**Содержание**

|  |  |
| --- | --- |
|  | Стр. |
| Введение | 4 |
| 1 Общая часть | 5 |
| 1.1 Описание предметной области | 5 |
| 1.2 Анализ существующей ситуации | 6 |
| 1.3 Постановки задачи | 6 |
| 1.4 Анализ существующих разработок и обоснование необходимости разработки | 7 |
| 2 Специальная часть | 10 |
| 2.1 Выбор технологии и инструментальных средств | 10 |
| 2.1.1 Выбор подхода к разработке | 10 |
| 2.1.2 Выбор средства разработки и языка программирования | 11 |
| 2.2 Разработка спецификации | 11 |
| 2.2.1 Разработка диаграмм вариантов использования | 11 |
| 2.2.2 Разработка диаграмм последовательностей системы | 13 |
| 2.2.3 Разработка диаграмм пакетов | 14 |
| 2.2.4 Разработка диаграмм классов | 15 |
| 2.2.5 Разработка диаграмм состояний | 15 |
| 2.2.6 Разработка диаграмм деятельности | 16 |
| 2.2.7 Разработка диаграмм размещения | 17 |
| 2.2.8 Разработка инфологической модели базы данных | 17 |
| 2.2.9 Разработка физической модели базы данных | 18 |
| 2.3 Проектирование программного модуля | 18 |

|  |  |
| --- | --- |
| 2.3.1 Разработка алгоритмов реализации основных функций программного обеспечения | 18 |
| 2.3.2 Проектирование пользовательского интерфейса | 19 |
| 2.4 Реализация программного обеспечения на выбранном языке программирования и в выбранной среде разработки | 21 |
| 2.5 Выбор стратегии тестирования, разработка тестов, тестирование и отладка программного обеспечения | 23 |
| 3 Безопасность жизнедеятельности | 30 |
| 3.1 Требования к рабочим помещениям и рабочим местам оператора ПК | 30 |
| 3.2 Расчет искусственной освещенности рабочего места оператора ПК | 31 |
| 3.3 Техника безопасности при работе на ПК | 33 |
| Заключение | 35 |
| Список использованных источников | 36 |
| Приложение А Техническое задание | 38 |
| Приложение Б Фрагмент листинга программы | 39 |

# Введение

# Общая часть

Современные образовательные учреждения стремятся интегрировать информационные технологии в учебный процесс, что позволяет повысить эффективность обучения и упростить взаимодействие между студентами и преподавателями. Электронный дневник — это цифровая платформа, которая предоставляет студентам доступ к важной информации о учебном процессе, включая оценки, расписание и контактные данные учителей. Данное техническое задание описывает разработку электронного дневника с акцентом на безопасность, удобство использования и функциональность.

## Описание предметной области

Предметной областью данного ПО является автоматизация процессов учета оценок обучающегося.

Основная задача программного модуля – повысить эффективность и оптимизировать просмотр оценок.

Электронный дневник предназначен для студентов различных образовательных учреждений (школы, колледжи, университеты). Основные функции системы включают:

Авторизация пользователей: Студенты могут входить в систему с использованием пароля или через систему Госуслуг.

Просмотр оценок: Студенты могут видеть свои текущие оценки за уроки и зачеты, а также общее количество баллов.

Расписание занятий: Доступ к актуальному расписанию с возможностью фильтрации по предметам и преподавателям.

Информация о преподавателях: Возможность просмотра профилей учителей, включая их контактные данные и предметы, которые они ведут.

Коммуникация: Функции для обмена сообщениями между студентами и преподавателями.

## Анализ существующей ситуации

В настоящее время многие учебные заведения используют устаревшие системы управления учебным процессом. Наиболее распространенные проблемы:

Ограниченный функционал: Многие системы не позволяют студентам взаимодействовать друг с другом или с преподавателями.

Безопасность данных: Существующие решения часто не обеспечивают достаточный уровень защиты личной информации.

Проблемы с доступностью: Некоторые системы не адаптированы под мобильные устройства, что ограничивает доступ студентов к информации.

## Постановка задачи

Основной целью данной работы является разработка эффективного и функционального программного модуля для комплексного учета и распределения информации о студентах. Данный модуль должен стать важным инструментом в оптимизации и автоматизации..

Основная задача разработки электронного дневника заключается в создании интуитивно понятной и безопасной платформы, которая позволит:

Студентам легко получать доступ к своим оценкам и расписанию.

Преподавателям управлять данными о студентах и оценках.

Обеспечить высокую степень безопасности данных пользователей.

1.4 Анализ существующих разработок и обоснование необходимости разработки

Существующие решения часто не соответствуют современным требованиям пользователей. Необходима разработка нового электронного дневника, который будет:

Интуитивно понятным и удобным в использовании.

Безопасным и защищенным от несанкционированного доступа.

Интегрированным с другими системами (например, Госуслугами) для упрощения процесса авторизации.

# Специальная часть

## Выбор технологий и инструментальных средств

SQL для создания и управления базами данных, использования MySQL или PostgreSQL для хранения и обработки данных.

C# для создания программного модуля, использования .NET Framework для создания веб-приложения и Entity Framework для взаимодействия с базой данных.

### 2.1.1 Выбор подхода к разработке

Выбор подхода к разработке:  
При разработке программного модуля для учета и распределения сотрудников охранной службы объектно-ориентированный подход также является оптимальным выбором.   
В объектно-ориентированном подходе уделяется внимание четкому разделению компонентов системы и их функций, что обеспечивает более удобное управление и масштабирование проекта. Использование классов и объектов в объектно-ориентированной разработке позволяет создать структурированную систему, где каждый компонент отвечает за определенные задачи и имеет четко определенный интерфейс взаимодействия с другими компонентами.  
Объектно-ориентированный подход также способствует повышению гибкости и расширяемости системы, позволяя легко добавлять новый функционал или вносить изменения без необходимости переписывать весь код. Это особенно важно для проектов, где требуется постоянное развитие и адаптация к изменяющимся условиям.  
В рамках объектно-ориентированного подхода также возможно эффективно использовать принципы SOLID, что способствует созданию более надежного и поддерживаемого программного кода.

### 2.1.2 Выбор среды разработки и языка программирования

Язык C# и Visual Studio Community 2022 также являются отличным выбором для разработки программного модуля для разработки программного модуля для электронного дневника C# обладает высокой производительностью, хорошей поддержкой объектно-ориентированной парадигмы, а также широким набором библиотек и фреймворков, что позволит разработать функциональный и эффективный модуль.  
Visual Studio Community 2022, в свою очередь, обеспечивает удобную среду разработки с множеством инструментов для ускорения процесса создания и отладки кода. Отличный отладчик, поддержка Git, автоматические рефакторинги и другие возможности помогут сделать процесс разработки проекта более удобным и эффективным.  
Сочетание C# и Visual Studio Community 2022 позволит создать модуль, который будет удобным в поддержке, расширении и развитии, что важно для программного обеспечения, предназначенного для мониторинга и анализа работы охранной службы.

## Разработка спецификаций

### 2.2.1 Разработка диаграмм вариантов использования

Диаграммы вариантов использования будут описывать основные сценарии взаимодействия пользователей с системой:

- Студент входит в систему через авторизацию.

- Студент просматривает свои оценки за уроки.

- Преподаватель обновляет оценки за занятия.

### 2.2.2 Разработка диаграмм последовательностей системы

Рисунок 2 – Диаграмма последовательностей системы.

1. "Сбор информации" - включает этапы анализа потребностей, определения требований, сбора информации и планирования.  
2. "Проектирование системы" - включает создание архитектуры, дизайна пользовательского интерфейса, базы данных и других компонентов системы.  
3. "Разработка" - этап создания и программирования системы, тестирования и интеграции различных компонентов.  
4. "Тестирование и отладка" - проверка функциональности системы, выявление и исправление ошибок.  
5. "Внедрение и обучение" - этап установки системы на рабочие места, обучения пользователей и перехода к реальной эксплуатации.  
6. "Поддержка и сопровождение" - обеспечение непрерывной работы системы, решение проблем и внесение необходимых изменений.

### 2.2.3 Разработка диаграммы пакетов

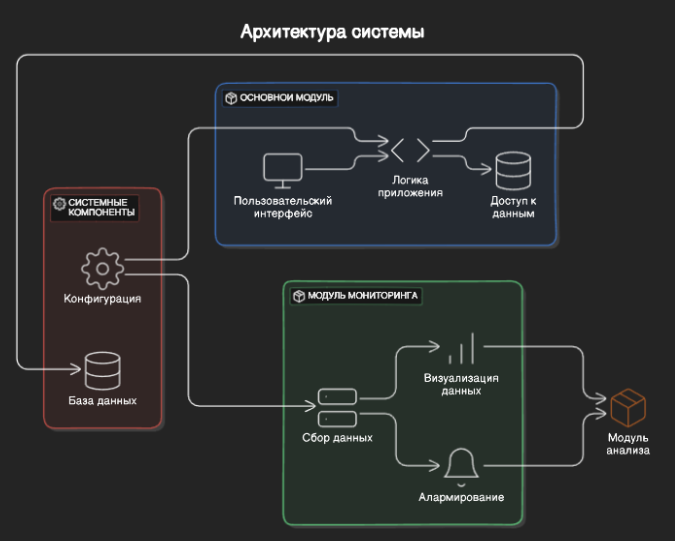
Для данной разработки построена диаграмма пакетов, которая показана ниже на рисунке 3. Она показывает зависимости между частями.

Рисунок 3 – Диаграмма пакетов.

### 2.2.4 Разработка диаграмм классов

Диаграммы классов будут описывать основные сущности системы (студенты, преподаватели, предметы) и их взаимосвязи:

- Класс "Студент" будет содержать атрибуты (имя, фамилия, ID, список оценок).

### 2.2.5 Разработка диаграммы состояний

Рисунок 5 – Диаграмма состояний.

Данная диаграмма позволяет лучше понимать работу системы и последовательность её действий в зависимости от введенных пользователем данных.

### 2.2.6 Разработка диаграммы деятельности

Рисунок 6 – Диаграмма деятельности.

Эта диаграмма деятельности описывает этапы создания программного модуля, который позволяет автоматически анализировать работу охранной службы. Модуль позволяет выявлять нарушения, определять риски, создавать отчеты и оповещать ответственных лиц о нештатных ситуациях.

### 2.2.7 Разработка диаграммы размещения



Рисунок 7 – Диаграмма размещения.

## Проектирование программного модуля

### 2.3.1 Разработка алгоритмов реализации основных функций программного обеспечения

Алгоритм начинается с начинается с запуска ПО.

Далее идет сбор информации полученной от пользователя.

Следующим этапом является анализ полученной информации

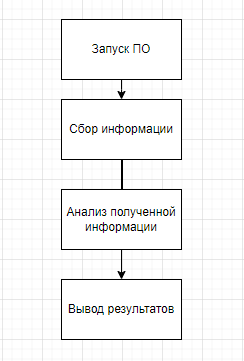
Дальше происходит проанализированный вывод результатов

Рисунок 9 – Алгоритм получения результатов.

### 2.3.2 Проектирование пользовательского интерфейса

Процедурно-ориентированные интерфейсы используют традиционную модель взаимодействия с пользователем, основанную на понятиях «процедура» и «операция». В рамках этой модели программное обеспечение предоставляет пользователю возможность выполнения некоторых действий, для которых пользователь определяет соответствующие данные и следствием выполнения которых является получение желаемых результатов.

Объектно-ориентированные интерфейсы используют несколько иную модель взаимодействия с пользователем, ориентированную на манипулирование объектами предметной области. В рамках этой модели пользователю предоставляется возможность напрямую взаимодействовать с каждым объектом и инициировать выполнение операций, в процессе которых взаимодействуют несколько объектов.

Для того, чтобы понять какой интерфейс использовать для разработки проведена небольшая сравнительная характеристика ниже в таблице 1.

Таблица 1 – Сравнительная характеристика интерфейсов.

|  |  |  |
| --- | --- | --- |
| № | **Процедурно-ориентированные пользовательские интерфейсы** | **Объектно-ориентированные пользовательские интерфейсы** |
|  | | |
| 1 | Обеспечивают пользователей функциями, необходимыми для выполнения задач | Обеспечивают пользователям возможность взаимодействия с объектами |
| 2 | Акцент делается на задачи | Акцент делается на входные данные и результаты |
| 3 | Пиктограммы представляют приложения, окна или операции | Пиктограммы представляют объекты |
| 4 | Содержание папок и справочников отображается с помощью таблиц и списков | Папки и справочники являются визуальными контейнерами объектов |

У данных интерфейсов также есть типы.

Процедурно-ориентированные и объектно-ориентированные интерфейсы делятся на 4 типа: «примитивные», меню и со свободной навигацией, а также прямого манипулирования. Их сравнение приведено ниже.

Примитивным называют интерфейс, который организует взаимодействие с пользователем в консольном режиме. Обычно такой интерфейс реализует конкретный сценарий работы программного обеспечения.

Интерфейс-меню в отличие от примитивного интерфейса позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой. Эти интерфейсы предполагают реализацию множества сценариев работы, последовательность действий в которых определяется пользователем.

Интерфейсы со свободной навигацией также называют графическими пользовательскими интерфейсами - что пользователь видит на экране, то он и получит при печати. Эти названия подчеркивают, что интерфейсы данного типа ориентированы на использование экрана в графическом режиме с высокой разрешающей способностью.

Объектно-ориентированные интерфейсы пока представлены одним видом: интерфейс прямого манипулирования. Этот тип интерфейса предполагает, что взаимодействие пользователя с программным обеспечением осуществляется посредством выбора и перемещения пиктограмм, соответствующих объектам предметной области.

Для данного ПО был выбран интерфейс-меню, так как в отличие от других интерфейсов он позволяет пользователю выбирать необходимые операции из специального списка, выводимого ему программой.

## 2.4 Реализация программного обеспечения на выбранном языке программирования и в выбранной среде разработки

Для реализации программного обеспечения на языке C# для автоматизированной системы мониторинга занятости стояночных мест при помощи программной можно использовать различные среды разработки, такие как Microsoft Visual Studio, JetBrains Rider, SharpDevelop.

Разработка программы в среде Visual Studio будет представлять собой следующие шаги:

1. Создание проекта.

В меню выбирается пункт "Создать проект", а затем - тип проекта, например, "Windows Forms App".

1. Добавление нужных файлов.

В проект необходимо добавить файлы, отвечающие за обработку видео, выделение областей и прочие нужные для реализации функции.

1. Написание кода.

Код программы написанный на языке C# реализует основные функции, описанные выше. В коде используются методы обработки данных о видео или фотографиях, методы для выделения объектов на изображении, алгоритмы классификации, обучения нейронных сетей и многое другое.

1. Тестирование и отладка.

После написания кода программа должна быть протестирована и отлажена. Можно использовать готовые тестовые данные или создать свои, чтобы проверить работу программы.

1. Компиляция и сборка.

Если все тесты пройдены успешно, можно скомпилировать и собрать программу. В результате получится файл с расширением ".exe", который пользователь может запустить на своем компьютере.

1. Развертывание.

Если программа должна быть доступна для пользователей, нужно развернуть ее на соответствующих платформах. Это может быть установка на компьютер пользователя, загрузка на сервер, доступ через Интернет и т.д.

Таким образом, реализация программного обеспечения на языке C# для автоматизированной системы мониторинга занятости стояночных мест при помощи программной видеоаналитики требует ряда шагов и использует средства разработки, доступные на рынке. Важно понимать, что конечное решение может сильно отличаться в зависимости от конкретных потребностей и условий задачи.

## 2.5 Выбор стратегии тестирования, разработка тестов, тестирование и отладка программного обеспечения

Определение стратегии тестирования для обеспечения качества программного обеспечения:

- Проведение юнит-тестирования для проверки отдельных компонентов системы.

- Интеграционное тестирование для проверки взаимодействия между модулями.

Тестирование должно включать следующие этапы:

1. Разработка тестов: Написание тестов на основе требований к функционалу.

2. Тестирование: Проведение тестов на различных этапах разработки.

3. Отладка: Исправление выявленных ошибок до выхода продукта на рынок.

# 3 Безопасность жизнедеятельности

## 3.1 Требования к рабочим помещениям и рабочим местам оператора ПК

Рабочие помещения и рабочие места оператора ПК должны соответствовать определенным требованиям, которые обеспечат комфортную и безопасную работу. К ним относятся:

1. Площадь помещения должна составлять не менее 6 квадратных метров на одно рабочее место.
2. В помещении должно быть естественное и искусственное освещение, соответствующее нормам СанПиН 2.2.2./2.4.1340-03 "Гигиенические требования к условиям труда при использовании персональных компьютеров".
3. Температура в помещении должна быть в пределах 18-24 ° C, а влажность - в пределах 40-60%.
4. Не должно быть шума или других нежелательных звуковых эффектов.
5. В помещении должно быть оборудование для обеспечения пожарной безопасности и пути эвакуации.
6. Мебель должна соответствовать эргономическим стандартам (регулируемая высота, регулируемый наклон, спинка и сиденье, а также подставки для ног и т.д.).
7. В помещении должно быть специальное оборудование для защиты здоровья (экраны, фильтры, держатели документов и т.д.).
8. Параметры компьютера должны соответствовать установленным нормам.
9. Работа оператора ПК должна проходить в микроклимате, обеспечивающем наиболее комфортные условия для зрения (уровень освещенности должен быть не менее 300 люкс).

Необходимая пространственная ориентация должна быть обеспечена с помощью различных способов размещения монитора и клавиатуры.

Общие требования к использованию компьютерных технологий:

1. Защита глаз.
2. Защита органов слуха.
3. Защита пищеварительной системы.
4. Защита кожи.

Составление графика работы и перерывы в течение дня.

## 3.2 Расчет искусственной освещенности рабочего места оператора ПК

Расчет искусственной освещенности рабочего места оператора ПК является необходимым для обеспечения комфортных условий работы с учетом световых параметров, необходимых для нормального функционирования зрительного аппарата человека.

Согласно требованиям СанПиН 2.2.2./2.4.1340-03, искусственное освещение должно обеспечивать не менее 300 лк на поверхности стола и не менее 500 лк на поверхности клавиатуры и экрана монитора.

Для расчета искусственной освещенности рабочего места оператора ПК необходимо знать:

1. Площадь рабочего помещения (S).
2. Коэффициент использования светового потока (K) - показатель, показывающий, какая часть из общей мощности светильника будет использоваться для освещения рабочей поверхности.
3. Световой поток, выделяемый одним светильником (F).
4. Коэффициент освещенности помещения, отношение светового потока к площади помещения (E=F/S).

Искусственная освещенность оператора ПК рассчитывается по формуле:

*E x K x η x T = L, (3)*

гдеE - коэффициент освещенности помещения, лк/м²;

K - коэффициент использования светового потока, принимается примерно равным 0,5;

η - коэффициент светопропускания светопрозрачных элементов в помещении (окна и т.п.), принимается в пределах 0,7-0,9;

T- коэффициент светового рассеивания стен белого цвета, имеющих светлую поверхность или покрытие, принимается равным 0,7 - 0,9;

L - необходимая освещенность поверхности рабочего места оператора ПК, лк.

Согласно стандартам, коэффициент освещенности для помещения размером 15 квадратных метров должен составлять не менее 300 лк/м². Для достижения этого значения и энергоэффективной работы светильника, мы можем рассчитать оптимальную световую мощность и эффективность светильника.

Подставляем значения в формулу:

*300 лк/м² × 15 м² = 4500 лк*

Для достижения необходимой световой мощности в 4500 лм при коэффициенте использования светового потока K=0,5, мы можем пересчитать эффективность светильника по формуле:

*P = (F × K) / η*

*η = (F × K) / P*

*η = (4500 лм × 0,5) / 50 Вт = 45*

Таким образом, чтобы обеспечить достаточный уровень освещенности помещения 15 кв. м с общим световым потоком 4500 лм, достаточно использовать светильники с эффективностью 45 лм/Вт и мощностью 50 Вт. Кроме того, для получения энергоэффективного решения, можно использовать светодиодные светильники, которые потребляют меньше энергии при той же световой мощности.

## 3.3. Техника безопасности при работе на ПК

Современная жизнь невозможна без использования компьютеров и интернета. Тем не менее, работа на ПК может представлять опасность для здоровья человека, если не соблюдать правила безопасности. Такие заболевания, как синдром карпального канала, синдром "сухого глаза", головные боли и шум в ушах, связаны с длительным пребыванием за компьютером. В связи с этим, необходимо соблюдать следующие правила техники безопасности при работе на ПК:

1. Регулярно делать перерывы. При работе за компьютером необходимо делать перерыв каждые 45-60 минут. Во время перерывов рекомендуется выполнять упражнения для глаз и рук.
2. Правильно настроить рабочее место. Рабочее место должно быть правильно настроено, чтобы минимизировать риск различных заболеваний. Клавиатура должна быть расположена на уровне локтя, а экран - на уровне глаз.
3. Использовать эргономическую мебель. Для работы за компьютером рекомендуется использовать эргономические кресла и столы, которые помогают поддерживать правильную позу.
4. Соблюдать правильный режим работы. Необходимо соблюдать правильный режим работы и сна, чтобы избежать усталости и стресса, которые могут привести к заболеваниям.
5. Использовать программы для защиты глаз. Существует множество программ, которые помогают защитить глаза при работе за компьютером. Они уменьшают яркость экрана, фильтруют синий свет и т.д.
6. Использовать антивирусное программное обеспечение. Антивирусное программное обеспечение помогает защитить компьютер от вирусов и злонамеренных программ, которые могут повредить систему или украсть личную информацию.
7. Избегать монотонной нагрузки. Повторяющиеся действия могут привести к снижению производительности и здоровью. Для этого рекомендуется использовать различные программы и ресурсы, менять виды деятельности, чтобы предотвратить нагрузку на конкретную группу мышц.
8. Использовать безопасные пароли. Для защиты личной информации и конфиденциальных данных следует использовать сложные пароли, которые не легко угадать или взломать. Рекомендуется использовать комбинации букв, цифр и символов.
9. Создавать резервные копии данных. Для сохранения важной информации необходимо регулярно создавать резервные копии данных, чтобы предотвратить потерю или повреждение важных файлов.
10. Обновлять программное обеспечение. Регулярное обновление программного обеспечения помогает устранять уязвимости и предотвращать атаки злонамеренных программ. Рекомендуется устанавливать обновления операционной системы, браузера и антивирусного ПО.
11. Использовать безопасную сеть. При работе на публичных сетях необходимо использовать VPN для защиты информации, передаваемой между компьютером и сервером или другим устройством.
12. Избегать использования нелицензионного программного обеспечения. Использование нелицензионного программного обеспечения может привести к нарушению законодательства и повышенному риску заражения вирусами и злонамеренными программами.

В целом, соблюдение правил техники безопасности при работе на ПК имеет важное значение для сохранения здоровья и безопасности человека. Рекомендуется следовать простым правилам, чтобы избежать проблем, связанных с длительным пребыванием за компьютером.