**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПБПУ»)

**Институт среднего профессионального образования**

|  |  |
| --- | --- |
|  | **Допустить к защите**  Заместитель директора по  учебно-методической работе  \_\_\_\_\_\_\_\_\_\_\_\_ Е. Г. Конакина \_  (подпись) (И.О.Ф.)  «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
|  |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема «Разработка модуля учёта прохождения практик студентами в СПб ИАЦ»

специальность 09.02.07 группа 42919/4

Студент (ка) Коннова А.Е.

(подпись) (ФИО)

Руководитель Стариков А. Ю.

(подпись) (ФИО)

Санкт-Петербург­­

2023

СОДЕРЖАНИЕ

[Введение 4](#_Toc136869181)

[1. Общая часть 5](#_Toc136869182)

[1.1 Анализ предприятия 5](#_Toc136869183)

[1.2 Описание предметной области 5](#_Toc136869184)

[1.3 Постановка задач для разработки проекта 6](#_Toc136869185)

[1.4 Анализ требований 6](#_Toc136869186)

[1.5 Постановка задачи 10](#_Toc136869187)

[1.6 Анализ рынка существующих решений 11](#_Toc136869188)

[1.7 Анализ целесообразности выполнения проекта 13](#_Toc136869189)

[1.8 Анализ и выбор методов решения 14](#_Toc136869190)

[1.9 Анализ и выбор средств разработки 14](#_Toc136869191)

[2. Специальная часть 23](#_Toc136869192)

[2.1 Описание структуры приложения 23](#_Toc136869193)

[2.2 Проектирование программы 23](#_Toc136869194)

[2.3 Проектирование пользовательского интерфейса 26](#_Toc136869195)

[2.4 Проектирование базы данных 31](#_Toc136869196)

[2.5 Разработка программы 34](#_Toc136869197)

[2.6 Описание работы программы 36](#_Toc136869198)

[2.7 Тестирование программы 40](#_Toc136869199)

[3 Экономическая часть 45](#_Toc136869200)

[3.1 Расчёт экономической эффективности от внедрения программного продукта 45](#_Toc136869201)

[3.2 Расчет капитальных затрат на проектирование и внедрение 46](#_Toc136869202)

[3.3 Расходы на содержание персонала по обслуживанию программного продукта 49](#_Toc136869203)

[3.4 Расходы на функционирование программы 50](#_Toc136869204)

[3.5 Расчет накладных расходов 50](#_Toc136869205)

[3.6 Прочие расходы 51](#_Toc136869206)

[3.7 Расчет экономии за счет повышения производительности труда пользователя (3.Н) 52](#_Toc136869207)

[4 Техника безопасности и охрана труда 54](#_Toc136869208)

[4.1 Анализ условий труда программиста, работающего в «СПб ИАЦ» 54](#_Toc136869209)

[4.2 Расчет искусственного освещения рабочего места 56](#_Toc136869210)

[4.3 Электробезопасность на предприятии 58](#_Toc136869211)

[4.4 Обеспечение пожарной безопасности 60](#_Toc136869212)

[Заключение 63](#_Toc136869213)

[Список использованных источников 64](#_Toc136869214)

[Приложение А (Справочное) Код Программы 65](#_Toc136869215)

ВВЕДЕНИЕ

Компания «СПб ИАЦ» открыта осенью 1980 года. Организация, работающая в области информатизации и информационного обеспечения государственной власти Санкт-Петербурга и других организаций, предоставления услуг в сфере создания и использования современных информационных и телекоммуникационных систем, средств и технологий.

Целью данного дипломного проекта является упрощение процесса и организации производственных практик руководителями соответствующих практик.

Задачами данного дипломного проекта являются:

* анализ предприятия СПб ГУП «Санкт-Петербургский информационно-аналитический центр»;
* анализ задач для разработки проекта;
* описание задач проекта;
* описание требований к проекту;
* анализ и выбор методов решения;
* анализ и выбор средств разработки;
* проектирование;
* разработка;
* описание работы программы;
* тестирование.

1. Общая часть
   1. Анализ предприятия

Санкт-Петербургский информационно-аналитический центр (СПб ИАЦ) — предприятие, работающее в области информатизации и информационного обеспечения органов государственной власти Санкт-Петербурга и других организаций, а также предоставления услуг в сфере создания и использования современных информационных и телекоммуникационных систем, средств и технологий.

Направления деятельности данного предприятия включают в себя построение единого информационного пространства органов государственной власти Санкт-Петербурга, оказание услуг в области информатизации и информационного обеспечения, внедрение информационных технологий в деятельность исполнительных органов государственной власти Санкт-Петербурга, а также множество других видов деятельности.

Например, одно из направлений заключает в себе разработку технических заданий, проектных решений и рабочей документации на автоматизированные системы в защищенном исполнении, телекоммуникационные системы (в том числе защищенные с использованием криптографических средств), в которых обрабатывается информация, доступ к которой ограничен федеральными законами и/или обладателем информации.

Одним из новых направлений стал проект разработки программного обеспечения для учета прохождения практики студентами.

* 1. Описание предметной области

Информационно-аналитический центр проводит практику для студентов разных учебных заведений.

В настоящее время руководитель практики вручную договаривается о количестве мест для студентов, назначает задачи и собирает данные о студентах.

В зависимости от трудовых потребностей на производстве руководить практики договаривается с учебными заведениями о количестве студентов, которое предприятие готово принять.

Система учета прохождения практик должна обеспечивать добавление информации о студентах, удаление или изменение информации о задачах.

Система учета по требованию пользователя формирует и позволяет скачать таблицы текстовым файлом.

Система осуществляет поиск информации о студентах.

* 1. Постановка задач для разработки проекта

Для оптимизации работы со студентами, проходящими практику на производствах, необходимо разработать веб-сайт.

Конечной целью проекта является реализация веб-приложение модуля производственного предприятия, как централизованной площадки взаимодействия.

Для реализации данного проекта необходимо провести анализы требований, анализ и выбор средств разработки, анализ рынка конкурентов.

* 1. Анализ требований
     1. **Анкетирование заинтересованных лиц**

Анкетирование руководителя практик

* Какие ваши основные обязанности в данной области?

Обеспечение общего контроля за деятельностью студентов, проходящих у нас практику, распределение нагрузки.

* Как Вы будете входить в свою учетную запись?

Я думаю, это будет лучше сделать по логину-пароль, по необходимости будем создавать личные кабинеты и так работать.

* Как список задач для студентов будет пополняться?

Я хотел бы самостоятельно формировать список задач через удобный интерфейс.

* Какие виды отчетов Вам были бы нужны?

Статистика по проделанной работе, по количеству студентов, проходящих у нас практику.

* + 1. **Метод опорных точек**

На основе анкетирования заинтересованных лиц, составлен метод опорных точек, где Руководитель практик и Сотрудники — это опорные точки, а остальное сервисы (Рисунок 1).

Изображение выглядит как текст, снимок экрана, Шрифт, круг

Автоматически созданное описание

Рисунок 1 – Метод опорных точек

* + 1. **Пользовательские истории**

Для формирования требований были созданы пользовательские истории:

* Я как руководитель практик, хочу, чтобы отображался список всех прошедших и предстоящих практик;
* Я как руководитель практик, хочу, чтоб у меня была возможность отмечать некоторых студентов в «избранное»;
* Я как руководитель практик, хочу, чтоб у меня была возможность просматривать информацию о студентах;
* Я как руководитель практик, хочу, чтобы приходили уведомления с предстоящей практикой за неделю;
* Я как руководитель практик, хочу, чтоб в уведомление о практики была информация: ФИО студентов, данные руководителя практики от учебного учреждения, даты практик, основная задача на практику;
* Я как руководитель практик, хочу, чтоб была возможность сортировать студентов по курсам и дате практик;
* Я как руководитель практик, хочу, чтобы возможность сортировать задачи по датам;
* Я как руководитель практик, хочу, чтобы возможность фильтровать задачи по отделам;
* Я как руководитель практик, хочу иметь возможность связываться с руководителями практик от учебного предприятия;
* Я как руководитель практик, хочу, чтоб на главной странице отображался календарь с напоминаниями;
* Я как руководитель практик, хочу видеть статистику за разные периоды времени;
* Я как руководитель практик, хочу, чтоб была возможность выбирать по каким параметрам смотреть статистику;
* Я как руководитель практик, хочу, чтоб была возможность входить в свою личную учетную запись;
* Я как руководитель практик, хочу, чтоб вход в учетную запись осуществлялся чрез логин пароль;
* Я как руководитель практик, хочу, чтобы сотрудники могли оставлять примечания на задачах или сотрудниках.
  + 1. **Построение диаграммы вариантов использования**

Диаграмма вариантов использования в UML — диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне (Рисунок 2).

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 2 – Диаграмма вариантов использования

* 1. Постановка задачи

Проведя анализ требований, описанных выше, получилось сформулировать следующие требования.

* + 1. **Функциональные требования**

Программный продукт должен:

* отображать данные из БД о студентах в виде таблицы;
* отображать данные из БД о задачах в виде таблицы;
* добавлять, редактировать и удалять записи;
* производить поиск по записям;
* производить сортировку по нескольким параметрам;
* иметь функцию выгрузки данных текстовым файлом;
* иметь возможность отображать статистику по выбранным параметрам;
* иметь возможность просматривать историю о всех прошедших и будущих практиках.
  + 1. **Требования к техническим средствам**

Серверная версия программы способна осуществлять работу на машинах средней производительности:

* + - Процессор с частотой выше 1 ГГц x64;
    - Оперативная память от 512 Мб;
    - Место на жестком диске от 4.5 Гб.
    1. **Требования к информационной и программной совместимости**

Клиентской версии программы необходимо:

* + - Браузер Яндекс (последняя версия).

Серверной версии программы необходимо:

* Поддержка Microsoft SQL Server;
* Поддержка Microsoft .NET Framework;
* Веб сервер IIS.
  1. Анализ рынка существующих решений

Для лучшего понимания темы получилось ознакомиться с уже существующими в интернете аналогами, но, к сожалению, раннее никто не занимался данным вопросом и не создавал ничего подходящего.

Удалось на собственном опыте ощутить какие «виртуальные помощники» можно использовать для учета и контроля производственной практики.

Для организации работы с практиками можно использовать различные социальные сети, утилиты, установленные на ваших компьютерах или телефонах, облачные платформы.

Один из примеров – это кроссплатформенная система мгновенного обмена информацией – Telegram (Рисунок 3). При учете возможностей, которые реализованы в данной системе руководители практик могут создавать чаты и вручную писать некоторую информацию, которую невозможно даже никак отсортировать.

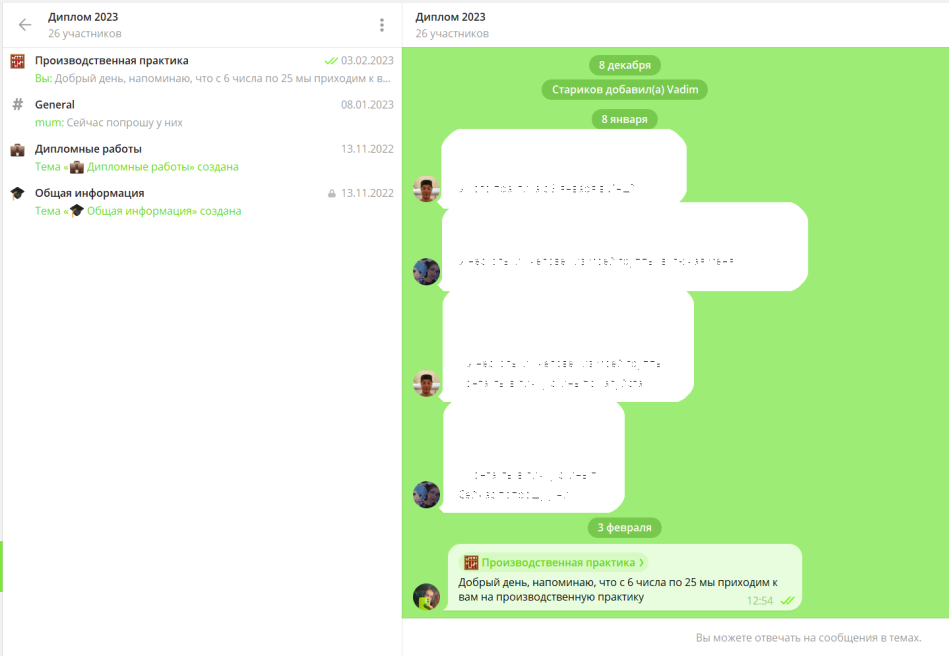


Рисунок 3 - Telegram

Еще одним более или менее подходящим инструментом является «Гугл диск» (Рисунок 4). При помощи данного инструмента удобно создавать таблицы, в которых можно вести учет студентов и задач, а также создавать диаграммы для статистики по выбранным параметрам. В дополнение к этим возможностям есть возможность добавлять доступ коллегам.

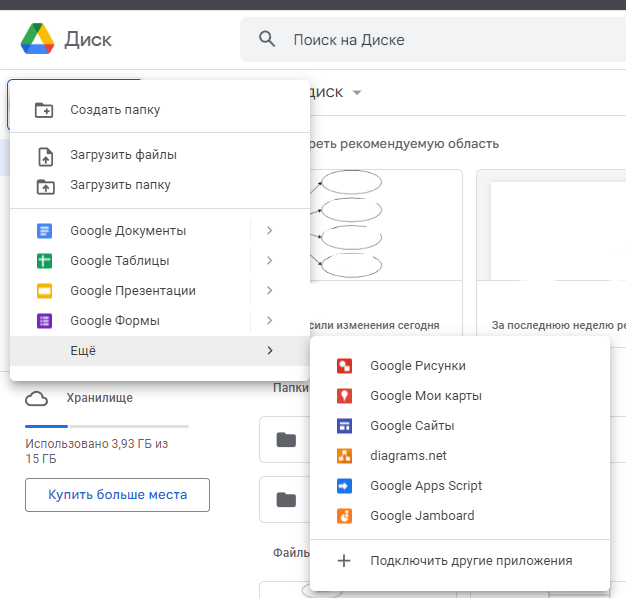


Рисунок 4 – Гугл диск

Всем этим можно пользоваться, но удобнее, если это все сосредоточенно в одном месте, и нет необходимости запоминать больше информации.

* 1. Анализ целесообразности выполнения проекта

Актуальность данного проекта обусловлена желанием оптимизировать работу руководителям практик и помочь предприятиям заблаговременно распределять доступные задачи.

* 1. Анализ и выбор методов решения

Для реализации проекта было принято решение работать над веб-приложение, т.к. некоторые сотрудники предприятия работают удаленно, для них должен быть доступ таким же простым, как и для сотрудников, работающих из офиса.

Веб-приложение – универсально и просто в обслуживание, не требует скачиваний и обновлений со стороны юзера.

* 1. Анализ и выбор средств разработки

Для разработки дипломного проекта было принято решение об использовании следующих технологий.

Для разработки приложения была выбрана Visual Studio в качестве среды разработки. Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений.

Данная среда разработки хороша тем, что:

* большое количество настроек среды разработки под «себя», благодаря встроенным механизмам и доступным дополнениям;
* свой компилятор – MSVC;
* поддержка множества языков при разработке;
* платные версии могут предоставляться учебным заведениям и студентам бесплатно;
* легкий интерфейс.

Недостатки:

* тяжеловесная IDE, достаточно требовательная к железу;
* достаточно маленькая скорость запуска проектов и приложений;
* версии visual studio professional и visual studio enterprise являются платными.
  + 1. **СУБД**

Система управления базами данных (СУБД) – это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными. Иными словами, СУБД — это набор программ, позволяющий организовывать, контролировать и администрировать базы данных. Большинство сайтов не могут функционировать без базы данных, поэтому СУБД используется практически повсеместно.

Среди них для рассмотрения выбраны MySQL и Microsoft SQL Server.

* + - 1. **MySQL**

MySQL - одна из самых популярных СУБД для веб-приложений. Фактически, является стандартом для веб-серверов, которые работают под управлением операционной системы Linux.

MySQL – распространяется, в основном, бесплатно, но при этом для нее постоянно выходят новые версии, расширяя функционал и улучшая безопасность. В бесплатной версии наибольший упор делается на скорость и надежность.

Разработку и поддержку MySQL осуществляет корпорация Oracle.

Эта СУБД позволяет выбирать различные движки для системы хранения, которые позволяют менять функционал инструмента и выполнять обработку данных, хранящихся в различных типах таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц. Она также имеет простой в использовании интерфейс, и пакетные команды, которые позволяют удобно обрабатывать огромные объемы данных. Система невероятно надежна и не стремится подчинить себе все доступные аппаратные ресурсы.

Достоинства MySQL:

- бесплатное распространение;

- доступная документация;

- поддержка набора пользовательских интерфейсов;

- возможность работы с другими базами данных, включая DB2 и Oracle.

Недостатки:

- необходимость настройки для выполнения несложных задач;

- отсутствие встроенной поддержки xml или olap;

- платная поддержка для бесплатной версии.

Подходит для организаций, которым требуется надежный инструмент управления базами данных, но бесплатный.

* + - 1. **Microsoft SQL Server**

Ещё одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно.

Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

К достоинствам MS SQL Server можно отнести:

* простота использования;
* возможность регулирования и отслеживания уровней производительности;
* взаимодействие с другими продуктами Microsoft.

Недостатки:

* высокая цена для юридических лиц;
* возможность использования все доступные ресурсы при тщательной настройке;
* подходит для: крупных организаций, которые уже используют ряд продуктов Microsoft.

По итогу анализа в качестве СУБД выбрана Microsoft SQL Server, которая отлично совместима с выбранной средой разработки.

# Язык программирования

Язык для разработки приложения выбираю в зависимости от масштаба проекта и его задач.

# JavaScript

JavaScript – язык, который в своей основе содержит скрипт. Это – документ, описывающий действия, необходимые для обработки и выполнения. Работать со скриптами сможет каждый браузер, что делает язык кроссплатформенным.

Наибольшая востребованность у этого языка наблюдается в веб-разработке. Программист на JS может создать не только приложение «для браузера», но и небольшую игру, а также полноценный веб-сайт или одностраничник. Клиенты и серверы для JS – ключевые сферы применения на практике.

К преимуществам JavaScript можно отнести следующие моменты:

* наличие полной интеграции с версткой страниц и серверной частью;
* рациональность применения и простоту. Решить элементарную задачу при помощи JS удается за несколько минут;
* сам ЯП обладает понятным синтаксисом, который не требует углубленного изучения;
* скорость работы. JS может сделать итоговое приложение более быстрым и удобным;
* производительность;
* комфортность использования пользовательских интерфейсов.

Почему нужно использовать JavaScript, понятно – это удобный и функциональный язык. Но у него есть свои недостатки. К ним относят:

* отсутствие возможности чтения и загрузки документов. Основная причина наличия этого минуса – соображения безопасности;
* отсутствие удаленного доступа. Полноценно для сетевого ПО соответствующий язык не используется;
* вольную трактовку типов. Нестыковки в коде будут игнорироваться;
* доступность для конкурентов. Связано это с высокой читаемостью исходного кода JavaScript.
  + - 1. **C #**

C# — это язык программирования от компании Microsoft. Изначально его создавали для проектов под Windows, но теперь это по-настоящему универсальный язык: на нём пишут игры, десктопные приложения, веб-сервисы, нейросети и даже графику для метавселенных[1].

Он предусматривает следующие преимущества:

* строгую типизацию;
* функциональность;
* достаточно мощный инструментарий;
* стабильную работу через Visual Studio;
* компактный и легко читаемый код;
* понятный даже новичкам синтаксис;
* налажена работа обработки исключений, а также наличие сборщика мусора.

Си Шарп – язык кроссплатформенного характера. Несмотря на все свои преимущества, недостатки у программирования на нем тоже есть. К ним относят:

* синтаксис, безграмотное применение которого провоцирует разного рода ошибки утилит;
* библиотеки, которые не подойдут для выполнения специфических задач;
* читаемость кода – конкуренты смогут с легкостью «разобрать» программное обеспечение и изучить его состав.

По итогу сравнений двух языков программирования, для дальнейшей работы над проектом был выбран C#.

* + - 1. **XAML**

XAML — это декларативный язык разметки. Как и в случае с моделью программирования .NET, XAML упрощает создание пользовательского интерфейса для приложения .NET. Можно создать видимые элементы пользовательского интерфейса в декларативной XAML-разметке, а затем отделить определение пользовательского интерфейса от логики времени выполнения, используя файлы кода программной части, присоединенные к разметке с помощью определений разделяемых классов.

* + 1. **Фреймворк**

Фреймворки — это программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули, а все специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надёжность решений.

Веб-фреймворк — это платформа для создания сайтов и веб-приложений, облегчающее разработку и объединение разных компонентов большого программного проекта. За счёт широких возможностей в реализации бизнес-логики и высокой производительности эта платформа особенно хорошо подходит для создания сложных сайтов, бизнес-приложений и веб-сервисов.

* + - 1. **WPF**

Платформа Windows Presentation Foundation (WPF) позволяет создавать клиентские приложения для настольных систем Windows с привлекательным пользовательским интерфейсом.

В основе WPF лежит независимый от разрешения векторный модуль визуализации, использующий возможности современного графического оборудования. Возможности этого модуля расширяются с помощью комплексного набора функций разработки приложений, которые включают в себя язык XAML, элементы управления, привязку к данным, макет, двумерную и трехмерную графику, анимацию, стили, шаблоны, документы, мультимедиа, текст и типографические функции. WPF является частью .NET, поэтому вы можете создавать приложения, включающие другие элементы .NET API.

* + - 1. **Windows Forms**

Windows Forms — это платформа пользовательского интерфейса для создания классических приложений Windows. Она обеспечивает один из самых эффективных способов создания классических приложений с помощью визуального конструктора в Visual Studio. Такие функции, как размещение визуальных элементов управления путем перетаскивания, упрощают создание классических приложений.

В Windows Forms можно разрабатывать графически сложные приложения, которые просто развертывать, обновлять, и с которыми удобно работать как в автономном режиме, так и в сети. Приложения Windows Forms могут получать доступ к локальному оборудованию и файловой системе компьютера, на котором работает приложение. С помощью такой среды разработки, как Visual Studio, можно создавать интеллектуальные клиентские приложения Windows Forms, которые отображают информацию, запрашивают ввод от пользователей и обмениваются данными с удаленными компьютерами по сети.

* + - 1. **ASP.NET**

ASP.NET — свободно распространяемая платформа для разработки динамических сайтов и веб-приложений, созданная компанией Microsoft и являющаяся частью NET.Framework. Она является дальнейшим развитием более старой технологии [Microsoft ASP](https://blog.skillfactory.ru/glossary/asp-net/) и сохранила многие ее некоторые внешние признаки и функции, что упрощает переход разработчиков на использование нового инструмента.В качестве языков программирования для разработки приложений на ASP.NET Core используются C# и F#. Для разработки дизайна пользовательского интерфейса используется HTML, CSS и JavaScript.

Проведя анализ трех Фреймворков, было решено, что проект будет разрабатываться на WPF, т.к. привязка данных в WPF превосходит то, что предлагает Windows Forms, WPF проще, чем ASP NET. Большим плюсом является наличие знаний о работе с WPF.

Кроме того, еще одним преимуществом WPF является то, что его легко изучать и применять на практике, поскольку его функции не сложны, и он широко используется для создания приложений и связанных с ними работ.

# Графический редактор

Для создания диаграмм был использован инструмент проектирования Draw.io. Богатая функциональность Draw.io позволяет пользователям эффективно отслеживать и восстанавливать изменения.

Draw.io – это сервис, предназначенный для формирования диаграмм и схем. Сервис разделён на три части – меню, панель объектов и сам документ. Напоминает MS Visio и возможно сделан под него, но приложение от Microsoft – платная, а онлайн-сервис Draw.io – бесплатный. С помощью веб-сервиса Draw.io можно создавать: диаграммы, UML-модели, графики, блок-схемы, формы.

1. Специальная часть
   1. Описание структуры приложения

Веб-приложение состоит из трех основных компонентов:

-Client, т.е часть, отвечающая за графическое представления сервиса;

-Server, т.е часть, обрабатывающая запросы, поступающие из клиентской части сервиса;

-BaseDate, база данных;

Взаимосвязь компонентов веб-сервиса представлена на рисунке 5.

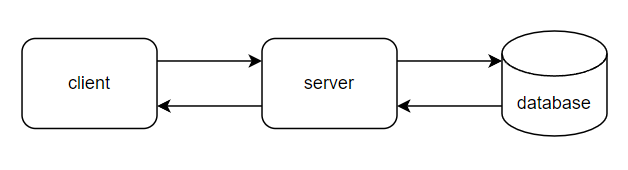


Рисунок 5 – Структура веб-приложения

* 1. Проектирование программы
     1. **Построение диаграммы состояний**

Диаграммы состояний применяются для того, чтобы объяснить, каким образом работают сложные объекты (Рисунок 6).

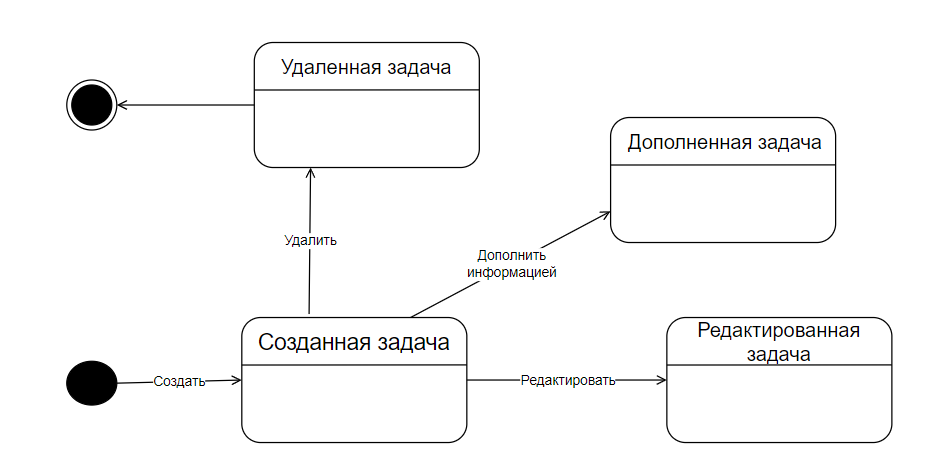


Рисунок 6 – Диаграмма состояний для объекта «Задача»

* + 1. **Построение диаграммы последовательностей**

Диаграммы последовательностей используются для уточнения диаграмм прецедентов, более детального описания логики сценариев использования. Это отличное средство документирования проекта с точки зрения сценариев использования. Данная диаграмма последовательностей представляет процесс входа в систему пользователя (Рисунок 7).

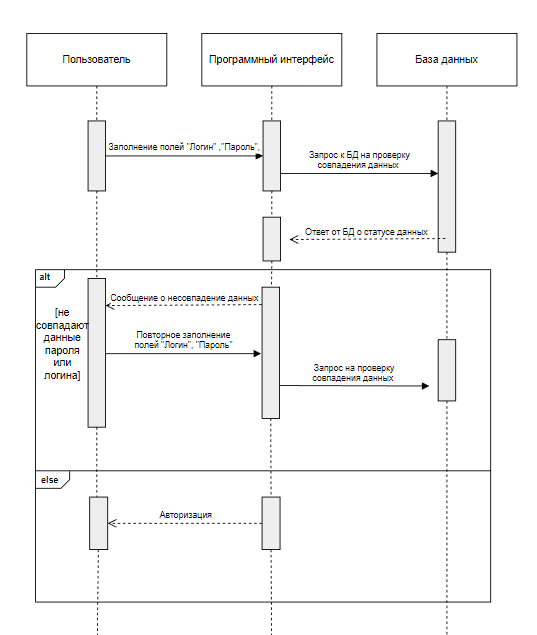


Рисунок 7 – Диаграмма последовательностей для процесса входа в систему

* + 1. **Построение диаграммы активности**

Диаграммы активностей (Activity Diagrams) являются представлением алгоритмов неких действий (активностей), выполняющихся в системе. Данная диаграмма активности создана для отображения процесса скачивания отчета (Рисунок 8).

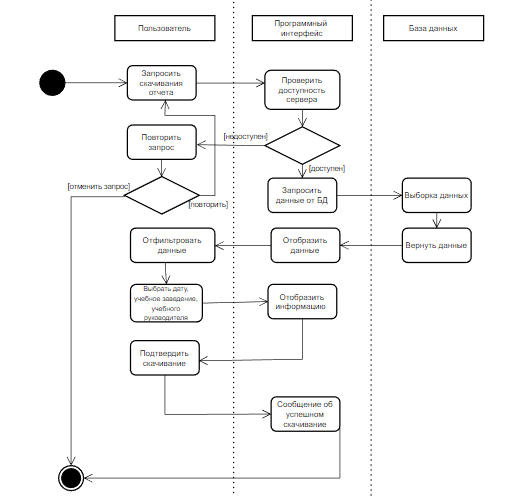


Рисунок 8 – Диаграмма активности процесса скачивания отчета

* 1. Проектирование пользовательского интерфейса
     1. **Карта навигаций сайта**

Для наглядности структуры сайта создается карта сайта (Рисунок 9). Карта сайта представляет собой схему, отображающая связь между страницами сайта.

Изображение выглядит как диаграмма, План, текст, Технический чертеж

Автоматически созданное описание

Рисунок 9 – Иерархия меню

* + 1. **Макеты графического интерфейса**

Для подробного представления страниц сайта создаются схематические иллюстрации страниц (форм), на которых отображено расположение и содержание отдельных элементов интерфейса пользователя (Рисунки 10 - 16).



Рисунок 10 – Окно страницы входа



Рисунок 11 – Окно регистрация



Рисунок 12 – Страница личного профиля

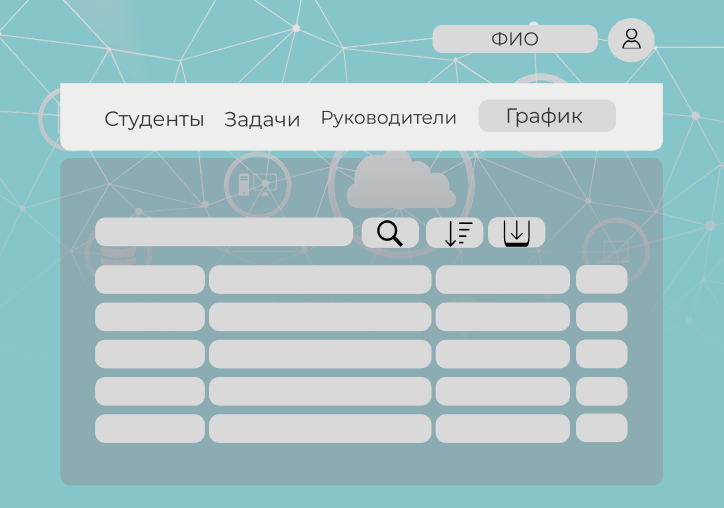


Рисунок 13 – Окно вкладки «График»

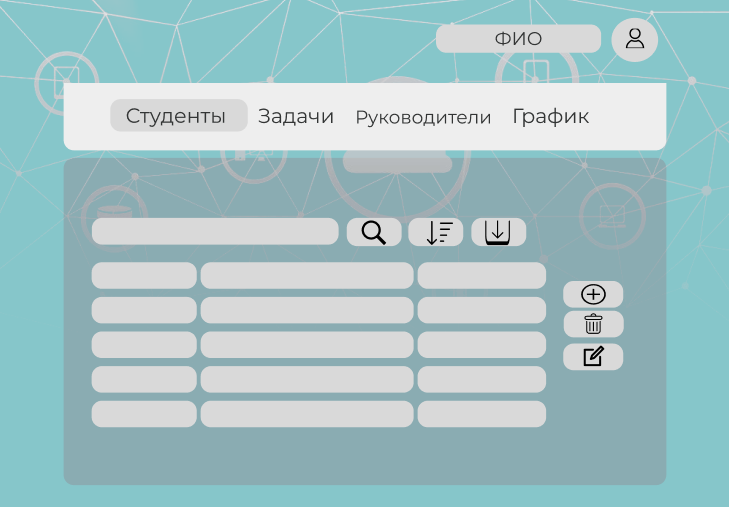


Рисунок 14 – Окно вкладки «Студенты»

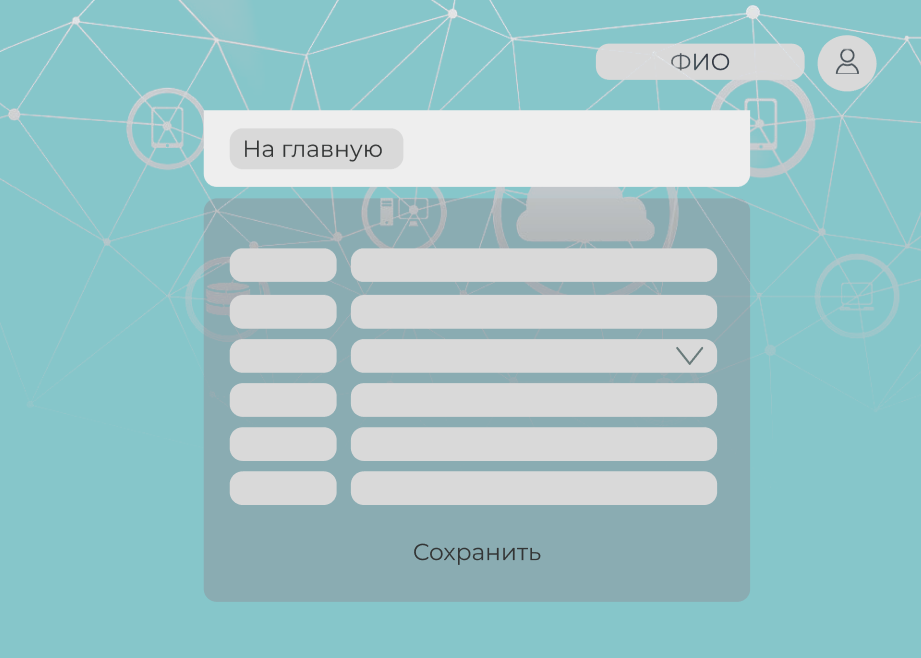


Рисунок 15 – Окно добавления новых записей

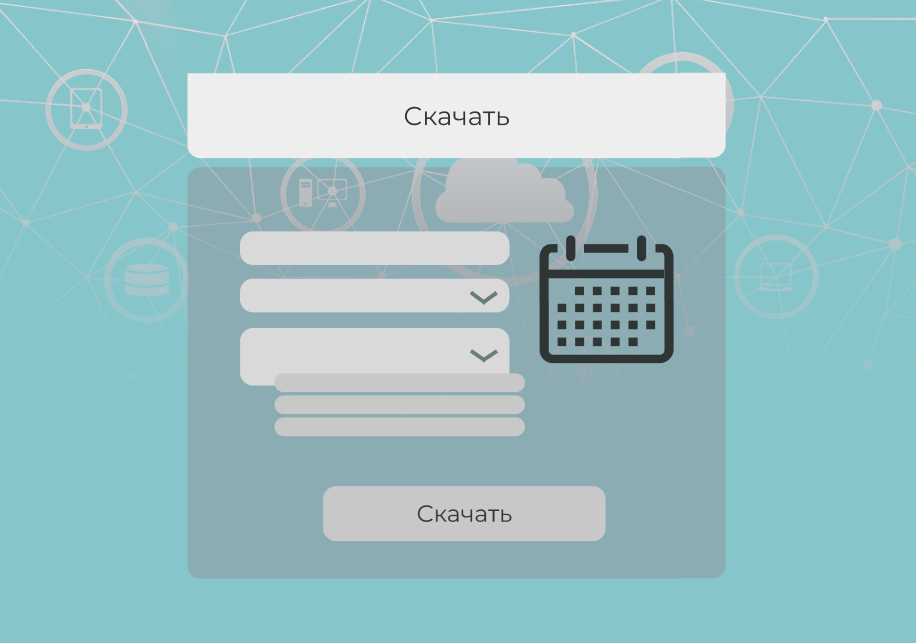


Рисунок 16 – Окно скачивания отчетов

* 1. Проектирование базы данных
     1. **Схема базы данных**

Схема базы данных представляет собой логическую конфигурацию реляционной базы данных. Схема представлена в виде наглядного представления БД, отображает связь между студентами, учебными заведениями, задачами, графиком и пользователями (Рисунок 17).

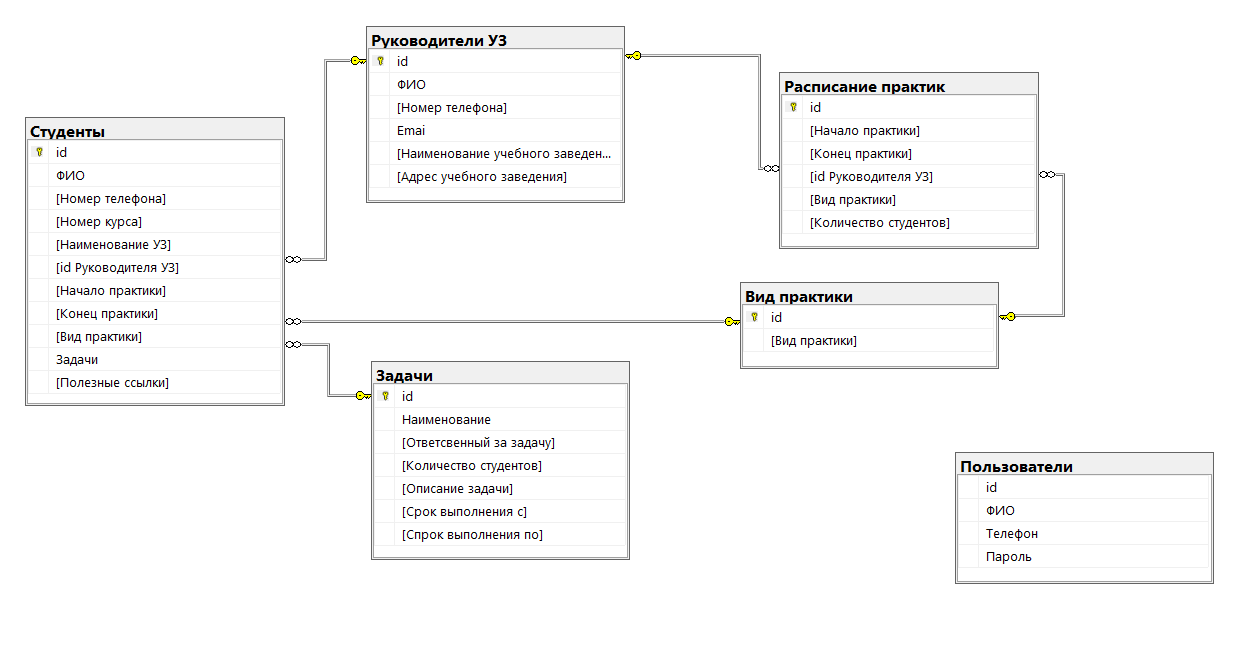


Рисунок 17 – Схема базы данных

* + 1. **Словарь данных**

Словарь данных, разработанный по имеющиеся базе данных представлено ниже (Таблицы 1-6)

Таблица 1 – Описание сущности Вид практики

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Тип** | **NULL** | **Ограничения** |
| Id | Int | Not null | PK |
| Вид практики | nvarchar(50) | Not null | Нет |

Таблица 2 – Описание сущности Руководители УЗ

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Тип** | **NULL** | **Ограничения** |
| Id | Int | Not null | PK |
| ФИО | nvarchar(MAX) | Not null | Нет |
| Номер телефона | nvarchar(MAX) | Not null | Уникален |
| Email | nvarchar(MAX) | Not null | Уникален |
| Наименование учебного заведения | nvarchar(MAX) | Not null | Нет |
| Адрес учебного заведения | nvarchar(MAX) | Not null | Нет |

Таблица 3 – Описание сущности Расписание практик

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Тип** | **NULL** | **Ограничения** |
| Id Заявки | Int | Not null | PK |
| Начало практики | data | Not null | Нет |
| Конец практики | data | Not null | Нет |
| Id Руководителя УЗ | Int | Not null | FK Руководители УЗ |
| Вид практики | Int | Not null | FK Вид практик |
| Количество студентов | Int | Null | Нет |

Таблица 4 – Описание сущности Пользователи

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Тип** | **NULL** | **Ограничения** |
| Id | Int | Not null | PK |
| ФИО | nvarchar(MAX) | Not null | Нет |
| Телефон | nvarchar(50) | Not null | Уникален |
| Пароль | nvarchar(50) | Not null | Захэширован |

Таблица 5 – Описание сущности Задачи

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Тип** | **NULL** | **Ограничения** |
| Id | Int | Not null | PK |
| Наименование | nvarchar(MAX) | Not null | Нет |
| Ответственный за задачу | nvarchar(MAX) | Null | Нет |
| Количество студентов | Int | Null | Нет |
| Описание задачи | nvarchar(MAX) | Null | Нет |
| Срок выполнения с | date | Not null | Нет |
| Срок выполнения по | date | Not null | Нет |

Таблица 6– Описание сущности Студенты

| **Название** | **Тип** | **NULL** | **Ограничения** |
| --- | --- | --- | --- |
| Id | Int | Not null | PK |
| ФИО | nvarchar(MAX) | Not null | Нет |
| Номер телефона | nvarchar(50) | Not null | Нет |
| Номер курса | Int | Not null | Нет |
| Наименование УЗ | nvarchar(MAX) | Not null | Нет |
| id Руководителя УЗ | Int | Not null | FK Руководители УЗ |
| Начало практики | date | Not null | Нет |
| Конец практики | date | Not null | Нет |
| Вид практики | Int | Not null | FK Вид практики |
| Задачи | Int | Not null | FK Задачи |
| Полезные ссылки | nvarchar(MAX) | Not null | Нет |

* 1. Разработка программы

После проектирования базы данных начинается разработка программы, то есть ПО для системы учета практик на производстве.

В коде окна входа прописана проверка на совпадения пароля и логина, при несовпадении данных осуществление перехода на главную страницу не осуществляется (Рисунок 18). Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 18

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 19

Метод «DoRequest» позволяет выполнять SQL-запросы к базе данных, используя подключение, которое было создано ранее, без возврата результата. Этот метод полезен, когда нужно выполнить запрос, который не требует возврата значений, например, запрос на изменение данных в базе данных.

Метод «GetRequest» позволяет выполнять SQL-запросы к базе данных, используя подключение, которое было создано ранее, и возвращать результат выполнения запроса в виде строки.

Метод «GetTable» представляет собой статический метод, который принимает строковый аргумент - SQL-запрос к базе данных, и возвращает результат выполнения этого запроса в виде объекта DataTable.

* 1. Описание работы программы

При запуске программы открывается окно входа, для продолжения работы ожидается ввод логина и пароля, имеющихся в базе данных (Рисунок 20).

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 20 – Вход

После удачной авторизации пользователь попадает на главную страницу (Рисунок 21).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 21 – Главная страница

Пользователь может просматривать информацию о студентах и задачах. Он свободно может работать с таблицами: сортировать, добавлять, удалять, редактировать информацию (Рисунки 22–23).

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 22 – Окно «Студенты»

Изображение выглядит как текст, снимок экрана, дисплей, число

Автоматически созданное описание

Рисунок 23 – Окно «Задачи»

Также пользователь имеет доступ к информации об учебных заведениях и графике. Список возможностей на этих вкладках немного ограничен, пользователю доступна сортировка (Рисунок 24–25).

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 24 – Окно «Учебные заведения»

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 25 – Окно «График»

На вкладке «Личный профиль» пользователю в будущем будет доступна функция смены пароля, сейчас же в данном окне указана краткая информация о пользователе (Рисунок 26).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 26 – Личный профиль

У пользователя имеется ряд возможностей при работе с таблицами: сортировка записей (Рисунок 27), добавление записей (Рисунок 28).

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 27 – Функция «Сортировка»

Изображение выглядит как текст, число, линия, Шрифт

Автоматически созданное описание

Рисунок 28 – Добавление записей в таблицу

* 1. Тестирование программы

Тестирование — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов. В данном проекте тестирование проводится посредством тестового случая (Тест-case).

Для ручного функционального тестирования были разработаны тест-кейсы. Тестирование через тест-кейсы — это формально описанный алгоритм тестирования программы, специально созданный для определения возникновения в программе определённой ситуации, определённых выходных данных.

В ходе данного тестирования была проверена работоспособность функции создания, удаления и сортировки записей, загрузка фотографий и соблюдения маски ввода дат (Таблицы 7-11). Успешно пройдены все тесты.

Таблица 7 – Тест-кейс 1

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_1 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка работоспособности функции «Удаления» |
| **Краткое изложение теста** | Тест должен проверить корректную работу функции удаления, при нажатии на кнопку должно открываться сообщение подтверждения удаления. |
| **Этапы теста** | 1. Запустить приложение 2. Зайти на любую вкладку 3. Выбрать нужную запись 4. Нажать на кнопку «Удалить» 5. Подтвердить удаление, нажать на кнопку «Удалить» |
| **Тестовые данные** | Отсутствуют |
| **Ожидаемый результат** | Появления подтверждающего сообщения для безопасного удаления данных. |
| **Фактический результат** | Результат соответствует ожидаемому |
| **Статус** | Пройдено |
| **Предварительное условие** | - |
| **Постусловие** | Система работает стабильно |
| **Примечания/комментарии** | - |

Таблица 8 – Тест-кейс 2

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_2 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка работоспособности сортировки на странице «Студенты» |
| **Краткое изложение теста** | Тест должен проверить корректную работу сортировки на странице «Студенты» по алфавиту |
| **Этапы теста** | 1. Запустить приложение 2. Зайти на вкладку «Студенты» 3. Нажать на кнопку сортировки |
| **Тестовые данные** | Отсутствуют |
| **Ожидаемый результат** | При нажатии на кнопку фамилии сортируются от «я» до «а», при повторном нажатии сортировка происходит от «а» до «я» |
| **Фактический результат** | Результат соответствует ожидаемому |
| **Статус** | Пройдено |
| **Предварительное условие** | - |
| **Постусловие** | Система работает стабильно |
| **Примечания/комментарии** | По умолчанию сортировка происходит в формате от «а» до «я» |

Таблица 9 – Тест-кейс 3

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_3 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка работоспособности поиска на странице «Задачи» |
| **Краткое изложение теста** | Тест должен проверить корректную работу поиска на странице «Задачи» по первым буквам наименования конкретной задачи |
| **Этапы теста** | 1. Запустить приложение 2. Зайти на вкладку «Задачи» 3. В поисковой строке начать вводить название |
| **Тестовые данные** | Отсутствуют |
| **Ожидаемый результат** | При вводе первых букв в таблице должны отображаться только подходящие по названию записи |
| **Фактический результат** | Результат соответствует ожидаемому |
| **Статус** | Пройдено |
| **Предварительное условие** | - |
| **Постусловие** | Система работает стабильно |
| **Примечания/комментарии** | - |

Таблица 10 – Тест-кейс 4

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_4 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка работоспособности загрузки фото в личный профиль. |
| **Краткое изложение теста** | Тест должен проверить возможность добавления фотографии пользователя. |
| **Этапы теста** | 1. Запустить приложение 2. Зайти на вкладку «Личный профиль» 3. Нажать кнопку «Загрузить фото» 4. Выбрать картинку в проводнике 5. Нажать «Открыть» 6. Нажать кнопку «Сохранить» |
| **Тестовые данные** | Новый файл |
| **Ожидаемый результат** | После сохранения файла в личном профиле пользователя обновленное фото. |
| **Фактический результат** | Результат соответствует ожидаемому |
| **Статус** | Пройдено |
| **Предварительное условие** | Желаемое фото должно заранее сохранено на компьютер |
| **Постусловие** | Система работает стабильно |
| **Примечания/комментарии** | - |

Таблица 11 – Тест-кейс 5

|  |  |
| --- | --- |
| Тестовый пример **#** | TC\_UI\_5 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Проверка работоспособности функции смены пароля |
| **Краткое изложение теста** | Тест должен проверить корректную работу функции смены пароля в личном профиле пользователя |
| **Этапы теста** | 1. Запустить приложение 2. Открыть окно личного профиля 3. Нажать на кнопку сменить пароль 4. В открывшемся окне ввести старый и новый пароль 5. Нажать кнопку «Сохранить» |
| **Тестовые данные** | Отсутствуют |
| **Ожидаемый результат** | Пароль должен обновиться |
| **Фактический результат** | Результат соответствует ожидаемому |
| **Статус** | Пройдено |
| **Предварительное условие** | - |
| **Постусловие** | Система работает стабильно |
| **Примечания/комментарии** | - |

1. Экономическая часть
   1. Расчёт экономической эффективности от внедрения программного продукта

Санкт-Петербургский информационно-аналитический центр (СПб ИАЦ) — предприятие, работающее в области информатизации и информационного обеспечения органов государственной власти Санкт-Петербурга и других организаций, а также предоставления услуг в сфере создания и использования современных информационных и телекоммуникационных систем, средств и технологий.

Цель работы: «Разработать модуль учета прохождения практик студентами на предприятие», которая обеспечивает:

* регистрацию и авторизацию пользователей;
* создание, изменение и удаление информации об учебных заведениях;
* создание, изменение и удаление практик;
* составление отчетов;
* редактирование данных студентов.

Снижение трудозатрат на предприятии возможно за счет автоматизации работы с документами, снижение затрат на поиск и обработку информации.

Критерием эффективности создания и внедрения новых средств автоматизации является ожидаемый экономический эффект, он определяется по формуле 1.

, (1)

где – годовая экономия;

= 0,15 – нормативный коэффициент;

– капитальные затраты на проектирование и внедрение, включая первоначальную стоимость программы.

Годовая экономия складывается из экономии эксплуатационных расходов и экономии в связи с повышением производительности труда пользователя, таким образом получаем формулу:

, (2)

где и – эксплуатационные расходы до и после внедрения разрабатываемой программы;

– экономия от повышения производительности труда пользователей.

* 1. Расчет капитальных затрат на проектирование и внедрение

В случае если оценивать экономический эффект с учетом всех деталей, то капитальные затраты на проектирование и внедрение рассчитываются с учетом длительности на этом этапе.

Рассмотрим подробнее расчет капитальных затрат на проектирование и внедрение системы автоматизации. Под проектированием понимается совокупность работ, которые необходимо выполнить, чтобы спроектировать системы, часть системы или поставленную задачу. Под внедрением понимается совокупность (комплекс) работ по вводу в промышленную эксплуатацию системы с возможными ее доработками.

Для расчёта затрат на этапе проектирования необходимо определить продолжительность каждой работы, начиная с составления ТЗ, заканчивая оформлением документа.

Продолжительность работ определяется на основании экспертных оценок по формуле:

, (3)

где – ожидаемая продолжительность работ;

и – наибольшая и наименьшая по мнению эксперта длительность работ.

Данные расчетов ожидаемой длительности работ приведены в таблице 12.

Таблица 12 – Длительность работ на этапе проектирования

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование работ** | **Длительность работ, дней** | | |
| **Мин.** | **Макс.** | **Ожид., To** |
| Разработка технического задания | 3 | 6 | 4 |
| Анализ технического задания | 6 | 10 | 8 |
| Разработка алгоритма | 4 | 6 | 5 |
| Создание разметки | 5 | 7 | 5 |
| Программирование | 20 | 24 | 23 |
| Тестирование | 6 | 12 | 9 |
| Отладка программы | 10 | 16 | 13 |
| Оформление документации | 3 | 4 | 3 |
| **Всего:** |  |  | **70** |

Капитальные затраты на этапе проектирования Кn рассчитываются по формуле:

, (4)

где – заработная плата специалиста на всех этапах проектирования и внедрения;

– затраты на использования ЭВМ на этапе проектирования и внедрения;

– накладные расходы на этапе проектирования и внедрения.

Одним из основных видов затрат на этапе проектирования является заработная плата специалиста, которая определена или рассчитывается по формуле:

*,* (5)

где – заработная плата разработчика на этапе проектирования;

*–* дневная заработная плата на этапе проектирования;

*–* общая ожидаемая длительность работ (в днях, берем из таблицы 13);

*–* процент отчисления на социальное страхование;

*–* процент премий (если предусмотрено)

Средняя зарплата программиста в день составляет 3000 р. Работодатель отчисляет 30% в фонд социального страхования от назначенной им заработной платы (премии не предусмотрены). = 3000\*70\*(30%) = *273000 р.*

В общем случае расходы на машинное время состоят из расходов на процессорное и дисплейное время. Расходы на машинное время рассчитываются по формуле:

*,* (6)

где и – стоимости одного часа процессорного и дисплейного времени;

и – процессорное и дисплейное время необходимое для решения задач (в часах).

При расчете Мп следует учитывать время на подготовку исходных текстов программ, их отладку и решение контрольных примеров. Так как программа разработана на быстродействующих компьютерах, то в дополнительном процессорном времени необходимости нет, таким образом значения Cп и tп принимаются нулевыми.

Из общего времени – 70 дней приходится на использование ПК. Стоимость одного часа процессорного и дисплейного времени составляет 20 рублей. Предположим, что времени на разработку в день уходит 6 часов, тогда M = 20 \* 6 \* 70 = 8400 руб.

Расходы на эксплуатационные принадлежности на этапе проектирования (накладные расходы, Н) определяется подсчетом затрат на их приобретение, в данном случае это будут затраты интернет.

Стоимость интернета – 467 руб./мес., то есть 15 руб./день (70 \* 15 = 1050 руб.)

Итого H = 1050 руб., следовательно, капитальные затраты на этапе проектирования составят: Кп = 273000 + 8400 + 1050 = 282450 руб.

* 1. Расходы на содержание персонала по обслуживанию программного продукта

Расходы по различным видам работающих определяются по формуле:

, (7)

где ni – численность персонала i вида, связанного с выполнением работ;

zi – среднегодовая заработная плата работника i вида;

Ac – процент отчислений на социальное страхование;

An – среднегодовой процент премии (если предусмотрена).

Предполагается, что с разработанной программой будет работать 1 человек. Также предполагается, что сопровождение и устранение неполадок в системе будет занимать около 4-х часов в неделю. Заработная плата – 3000 рублей в день при 8-часовом рабочем дне (375 рублей в час). За месяц при 4-х рабочих часах в неделю – 375 \* 4 \* 4 = 6000 рублей в месяц. Премии не предусмотрены. Среднегодовая заработная плата равна 72000 рублей в год, следовательно, Z = 72000 \* 1,30 = 93600 рублей.

* 1. Расходы на функционирование программы

Расходы на функционирование программы складываются из затрат на машинное время и затрат на эксплуатацию различных принадлежностей (накладные расходы). Из формулы (6) произведем расчет расходов на функционирование программы.

При этом, можно оценить расходы до внедрения (М\_1) и после внедрения (М\_2) и сравнить полученные значения. При внедрении программы уменьшается время работы с одной и той же задачей, за счет этого уже появляется экономия.

До внедрения программы решением задачи занимались на компьютере, но вручную проводили расчеты и анализ с использованием различных программ, позволяющих хранить и обрабатывать данные. В качестве исходных данных будем использовать данные, приведенные в таблице 2.

Работа за компьютером для решения задачи одним человеком занимает около 3-х часов в день. В году 240 рабочих дней, следовательно, М\_1=20\*3\*240 = 14400 руб.

Работа за компьютером для решения задачи при работе с программой одного человека в среднем занимает около 1 часа в день (таб.2). В соответствии с этим, расходы на функционирование программы составляют: М\_2=20\*1\*240 = 4800 руб.

* 1. Расчет накладных расходов

Затраты на накладные расходы определяются подсчетом стоимости на их приобретение. В данном случае это будут затраты на интернет, дополнительные программные средства.

* стоимость среды разработки (ms visual studio – 2490 рублей);
* стоимость интернета – тариф 467 рублей в месяц, то есть 15 рублей в день, 70\*15= 1050.

Итого Н = 3540 рублей.

* 1. Прочие расходы

Прочие расходы составляют от 1 до 3-х процентов от суммы всех эксплуатационных расходов. Прочие расходы до внедрения программы определяются по формуле:

Р\_пр1=(Z+M\_1+H)\* 0,03, (8)

Соответственно, прочие расходы до внедрения программы равны:

Р\_пр1=(93600+14400+3540)\*0,03=3346 руб.

Прочие расходы после внедрения программы определяются по формуле:

Р\_пр2=(Z+M\_2+H)\* 0,03 , (9)

Соответственно прочие расходы после внедрения программы равны:

Р\_пр2=(93600+4800+3540)\*0,03=3058 руб.

Таким образом, эксплуатационные расходы составляют:

* до внедрения программы (формула 10)
* после внедрения программы (формула 11)

(10)

(11)

114886 руб.

104998 руб.

* 1. Расчет экономии за счет повышения производительности труда пользователя (3.Н)

Если пользователь i вида работ с применением программы экономит Ti часов, то повышение производительности труда Рi (в процентах) определяется по формуле:

Рi=((∆Tj) / (Fj \* ∆Tj )), (12)

где F\_j – время, которое планировалось пользователем для выполнения работы j вида до внедрения программы (в минутах);

∆T\_j – экономия времени (в минутах).

Таблица 13 – Работы пользователей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Вид работ** | **До автоматизации, мин (Fj)** | **После автоматизации, мин (Ti)** | **Экономия времени, мин**  **(Fj – Ti)** | **Повышение производительности труда Pi**  **(в %)** |
| Ввод информации | 40 | 20 | 20 | 100 |
| Проведение расчетов | 20 | 5 | 15 | 300 |
| Подготовка и печать отчетов | 20 | 10 | 10 | 100 |
| Анализ и выборка данных | 60 | 20 | 40 | 200 |

Экономия, связанная с повышением производительности труда пользователя РП определяется по формуле:

*,* (13)

где – заработная плата пользователя за месяц.

Предполагается, что заработная плата пользователя программы составляет 50000 рублей в месяц. Экономия, связанная с повышением производительности труда пользователя, будет равна:

= 50.000 \* (700/100) = 350000, тогда

Эр = (114886 –104998) +350000 = 359888 рублей (по формуле 2).

В итоге получаем следующую ожидаемую экономическую эффективность:

Согласно анализу экономической эффективности, использование автоматизации принесет выгоду, особенно в долгосрочной перспективе. Внедрение автоматизации может привести к корректировке процессов, ускоряя выполнение задач. Что позволит сотрудникам обрабатывать больше информации в рабочее время. Руководители смогут тратить меньше времени на проверку документации и оформлении отчетности.

Важно отметить, что если одно программное решение автоматизирует различные подразделения, то снижаются затраты на организацию документооборота между ними. Это приводит к сокращению как временных, так и материальных затрат.

1. Техника безопасности и охрана труда
   1. Анализ условий труда программиста, работающего в «СПб ИАЦ»

Комфорт и безопасность рабочего места для программиста — это один из главных критериев эффективности работы любого работника. Рабочее место программиста — это его рабочий кабинет, либо часть помещения, в котором он проводит большую часть работы. Совершенно очевидно, что чем лучше приспособлено рабочее место для выполнения функций программиста, тем производительнее и эффективнее будет его трудовая деятельность.

Главными критериями комфорта являются: удобство рабочего кресла, вентиляция воздуха, температурный режим, эстетические характеристики офиса и цвет мебели, комфортное освещение.

В офисе стоят кресла с регулируемой высотой, спинкой и подлокотниками. Тем самым кресло можно отрегулировать для каждого человека отдельно. Что позволяет минимизировать риск заработка заболеваний связанные с расстройством позвоночника и суставами. Кресла окрашены в черный цвет, благодаря этому не раздражают глаза и не отвлекают от работы. Важно отметить, что кресла имеют поворотный механизм, что позволяет делать поворот на 360 градусов. После регулировки, кресло надежно фиксируется. Подлокотники регулируются по высоте и внутреннему расстоянию между ними. Спинка кресла регулируется по углу наклона.

Имеются подставки для ног, которые регулируются по высоте в и имеют угол наклона опорной поверхности до ~20 градусов. Имеет ширину 300 миллиметров и глубину 450 миллиметров. Поверхность подставки имеет рифлёную текстуру. Также предусмотрен бортик у переднего.

Рабочий стол программиста обеспечивает возможность размещение всего необходимого оборудования. В офисе стоят регулируемые рабочие столы, высота которых, может изменяться. Высота регулируется легко из положения сидя, надежно фиксируется и имеет легкое управление. Глубина рабочего стола составляет 800 миллиметров, а ширина 1600 миллиметров. Имеет пространство для ног. Рабочий стол имеет гладкие углы, покрытие стола состоит из диффузно отражающего материала с коэффициентом отражения 0,50.

Для создания и поддержания независимо от наружных условий оптимальных значений температуры, влажности, чистоты воздуха, в холодное время используется водяное отопление, в теплое время года применяется кондиционирование воздуха.

Кабинет, в котором расположено рабочее место, представляет собой помещение площадью 20 квадратных метров и высотой 3 метра.

У каждого программиста имеется ЭВМ, со средними характеристиками на 2021 позволяющими решать разные рабочие задачи с максимальным комфортом и быстродействием.

Монитор, а чаще всего 2 монитора находятся на столе в таком положении, чтобы человек мог распознавать изображение на любой части монитора, без необходимости изменения положения головы. Дисплей установлен ниже уровня глаз человека, угол наблюдения экрана человеком относительно горизонтальной линии взгляда 60 градусов. Монитор соответствует уровню излучения и визуальным параметрам – по ГОСТ Р 50948.

Клавиатура располагается так, чтобы обеспечивалась оптимальная видимость экрана. Имеет возможность свободного перемещения. Клавиатура обычно расположена на расстоянии ~ от 100 миллиметров до 300 миллиметров от переднего края.

В кабинете, где работают программисты также предусмотрена оргтехника. Дабы ограничить сотрудников от вредных токсичных газов, которые выделяются при работе, шумов и вибраций комплектующая оргтехника расположена достаточно далеко от самих специалистов. Данные факторы оказывают гораздо меньшее негативное влияние, нежели излучение от компьютера, но также являются вредными и могут нанести вред здоровью сотрудника.

Режим работы программиста - односменный с пятидневной рабочей неделей и нормированным рабочим днем. Продолжительность ежегодного отпуска - 28 календарных дней.

У разработчиков имеется время для обеда, в промежутке от 13:00 до 14:00. Также имеются перерывы в течении 15 минут в каждом рабочем часу. Рабочий день длится от 08:00 до 18:00.

Офис расположен рядом с торговыми центрами и рядом с офисом развита инфраструктура. Офис расположен рядом с метро, что позволяет удобно добираться. Имеется рядом территория с лавочками и деревьями. Позволяющая проводить время на улице во время перерывов.

* 1. Расчет искусственного освещения рабочего места

Через зрительный анализатор человек получает около 80% из общего объема информации. Качество поступающей информации во многом зависит от освещения: неудовлетворительное количественно или качественно, оно не только утомляет зрение, но и вызывает утомление организма в целом.

Кроме того, нерациональное освещение может явиться причиной травматизма: плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени ухудшают видимость настолько, что вызывают полную потерю ориентировки работающих, снижают производительность труда и увеличивают брак продукции.

Поэтому необходимо достаточное искусственное освещение для рабочего места.

Условия работы для расчета показателя освещенности рабочего места программиста являются:

* помещение размером 20 квадратных метров;
* высота помещения 3 метра, длина 5 метра, ширина 4 метров;
* лампы люминесцентные (дневного света) в количестве 5 штук, новые;
* крупногабаритная мебель отсутствует.

Расчет общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента использования светового потока. Световой поток (лм) одной лампы рассчитывается по следующей формуле:

 (1)

где Ен - нормируемая минимальная допустимая освещенность по СП 1.3330.2016, для помещения 2000 лк;

S - площадь освещаемого помещения, м2;

z - коэффициент неравномерности освещения, z = 1,1;

K - коэффициент запаса, учитывающий снижение освещенности в процессе эксплуатации вследствие загрязнения и старения ламп и светильников, а также снижения отражающих свойств поверхностей помещения для кабинета будет равен 1,3;

Nc - число светильников в помещении;

- коэффициент затенения, обычно 1;

i - коэффициент использования светового потока.

Коэффициент использования светового потока, давший название методу расчета, определяется по индексу помещения i в зависимости от типа светильника и коэффициентов отражения света от потолка, стен и пола:

 (2)

где А, В - длина и ширина помещения в плане, м;

Нс - высота подвеса светильников над рабочей поверхностью, м.

лм

Согласно СП 52.13330.2016 величина полученного светового потока достаточно для данного помещения.

* 1. Электробезопасность на предприятии

При поступлении на работу сотрудник должен пройти инструктаж по технике безопасности. Работника знакомят с основными правилами по технике безопасности. Обычно инструктаж занимает около часа. За это время, работникам объясняют все основные моменты, к примеру, в случае поломки электронного устройства, звонить на номер дежурного электрика.

Инструктаж по технике безопасности при выполнении конкретной работы проводит руководитель по технике безопасности соответствующего производственного участка, показывая инструктируемому правильные безопасные приемы работы.

Повторный инструктаж проводится для рабочих независимо от их квалификации, стажа и опыта работы, не реже 3 раза в 6 месяцев.

При нарушении работающими правил и инструкций по технике безопасности, технологической и производственной дисциплины, а также в случаях изменения технологического процесса или вида работы проводят дополнительные инструктажи.

Работник на рабочем месте должен соблюдать общие меры электробезопасности. При этом запрещается:

* прикасаться к задней панели системного блока при включенном питании;
* допускать попадание влаги на поверхность устройств;
* производить самостоятельно вскрытие оборудования;
* пользоваться неисправными электроприборами и электропроводкой;
* ремонтировать электроприборы самостоятельно;
* наступать на переносимые электрические провода, лежащие на полу.

При работе с электроприборами и оргтехникой:

* автоматические выключатели и электрические предохранители должны быть всегда исправны.
* изоляция электропроводки, электроприборов, выключателей, штепсельных розеток, ламповых патронов и светильников, а также шнуров, с помощью которых включаются в электросеть электроприборы, должны быть в исправном состоянии.
* электроприборы необходимо хранить в сухом месте, избегать резких колебаний температуры, вибрации, сотрясений.

работодатель обязан обеспечить:

* содержание электроустановок в работоспособном состоянии и их эксплуатацию в соответствии с требованиями нормативно-технических документов;
* своевременное и качественное проведение технического обслуживания, планово-предупредительного ремонта, испытаний, модернизации и реконструкции электроустановок и электрооборудования;
* подбор электротехнического и электротехнологического персонала, периодические медицинские осмотры работников, проведение инструктажей по безопасности труда, пожарной безопасности;
* обучение и проверку знаний электротехнического и электротехнологического персонала;
* надежность работы и безопасность эксплуатации электроустановок;
* охрану труда электротехнического и электротехнологического персонала;
* охрану окружающей среды при эксплуатации электроустановок;
* учет, анализ и расследование нарушений в работе электроустановок, несчастных случаев, связанных с эксплуатацией электроустановок, и принятие мер по устранению причин их возникновения;
* представление сообщений в органы госэнергонадзора об авариях, смертельных, тяжелых и групповых несчастных случаях, связанных с эксплуатацией электроустановок;
* разработку должностных, производственных инструкций и инструкций по охране труда для электротехнического персонала;
* укомплектование электроустановок защитными средствами, средствами пожаротушения и инструментом;
* учет, рациональное расходование электрической энергии и проведение мероприятий по энергосбережению;
* проведение необходимых испытаний электрооборудования, эксплуатацию устройств молниезащиты, измерительных приборов и средств учета электрической энергии;
* выполнение предписаний органов государственного энергетического надзора.

Существует очень много видов опасностей при работе с электрическими приборами и электроустановками, поэтому нужно соблюдать все меры предосторожности и каждый работающий должен уметь оказывать первую доврачебную помощь.

* 1. Обеспечение пожарной безопасности

Инструктаж по пожарной безопасности проводится по программе, разработанной инженером по охране труда организации, с учетом требований стандартов, правил, норм и инструкций о мерах пожарной безопасности. Продолжительность инструктажа устанавливается в соответствии с утвержденной программой. Инструктаж по пожарной безопасности, как правило, проводится совместно с инструктажем по технике безопасности и в те же сроки.

Инструктаж по пожарной безопасности проходят все работники организации, независимо от их образования, стажа работы по данной профессии или должности, временные работники, командированные, обучающиеся и студенты, прибывшие на практику.

О проведении инструктажа по пожарной безопасности работник, проводивший инструктаж, делает запись в журнале учета проведения инструктажей по пожарной безопасности, с обязательной подписью инструктируемого и инструктирующего. При регистрации внепланового инструктажа по пожарной безопасности указывают причину его проведения.

Все работники организаций должны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении специфики работы проходить дополнительное обучение предупреждению и тушению возможных пожаров в порядке, установленном руководителем.

На проведение первичного противопожарного инструктажа необходимо отводить не менее 1 ч. Инструктируемые должны ознакомиться:

* с действующими на объекте правилами пожарной безопасности и инструкциями;
* с производственными участками, наиболее опасными в пожарном отношении, где запрещается курить, применять открытый огонь;
* с возможными причинами возникновения пожара и мерами его предупреждения;
* с практическими действиями в случае возникновения пожара - вызов пожарной помощи, использование первичных средств пожаротушения, место расположения ближайшего телефона и ознакомление с правилами поведения в случае возникновения пожара, эвакуации людей и материальных ценностей.

При первичном инструктаже инструктирующий обязан рассказать о производственных установках с повышенной пожарной опасностью, мерах предотвращения пожаров и загораний, указать место курения, ознакомить вновь поступившего с имеющимися на объекте средствами пожаротушения, показать ближайший телефон и объяснить правила поведения в случае возникновения пожара.

Проведение противопожарного инструктажа в обязательном порядке должно сопровождаться практическим показом способов использования имеющихся на объекте средств пожаротушения (противогазы, респираторы, огнетушители и т.д.).

Вывод:

Соблюдение рассмотренных в данном разделе мероприятий позволяет снизить утомляемость и травматизм, повысить производительность труда, обеспечить комфортные условия трудовой деятельности.

ЗАКЛЮЧЕНИЕ

В данной работе был разработан и реализован проект веб-приложения для учета прохождения практик, также рассмотрены методы автоматизации и оптимизации процесса.

При разработке были исследованы потенциальные проблемы и произведен анализ аналогичных существующих решений, было приведено обоснование предлагаемого решения.

В экономической части приведены расчеты экономического эффекта, который может быть достигнут при внедрении разработанного программного обеспечения.

В разделе «Техника безопасности и охрана труда» были обоснованы инженерно-технические мероприятия по обеспечению безопасности и охране труда при использовании разработанного программного продукта на вычислительной технике.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Самоучитель WPF [Электронный ресурс] / wpfbook.ru : Влад Мержевич. – Электрон. дан. – [Б. м. : б. и.], 2021. – Режим доступа: http://wpfbook.ru/samwpf5, свободный. – Загл. с экрана.
2. Документация по C# [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/,свободный – Загл. с экрана.
3. «Microsoft Learn» [Электронный ресурс]. – Режим доступа: https://learn.microsoft.com/ru-ru/sql, свободный. – Загл. с экрана.
4. Документация для построения диаграммы последовательности [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/32/32/lecture/1014>, свободный. – Загл. с экрана.
5. Общая информация о диаграмме состояния [Электронный ресурс]. – Режим доступа: <https://studopedia.ru/11_96547_ponyatie-o-diagrammah-sostoyaniya.html>, свободный. – Загл. с экрана.
6. Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М. : КНОРУС, 2018.- 488 с.
7. Дейт, К. Дж. Введение в системы баз данных – СПб.: Вильямс, 2006. – 1072 с.
8. Климов, А. C#. Советы программистам / А. Климов. - М.: БХВ-Петербург, 2018. - 544 c.
9. Секунов, Н. Самоучитель C# / Н. Секунов. - М.: БХВ-Петербург, 2017. - 576 c.
10. Карвин, Билл Программирование баз данных SQL. Типичные ошибки и их устранение / Билл Карвин. - М.: Рид Групп, 2018. - 336 c.
11. Маркин, А. В. Построение запросов и программирование на SQL. Учебное пособие / А.В. Маркин. - М.: Диалог-Мифи, 2014. - 384 c.
12. Фленов, Михаил Библия C# / Михаил Фленов. - М.: БХВ-Петербург, 2021. - 560 c.

ПРИЛОЖЕНИЕ А  
(справочное)  
Код программы

Код формы авторизации:

using System.Windows;

namespace Диплом\_пробник

{

public partial class Производство\_вход : Window

{

public Производство\_вход()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string loginQuery = $"SELECT id FROM Пользователи WHERE Телефон = '{loginBox.Text}' and Пароль = HASHBYTES('SHA2\_256', Convert(varchar(MAX), '{passwordBox.Text}'))";

string id = SqlQuery.GetRequest(loginQuery);

if (id != null)

{

Производство\_главная\_страница mainForm = new Производство\_главная\_страница(id);

mainForm.Show();

this.Hide();

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

Регистрация mainForm = new Регистрация();

mainForm.Show();

}

}

}

Код главной страницы:

using System;

using System.Data;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using Microsoft.Office.Interop.Excel;

namespace Диплом\_пробник

{

public partial class Производство\_главная\_страница : System.Windows.Window

{

Microsoft.Office.Interop.Excel.Application excel = null;

Workbook wb = null;

object missing = Type.Missing;

Worksheet ws = null;

private string \_id;

public Производство\_главная\_страница(string id)

{

InitializeComponent();

\_id = id;

FIOBox.Text = SqlQuery.GetRequest($"SELECT ФИО FROM Пользователи WHERE id = {\_id}");

}

private void StudentButtonClick(object sender, RoutedEventArgs e)

{

dataGridMain.ItemsSource = SqlQuery.GetTable("Select \* from [Студенты]").DefaultView;

}

private void TasksButtonClick(object sender, RoutedEventArgs e)

{

dataGridMain.ItemsSource = SqlQuery.GetTable("Select \* from [Задачи]").DefaultView;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

dataGridMain.ItemsSource = SqlQuery.GetTable("Select \* from [Руководители УЗ]").DefaultView;

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

Window1 mainForm = new Window1(\_id);

mainForm.Show();

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

dataGridMain.ItemsSource = SqlQuery.GetTable("Select \* from [Расписание практик]").DefaultView;

}

private static void SaveToExcel(System.Data.DataTable dataGridView1)

{

Microsoft.Office.Interop.Excel.Application excel = null;

Microsoft.Office.Interop.Excel.Workbook wb = null;

object missing = Type.Missing;

Microsoft.Office.Interop.Excel.Worksheet ws = null;

Microsoft.Office.Interop.Excel.Range rng = null;

// collection of DataGrid Items

var dtExcelDataTable = dataGridView1;

excel = new Microsoft.Office.Interop.Excel.Application();

wb = excel.Workbooks.Add();

ws = (Microsoft.Office.Interop.Excel.Worksheet)wb.ActiveSheet;

ws.Columns.AutoFit();

ws.Columns.EntireColumn.ColumnWidth = 25;

// Header row

for (int Idx = 0; Idx < dtExcelDataTable.Columns.Count; Idx++)

{

ws.Range["A1"].Offset[0, Idx].Value = dtExcelDataTable.Columns[Idx].ColumnName;

}

// Data Rows

for (int Idx = 0; Idx < dtExcelDataTable.Rows.Count; Idx++)

{

ws.Range["A2"].Offset[Idx].Resize[1, dtExcelDataTable.Columns.Count].Value = dtExcelDataTable.Rows[Idx].ItemArray;

}

excel.Visible = true;

wb.Activate();

}

private System.Data.DataTable ConvertDataGridToDataTable(DataGrid dataGrid)

{

System.Data.DataTable dataTable = new System.Data.DataTable();

// Create columns in DataTable

foreach (var column in dataGrid.Columns)

{

if (column is DataGridColumn dataGridColumn)

{

dataTable.Columns.Add(dataGridColumn.Header.ToString());

}

}

// Populate data rows

foreach (var item in dataGrid.Items)

{

DataRow dataRow = dataTable.NewRow();

foreach (var column in dataGrid.Columns)

{

if (column is DataGridColumn dataGridColumn)

{

// Get the binding property name

string bindingPropertyName = ((Binding)dataGridColumn.ClipboardContentBinding).Path.Path;

// Get the value of the corresponding property from the item

var cellValue = item.GetType().GetProperty(bindingPropertyName)?.GetValue(item, null);

// Set the value in the DataRow

dataRow[dataGridColumn.Header.ToString()] = cellValue;

}

}

// Add the DataRow to DataTable

dataTable.Rows.Add(dataRow);

}

return dataTable;

}

public void ExportDataTableToExcel(System.Data.DataTable dataTable, string filePath)

{

// Create a new Excel application and workbook

wb = excel.Workbooks.Add();

ws = wb.ActiveSheet;

// Set the column headers

for (int col = 0; col < dataTable.Columns.Count; col++)

{

ws.Cells[1, col + 1] = dataTable.Columns[col].ColumnName;

}

// Set the data rows

for (int row = 0; row < dataTable.Rows.Count; row++)

{

for (int col = 0; col < dataTable.Columns.Count; col++)

{

ws.Cells[row + 2, col + 1] = dataTable.Rows[row][col];

}

}

}

private void ReleaseObject(object obj)

{

try

{

//System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);

obj = null;

}

catch (Exception ex)

{

obj = null;

}

finally

{

GC.Collect();

}

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

ReleaseObject(wb);

ReleaseObject(wb);

ReleaseObject(excel);

System.Data.DataTable dataTable = ConvertDataGridToDataTable(dataGridMain);

SaveToExcel(dataTable);

}

}

}

Код личного профиля:

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

namespace Диплом\_пробник

{

/// <summary>

/// Логика взаимодействия для Window1.xaml

/// </summary>

public partial class Window1 : Window

{

public Window1(string id)

{

InitializeComponent();

nameBox.Text = SqlQuery.GetRequest($"SELECT ФИО FROM Пользователи WHERE id = {id}");

phoneBox.Text = SqlQuery.GetRequest($"SELECT Телефон FROM Пользователи WHERE id = {id}");

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

}

}

Код класса SqlQuery  
  
  
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;

namespace Диплом\_пробник

{

public static class SqlQuery

{

private static string \_connectionString = "Data Source = LAPTOP-V39BIFQ3\\SQLEXPRESS; Initial catalog = ППП; Integrated Security = True;";

public static SqlConnection \_sqlConnection = new SqlConnection(\_connectionString);

public static void DoRequest(string request)

{

\_sqlConnection.Open();

SqlCommand cmd = new SqlCommand(request, \_sqlConnection);

cmd.ExecuteNonQuery();

\_sqlConnection.Close();

}

public static string GetRequest(string request)

{

\_sqlConnection.Open();

SqlCommand cmd = new SqlCommand(request, \_sqlConnection);

object result = cmd.ExecuteScalar();

\_sqlConnection.Close();

if (result != null)

return result.ToString();

else

return null;

}

public static DataTable GetTable(string request)

{

\_sqlConnection.Open();

DataTable dt = new DataTable();

SqlDataAdapter adapter = new SqlDataAdapter(request, \_sqlConnection);

adapter.Fill(dt);

\_sqlConnection.Close();

return dt;

}

public static Dictionary<string, string> GetComboBox(string request)

{

Dictionary<string, string> elements = new Dictionary<string, string>();

\_sqlConnection.Open();

SqlCommand cmd = new SqlCommand(request, \_sqlConnection);

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

elements.Add(reader.GetInt32(0).ToString(), reader.GetString(1));

}

\_sqlConnection.Close();

return elements;

}

}

}