЕНИЕ | |
| Б..... | 72 |
| ПРИЛОЖЕНИЕ | |
| В..... | 96 |

ВВЕДЕНИЕ

1. Актуальность программы

В современном мире эффективное управление системами контроля доступа становится все более важным для обеспечения безопасности и организации работы на объектах. Точность и надежность в мониторинге, анализе данных и принятии решений по контролю доступа напрямую влияют на безопасность и эффективность функционирования объектов. В связи с этим разработка и использование программного обеспечения для управления камерами видеонаблюдения и контроля пропускных пунктов становятся ключевыми аспектами в современных системах безопасности.

2.1 Цели и задачи проекта

Цель данного проекта - создание системы для управления камерами видеонаблюдения и контроля пропускных пунктов.

Система включает модули для отображения видеопотоков, регистрации входа и выхода, а также анализа данных о перемещениях.

Основные задачи проекта включают:

Разработку надежной и масштабируемой архитектуры информационной системы для управления камерами и контроля доступа, обеспечивающей эффективное функционирование и обработку данных о перемещениях.

Реализацию функционала для учета всех событий доступа, включая регистрацию входа и выхода, управление правами доступа, а также ведение истории событий.

Создание удобного пользовательского интерфейса, который позволит сотрудникам службы безопасности легко просматривать видеопотоки, анализировать данные о перемещениях и формировать необходимые отчеты.

Интеграцию с существующими системами безопасности и управления объектами, такими как системы учета сотрудников или управления охраной, для обеспечения единой базы данных и согласованности информации между различными подразделениями.

3. Обзор существующих аналогов и их недостатки

Существующие аналоги программного обеспечения для управления системами видеонаблюдения и контроля доступа часто не удовлетворяют потребностям пользователей из-за их ограниченной гибкости и недостаточной точности в учете специфических требований. Ограниченный функционал и недостаточная возможность интеграции с другими системами могут препятствовать эффективному управлению безопасностью объектов. В результате, разработка собственной информационной системы становится необходимой для обеспечения оптимального контроля доступа и удовлетворения уникальных потребностей каждого объекта.

4. Методы разработки программного обеспечения, использованные в проекте

При разработке программного обеспечения для управления камерами видеонаблюдения и контроля доступа были использованы современные методы программирования и технологии, среди которых важное место занимает фреймворк FastAPI. Этот выбор обеспечил высокую степень надежности и безопасности системы, а также позволил эффективно реализовать требования по интерфейсу и функциональности.

В качестве основного метода разработки была выбрана итеративная модель, которая способствует постепенному совершенствованию и доработке функционала системы с учетом специфики работы служб безопасности. Этот подход позволяет быстрее реагировать на изменения в требованиях, учитывать обратную связь от пользователей и постепенно совершенствовать продукт, обеспечивая лучшее соответствие потребностям бизнеса.

2. ПРОЕКТИРОВАНИЕ И КОМПЛЕКСНОЕ МОДЕЛИРОВАНИЕ ПРИЛОЖЕНИЯ

2.1 Характеристика и структура программы

В информационную систему управления камерами видеонаблюдения и контроля доступа входят следующие функциональные подсистемы:

Подсистема видеонаблюдения: обеспечивает отображение и запись видеопотоков с камер.

Подсистема обработки данных: включает данные о входе и выходе сотрудников, а также события доступа. Она обеспечивает корректное заполнение и редактирование данных, а также удаление записей по мере необходимости.

Подсистема управления доступом: отвечает за управление правами доступа и настройку уровней безопасности для различных категорий пользователей.

Подсистема координации: организует слаженную работу всех подсистем, обеспечивая синхронизацию данных и координацию действий между различными подразделениями для повышения общей эффективности.

Подсистема хранения данных: отвечает за хранение всей информации, связанной с видеонаблюдением и контролем доступа, включая записи событий, настройки камер и другие документы.

Информационный обмен между компонентами системы осуществляется через API, реализованный с использованием FastAPI, что обеспечивает высокую скорость и эффективность взаимодействия.

Характеристики подсистем:

Подсистема видеонаблюдения: позволяет легко управлять видеопотоками, обеспечивая доступ к записям в реальном времени.

Подсистема обработки данных: включает в себя данные о присутствии, событиях доступа и других параметрах. Она обеспечивает корректное заполнение и редактирование данных, а также удаление записей по мере необходимости.

Подсистема управления доступом: позволяет настраивать права доступа для различных пользователей и групп, обеспечивая безопасность и контроль.

Подсистема координации: обеспечивает синхронизацию данных и координацию действий между различными подразделениями компании, что способствует повышению общей эффективности.

Подсистема хранения данных: отвечает за хранение всей информации, связанной с видеонаблюдением и контролем доступа, включая записи событий и настройки системы.

Эти подсистемы работают в тесной взаимосвязи, обеспечивая точный учет и контроль доступа, что способствует оптимизации рабочих процессов и повышению общей безопасности объекта.

2.2 Описание основных особенностей программы

Архитектура разрабатываемой программы для управления камерами видеонаблюдения и контроля доступа основана на передовых принципах проектирования и использует современные технологии для обеспечения эффективности, масштабируемости и безопасности. Основой архитектуры является фреймворк FastAPI, который обеспечивает высокую производительность и простоту в разработке RESTful API.

FastAPI — это современный веб-фреймворк для создания API на Python, который позволяет быстро разрабатывать приложения с высокой производительностью. Он поддерживает асинхронное программирование и автоматическую генерацию документации, что упрощает процесс разработки и интеграции.

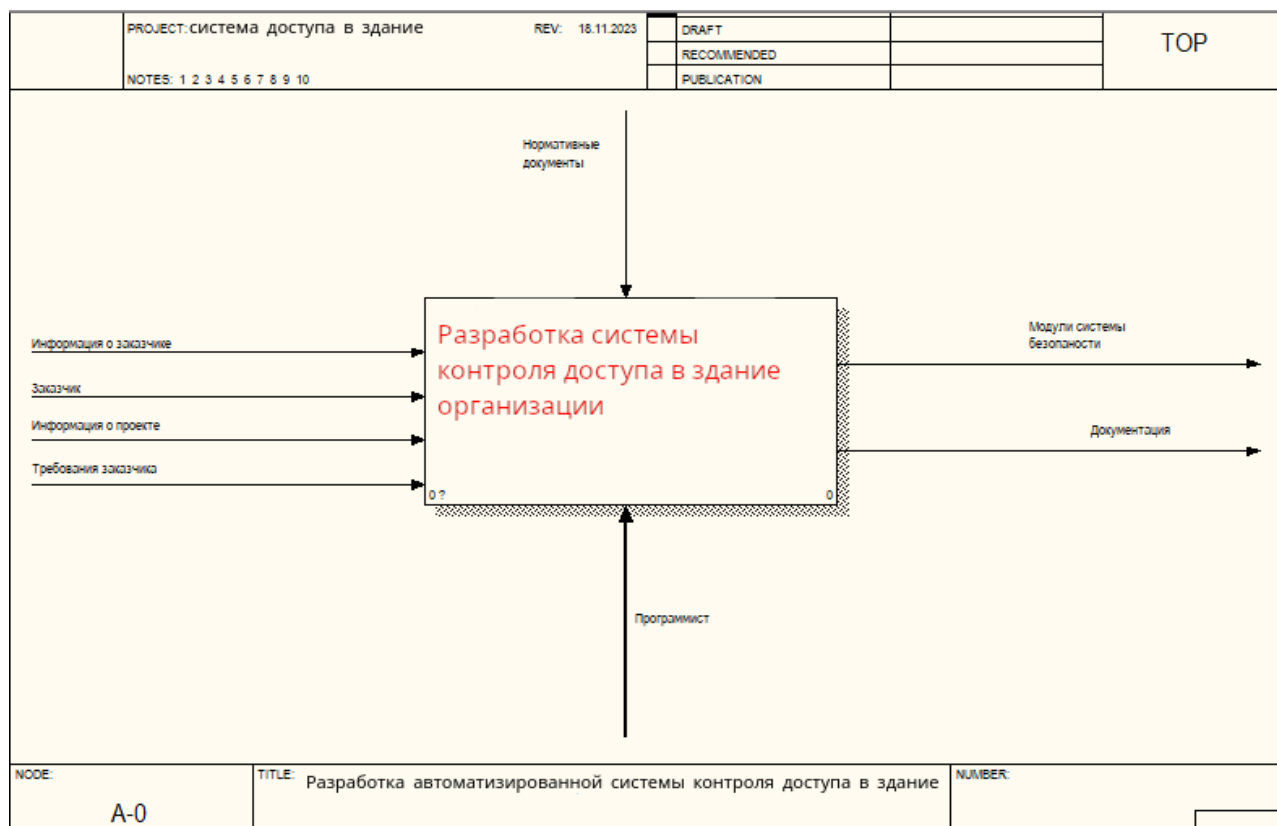
MongoDB — это NoSQL база данных, которая обеспечивает гибкость в хранении данных и масштабируемость. Она идеально подходит для хранения неструктурированных данных, таких как записи событий и настройки системы, что делает ее отличным выбором для приложений, требующих высокой производительности и быстрого доступа к данным.

Таким образом, использование FastAPI и MongoDB в проекте позволяет создать эффективную, масштабируемую и безопасную систему для управления камерами видеонаблюдения и контроля доступа, соответствующую современным требованиям безопасности и удобства использования.

2.3 Функциональное моделирование в методике IDEF0

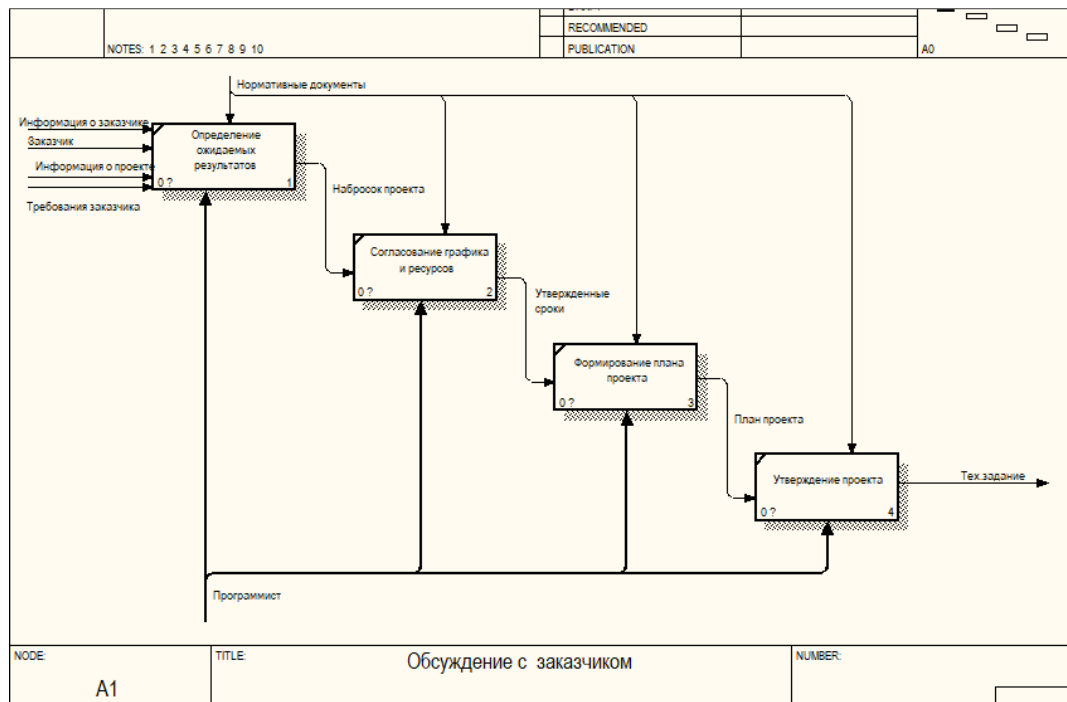
DEF0 используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции.

2.4.1 Контекстная диаграмма

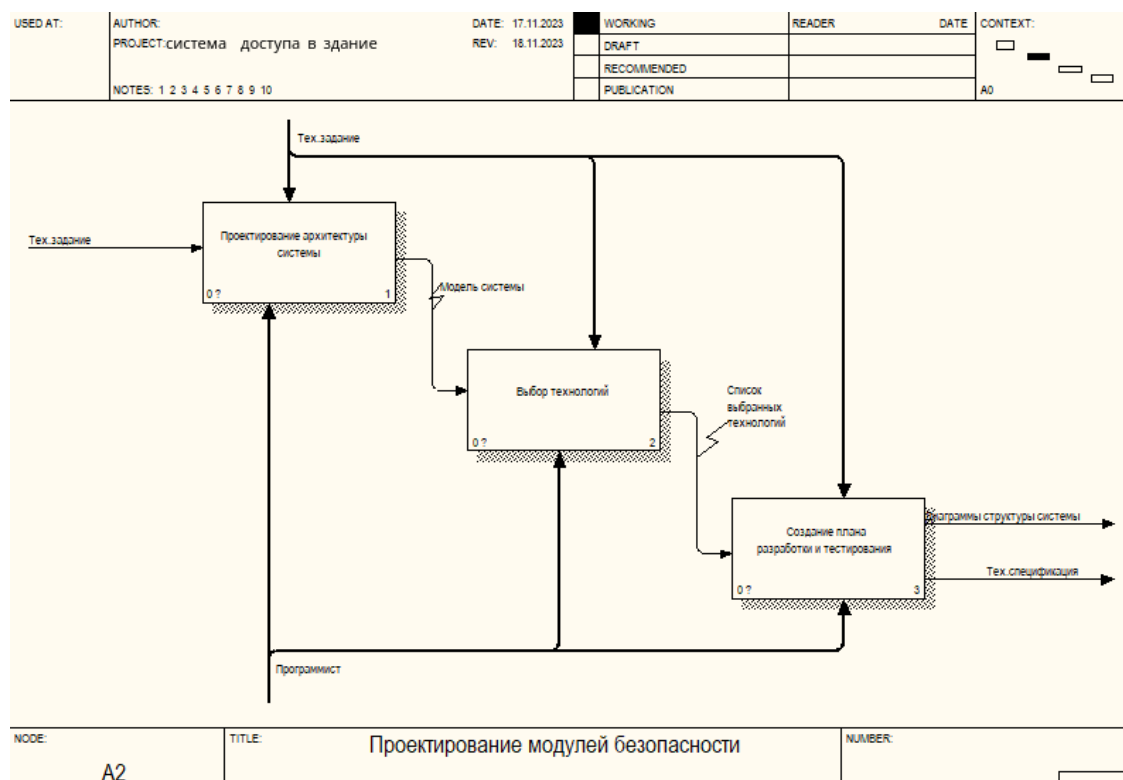


2.4.3 Декомпозиция контекстной диаграммы

Декомпозиция контекстной диаграммы в методологии IDEF0 — это процесс детализации и разделения высокого уровня представления системы на более мелкие и управляемые части. Контекстная диаграмма (или A-1 диаграмма) в IDEF0 показывает основную функцию системы, взаимодействие системы с внешними объектами (входы, выходы, механизмы и управления), но не вдаётся в детали внутренней структуры системы. В данном случае мы декомпозировали систему на 4 основных задачи.

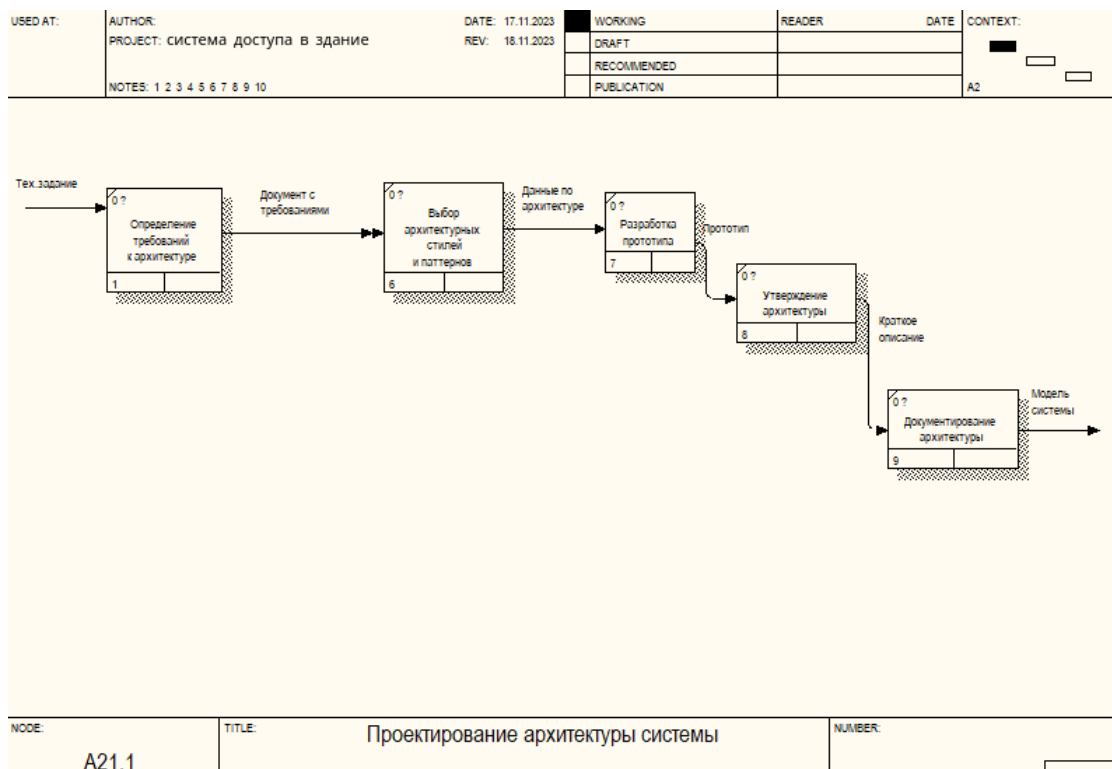


Декомпозиция задачи «обсуждение с заказчиком» на 4 подзадачи.



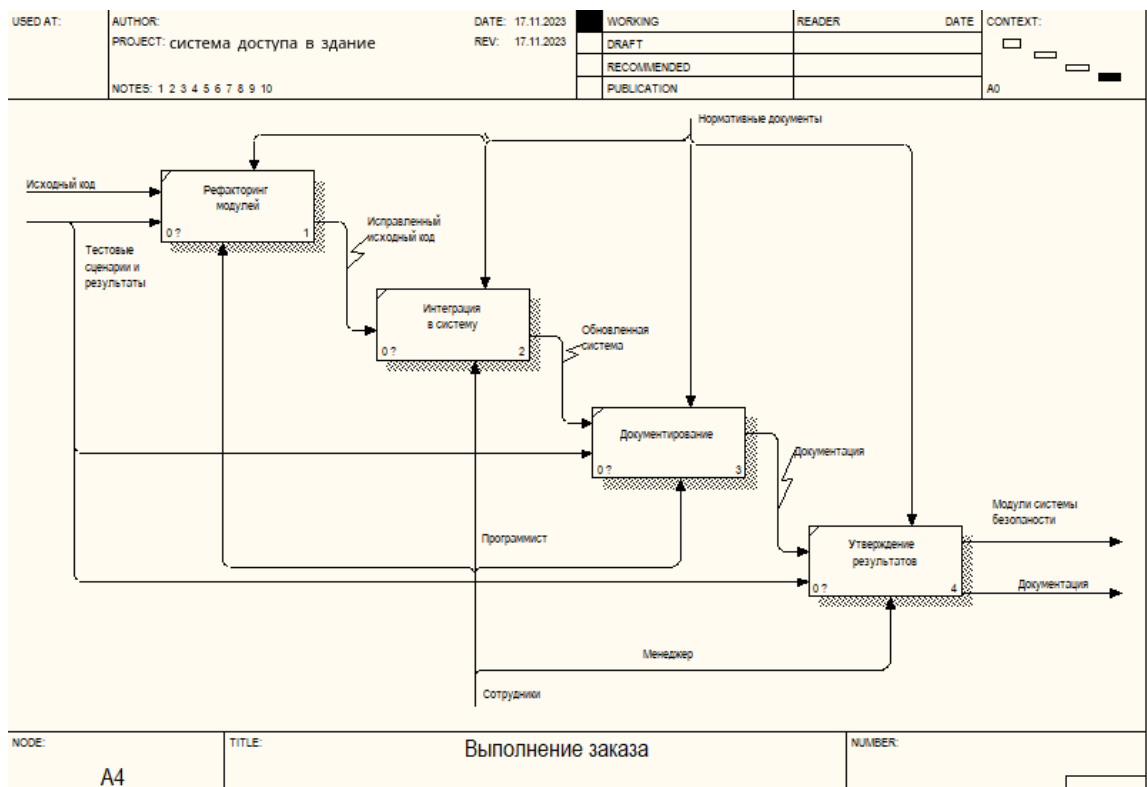
2.4.4 Декомпозиция задачи A2

Декомпозиция задачи «оформить перевод» на 3 подзадачи.

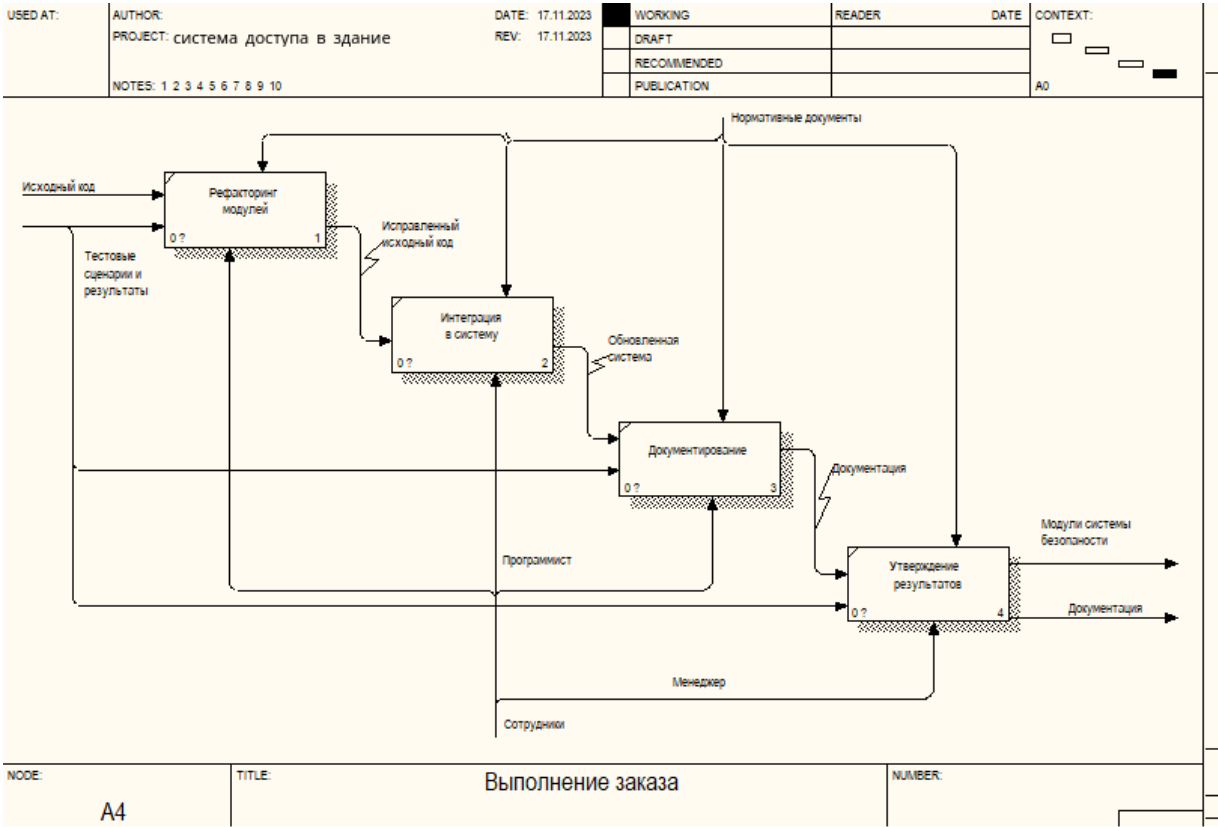


2.4.5 Декомпозиция задачи АЗ

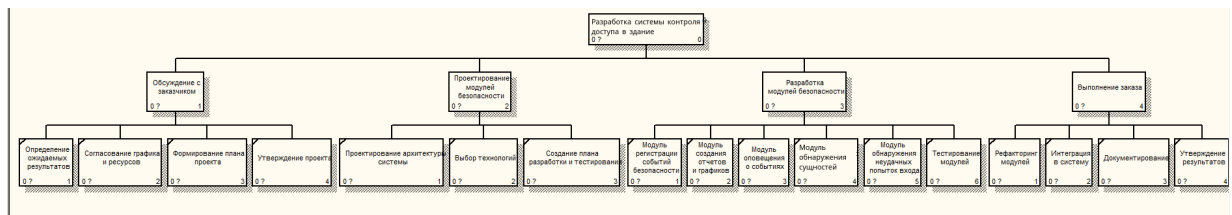
Декомпозиция задачи «разработка модулей учета и контроля» на 5 подзадач.



2.4.6 Декомпозиция задачи A4
Декомпозиция задачи «Выполнение заказа» на 3 подзадачи.



2.4.7 Дерево узлов



2.5 Анализ результатов функционального моделирования в методике IDEF

В результате функционального моделирования была создана контекстная диаграмма, которая рассматривает систему как единый функциональный блок, затем были созданы диаграммы декомпозиции, которые отражают взгляд на систему с разных точек зрения, также было создано дерево узлов, которое показывает иерархию работ в модели и позволяет рассмотреть всю модель целиком.

В работе использовалась нотация IDEF0, для моделирования бизнеспроцессов, IDEF3, для моделирования логики взаимодействия бизнеспроцессов, а также DFD, для моделирования системы с точки зрения хранения, обработки и передачи данных.

2.6 Объектное моделирование в методике UML

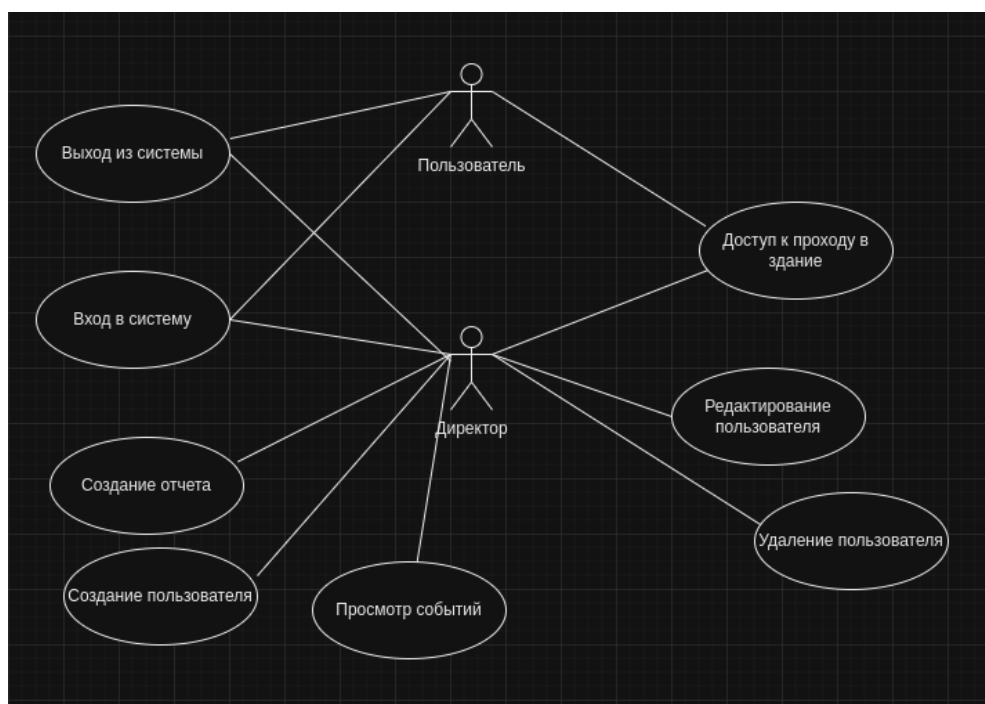
2.6.1 Диаграмма вариантов использования

На диаграмме вариантов использования UML изображены основные взаимодействия между пользователями и системой учета и контроля рабочего времени.

В диаграмме выделены три ключевых актора: "Директор компании", "Руководитель", "Сотрудник". Варианты использования показывают, какие функции доступны каждому актору.

Диаграмма наглядно демонстрирует, как различные пользователи и системы взаимодействуют с системой, что помогает понять общую функциональность системы и выявить основные сценарии использования для дальнейшего проектирования и тестирования.

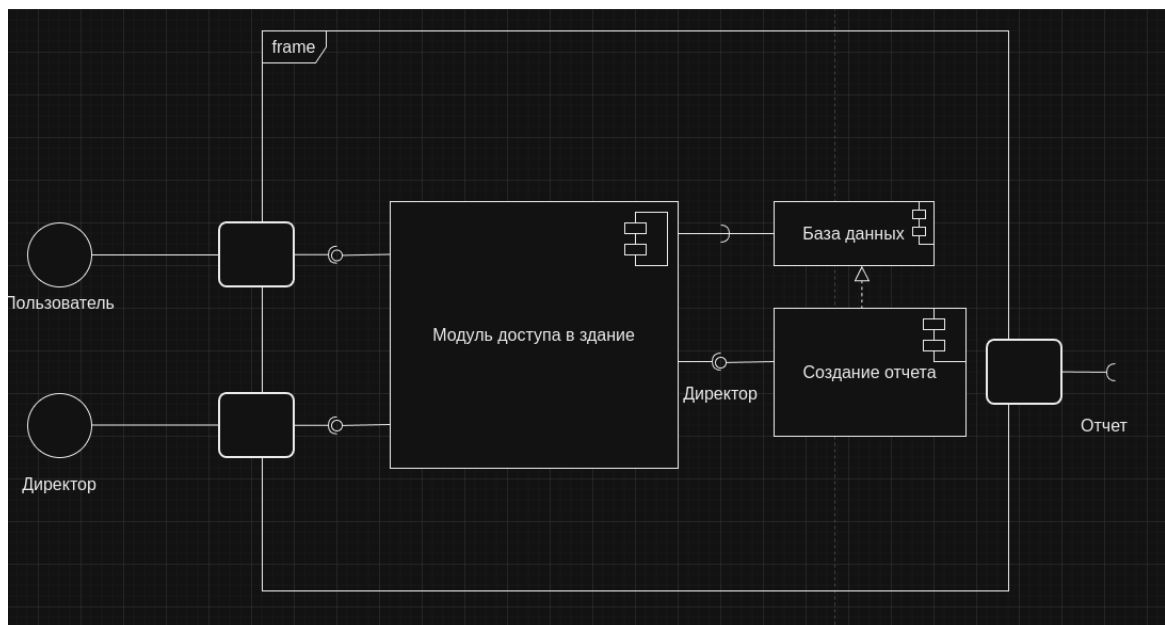
Рисунок 1 Диаграмма вариантов использования



2.6.2 Диаграмма компонентов

Диаграмма компонентов UML используется для визуализации и понимания структуры программной системы на уровне высокоуровневых компонентов и их взаимосвязей. Она отображает различные программные компоненты, такие как модули, библиотеки, и их интерфейсы, а также зависимости между ними. Эта диаграмма помогает определить, как различные части системы взаимодействуют друг с другом, выявить потенциальные проблемы с интеграцией и зависимостями, и спланировать модульность и повторное использование компонентов. Благодаря диаграммам компонентов можно эффективно управлять сложностью системы, улучшать её модульность и способствовать ясному разделению ответственности между различными частями проекта.

Рисунок 2 Диаграмма компонентов



2.6.3 Диаграмма пакетов

Диаграмма пакетов UML используется для организации и группировки различных элементов модели, в логически связанные группы, облегчая понимание и управление сложными системами. Эта диаграмма помогает визуализировать структуру системы на более высоком уровне абстракции, показывая, как пакеты взаимодействуют друг с другом через зависимости и отношения.

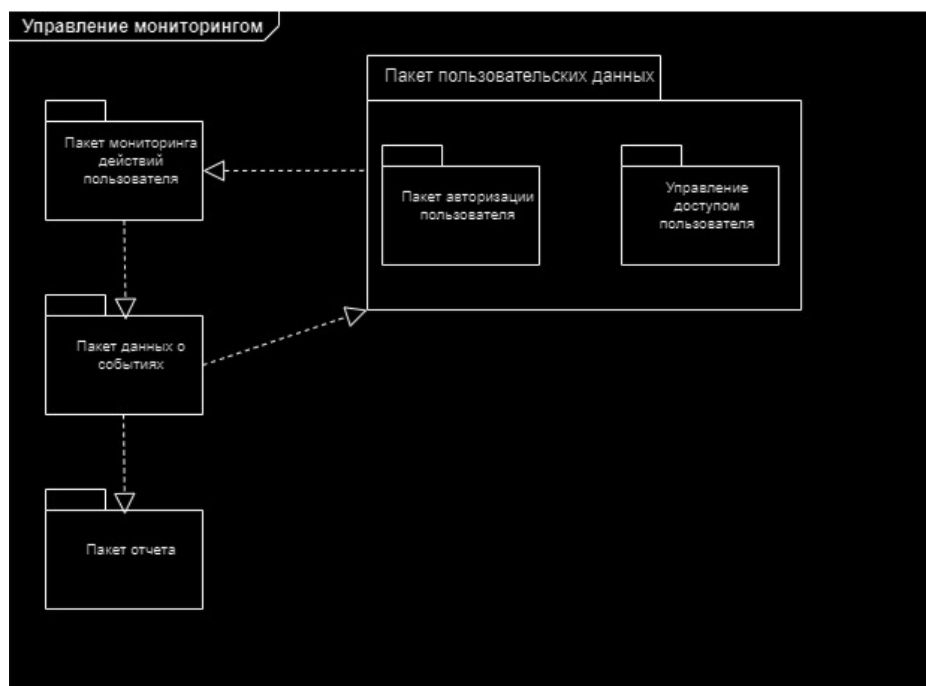


Рисунок 3 Диаграмма пакетов

2.6.4 Диаграмма размещения

Диаграмма размещения UML используется для моделирования физического развёртывания артефактов программного обеспечения на узлах аппаратного обеспечения. Она показывает, как программные компоненты и сервисы распределены по физическим устройствам, таким как серверы, компьютеры и другие аппаратные узлы, а также описывает связи и взаимодействия между ними. Эта диаграмма полезна для визуализации и анализа архитектуры системы с точки зрения её физической реализации, что помогает в планировании развёртывания, оценке производительности, надёжности и масштабируемости системы.

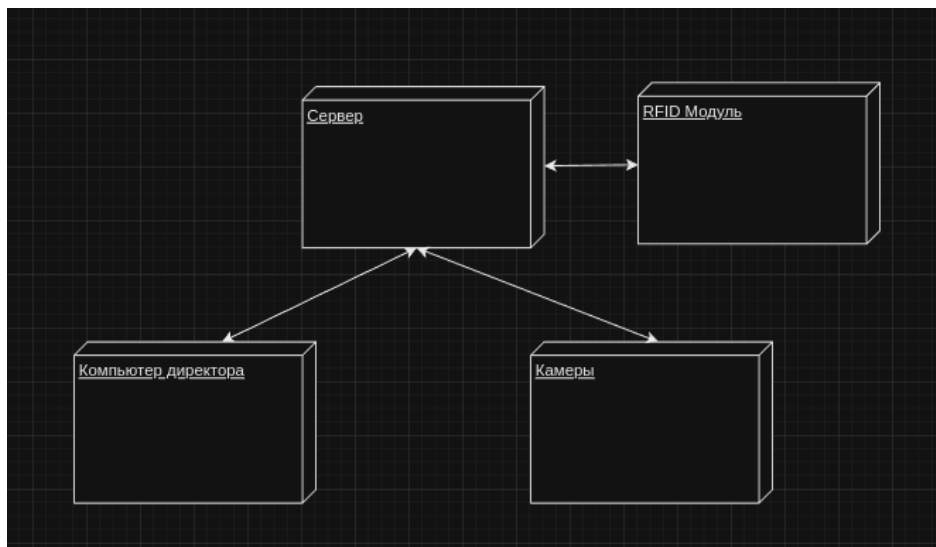


Рисунок 4 Диаграмма размещения

Разработка программы

1. Описание ключевых классов и методов

/auth/login — авторизация

/auth/logout — выход из системы

/auth/get_user - Получить данные о пользователе

/auth/add_user - Добавление нового пользователя

/auth/remove_user - Удаление пользователя

2. Особенности реализации на FastAPI и MongoDB

В FastAPI и MongoDB представлены ряд особенностей и улучшений, которые способствуют удобству разработки и повышают производительность приложений. Некоторые из ключевых аспектов разработки с использованием этих технологий включают:

Асинхронное программирование: FastAPI поддерживает асинхронные функции, что позволяет обрабатывать множество запросов одновременно. Это значительно повышает производительность приложения, особенно при работе с большим количеством пользователей и запросов к базе данных.

Простота и скорость разработки: FastAPI позволяет быстро создавать RESTful API с минимальными усилиями. Автоматическая генерация документации (Swagger и ReDoc) упрощает процесс разработки и тестирования API, что позволяет разработчикам сосредоточиться на бизнес-логике приложения.

Модульность и расширяемость: FastAPI поддерживает модульную архитектуру, что позволяет легко добавлять новый функционал и расширять возможности приложения по мере необходимости. Это особенно полезно для проектов, которые могут расти и изменяться со временем.

Гибкость хранения данных: MongoDB, как NoSQL база данных, предоставляет гибкость в хранении неструктурированных данных. Это позволяет разработчикам легко

адаптировать структуру данных под изменяющиеся требования приложения без необходимости сложных миграций.

Быстрая интеграция с клиентами: FastAPI позволяет легко интегрировать фронтенд-приложения с бэкендом, обеспечивая высокую скорость обмена данными. Это особенно важно для приложений, требующих быстрого реагирования на действия пользователей.

Поддержка различных форматов данных: FastAPI поддерживает работу с различными форматами данных, такими как JSON, что упрощает взаимодействие с клиентскими приложениями и сторонними сервисами.

Удобство разработки и отладки: FastAPI предоставляет инструменты для отладки и тестирования, что упрощает процесс разработки. Возможность использования встроенных тестов позволяет разработчикам быстро проверять функциональность приложения.

Поддержка сообщества и документация: FastAPI имеет активное сообщество разработчиков и обширную документацию, что обеспечивает поддержку и помощь в решении проблем. Это позволяет разработчикам быстро освоить инструменты и начать работу над проектами.

3. Интеграция с базами данных

Для управления данными в системе видеонаблюдения и контроля доступа я использовала MongoDB в качестве системы управления базами данных. Этот выбор обусловлен гибкостью, масштабируемостью и возможностью работы с неструктурированными данными, что критически важно для обеспечения точности и целостности данных в приложении.

Интеграция с MongoDB осуществляется с использованием библиотеки Motor, которая предоставляет асинхронный драйвер для работы с MongoDB в Python. Это позволяет эффективно обрабатывать запросы к базе данных без блокировки основного потока выполнения приложения. Конфигурация соединения с базой данных производится через соответствующие параметры в коде приложения.

Для обеспечения целостности данных и их последовательности в базе данных реализованы соответствующие ограничения и индексы. Это позволяет оптимизировать запросы и улучшить производительность системы, что особенно важно при работе с большим объемом данных о событиях доступа и видеозаписях.

ТЕСТИРОВАНИЕ ПРОГРАММЫ

1. Методы тестирования, использованные для проверки функциональности и надежности программы

Тестирование программы для управления камерами видеонаблюдения и контроля доступа было организовано с использованием методов ручного функционального тестирования, что позволило тщательно проверить все аспекты функциональности приложения. Процесс тестирования включал в себя ряд задач, направленных на верификацию соответствия программы требованиям спецификации и ожиданиям пользователей. Тестирование осуществлялось согласно заранее разработанным тест-кейсам, которые покрывали все ключевые функции программы, такие как:

- Вход в систему
- Добавление записи о событии доступа
- Изменение данных о пользователе или событии
- Удаление записи о событии доступа
- Генерация отчетов о доступе

В ходе тестирования особое внимание уделялось проверке граничных условий и нестандартных сценариев использования, чтобы обеспечить устойчивость и надежность программы в различных условиях эксплуатации. Тестирование показало хорошую функциональность и производительность приложения, а также подтвердило его способность корректно обрабатывать ошибки и исключения.

2. Результаты тестирования и обработка ошибок

В процессе тестирования выявленные недочеты и ошибки регистрировались для последующей обработки. Эти данные использовались для корректировки кода и улучшения общей стабильности системы. Каждая ошибка анализировалась, и на основании анализа принимались меры по ее исправлению, что включало рефакторинг кода, улучшение логики работы функций и оптимизацию запросов к базе данных. Результаты исправлений повторно проверялись, чтобы убедиться в их эффективности и отсутствии регрессии в уже проверенных частях программы.

Таким образом, тщательно организованное тестирование и последующая оптимизация программы позволили значительно повысить ее качество и удобство использования, обеспечивая высокую удовлетворенность конечных пользователей.

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

1. Описание интерфейса и его удобства для пользователя

Пользовательский интерфейс программы для управления камерами видеонаблюдения и контроля доступа разработан с учетом принципов удобства использования и интуитивной навигации. Интерфейс представляет собой чистое и минималистичное оформление, обеспечивающее быстрый доступ к основным функциям, таким как вход в систему, добавление записи о событии доступа, изменение данных, удаление записи и создание отчетов. Это позволяет пользователям эффективно взаимодействовать с системой и быстро осуществлять необходимые операции без лишних усилий.

Основные элементы интерфейса:

1. Добавить запись о событии доступа:

- Откройте интерфейс вашего приложения для учета событий доступа.
- Создайте новую форму или раздел для добавления записи.
- Предоставьте пользователю поля для ввода информации о событии, такие как имя пользователя, время доступа, тип события (вход/выход) и комментарии.
- Убедитесь, что данные введены корректно и сохраняйте запись в базе данных вашего приложения.

2. Изменить данные о событии:

- Найдите запись, данные о которой вы хотите изменить.
- Откройте форму или раздел редактирования записи.
- Позвольте пользователю изменить необходимые поля, такие как время доступа или комментарии.
- После внесения изменений убедитесь, что данные сохранены правильно.

3. Удалить запись о событии:

- Найдите запись, которую вы хотите удалить.
- Предоставьте пользователю опцию удаления.

- Подтвердите действие удаления с помощью подтверждения от пользователя.

- Удалите запись из базы данных и обновите интерфейс приложения.

4. Создать отчет по данным доступа:

- Определите, какие данные должны включаться в отчет.
- Разработайте механизм для выбора параметров отчета (например, диапазон дат или тип событий).

- Создайте шаблон отчета, который будет содержать необходимую информацию.

- Предоставьте пользователю возможность сгенерировать отчет с выбранными параметрами.

- Убедитесь, что отчет генерируется корректно и содержит актуальные данные.

Таким образом, интерфейс программы разработан с целью обеспечить максимальное удобство для пользователей, позволяя им эффективно управлять доступом и легко получать нужную информацию, что способствует повышению общей продуктивности работы.

2. Скриншоты интерфейса и его удобства для пользователя

Вход в систему

Authorization

Login

Password

[Forgot your password?](#)

Login

Рисунок 6.1. - Ввести логин и пароль.

Камеры

Пользователи

Журнал аудита

Пропуска

Пользователи

| | | | |
|---------|----------------|------|-----------------------|
| login | password | role | Добавить пользователя |
| Удалить | 100LAR admin | | |

Рисунок 6.2.-Страница списка пользователей

Камеры

Пользователи

Журнал аудита

Пропуска

Журнал аудита

Дата в формате YYYY-MM-DD

0000-00-00

Сгенерировать
отчёт

2024-09-05T14:07:29.521000 | FwQfm8mGoWU6EQ | test

2024-09-05T16:02:10.889000 | FwQfm8mGoWU6EQ | test

2024-09-05T16:02:12.671000 | FwQfm8mGoWU6EQ | test

2024-09-05T16:02:12.978000 | FwQfm8mGoWU6EQ | test

2024-09-05T16:02:13.279000 | FwQfm8mGoWU6EQ | test

2024-09-05T16:02:13.637000 | FwQfm8mGoWU6EQ | test

2024-09-05T16:02:13.865000 | FwQfm8mGoWU6EQ | test

Рисунок 6.3.-Страница списка аудита

Камеры

Пользователи

Журнал аудита

Пропуска

Пропуска

name

Добавить пропуск

66d98eee1926882dd1accfa1 | test | FwQfm8mGoWU6EQ

66d9ad8eadf597d4748be93a | 123 | 9UJ58UwXsb4ULg

66d9ad91adf597d4748be93b | 123 | s0D3C6FNWKLBDg

Рисунок 6.4.-Страница списка карт

ЗАКЛЮЧЕНИЕ

В заключении проекта по созданию программы для управления камерами видеонаблюдения и контроля доступа можно отметить успешное достижение поставленных целей. Разработанный пользовательский интерфейс представляет собой интуитивно понятное и удобное в использовании приложение, которое обеспечивает эффективное управление процессами видеонаблюдения и контроля доступа на объекте. Интеграция с базой данных MongoDB с использованием FastAPI обеспечивает надежное хранение и обработку данных, а также поддержку гибкости в работе с неструктурированными данными.

Проект выделяется чистым и минималистичным дизайном пользовательского интерфейса, который способствует удобству работы и повышению эффективности пользователей. Все ключевые функции, такие как добавление записей о событиях доступа, изменение данных и генерация отчетов, реализованы с акцентом на простоту и доступность.

В целом, разработанный проект отвечает требованиям и ожиданиям заказчика, предоставляя надежное и удобное решение для управления системами видеонаблюдения и контроля доступа. Успешное завершение проекта подтверждает его высокую функциональность и способность адаптироваться к изменяющимся потребностям пользователей, что делает его ценным инструментом для обеспечения безопасности и организации работы на объекте.

ПРИЛОЖЕНИЕ

1. Код программы

В репозитории на github:

<https://github.com/4LAR/Programming-GUAP/tree/main/ssd/kusa4>

2. Руководство пользователя

\

УТВЕРЖДАЮ
Генеральный
директор ИП «WatchGuard»
_____ Столяров
“ _____ ” _____

**СПЕЦИАЛЬНОЕ
ОБЕСПЕЧЕНИЕ**

ПРОГРАММНОЕ

**«СИСТЕМА ДЛЯ УЧЕТА КОНТРОЛЯ ДОСТУПА
В ЗДАНИЕ»**

Руководство пользователя Лист
утверждения

Руководитель разра

Начальник отдела

_____Столяров Н

“ _____ ”

Ответственный

ИСПОЛНИТЕЛЬ

Начальник группы

разработки ОРСПО

_____ Столяров

“ _____ ”

2024

УТВЕРЖДЕНО

**СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ
«СИСТЕМА ДЛЯ УЧЕТА КОНТРОЛЯ
ДОСТУПА»**

Руководство пользователя

Листов 24

2024

АННОТАЦИЯ

В данном программном документе представлено руководство пользователя по настройке и использованию специализированного программного обеспечения "Система учета доступа и видеонаблюдения".

В разделе "Введение" содержится общая информация о руководстве, его цели и структура, что позволяет пользователям быстро ориентироваться в содержании документа.

Раздел "Назначение и условия применения" описывает основные функции системы, такие как управление камерами видеонаблюдения, контроль доступа и формирование отчетов, а также условия, необходимые для её эффективного функционирования, включая технические требования и ограничения.

В разделе «Подготовка к работе» объясняется, как правильно установить и настроить систему перед началом её использования, включая состав и содержание дистрибутивного носителя данных и порядок загрузки программ.

Раздел «Описание операций» представляет детальное описание всех операций, которые можно выполнять с помощью системы, включая регистрацию событий доступа, управление данными пользователей и формирование отчетов о доступе.

Раздел «Аварийные ситуации» содержит информацию о том, как распознать, предотвратить и устранить возможные аварийные ситуации или технические неисправности, которые могут возникнуть в процессе эксплуатации системы.

В разделе «Рекомендации по освоению» предложены методы и подходы для более быстрого и эффективного освоения системы, включая контрольные примеры для практики.

Документ "Руководство пользователя" оформлен в соответствии с требованиями стандартов ЕСПД (ГОСТ 19.101-77, ГОСТ 19.103-77, ГОСТ 19.104-78, ГОСТ 19.105-78, ГОСТ 19.106-78, ГОСТ 19.504-79, ГОСТ 19.604-78).

Список стандартов:

ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.

ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.

ГОСТ 19.104-78 ЕСПД. Основные надписи.

ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.

ГОСТ 19.106-78 ЕСПД. Общие требования к программным документам, выполненным печатным способом.

ГОСТ 19.504-79 ЕСПД. Руководство программиста. Требования к содержанию и оформлению.

ГОСТ 19.604-78 ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом.

ВВЕДЕНИЕ

1.1. Область применения

Требования настоящего руководства применяются в следующих условиях эксплуатации системы:

- Предварительные комплексные испытания перед внедрением системы в эксплуатацию.
- Опытная эксплуатация для оценки функциональности и надежности системы.
- Приемочные испытания для подтверждения соответствия системы установленным требованиям.
- Промышленная эксплуатация для регулярного учета рабочего времени в рамках деятельности организации.

1.2. Краткое описание возможностей

Система учёта и контроля пропускного пункта предназначена для автоматизации процессов регистрации и контроля пропусков

- Добавление новой операции

1.3. Уровень подготовки пользователя

Для эффективного использования системы требуется:

- Опыт работы с операционными системами, такими как MS Windows или что угодно.
- Навыки работы с офисными программами для ввода данных и работы с отчетами.
- Понимание основ учета рабочего времени и принципов формирования табелей.
- Способность к быстрому освоению программного обеспечения специализированного характера.

1.4. Перечень эксплуатационной документации

Перед началом работы с системой рекомендуется ознакомиться с следующей документацией:

- Инструкция по безопасности работы с электронными системами учёта.
- Правила внутреннего распорядка организации, касающиеся учета рабочего времени.

2. НАЗНАЧЕНИЕ И УСЛОВИЯ

Система WatchGuard предназначена для автоматизации процессов наблюдения и контроля на пропускных пунктах. Это включает в себя регистрацию времени входа и выхода сотрудников, мониторинг активности, а также формирование необходимой отчетности для руководства и службы безопасности.

2.1 Виды деятельности и функции

Система предназначена для использования в любых организациях, где требуется точный и эффективный контроль доступа и наблюдение за перемещением сотрудников.

Особенно актуальна для компаний с высоким уровнем безопасности, а также в условиях, где необходим строгий контроль за доступом и его оптимизация.

2.2 Технические условия

Вид ЭВМ и конфигурация: Система требует компьютера с операционной системой Microsoft Windows 10 или новее, стабильного интернет-соединения для доступа к облачной версии или локальной сети для корпоративной версии.

Операционная среда: Система рассчитана на работу в сетевом режиме с использованием современных веб-браузеров, таких как Google Chrome или Mozilla Firefox.

База данных: Для хранения данных используется надежная и безопасная база данных, обеспечивающая быстрый доступ и сохранность информации.

2.3 Требования к пользователям

Пользователи должны иметь базовые навыки работы на ПК и понимание основ работы с веб-браузерами.

Для администраторов и сотрудников службы безопасности рекомендуется дополнительное обучение по настройке и использованию системы для полноценного управления доступом и отчетности.

2.4 Условия доступа

Доступ к системе предоставляется через персонализированные учетные записи с соответствующими уровнями прав доступа, зависящими от роли пользователя в организации.

Система WatchGuard идеально подходит для компаний, стремящихся повысить эффективность управления доступом и обеспечить точность данных для последующего анализа и принятия решений.

3. ПОДГОТОВКА К РАБОТЕ

3.1 Состав и содержание дистрибутивного носителя данных

Для работы с системой WatchGuard необходимо следующее программное обеспечение:

- ПК на любой ОС

3.2 Порядок загрузки данных и программ

Для начала работы с системой WatchGuard выполните следующие шаги:

- 1) Откройте приложение WatchGuard
- 2) Зарегистрируйтесь или войдите в систему, используя ваш логин и пароль.
- 3) После входа в систему, при первом обращении может потребоваться настройка пользовательского интерфейса или загрузка необходимых компонентов системы, что произойдет автоматически.

3.3 Порядок проверки работоспособности

Для проверки работоспособности системы WatchGuard рекомендуется следовать этим шагам:

- 1) Входим в систему

Authorization

Login

Password

[Forgot your password?](#)

Login

Рисунок 3.1

Ввести логин и пароль администратора или сотрудника(рис. 3.1).

2) Проверить наличия всех функций

Камеры

Пользователи

Журнал аудита

Пропуска

Пропуска

Добавить пропуск

66d98eee1926882dd1accfa1 | test | FwQfm8mGoWU6EQ

66d9ad8eadf597d4748be93a | 123 | 9UJ58UwXsb4ULg

66d9ad91adf597d4748be93b | 123 | s0D3C6FNWKLBDg

Рисунок 3.2

4. ОПИСАНИЕ ОПЕРАЦИЙ

4.1 Выполняемые функции и задачи

WatchGuard выполняет функции и задачи, приведенные в таблице ниже:

| ФУНКЦИИ | ЗАДАЧИ | ОПИСАНИЕ |
|---|--|--|
| Обеспечивает возможность добавления записи о проходе | Добавляет информацию о новом проходе | Пользователь может добавить запись о новом проходе в систему. |
| Обеспечивает возможность изменения записи о проходе | Изменяет информацию о проходе | Пользователь может изменить существующую запись о проходе. |
| Обеспечивает возможность удаления записи о проходе | Удаляет информацию о проходе | Пользователь может удалить выбранную запись о проходе из системы. |
| Обеспечивает возможность формирования отчета | Создает отчет о проходах | Пользователь может создать отчет на основе данных о проходах. |
| Обеспечивает просмотр всей базы данных | Визуализируются данные о проходах | Пользователь может просмотреть все записи о проходах в системе. |
| Обеспечивает возможность для поиска нужной записи о проходе | Визуализирует данные о нужных проходах | Пользователь может осуществлять поиск по записям о проходах для быстрого доступа к необходимой информации. |

4.2 Описание операций технологического процесса обработки данных, необходимых для выполнения задач.

5. Добавить операцию:
 - Откройте интерфейс вашего приложения.
 - Создайте новую форму или раздел для добавления операции.
 - Предоставьте пользователю поля для ввода информации о новой операции, такие как название, условия выполнения, подготовительные и основные действия, заключительные действия, ресурсы, расходуемые на операцию.
 - Убедитесь, что данные введены корректно и сохраняйте операцию в базе данных вашего приложения.
6. Изменить данные об операции:
 - Найдите операцию, данные о которой вы хотите изменить.
 - Откройте форму или раздел редактирования операции.
 - Позвольте пользователю изменить необходимые поля, такие как условия выполнения, действия и ресурсы.
 - После внесения изменений убедитесь, что данные сохранены правильно.
7. Удалить операцию:
 - Найдите операцию, которую вы хотите удалить.
 - Предоставьте пользователю опцию удаления.
 - Подтвердите действие удаления с помощью подтверждения от пользователя.
 - Удалите операцию из базы данных и обновите интерфейс приложения.
8. Создать отчет по данным:
 - Определите, какие данные должны включаться в отчет.
 - Разработайте механизм для выбора параметров отчета (например, диапазон дат или тип операций).

- Создайте шаблон отчета, который будет содержать необходимую информацию.
- Предоставьте пользователю возможность сгенерировать отчет с выбранными параметрами.
- Убедитесь, что отчет генерируется корректно и содержит актуальные данные.

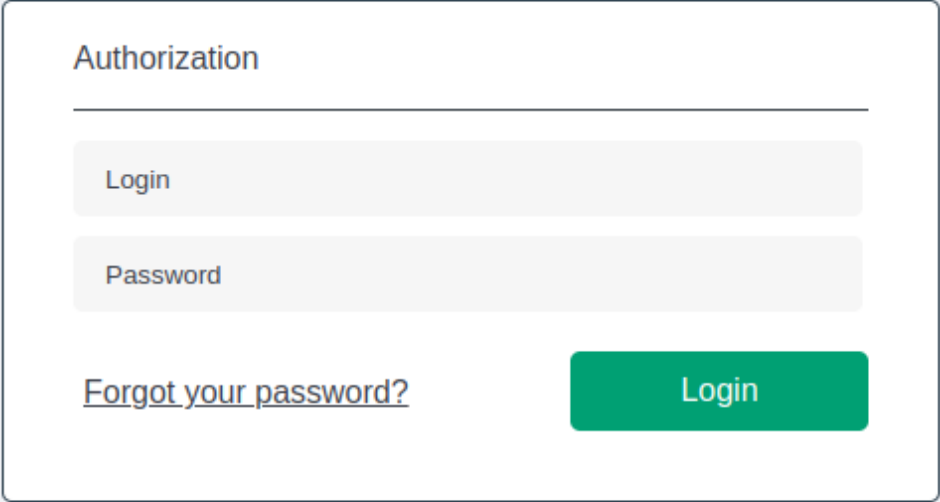
5. АВАРИЙНЫЕ СИТУАЦИИ

Все аварийные ситуации можно увидеть в таблице ниже:

| Класс ошибки | Ошибка | Описание ошибки | Требуемые действия пользо вателя при возникно вании ошибки |
|---------------------------------|---|--|--|
| Сбой при переходе функцию | Функция не найдена. Невозможно выполнить действие | Возможные проблемы настройками приложения. | Проверьте правильность данных или наст ройки приложения. |
| Сбой при входе в систему | Неправильный пароля или логина | Возможные проблемы неправильным вводом пароля или логина | Проверьте правильность логина и пароля. |

6. РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ

6.1 Вход в систему



Authorization

Login

Password

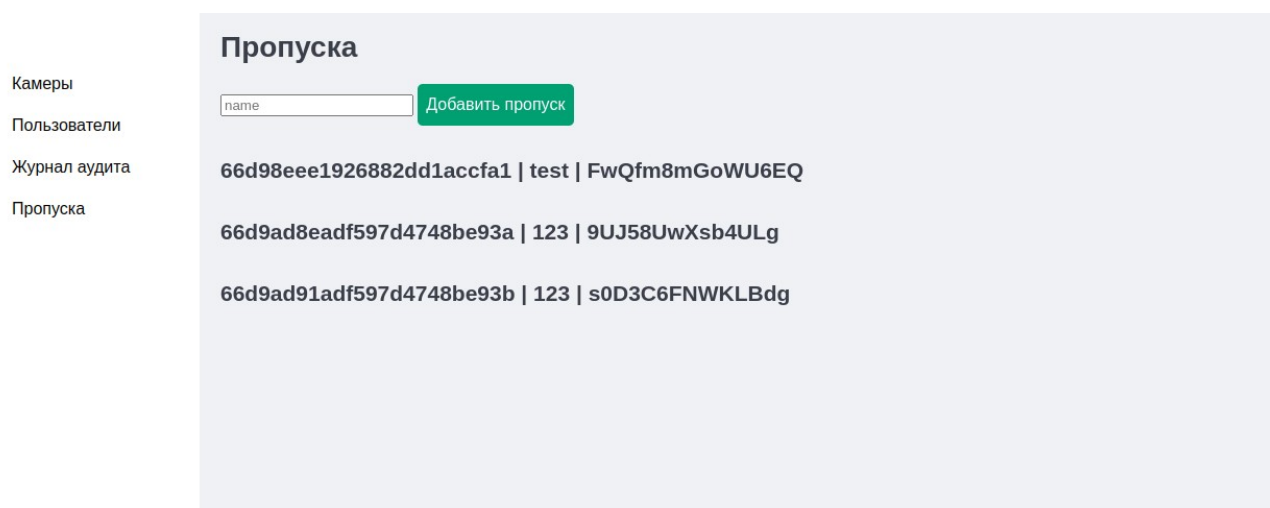
[Forgot your password?](#)

Login

Рисунок 6.1.

Ввести логин и пароль.

Окно главного меню программы (рис 6.2)



Камеры

Пользователи

Журнал аудита

Пропуска

Пропуска

Добавить пропуск

66d98eee1926882dd1accfa1 | test | FwQfm8mGoWU6EQ

66d9ad8eadf597d4748be93a | 123 | 9UJ58UwXsb4ULg

66d9ad91adf597d4748be93b | 123 | s0D3C6FNWKLBDg

Рисунок 6.2.

3. Руководство программиста

УТВЕРЖДАЮ
Генеральный
директор ИП «WatchGuard»
_____ Столяров
“ _____ ”

СПЕЦИАЛЬНОЕ ОБЕСПЕЧЕНИЕ

ПРОГРАММНОЕ

«СИСТЕМА ДЛЯ УЧЕТА КОНТРОЛ ДОСТУПА В ЗДАНИЕ»

Руководство программиста Лист
утверждения

Руководитель разра

Начальник отдела

____ Столяров Н

“ ”

Ответственный

ИСПОЛНИТЕЛЬ

Начальник группы

разработки ОРСПО

_____ Столяров

“ _____ ”

2024

УТВЕРЖДЕНО

**СПЕЦИАЛЬНОЕ ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ
«СИСТЕМА ДЛЯ УЧЕТА КОНТРОЛЯ ДОСТУПА
В ЗДАНИЕ»**

Руководство программиста

Листов 21

2024

АННОТАЦИЯ

В данном программном документе представлено руководство программиста по настройке и использованию специализированного программного обеспечения "Система учета рабочего времени".

В разделе "Назначение и условия применения программы" описаны основные функции и задачи системы, а также условия, необходимые для её функционирования (объем оперативной памяти, требования к компонентам и параметрам периферийных устройств, программное обеспечение и прочее).

Раздел "Характеристика программы" содержит описание ключевых характеристик и особенностей программы, таких как режимы работы, инструменты контроля корректности выполнения задач и возможности самовосстановления системы.

В разделе "Входные и выходные данные" дается описание организации данных, которые система принимает и выдает в процессе работы.

Раздел "Сообщения" включает тексты системных сообщений, предназначенных для программистов и операторов, их содержание и рекомендации по действиям в ответ на эти сообщения.

Документ "Руководство программиста" оформлен в соответствии с требованиями стандартов ЕСПД (ГОСТ 19.101-77, ГОСТ 19.103-77, ГОСТ 19.104-78, ГОСТ 19.105-78, ГОСТ 19.106-78, ГОСТ 19.504-79, ГОСТ 19.604-78).

-
- ¹⁾ ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов
- ²⁾ ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов
- ³⁾ ГОСТ 19.104-78* ЕСПД. Основные надписи
- ⁴⁾ ГОСТ 19.105-78* ЕСПД. Общие требования к программным документам
- ⁵⁾ ГОСТ 19.106-78* ЕСПД. Общие требования к программным документам, выполненным печатным способом
- ⁶⁾ ГОСТ 19.504-79* ЕСПД. Руководство программиста. Требования к содержанию и оформлению
- ⁷⁾ ГОСТ 19.604-78* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом

1.1 WatchGuard предназначена для решения задач автоматизации учета инвентаризации и состава сотрудников, в том числе для решения следующих задач:

- добавление персонала и распределение ролей
- составление отчетностей
- просмотр аудита
- просмотр камер

1.2 Функции, выполняемые программой

- Получение и отправка http запросов
- Работа с Базой данных

1.3 Условия необходимые для выполнения программы

1.3.1 Операционная система

Для корректной работы системы необходимо устройство на операционной системе Linux на дистрибутиве Debian или Ubuntu

- Пакет Инструментов для Python

Необходим ЯП python 3.7 или выше и набор модулей описанных в файле requirements.txt в корне папки исходного кода проекта

- База данных

Для ведения базы данных необходимо подключение к удаленной mongoDB

- Объем оперативной памяти

Рекомендуемый объем оперативной памяти 1ГБ или выше

- Требования к составу периферийных устройств

Особые требования не предъявляются

- Требования к программисту

Программист должен обладать практическими навыками работы в терминале bash и иметь навыки по работе с Python и MongoDB, HTML, Css, JavaScript, GIT.

- **Характеристики программы**

2.1 Описание основных характеристик программы

Описание характеристик программы описаны в документе [руководство пользователя](#)

- Режим работы программы

Программа работает в терминале, доступ к интерфейсу обеспечивается через браузер по ip адресу и 8080 порту

- Средства контроля правильности выполнения программы

Контроль правильности выполнения АСИУС осуществляется встроенными средствами программного обеспечения в виде протоколирования событий

- Описание особенностей программы

При запуске программы поднимается http сервер на FastAPI и происходит подключение к базе MongoDB

- Самовостанавливаемость программы обеспечивается стандартными средствами операционной системы

- **Загрузка и запуск программы**

3.1 Загрузка исходного кода

в терминале Bash выполнить команду

```
git clone <<здесь репа>>
```

- Python, установка зависимостей и запуск проекта

в терминале Bash выполнить команды

```
sudo apt-get update
sudo apt-get install python
cd app
pip install -r requirements.txt
python3 app
```

- MongoDB

Установить локально Базу данных и подключиться к ней с помощью Compas

Руководство по установке Mongo DB

И добавить в Базу данных три пустые коллекции catalog, users, change_log.

После установки выполнить в bash команды

```
Cd app  
Nano .env
```

И внести изменения в строчку MONGODB ="ip" указав ip адрес сервера базы данных

- **Входные и выходные данные**

- Программа не требует никаких входных данных, весь вывод работы системы сохраняется в файл nohup.out в корне проекта.

Для просмотра выходных данных нужно выполнить команду

Bash

```
cd inventory  
cat nohup.out
```

