**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

**Государственное бюджетное профессиональное образовательное   
учреждение города Москвы**

**«Колледж автоматизации и информационных технологий № 20»**

**ДОПУЩЕН(А) К ЗАЩИТЕ**

Зам. директора по ККО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ткаченко В.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Тема: «Разработка информационного ресурса для обучения и дистанционного тестирования сотрудников»**

Выпускная квалификационная работа выполнена в форме:

**дипломного проекта**

обучающимся группы ИСП-411 Кузнецовым Матвеем Сергеевичем

(Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (дата)

по специальности

09.02.07 Информационные системы и программирование

(код и наименование специальности)

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_ Кузьмина О.Б. «\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

(подпись) (И.О. Фамилия)

Консультант по

экономической части \_\_\_\_\_\_\_\_\_\_\_\_\_ Носаченко Е.А. «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2022 г.

(подпись) (И.О. Фамилия)

**Согласовано**

Заведующий УО «Юниор»

\_\_\_\_\_\_\_\_\_\_\_ Шеренговский М.Ф.

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

2023 г.

**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**

**Государственное бюджетное профессиональное образовательное   
учреждение города Москвы**

**«Колледж автоматизации и информационных технологий № 20»**

|  |  |
| --- | --- |
| РАССМОТРЕНО и ОДОБРЕНО  НА ЗАСЕДАНИИ СЦН ПО НАПРАВЛЕНИЮ «ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»  Протокол №\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Скиллсменеджер СЦН \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Б. Кузьмина | УТВЕРЖДАЮ  Заведующий отделением  \_\_\_\_\_\_\_\_\_\_\_ С.В. Лукьянов  «\_\_\_» \_\_\_\_\_\_ 2023 г. |
|  |  |

**ЗАДАНИЕ**

НА ДИПЛОМНЫЙ ПРОЕКТ ОБУЧАЮЩЕМУСЯ (ЙСЯ)

Кузнецову Матвею Сергеевичу

(фамилия, имя, отчество)

|  |  |
| --- | --- |
| 1. Тема | Разработка информационного ресурса для обучения и дистанционного тестирования сотрудников |

Тема утверждена приказом по колледжу 28.11.2022 г. № 193-3/лу.

1. Срок сдачи обучающимся законченного дипломного проекта: \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.
2. Исходные данные к дипломному проекту: Техническое задание на разработку программного продукта, тестовые наборы данных для базы данных, тестовые наборы данных проектной документации заказов, принципы стилистического оформления пользовательского интерфейса.
3. Перечень вопросов, подлежащих разработке в дипломном проекте:
   1. Провести анализ предметной области;
   2. Определить основные требования к функциональности приложения;
   3. Ознакомиться с аналогами разрабатываемого программного продукта;
   4. Выполнить проектирование интерфейса и дизайна приложения, путём создания прототипа;
   5. Разработать проектную документацию;
   6. Разработать базу данных на основе проектной документации и имеющихся данных;
   7. Реализовать поэтапную разработку настольного приложения;
   8. Выполнить тестирование и отладку приложения;
   9. Разместить исходный код настольного приложения в системе контроля версий;
   10. Произвести расчет себестоимости программного проекта;
   11. Произвести расчет стоимости программного продукта;
   12. Произвести оценку экономической эффективности внедрения проекта.
4. Перечень иллюстрированных материалов (графические материалы с точным указанием обязательных чертежей или раздаточных материалов, с указанием наименования таблиц, рисунков или схем).

Use-case диаграмма, ERD диаграмма

Консультанты:

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел работы содержание определяется СЦН по направлению** | **Консультант по разделу (ФИО)** | **Дата** | **Подпись** |
| Раздел: Экономическая часть | Носаченко Елена Александровна |  |  |

|  |  |
| --- | --- |
| Руководитель дипломного проекта  \_\_\_\_\_\_\_\_\_\_\_ / Кузьмина О.Б. /  подпись Ф.И.О  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ | Задание принял (а) к исполнению обучающийся (щаяся)    \_\_\_\_\_\_\_\_\_\_\_ / Кузнецов М.С. /  подпись Ф.И.О  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ |

Оглавление

[Введение 6](#_Toc129605895)

[1. Теоретическая часть 8](#_Toc129605896)

[1.1. Создание образовательного ПО и его история: от механических машин до первых компьютеров 8](#_Toc129605897)

[1.2. Компьютерные учебные программы 9](#_Toc129605898)

[1.3. Инструменты для разработки настольных приложений. 12](#_Toc129605899)

[1.4. Платформа для разработки настольных приложений 13](#_Toc129605900)

[1.5. Среда разработки 15](#_Toc129605901)

[1.6. Обоснования необходимости разработки внутреннего ПО для тестирования сотрудников компании 16](#_Toc129605902)

[1.7. Требования к надежности 17](#_Toc129605903)

[1.8. Требования к функциональным характеристикам 18](#_Toc129605904)

[1.9. Требования к целостности данных 19](#_Toc129605905)

[ГЛАВА 3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ 21](#_Toc129605906)

[1.10. Цель и задачи экономического раздела 21](#_Toc129605907)

[Глава 4. Авторское право 22](#_Toc129605908)

[Заключение 23](#_Toc129605909)

[Список литературы 24](#_Toc129605910)

# **Введение**

С давних времен человечество постоянно обучалось. Изучение нового является основным двигателем прогресса, поэтому способы получения знаний менялись и формировались на протяжении тысячелетий.

Обучение-это процесс приобретения нового понимания, знаний, поведения, навыков, ценностей, установок и предпочтений.

В процессе постоянного накопления знаний человеку стало сложно передавать их в устной форме, поэтому были созданы различные способы хранения информации. Основным из таких способов является письменность. Человечество записывало те знания, что необходимо было передать другим или же обратиться к ним впоследствии.

Совершенствование, структурирование и оптимизация технологий письменного хранения данных позволяло выводить самообучение на новый уровень. В следствии чего количество потребляемой информации серьезно увеличилось, поэтому необходимо было автоматизировать данный процесс. Автоматизация стала одним из основных терминов в сфере IT технологий. Электронные системы обучения набирали популярность и увеличивали процесс изучения информации в несколько раз.

В наше время жизнь каждого человека строится на ежедневном поглощении информации из различных источников, поэтому актуальность автоматизированного обучения находится на очень высоком уровне. При приеме на работу очень неудобно лично обучать каждого сотрудника всем тонкостям работы в компании, поэтому куда эффективнее использовать готовые информационные ресурсы, после изучения которых сотрудник получает необходимые для продуктивной работы знания и навыки.

Основной целью данной дипломной работы является создание электронной системы обучения с системой дистанционного тестирования сотрудников. Она позволит организовывать внедрение новых специалистов в работу компании на высоком уровне. Отличительная черта данной системы является её полная автономность.

Основные задачи дипломной работы:

1. Изучить требования заказчика для их последующей реализации в программе;
2. Составить план действий по реализации проекта;
3. Схематично визуализировать вид приложения;
4. Реализовать запланированные возможности продукта;
5. Протестировать рабочие функции приложения.

# **1. Теоретическая часть**

## Создание образовательного ПО и его история: от механических машин до первых компьютеров

Сегодняшнее образовательное программное обеспечение (ПО) представляет собой комплекс приложений, которые предназначены для развития определенных навыков у студентов. Однако, история образования насчитывает более ста лет, и первые образовательные механические устройства появились в конце XIX века. Ранее, перед появлением механических устройств, главными источниками знаний были учителя и книги, но обучение было очень трудоемким и неэффективным.

Идея заменить преподавателей механическими учебными машинами появилась в связи с успехами промышленной революции, и была направлена на ускорение процесса обучения и повышение эффективности. Хотя идея заменить учителей механическими устройствами сегодня кажется наивной, она стала основой для развития современных технологий.

Первым образовательным механическим устройством для изучения грамматики стало устройство, разработанное Алкионом Скиннером и запатентованное им в 1866 году. В 1911 году Герберт Остин Эйкинс создал механическое устройство для обучения арифметике, чтению и правописанию. А в 1912 году Эдвард Ли Торндайк предложил новый подход к образованию, основанный на механической книге, где следующие разделы открываются только после того, как предыдущие были усвоены.

Сегодняшние технологии образования, такие как онлайн-курсы, электронные учебники, и программное обеспечение, являются продуктом продолжительного и эволюционного процесса, начавшегося с появления первых образовательных механических устройств в конце XIX века. Несмотря на прогресс, достигнутый в современных технологиях, эти первоначальные идеи остаются фундаментальными для развития современных методов обучения.

В 1924 году профессор Сидни Пресси из Огайо разработал систему обучения под названием Automatic Teacher, которая позволяла ученику выбирать правильный ответ из предложенных вариантов на барабане машины и отмечала количество верных ответов на счетчике, но не могла обучать, а только проверять знания путем прохождения тестов. Многие изобретатели начали создавать новые обучающие машины, объединяя опыт XIX века, идеи Торндайка и технологии нового времени. В 1954 году профессор Беррес Фредерик Скиннер сформулировал теорию программированного обучения, в которой главным компонентом обучающего устройства должна быть строгая программа с элементами для изучения и проверки материала, а сам процесс обучения является ступенчатым.

Скиннер представил "обучающую машину", которая печатала вопросы на бумажных карточках и демонстрировала их покадрово в специальном окошке, а ученик набирал ответ на клавиатуре устройства. Система Скиннера не только тестировала знания, но и обучала студентов, и была первым устройством, в котором удалось сочетать результаты теоретических изысканий в педагогической психологии с технологическими новшествами того времени. В 1960 году инженер Дональд Битцер создал компьютерную систему PLATO (Programmed Logic for Automated Teaching Operations), которая основывалась на теории программированного обучения и предоставляла студентам возможность изучения нескольких тематических курсов. В настоящее время система не используется, но компания PLATO Learning (ныне Edmentum) разрабатывает учебные курсы. В 60-е годы появилась критика новых технологий обучения, связанная с недостаточной методической и технической подготовкой учителей, дороговизной оборудования и возможной дегуманизацией образования. Однако дальнейшее развитие показало, что опасения были напрасными.

## Компьютерные учебные программы

КУП, также известные как компьютерные учебные программы, представляют собой программы, которые используются в учебном процессе и взаимодействуют с учащимся. Они включают в себя различные типы, такие как обучающие, тестирующие, тренажерные, моделирующие, учебные игры, прикладные и интегрированные программы. Обучающие программы, такие как автоматизированные учебные курсы (АУК), компьютерные учебники, программы искусственного интеллекта, компьютерные справочники и энциклопедии, могут использоваться для представления, проверки и оценки усвоения учебного материала.

АУК могут содержать вступительную часть, учебный материал и вопросы для проверки понимания. Они также могут предлагать предварительные тесты для определения уровня подготовки и дополнительные вопросы, помощь или правильные ответы в ответ на ответы учащихся. Обучение может осуществляться с помощью различных форматов, таких как текст, графика, мультимедиа и аудио- и видеоинформация.

Компьютерные учебники предназначены для самостоятельного изучения теоретического материала и могут быть построены на гипертекстовой основе. Они содержат структурированный учебный материал, представленный в виде интерактивных кадров с мультимедийными приложениями. Электронный учебник позволяет учащимся выбирать оптимальную траекторию изучения материала, регулировать темп и способ изложения в соответствии с их психофизиологическими особенностями восприятия.

Кроме того, в электронных учебниках может быть предусмотрена возможность протоколирования действий учащихся для анализа преподавателем. В настоящее время компьютерные учебники считаются наиболее развитым средством обучения.

КУП-ы, такие как компьютерные справочники и энциклопедии, предназначены исключительно для представления учебного материала и часто содержат большие объемы информации, поддерживаемые технологиями мультимедиа и гипертекста. Каждый КОП должен иметь справочную систему, которая может быть встроена в электронный учебник или являться самостоятельным элементом.

Искусственный интеллект включает два основных типа программ: программы, моделирующие учащегося, и экспертные системы. Разработчики обучающих программ ИИ считают важным, чтобы программа могла строить модель обучаемого, учитывая его способности и оптимальные режимы усвоения материала. После сравнения с образцовой моделью вырабатываются стратегии, методы и последовательность обучения.

Учебные программы искусственного интеллекта подразделяются на несколько типов, включая эвристические программы, реагирующие учебные среды, программы-инструкторы, программы-помощники и тренажерные программы. Эвристические программы используют методы обучения, которые дают общее направление решения задач. Реагирующая учебная среда позволяет исправлять ошибки и учить навыкам мышления, основанным на тьюторских правилах. Программа-инструктор оценивает действия обучаемого и моделирует его умения и навыки. Программа-помощник направляет процесс обучения, предлагая знания и помогая усвоить понятия. Тренажерные программы используются для практического закрепления учебного материала и тренировки операторов в различных технических системах. Компьютерные задачники могут также рассматриваться как тренажеры, которые позволяют отработать приемы решения типовых задач.

Компьютерное моделирование используется для имитации реальных объектов и процессов с целью изучения их свойств. Этот метод особенно полезен для моделирования сложных, опасных или дорогостоящих процессов, например, ядерных реакторов или превращения куколки в бабочку. Хотя компьютерные модели упрощают реальность, они позволяют обучаемым изучать объекты и процессы, управлять ими и понимать их особенности. Компьютерные модели не являются универсальными и предназначены для моделирования конкретных явлений. Они могут использоваться для демонстрации трудно воспроизводимых в учебной обстановке явлений и выяснения влияния параметров на изучаемые процессы и явления.

Учебные игры являются похожими на имитационные программы и помогают учащимся усваивать знания и навыки. Они могут быть разных типов, включая приключенческие, настольные, азартные, боевые и психомоторные игры. С развитием компьютеров в обучении, учебные заведения начали использовать прикладные программы общего назначения, называемые инструментальными средствами учащихся. Компьютеры могут использоваться для проведения расчетов, ввода данных, обучения основам управления роботами и аппаратами, хранения и поиска информации. Примерами прикладных программ, используемых в учебных заведениях, являются редакторы, базы данных и файловые системы, электронные динамические системы.

Редакторы используются для управления курсором на ранних стадиях обучения, редактирования документов, хранения данных, обучения последовательному изложению мыслей и составления различных текстовых документов. Базы данных и файловые системы используются для обучения пользованию готовыми файловыми системами, создания своих файлов, разработки, создания и использования файлов данных.

Интегрированные системы представляют собой программные комплексы, содержащие несколько типов компьютерных учебных программ, например, компьютерные учебники, экзаменаторы, тренажеры, программы моделирования и справочники.

## Инструменты для разработки настольных приложений.

C# (читается как "си шарп") - это язык программирования, разработанный компанией Microsoft в 2000 году. Он основан на языке C++, но содержит множество усовершенствований и расширений.

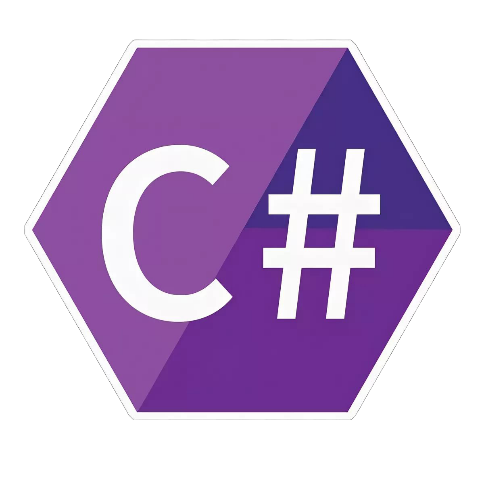


Рисунок 1 – Логотип языка программирования C#.

Преимущества языка C#:

Простота и удобство использования: C# имеет четкую и простую синтаксическую структуру, которая делает его более доступным для начинающих программистов. Большинство разработчиков на C# отмечают, что этот язык более удобен в использовании, чем C++.

Платформонезависимость: C# разрабатывался для использования на платформе .NET Framework, что позволяет создавать приложения, которые будут работать на любой платформе, поддерживающей эту технологию.

Безопасность: C# обладает встроенными средствами защиты от нежелательных действий, таких как переполнение буфера и другие уязвимости, что делает его более безопасным в использовании.

Обширная библиотека классов: .NET Framework включает обширную библиотеку классов, которая содержит множество готовых решений для программистов.

Недостатки языка C#:

Зависимость от платформы: Приложения, написанные на C#, требуют наличия установленного .NET Framework на компьютере пользователя. Это может создавать проблемы совместимости и использования на старых версиях операционных систем.

Производительность: Хотя C# проще и удобнее в использовании, чем C++, он может быть медленнее в работе в некоторых случаях, из-за того, что является языком со сборкой мусора и имеет другие особенности.

Ограничения: C# не является языком системного программирования и не подходит для написания низкоуровневых приложений.

Почему стоит выбрать C# для разработки настольных приложений?

Удобство использования: C# предоставляет более простой и понятный синтаксис, чем C++, что упрощает разработку приложений.

## Платформа для разработки настольных приложений

Существует множество платформ для разработки настольных приложений, и каждая из них имеет свои преимущества и недостатки. Одной из наиболее популярных и распространенных платформ для разработки настольных приложений является Windows Presentation Foundation (WPF).

WPF - это технология, которая была разработана компанией Microsoft и входит в состав .NET Framework. Она позволяет создавать приложения с привлекательным пользовательским интерфейсом, который поддерживает различные элементы управления, анимацию, графику и мультимедиа. Вот несколько преимуществ WPF:

Графический интерфейс. Один из основных преимуществ WPF - это его графический интерфейс, который позволяет создавать красивые и современные пользовательские интерфейсы. WPF предлагает множество элементов управления, которые легко настраиваются и поддерживают различные стили.

XAML. WPF использует язык разметки XAML (eXtensible Application Markup Language), который позволяет разработчикам создавать графические элементы и располагать их на форме без необходимости программирования на языке C#.

Анимация. WPF предоставляет мощный инструментарий для создания анимации и визуальных эффектов, которые могут быть использованы для улучшения пользовательского интерфейса.

Интеграция с другими технологиями Microsoft. WPF может быть интегрирован с другими технологиями Microsoft, такими как LINQ to SQL, Windows Communication Foundation (WCF) и Windows Workflow Foundation (WF).

Несмотря на все преимущества, у WPF есть и недостатки:

Сложность. WPF является сложной технологией, и ее изучение может занять много времени. Кроме того, при разработке на WPF может возникнуть необходимость в использовании других технологий Microsoft, что также может быть сложно для новичков.

Ограничения. WPF может быть ограничен в использовании на определенных платформах, таких как Linux и macOS.

Необходимость установки .NET Framework. Для запуска приложений, созданных на WPF, необходимо установить .NET Framework на компьютере пользователя.

В целом, WPF - это мощная и популярная платформа для разработки настольных приложений с красивым графическим интерфейсом.



Рисунок 2 – Логотип платформы для разработки WPF.

## Среда разработки

C# - это один из языков программирования, который часто используется для создания настольных приложений. Для разработки приложений на C# для платформы Windows Presentation Foundation (WPF) используются различные среды разработки (IDE), такие как Visual Studio и JetBrains Rider.

Visual Studio - это одна из самых популярных и мощных сред разработки для C#. Она предлагает множество функций для создания и отладки приложений, включая редактор кода, инструменты для проектирования пользовательского интерфейса и многие другие.

Для создания настольных приложений на C# с использованием WPF в Visual Studio требуется установить пакеты разработки .NET и средства разработки WPF. После установки пакетов и настройки среды разработки можно начать создавать приложения на C# с использованием WPF.

JetBrains Rider - это другая популярная среда разработки, которая поддерживает язык C# и WPF. JetBrains Rider предлагает набор инструментов для разработки приложений, включая редактор кода, инструменты отладки, анализ кода и многие другие.

Кроме того, существуют и другие среды разработки для C#, такие как SharpDevelop и Xamarin Studio. Они также поддерживают создание приложений на WPF, но они менее популярны, чем Visual Studio и JetBrains Rider.

В целом, выбор среды разработки для создания настольных приложений на C# с использованием WPF зависит от индивидуальных потребностей разработчика и предпочтений в использовании конкретных инструментов.

## Обоснования необходимости разработки внутреннего ПО для тестирования сотрудников компании

Разработка нового программного обеспечения для тестирования сотрудников и прохождения курсов является необходимой задачей для многих компаний. Существует несколько причин, почему компании могут решить разработать свою собственную систему тестирования и обучения сотрудников.

* Нестандартные требования - каждая компания имеет свои уникальные требования к системе тестирования и обучения сотрудников, которые могут отличаться от стандартных решений, доступных на рынке. Например, компания может иметь специфические бизнес-процессы или требования к обучению сотрудников, которые не учитываются в стандартных системах.
* Поддержка бренда - компания может заинтересована в создании своего бренда и идентичности для обучения сотрудников. Разработка собственной системы тестирования и обучения может помочь компании усилить свой бренд и создать единый стиль для обучения сотрудников.
* Контроль данных - компания может хотеть иметь полный контроль над данными своих сотрудников. Разработка собственной системы тестирования и обучения позволяет компании контролировать доступ к данным и обеспечить их безопасность.
* Гибкость - разработка собственной системы тестирования и обучения позволяет компании создавать гибкие и настраиваемые решения, которые отвечают конкретным потребностям компании.
* Экономия - разработка собственной системы тестирования и обучения может оказаться более экономически выгодным решением, чем покупка готового решения. Кроме того, компания может сохранить деньги, не платя за лицензии и обновления сторонних продуктов.
* Интеграция - разработка собственной системы тестирования и обучения позволяет компании интегрировать ее с другими системами, используемыми в компании, например, системами управления персоналом, CRM-системами и т.д.
* Адаптация к изменениям - разработка собственной системы тестирования и обучения позволяет компании быстро реагировать на изменения внутри компании и изменения в бизнес-среде.

## Требования к надежности

Надежность системы для тестирования сотрудников и прохождения курсов является одним из важнейших факторов, которые необходимо учитывать при ее разработке. Низкая надежность системы может привести к потере данных, сбоям в работе приложения или даже к разрушению всей системы. Поэтому необходимо учитывать следующие требования к надежности системы.

* Безопасность данных - система должна обеспечивать защиту данных, хранящихся в ней. Это включает в себя защиту от несанкционированного доступа, взломов и кражи данных. Для этого необходимо использовать механизмы шифрования, контроль доступа и многоуровневую аутентификацию.
* Резервное копирование данных - система должна регулярно создавать резервные копии данных, чтобы в случае сбоя или потери данных можно было бы их восстановить.
* Масштабируемость - система должна быть способна обрабатывать большое количество пользователей и запросов, особенно в периоды пиковой нагрузки. Для этого необходимо использовать масштабируемые архитектурные решения, такие как облачные сервисы.
* Отказоустойчивость - система должна быть способна обеспечивать работоспособность даже при сбоях в работе отдельных компонентов. Для этого необходимо использовать механизмы резервирования, балансировки нагрузки и автоматического восстановления после сбоев.
* Стабильность - система должна быть стабильной и надежной в работе, чтобы пользователи могли использовать ее без сбоев и ошибок. Для этого необходимо проводить тщательное тестирование системы перед ее запуском и регулярно проводить обновления и исправления ошибок.
* Доступность - система должна быть доступна для пользователей в любое время суток, поскольку многие пользователи могут использовать ее в свободное время или в ночное время. Для этого необходимо обеспечивать высокую доступность системы и минимальное время простоя в случае сбоев или обновлений.
* Удобство использования - система должна быть простой в использовании и интуитивно понятной для пользователей.

## Требования к функциональным характеристикам

Функциональные характеристики системы для тестирования сотрудников и прохождения курсов играют ключевую роль в обеспечении эффективности процесса обучения и оценки знаний сотрудников. Для того, чтобы система была полезной и удобной для пользователей, необходимо учитывать следующие требования к ее функциональным характеристикам.

* Управление курсами и контентом - система должна обеспечивать удобное управление курсами и контентом. Администраторы должны иметь возможность создавать и редактировать курсы, добавлять контент, обновлять их и определять доступ к ним.
* Разнообразие контента - система должна поддерживать разнообразный контент, включая видео, аудио, документы, изображения и интерактивные задания. Это позволит сотрудникам получать знания различными способами и в соответствии со своими потребностями.
* Оценка знаний - система должна иметь механизмы для оценки знаний сотрудников, такие как тесты и задания. Администраторы должны иметь возможность создавать и редактировать тесты, а также просматривать результаты прохождения тестов.
* Система обратной связи - система должна иметь механизмы для обеспечения обратной связи сотрудникам, чтобы они могли получать поддержку и помощь при необходимости. Это может быть реализовано через систему онлайн-чата или электронную почту.
* Мониторинг процесса обучения - система должна обеспечивать мониторинг процесса обучения, чтобы администраторы могли следить за прогрессом сотрудников и оценивать их успехи. Для этого необходимо иметь возможность просмотра статистики и отчетов о прохождении курсов и тестов.
* Гибкость и настраиваемость - система должна быть гибкой и настраиваемой, чтобы администраторы могли адаптировать ее под свои потребности. Например, система должна позволять настраивать права доступа и уровень сложности заданий.

## Требования к целостности данных

Целостность данных является одним из самых важных аспектов для системы тестирования сотрудников и прохождения курсов. Целостность данных обеспечивает правильность и точность информации, которую используют сотрудники для обучения и оценки своих знаний. Вот несколько требований к целостности данных, которые должны быть учтены при разработке системы тестирования сотрудников и прохождения курсов.

* Надежность системы хранения данных - система должна иметь надежную систему хранения данных, которая обеспечивает сохранность информации и защиту от несанкционированного доступа. Для этого необходимо использовать современные методы шифрования, резервное копирование данных и автоматическую проверку целостности данных.
* Защита данных от несанкционированного доступа - система должна иметь механизмы для защиты данных от несанкционированного доступа. Это может быть достигнуто через использование паролей, аутентификации по IP-адресу, двухфакторной аутентификации и других методов.
* Ограничение прав доступа - система должна иметь возможность ограничивать права доступа к данным в соответствии с ролями пользователей. Например, администраторы должны иметь больше прав, чем обычные пользователи.
* Ограничение возможности изменения данных - система должна иметь механизмы для ограничения возможности изменения данных пользователем. Например, администраторы должны иметь возможность редактировать курсы и тесты, но не результаты тестирования.
* Мониторинг целостности данных - система должна иметь механизмы для мониторинга целостности данных. Например, система должна предупреждать об изменениях данных и отправлять уведомления администраторам.
* Резервное копирование данных - система должна иметь механизмы для резервного копирования данных, чтобы обеспечить сохранность информации в случае сбоя системы или несанкционированного доступа.

Аудит доступа к данным - система должна иметь возможность аудита доступа к данным, чтобы администраторы могли просматривать журналы доступа и проверять, кто и когда получал доступ к данным.

# **ГЛАВА 3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ**

## Цель и задачи экономического раздела

В настоящем разделе выпускной квалификационной работы производится расчет затрат на разработку …

# **Глава 4. Авторское право**

# **Заключение**

# **Список литературы**

1. Виталий Трунин: Путь программиста T-SQL. Самоучитель по языку Transact-SQL, 2020, 204 стр.
2. Фримен А.: Entity Framework Core 2 для ASP.NET Core MVC для профессионалов, 2019, 626 стр.
3. Казанский А. А*.*: Программирование на Visual C#: учебное пособие для среднего профессионального образования, 2021, 192 стр.
4. Джепикс Филипп, Троелсен Эндрю: Язык программирования C# 7 и платформы .NET и .NET Core, 2018, 1328 стр.
5. Скит Джон: C# для профессионалов. Тонкости программирования, 2019, 608 стр.
6. Чаллавала Шаббир: MySQL 8 для больших данных, 2018 226 стр.
7. Уорд Брайан: Инновации SQL Server 2019. Использование технологий больших данных и машинного обучения, 2020, 408 стр.
8. https://metanit.com - Информационный ресурс «Метанит»
9. https://ru.wikipedia.org - Информационный ресурс «Википедия»
10. https://visualstudio.microsoft.com/ru/ - Интернет - магазин и разработок «Microsoft»
11. https://studopedia.ru/ - Учебные пособия и материалы для студентов «Студопедия»
12. https://www.jetbrains.com - Программные решения компании JetBrains
13. https://spravochnick.ru/programmirovanie/yazyki\_programmirovaniya/yazyk\_programmirovaniya\_basic/ - Информационный ресурс для программистов «Справочник»
14. <https://habr.com/ru/company/spbifmo/blog/476160/> - Рождение образовательного ПО и его история: от механических машин до первых компьютеров
15. <https://studfile.net/preview/8828003/page:10/> - Компьютерные учебные программы