**СОДЕРЖАНИЕ**

[Введение 6](#_Toc135474462)

[1 Аналитическая часть 7](#_Toc135474463)

[1.1 Описание предметной области 7](#_Toc135474464)

[1.2 Анализ существующих разработок и обоснование необходимости 8](#_Toc135474465)

[1.3 Постановка задачи 8](#_Toc135474466)

[1.4 Формализация и документирование требований к программному средству 9](#_Toc135474467)

[2 Проектирование программного средства 10](#_Toc135474468)

[2.1 Разработка модели данных 11](#_Toc135474469)

[2.1.1 Входная и выходная информация 12](#_Toc135474470)

[2.1.2 Функциональная модель 13](#_Toc135474471)

[2.1.3 Структура данных 14](#_Toc135474472)

[2.2 Иерархическая структура и ее характеристика 15](#_Toc135474473)

[2.3 Проектирование пользовательского интерфейса 16](#_Toc135474474)

[2.4 Выбор программных и аппаратных средств 19](#_Toc135474475)

[3 Разработка программного средства 22](#_Toc135474476)

[3.1 Описание основных алгоритмов 22](#_Toc135474477)

[3.2 Реализация взаимосвязи компонентов программного средства 23](#_Toc135474478)

[3.3 Защита информации 23](#_Toc135474479)

[3.4 Тестирование программного средства 25](#_Toc135474480)

[3.5 Разработка документации к программному средству 26](#_Toc135474481)

[4 Охрана труда, техника безопасности, противопожарные мероприятия и охрана окружающей среды 28](#_Toc135474482)

[4.1 Организация рабочего места для работающих на компьютере 28](#_Toc135474483)

[4.2 Производительность труда и рабочее место 30](#_Toc135474484)

[4.3 Зоны досягаемости рук в горизонтальной плоскости 31](#_Toc135474485)

[4.4 Требования безопасности при работе на персональных электронно-вычислительных машинах 32](#_Toc135474489)

[4.5 Гигиенические требования к правильной посадке при работе на компьютере 33](#_Toc135474497)

[5 Экономическая часть 35](#_Toc135474498)

[5.1 Характеристика проекта 35](#_Toc135474499)

[5.2 Определение трудоемкости создания программного продукта 36](#_Toc135474500)

[5.3 Определение полной себестоимости и отпускной цены программного продукта 37](#_Toc135474501)

[5.4 Оценка экономического  эффекта  от  внедрения программного продукта 42](#_Toc135474502)

[5.5 Выводы 43](#_Toc135474503)

[Заключение 44](#_Toc135474504)

[Список использованных источников 45](#_Toc135474505)

**Графическая часть**

Приложение А Техническое задание

Приложение Б Тестовые сценарии

Приложение В Руководство пользователя

Приложение Г Аннотация к программному средству

Приложение Д Листинг кода

Лист 1 Схема данных

Лист 2 Дерево сайта

# **ВВЕДЕНИЕ**

Минск – один из немногих городов в Беларуси, где есть метрополитен. Минское метро, с длиной линий более 36 километров и 31 станцией, крупнейшее в Беларуси, и ежедневно перевозит более 700 тысяч пассажиров. Со времен первого открытия в 1984 году метрополитен постоянно развивается, расширяется и модернизируется.

Общественный интерес в развитии минского метрополитена растет, и поэтому имеет большое значение разработка Интернет-ресурса, которое удобно и доступно обеспечивает доступ к информации о минском метрополитене и его истории.

В настоящее время виртуальные музеи становятся все более актуальными в свете быстрого развития информационных технологий и увеличения интереса к истории и культуре. Виртуальные музеи позволяют посетителям не только легко ознакомиться с экспонатами и историей, но и наслаждаться визуальным и звуковым оформлением, которое удобно представлено в Интернете.

Инновационный и уникальный подход для создания виртуального музея минского метрополитена, предлагаемый в данном дипломном проекте, не только обеспечивает комфорт и простоту использования приложения, но также стабильную работу и реализацию всех требуемых функций.

Большую роль виртуальные музеи и площадки для ознакомления с информацией о достопримечательностях играют в образовании, открывая доступ к объектам культурного наследия страны и отдельных ее регионов, они способствуют осуществлению образовательной и просветительской функции. Потенциал виртуальных музеев может быть применен как инструмент самообразования, так и в качестве образовательного материала в рамках учебной программы.

Целью дипломного проекта является разработка веб-приложения «Виртуальный музей минского метрополитена», предназначенное для ознакомления пользователя с историческими датами, событиями, фактами, мероприятиями, архивными фотографиями минского метрополитена.

Задачи дипломного проекта:

Пояснительная записка содержит разделы:

* «Аналитическая часть»: описание предметной области, анализ существующих разработок и обоснование необходимости, постановка задачи, формализация и документирование требований к программному средству.
* «Проектирование программного средства»: разработка модели данных, входная и выходная информация, функциональная модель, структура данных, иерархическая структура и ее характеристика, проектирование пользовательского интерфейса, выбор программных и аппаратных средств.
* «Разработка программного средства»: описание основных алгоритмов, методов и приемов разработки программных модулей, реализация взаимосвязи компонентов программного средства, защита информации, тестирование программного средства, разработка документации к программному средству.
* «Охрана труда, техника безопасности, противопожарные мероприятия и охрана окружающей среды»
* «Экономическая часть» должна включать в себя расчет материальных затрат, затраты на оплату труда и расчет себестоимости разрабатываемого программного продукта.

Последним пунктом пояснительной записки должно являться заключение, в котором должны содержатся выводы о разработке программного продукта, а также об экономической эффективности его эксплуатации.

1. **Аналитическая часть**
   1. **Описание предметной области**

Подробно изучив информацию, относящуюся к теме дипломного проекта, было принято решение реализовать удобное для использования средство, доступное пользователям сети интернет для поиска информации о различных достопримечательностях и культурно-исторических объектах в любое удобное время.

Для того, чтобы обеспечить возможность посещения ресурса с различных устройств была выбрана разработка веб-ресурса. В зависимости от технологий и структуры, использованных при создании веб-приложения, можно выделить 2 типа страниц:

1. статические страницы чаще всего разрабатываются средствами языка разметки гипертекста HTML (HyperText Markup Language), и частое изменение информации, размещенной на них, не предусматривается. На действия пользователя страница не будет реагировать и изменять свой контент.
2. динамические страницы создаются средствами HTML/CSS в сочетании с языками программирования такими как PHP и JavaScript, которые имеют возможность динамического изменения контента, размещенного на страницах ресурса.

Разрабатываемое веб-приложение относится к динамически изменяемой странице. Но, современные веб-приложения имеют несколько видов, которые необходимо изучить перед началом разработки. Веб-приложения разделяются на:

1. MPA (Multi Page Application) – это приложения, на котором основные ссылки открываются в разных окнах браузера. Это означает, что при каждом изменении данных или загрузке новой информации страница обновляется или открывает новую вкладку. Такие приложения более ресурсоемки, чем одностраничные, поэтому их использование целесообразно только в тех случаях, когда нужно отобразить объемный контент.
2. SPA (Single Page Application) – это одностраничное веб-приложение, которое загружается на одну HTML-страницу. Благодаря динамическому обновлению с помощью JavaScript, во время использования не требуется перезагрузка. На практике это означает, что пользователю предоставляется весь основной контент, а при прокрутке или переходах на другие страницы, вместо полной перезагрузки, нужные элементы просто подгружаются.
3. PWA (Progressive Web Application) – это веб-приложение, которое может быть использовано, как стандартное приложение. Они могут устанавливаться на главный экран мобильного устройства, отправлять push-уведомления и работать в офлайн-режиме, т.е. данное приложение может выполнять работу с локальными данными и отправлять их на сервер.

Из указанного выше описания можно определить, что необходимости в разработке полноценного PWA в данном случае нет, поэтому можно взять вторую по важности и сложности реализации концепцию работы прогрессивного веб-приложения, а именно работу в одном окне (SPA), что максимально удобно и предоставляет пользователю удобную навигацию между множеством вкладок, которые требуется открыть во время работы с приложением [1].

* 1. **Анализ существующих разработок и обоснование необходимости**

В настоящее время многие города по всему миру имеют свои собственные виртуальные музеи, которые позволяют жителям и туристам без труда ознакомиться с историей и культурой города даже не выходя из дома. Такие музеи дают возможность обществу получить более обширное и точное представление о прошлом и настоящем города, а также вносят большой вклад в развитие туризма. В связи с этим, было принято решение создать веб-приложение «Виртуального музея минского метрополитена».

Перед созданием виртуального музея минского метрополитена необходимо было провести анализ уже существующих проектов. Исследование показало, что большинство виртуальных музеев уже существующих городов ориентированы на туристов и посетителей, которые хотят узнать об истории и культуре города. Это позволяет развивать туризм и привлекать новых посетителей. Удобство виртуального музея не только заключается в том, что он доступен в любое удобное для посетителя время, но также в том, что содержание музея можно адаптировать под запросы и потребности пользователя. Большинство существующих виртуальных музеев оснащено интересными и уникальными экспонатами, а также снабжено разнообразными интерактивными функциями, которые позволяют полностью погрузиться в историю и культуру города.

Создание веб-приложения «Виртуальный музей минского метрополитена» обусловлено необходимостью развития туризма в городе Минск, а также удобства предоставления информации о метрополитене в первую очередь жителям города и туристам. Веб-приложение «Виртуальный музей минского метрополитена» станет полезной информационной площадкой для знакомства с этой транспортной системой, ее историей и устройством, его разработка позволит удовлетворить интересы широкого круга пользователей, а также сможет содействовать укреплению положительного имиджа города в глазах местных жителей и туристов. Таким образом, веб-приложение «Виртуальный музей минского метрополитена» станет важным инструментом для развития города и управления туристической индустрией.

* 1. **Постановка задачи**

Задачей данного дипломного проекта является разработка веб-приложения «Виртуальный музей минского метрополитена» с целью ознакомления пользователя с историческими датами, событиями, фактами минского метрополитена и обеспечение возможности удобного ее администрирования через панель администратора.

Целью разрабатываемого веб-приложения является:

1. обеспечение контроля и целостности данных со стороны администратора;
2. повышение качества и релевантности поиска информации об основании и развитии метрополитена;
3. обеспечение бесперебойного доступа к веб-приложению;
4. сокращение времени, необходимого для администрирования по сравнению с стандартными аналогами панелей администратора;
5. обеспечение удобства работы с интерфейсом как с клиентского, так и с администраторского доступа.
   1. **Формализация и документирование требований к программному средству**

Техническое задание (ТЗ) – исходный документ на проектирование технического объекта, устанавливающий: основное назначение разрабатываемого программного средства, его технические характеристики, показатели качества, требования, стадии создания документации и ее состав. В техническом задании описывается, что именно нужно заказчику в своем программном средстве.

При уточнении и формализации требований к программному средству были учтены следующие основные пункты:

1. сроки сдачи;
2. этапы разработки веб-приложения;
3. технические и технологические требования к разрабатываемому веб-приложению;
4. структура и описание контента веб-приложения;
5. входные и выходных данные;
6. процедуры тестирования и контроля качества.

Разработка ТЗ – важный подготовительный этап, от успешного выполнения которого зависит эффективность дальнейшего сотрудничества между заказчиком и исполнителем, а также качество полученного на выходе продукта.

Требования к разрабатываемому веб-приложению описаны в техническом задании Приложения А, которое соответствует стандарту IEEE 29148-2011 SyRS.

1. **Проектирование программного средства**

Проектирование веб-приложения, как и любого другого приложения, стоит начать с определения первоначальной цели и области решаемых задач, понять, к какому типу относятся потенциальные пользователи данного ресурса. На следующем этапе необходимо собрать требования к приложению, которое необходимо разработать. Уточнить цели и область решаемых задач и построить последовательную структуру работ.

Каждое веб-приложение отправляет HTTP-запросы на веб-сервер для получения данных. Программа под управлением веб-сервера использует ту или иную модель для хранения данных. В современном мире чаще всего используются базы данных, SQL или NoSQL.

Формально каждое веб-приложение можно разбить на 3 взаимно независимые части:

1. Модуль, который исполняется веб-браузером. Это приложение может быть написано на любом языке, который поддерживает браузер. Чаще всего используется язык JavaScript, как наиболее поддерживаемый и имеющий большую библиотечную поддержку, что позволяет существенно экономить бюджеты проектов.
2. Модуль, исполняемый на серверной стороне под управлением веб-сервера. Это приложение может быть написано на любом языке, интерпретацию которого поддерживает выбранный веб-сервер. Последнее время, в качестве языка программирования выбираются язык Java, PHP.
3. База данных. В этой области так же существует достаточно широкий выбор. Есть промышленные базы данных, такие как MSSQL, Oracle, PostgreSQL. Есть базы данных, такие как MySQL. База данных выбирается, основываясь на целях и области решаемых задач. Иногда реализация полноценной базы данных не требуется, так как это дополнительная нагрузка как по времени разработки, так и по задержкам на выполнение запросов в реализованной базе данных.

В случаях, если данные небольшого объема и разрабатываемая система хранения данных не подразумевает хранения большего объема, можно обойтись обычным массивом данных, вынесенном в отдельный файл, хранящийся в безопасном месте.

Разрабатываемое веб-приложение представляет из себя 2 экрана: пользовательский и администраторский. Поэтому, требуется обеспечить обмен данными между двумя этими экранами, чтобы внесенные изменения в панели администратора отображались на клиентский экран.

Для обеспечения обмена данными была реализована полная взаимосвязь клиентской и серверной части, которая имеет доступ к базе данных, содержащему детальную информацию о каждом субъекте информатизации.

* 1. **Разработка модели данных**

Модель данных разрабатываемого веб-приложение можно составить их 3 основных сущностей компонентов визуализации. К основным сущностям можно отнести:

1. тему;
2. подтему;
3. карточку подтемы.

Компоненты веб-приложения в свою очередь разделяются на 2 ветви:

1. компоненты пользователя – основные отображаемые каждому пользователю компоненты, содержимое которых пользователь изменить не может;
2. компоненты администратора – компоненты, доступ к которым имеет только администратор, предварительно выполнив авторизацию.

К компонентам пользователя относятся:

1. общий контейнер пользовательских компонентов – основной компонент, содержащий в себе все остальные компоненты, в котором преимущественно хранятся основные состояния приложения;
2. компонент тем – компонент, содержащий слайдер, отображающий только названия подтемы;
3. компонент слайдера – переиспользуемый компонент, отображающий информацию в виде блоков, которые можно пролистывать с помощью компонентов навигации по слайдеру;
4. компонент навигации по слайдеру – компонент управления, передающий новое состояние после выполненного действия – нажатия кнопок навигации;
5. контейнер со списком подтем – компонент, содержащий название и краткий текст подтем, выводящий информацию с помощью слайдера;
6. общий компонент для ссылок – компонент, использующийся, для контроля ссылок в компонентах;
7. компонент отображения конкретной темы – компонент, в который отображается вся общая информация по определенной подтеме;
8. контейнер со слайдером карточек подтем и блоком их отображения.

К компонентам панели администратора относятся:

1. компонент авторизации – форма для авторизации администратора;
2. основной компонент панели администратора – контейнер, содержащий информацию, размещенную на панели администратора;
3. компонент отображения тем – компонент с информацией, допустимой для изменения администратором, вложенными внутренними массивами второго и третьего уровня;
4. компонент отображения подтем – контейнер, размещенный в каждой теме, содержащий всю общую информацию о подтеме;
5. компонент контейнера карточек подтем – контейнер, необходимый для отображения карточек подтем;
6. компонент изменяемой карточки подтемы – компонент карточки подтемы, находящийся в состоянии изменения;
   * 1. **Входная и выходная информация**

Входная информация – это данные, которые поступают в веб-приложение из внешнего источника или от пользователя. Эти данные могут быть использованы в приложении для выполнения определенных задач и функций. Примером такой информации может служить: данные о пользователе, информация о продукте или услуге, результаты поиска и т.д.

Входными данными необходимыми для корректной работы веб-приложения «Виртуальный музей минского метрополитена» являются:

* база данных, содержащая информацию о станциях/линиях минского метрополитена, информацию с переводами веб-приложения на русский/белорусский/английский языки. Вид данных – база данных, формат данных – SQL, источник данных – файл на серверной части, режим ввода – автоматический, способ контроля – валидация при обработке, периодичность ввода – при каждой перезаписи содержимой базы данных;
* разрешение экрана пользователя, для определения стилей дизайна. Способ контроля – проверка браузером, вид данных – ширина и высота экрана в пикселях;
* руководство пользователя, необходимое для управления данными. Тип данных – текст, формат данных – doc, структура данных – файл, источник данных – документ, получаемый вместе с проектом.

Выходная информация – это результат работы веб-приложения, представленный в виде данных, графиков, таблиц, документов, изображений или другой информации, которая передается из приложения пользователю или внешней системе. Примерами выходной информации могут быть: отчеты, таблицы, графики, изображения, рекомендации и т.д.

Входная и выходная информация имеют критическое значение для работы веб-приложения. Если входные данные не соответствуют требованиям, то могут возникнуть ошибки в приложении или оно может не сработать как положено. Качество выходной информации зависит от правильной обработки входных данных и функциональности приложения. Чем более точна и правильна входная информация, тем более точна и качественная

Результатом работы веб-приложения является измененная база данных и дизайн макета веб-приложения, адаптированный под информацию из базы данных. Данная схема отображена на контекстной диаграмме типа IDEF0 на рисунке 1.

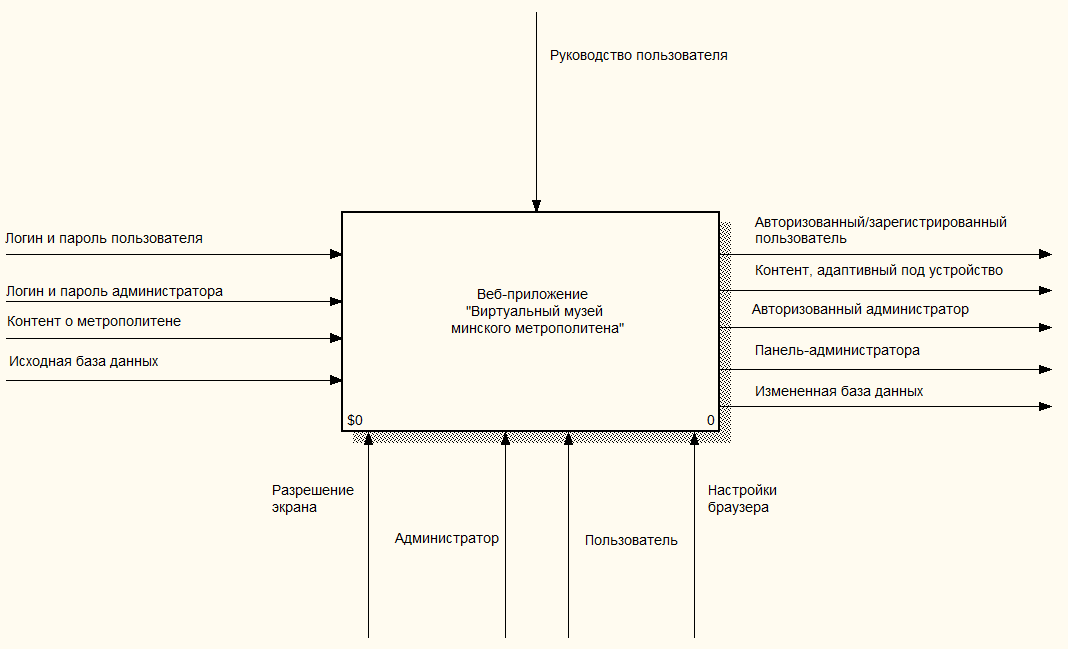


Рисунок 1 – Контекстная диаграмма IDEF0

* + 1. **Функциональная модель**

При разработке любого приложения, в том числе и веб-приложения, подходящим выбором считается разбиение кодовой базы на логические компоненты – страницы.

Результат выполнения этого этапа – общая модель объектов и связей между ними, функциональная модель в целом и каждой подсистемы, точно определенные с помощью CASE-средств интерфейсы, построение прототипов экранов, диалогов, отчетов.

IDEF0 – нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. Стандарт IDEF0 (Integration Definition for Function Modeling) утвержден в США в 1993 как стандарт обработки информации. К его особенностям можно отнести:

1. использование контекстной диаграммы;
2. поддержка декомпозиции;
3. доминирование;
4. выделение 4 типов стрелок.

Функциональные блоки на диаграммах изображаются прямоугольниками, означающими поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Имя работы должно быть выражено отглагольным существительным, обозначающим действие.

Благодаря ее использованию можно детально, разделяя по уровням вложенности, создать модель данных для понимания людей, занимающихся разработкой.

Функциональная модель была построена с предполагаемым порядком и расположением компонентов внутри клиентской части приложения и отображена в функциональной диаграмме нотации IDEF0, расположенной на рисунке 2.

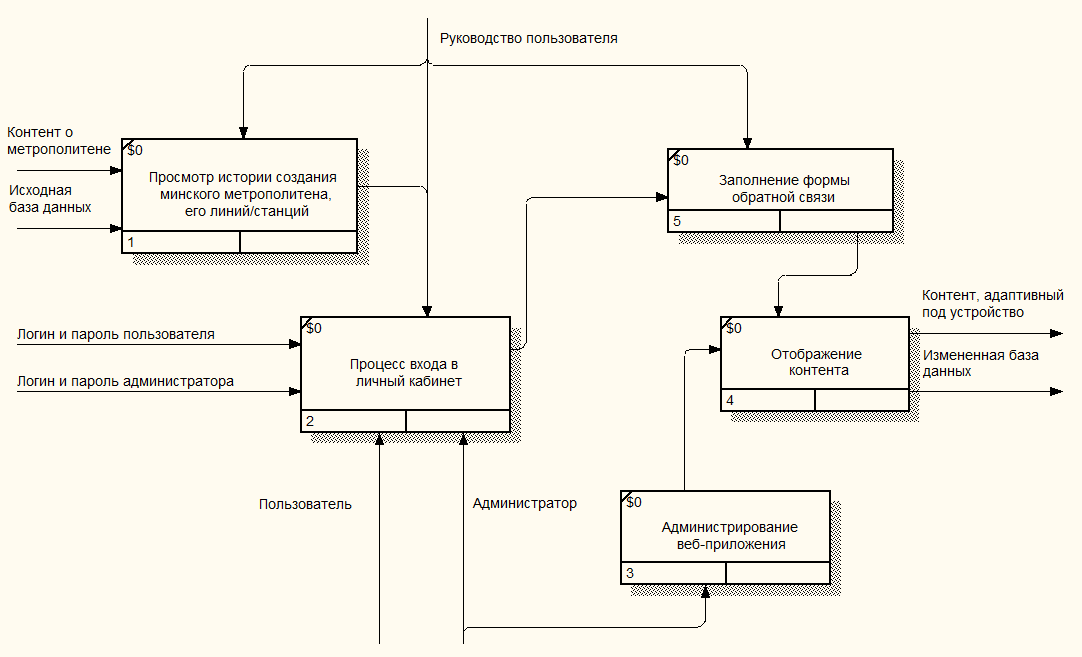


Рисунок 2 – Функциональная модель IDEF0

* + 1. **Структура данных**

Схему данных при построении данного веб-приложения можно отобразить, используя диаграмму взаимодействия компонентов, в которой можно увидеть все установленные связи между компонентами клиентской и серверной части. Данная диаграмма отображена на рисунке 8.

Вторым способом отображения схемы данных является представление в виде модели сущность-связь (ER-модель). Но, так как веб-приложение имеет всего 3 основные сущности, ее можно совместить с диаграммой классов, в которой размещается детальная информация о каждой сущности, изображенной на рисунке 3.

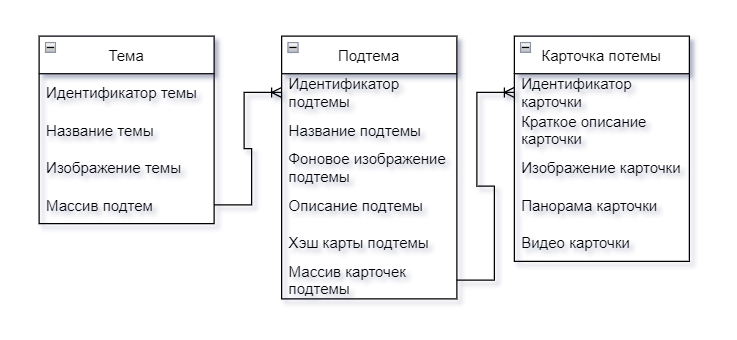


Рисунок 3 – Диаграмма отображаемых классов

* 1. **Иерархическая структура и ее характеристика**

Иерархическая структура работ (Work Breakdown Structure) – инструмент, позволяющий разбить проект на составные части. Она устанавливает иерархически структурированное распределение работ по реализации проекта. Данная структура сочетается с выбранным фреймворком разработки клиентской части, который также подразумевает разбиение приложения на компоненты.

В ходе построения WBS осуществляется последовательная декомпозиция проекта на подпроекты, пакеты работ различного уровня, пакеты детальных работ. Декомпозиция – это деление результатов проекта на меньшие, более управляемые компоненты до уровня пакетов работ. Пакеты работ обычно соответствуют самому нижнему уровню детализации и состоят из отдельных работ. Декомпозиция должна быть корректной, т.е. элементы любого уровня WBS должны быть необходимы и достаточны для создания соответствующего элемента верхнего уровня.

Документно-ориентированная структура массива данных веб-приложения позволяет построить явно выраженную и довольно простую иерархическую структуру субъектов реализации, которая имеет вид, указанный на рисунке В.2.

Также, иерархическую структуру можно построить и между компонентами интерфейса веб приложения используя диаграмму взаимодействия компонентов, изображенную на рисунке 4.

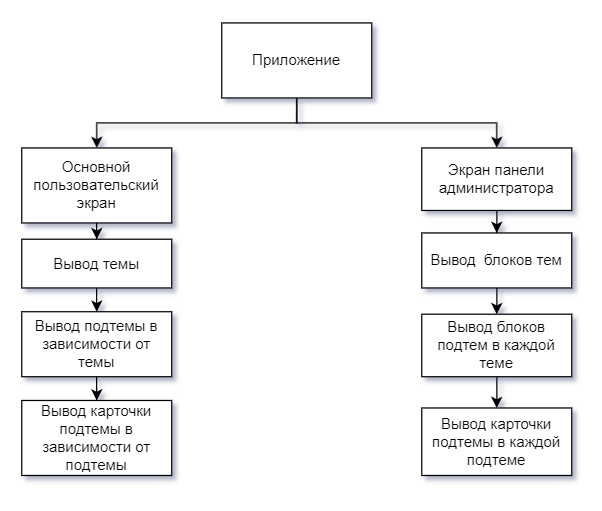


Рисунок 4 – Иерархическая структура основных компонентов

* 1. **Проектирование пользовательского интерфейса**

Пользовательский интерфейс представляет собой совокупность программных и аппаратных средств, обеспечивающих взаимодействие пользователя и вычислительной системы.

Проектирование пользовательского интерфейса – это создание тестовой версии приложения. Это начальный этап разработки пользовательского интерфейса, когда распределяются функции приложения по экранам, определяются макеты экранов, содержимое, элементы управления и их поведение.

Пользователь при обращении с интерфейсом должен представить себе, какая информация о выполняемой задаче у него существует, и в каком состоянии находятся средства, с помощью которых он будет решать данную задачу. Эффективность работы пользователя и его интерес обеспечивает правильно сформулированная методика разработки и проектирования пользовательского интерфейса.

Именно поэтому необходимо большое внимание уделять процессу построения пользовательских интерфейсов (UI) и выстраиванию пользовательского опыта в целом (UX). Проектирование UI – это не разовая фаза проекта, это непрерывный итерационный процесс, в который вовлечены бизнес-пользователи, UX-инженеры, дизайнеры и программисты.

UI (User Interface – дословно «пользовательский интерфейс») – то, как выглядит интерфейс, и то, какие физические характеристики приобретает. Определяет, какого цвета будет «изделие», удобно ли будет человеку выбрать кнопку, читабельным ли будет текст и т.п.

Визуально привлекательный и удобный пользовательский интерфейс – ключевой показатель качества сайта. В сочетании с грамотной структурой и навигацией по разделам ресурса, он способствует привлечению посетителей и улучшению функциональности веб-приложения.

Главная задача в проектировании веб-интерфейсов – максимально упростить UI пользователя, затрачивая минимум усилий для достижения конечной цели.

Весьма важно формализовать объект проектирования. От того, насколько полное формальное описание объекта зависит выбор метода решения задачи, а, следовательно, это позволяет определить, возможно ли применить средства вычислительной техники при проектировании.

Если задача не формализована, то проектировщик вынужден пользоваться каким-либо эвристическим методом решения задачи. Однако, когда задача полностью формализована (имеется полная математическая модель объекта проектирования), ее можно решать, используя стандартные методы проектирования.

Пользовательский интерфейс должен иметь четкую визуальное композицию, чтобы люди могли удобно использовать продукт. Золотое сечение часто применяется для эффективного размещения элементов пользовательского интерфейса.

Кроме того, схема золотое сечение поможет использовать изображения для веб-дизайна, балансируя в соответствии общей структурой сайта или приложения. Прежде всего, его можно использовать на этапе проектирования макета.

Для разработки интерфейса данного веб-приложения было просмотрено множество других разработок схожих по темам пользовательских интерфейсов. Из них были взяты некоторые успешно реализованные концепции, улучшающие пользовательское восприятие и опыт использования веб-приложений.

ОПИСАТЬ КАКИЕ ЭЛЕМЕНТЫ БЫЛИ ИСПОЛЬЗОВАНЫ ДЛЯ СОЗДАНИЯ ВБ, НИКАКИХ КАРТИНОК!

* 1. **Выбор программных и аппаратных средств**

Программной средой для разработки данного веб-приложения был использован редактор кода Visual Studio Code, который, путем установки дополнительных плагинов позволит разрабатывать проекты любого уровня сложности.

При выборе программных средств, были определены решения, с помощью которых возможно реализовать быструю, надежную и кодовую базу. Таким образом, для разработки клиентской части была выбрана библиотека React, которая легла в основу для разработки компонентной структуры пользовательского интерфейса данного веб-приложения.

Серверная часть веб приложения «Виртуальный музей минского метрополитена» тоже отличилась от стандартного вида back-end-разработки. Выбор лежал между стандартом прошлых лет в виде языка программирования PHP и фреймворком языка программирования JavaScript – Node.js. В итоге выбором стал Node.js из-за ряда объективных причин.

Обсуждая Node.js, необходимо упомянуть имеющуюся встроенную поддержку управления пакетами, применяемую на всех этапах разработки данного веб-приложения, для которой применяется инструмент NPM, по умолчанию присутствующий в любой установке Node.js.

Идея модулей NPM во многом схожа с Ruby Gems: это набор общедоступных компонентов для многократного использования, которые можно установить через онлайновый репозиторий; для них поддерживается управление версиями и зависимостями.

Также, на стороне сервера присутствует необходимое обеспечение для упрощения разработки в среде Node.js, которым является Express.js.

Express – это минималистичный и гибкий веб-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и веб-приложений. Он предоставляет тонкий слой фундаментальных функций веб-приложений, которые встроенными функциями Node.js.

1. **Разработка программного средства**
   1. **Описание основных алгоритмов**

Так как данное веб-приложение включает в себя необходимые для его работы функции, для их структуризации требуется четкое разделение по месту их применения, а так как большая часть приложения состоит из компонентов, вся бизнес-логика находится непосредственно в этих компонентах. Это позволяет не только структурировать данные, но и более удобно манипулировать функциями, находящимися в компонентах.

Описание функций и методов, реализованных для корректной работы бизнес-логики, проходило с использованием как формального описания, так и словесного.

Чтение и запись массива данных выполняются с помощью встроенного модуля «fs» для работы с файловой системой средствами, доступными на стороне сервера с помощью Node.js. На сервер поступает запрос на чтение или запись данных, принимается определенным эндпоинтом, считываются данные запроса, вызывается соответствующая функция, происходит считывание/запись данных, новые данные поступают в счм.

Загрузка изображения в тему/подтему/карточку подтемы происходит через универсальный компонент, обрабатывающий перемещение файлов с разрешением jpg/png/jpeg/avif. После успешного прохождения фильтрации типа внедренного файла происходит создание обертки для него в виде FormData, со встроенными методами, упрощающими транспортировку данных с клиентской на серверную часть. После оборачивания изображения в форму и добавления атрибута элемента, в который нужно добавить изображение, оно отправляется на сервер с помощью метода fetch.

На сервере эта форма также принимается эндпоинтом image. Следующим этапом проходит деструктуризация формы, чтение изображения, поиск расположения данного элемента в массиве данных. После поиска места в массиве проверяется, существует ли папка, находящаяся по пути в соответствии с темой.

Удаление изображений происходит по другому алгоритму, так как при удалении информация о месте удаления содержится в пути изображения, который и передается на сервер. На клиентской части запрашивается подтверждение на операцию удаления изображения.

При операции удалении темы должен не только измениться массив данных, но и удалиться соответствующая папка изображений, если она имеется, для предотвращения утечек памяти на стороне сервера. Во время перебора массива из него извлекается удаляемая тема, и отправляется соответствующий запрос на сервер с идентификатором массива.

На сервере проверяется наличие директории, соответствующей идентификатору удаляемой темы и, в случае успешного поиска, происходит его удалении всей директории.

Еще одной функцией является общая клиентская навигация и поиск по названию тем и подтем. Они реализованы в одном компоненте навигации и более важным является алгоритм поиска по массиву с вложенными подтемами.

Так как этот поиск должен выполнять поиск не только по темам, но и подтемам, ему одновременно требуется проверять условие, по которому результатом поиска будет найдена тема со всеми вложенными подтемами, так и проверять условия совпадения с названием подтемы.

С кодом описанных функций можно ознакомиться в листинге кода, расположенном в приложении Д.

* 1. **Реализация взаимосвязи компонентов программного средства**

В структуре данного веб-приложения присутствует множество компонентов с всевозможной информацией, передаваемой через связи между ними. В фреймворке React, применяемом на стороне клиента, информацию, передаваемую между компонентами принято называть пропсами [6].

Всего существует 3 случая связи между компонентами React:

1. общение от родительского к дочернему элементу;
2. общение от дочернего к родительскому элементу;
3. не связанные компоненты (любой компонент для любого компонента).

В случае взаимосвязи компонентов в данном приложении, в основном были использован первый вариант связи.

Во время программирования веб-приложения была реализована структура связей, отображенная на рисунке 8.

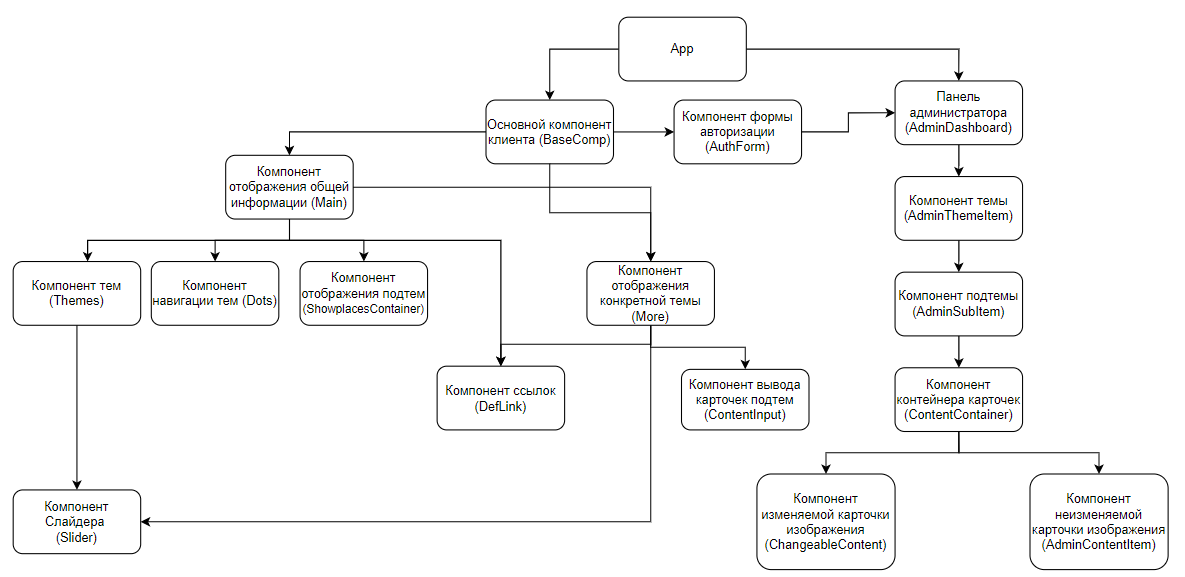


Рисунок 8 – Структура связей компонентов

* 1. **Защита информации**

Простые методики защиты приложения не требуют крупных финансовых вложений и доступны большинству владельцев веб-приложений. Среди них самые популярные:

1. аудит (превентивная мера);
2. использование защищенных протоколов передачи данных;
3. применение ПО, обеспечивающего безопасность.

Все способы необходимо применять в комплексе.

Прежде чем разрабатывать методику защиты веб-приложения от потенциальных угроз, сайт следует проверить на уязвимости. Проверка проводится ручным или автоматизированным способом. Программы, доступные в платной (SpecFlow) и бесплатной версиях (NUnit), протестируют приложение на основные риски. Такие программные продукты существуют в двух вариантах: Black hat, моделирующие действия взломщиков, и White hat, планомерно выявляющие все недочеты системы методом сканирования.

После исправления крупных уязвимостей сканирование следует провести повторно. Закончив автоматическую проверку, можно организовать взлом веб-приложения вручную. Для этого нужно изменить значение запросов POST (отправка данных на ресурс) и GET (запрос данных у ресурса) в HTTP. Лучше использовать прокси-сервер, перехватывающий HTTP-запросы. Также необходимо обойти валидацию данных (проверку соответствия запроса заданным требованиям) и внедрить на сайт SSL-инъекцию, перехватывающую данные пользователей. Если системы мониторинга показывают существенные уязвимости, необходимо усиливать защиту в выявленных областях.

Используя платные или бесплатные ресурсы мониторинга систем безопасности необходимо проверить гипотетическую возможность обойти требования обязательной аутентификации, предусмотренные для некоторых страниц веб-приложения. Для этого нужно использовать такие традиционные способы взлома как смена параметров URL (в частности, ID пользователя) или смена Cookie.

HTTPS

Вторым по популярности способом защитить данные пользователя после идентификации является применение защищенного протокола передачи данных HTTPS. Hyper Text Transfer Protocol Secure защищает информацию о пользователе веб-приложения при помощи шифрования трафика. Он обеспечивает сохранение конфиденциальности и целостности информации, не допуская утечку или подмену данных.

Большинство ресурсов использует HTTPS технологию давно, это стало хорошим тоном, подтверждающим готовность их владельцев защищать интересы клиентов. Поисковая система Google быстрее индексирует в выдаче сайты, использующие эту технологию.

HTTPS необходим, если пользователи передают сервису такие сведения, как:

1. номера кредитных карт;
2. персональные данные;
3. адреса страниц, на которые они заходят.

При генерации запроса с формы авторизации применяются cookie-файлы, они подлежат отправке на сервер при каждом запросе. При недостаточной защите веб-приложения ничего не мешает злоумышленнику перехватить файлы и изменить запрос, получив права пользователя. Применение HTTPS для каждой страницы сайта снизит степень этого риска.

Решить задачу несложно, потребуются следующие шаги:

1. сгенерировать SSL-сертификат (на некоторых ресурсах это делается бесплатно);
2. получить и установить сертификат;
3. подключить для веб-приложения поддержку HTTPS.

Дополнительной возможностью после настройки HTTPS станет применение Hyper Strict Transport Security (HSTS). Это опция принудительного использования протокола HTTPS, даже если сервер не поддерживает его применение. Однако и защищенные протоколы не дадут достаточной защиты веб-приложению, если само программное обеспечение устарело.

Избежать риска получится, если применять параметризованные запросы, в которых задействовано несколько языков программирования.

Работа с веб-сервисами требует использования широкого инструментария для обеспечения безопасности. Помимо основных перечисленных методов, часто применяются:

1. шифрование паролей;
2. избегание межсайтового скриптинга;
3. контроль загрузки файлов на сервер.

Совместное применение всех доступных решений обеспечит безопасность на максимально доступном уровне.

Составляющие Web-проекта с точки зрения обеспечения его безопасности:

1. Информационная среда Web-сервера:
2. операционная система;
3. Web-сервер;
4. среда программирования;
5. база данных.
6. Информационная среда администраторов Web-проекта:
7. КИС компании, управляющей сайтом;
8. система антивирусной защиты КИС;
9. средства защиты КИС от атак из Интернет;
10. средства защиты КИС от утечек информации.
11. Сторонние Web-приложения.

Стороннее Web-приложение можно рассматривать как проект в проекте, и поэтому задача обеспечения безопасности этих Web-приложений базируется на тех же методах и приемах, которые были изложены в предыдущих пунктах.

* 1. **Тестирование программного средства**

При разработке веб-приложения, как и любого другого приложения, обязательным условием является проведение его тестирования. В зависимости от степени доступа к коду системы можно выделить два типа функционального тестирования:

* тестирование black box (черный ящик) – проведение функционального тестирования без доступа к коду системы;
* тестирование white box (белый ящик) – функциональное тестирование с доступом к коду системы.

Функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного программного обеспечения (ПО) исходным функциональным требованиям заказчика. Проведение функционального тестирования – процесс, позволяющий проверить способность информационной системы работать в определенных условиях и решать задачи, нужные пользователям.

После проведения функционального тестирования были получены следующие результаты:

* метод черного ящика показал, что все основные требования к веб-приложению были выполнены в полном объеме;
* метод белого ящика выявил некоторые небольшие недоработки, которые сразу же были доработаны.

Таким образом, метод функционального тестирования позволил избежать некоторых ошибок и сбоев в работе данного веб-приложения, а также позволил убедиться в необходимости его применения в процессе разработки программных продуктов. Детальная информация о проведенных тестах размещена в Приложении Б.

Также, было проведено тестирование удобства использования разработанного клиентского интерфейса веб-приложения. Исходя из полученной информации после тестирования было определено следующее:

* при работе с веб-приложением пользователю достаточно пользоваться только мышью и клавиатурой;
* для выполнения основных пользовательских сценариев пользователю необходимо выполнить минимальное количество действий;
* интерфейс имеет оптимальный для чтения шрифт и размер символов.
  1. **Разработка документации к программному средству**

Проектная документация на любой программный продукт является необходимой вещью, так как является определяющей в процессе разработки и представляет собой то видение системы, которое будет в дальнейшем реализовано [3, с. 172]. Однако, не менее важны и эксплуатационные документы, которые позволят использовать и обслуживать систему правильно, эффективно и с минимальными затратами.

Одним из необходимых в процессе эксплуатации системы документов является «Руководство пользователя». Цель создания документа заключается в том, чтобы предоставить пользователю возможность самостоятельно решать свои прикладные задачи с помощью системы. Этой цели может служить и введение в предметную область, и ознакомление со всеми возможностями программы, и описание конкретных процедур решения задач, и приведение различных инструкций. Иногда руководство пользователя больше похоже на справочник, к которому можно обращаться в процессе работы, а иногда – на учебник, который позволяет изучить принципы работы с программой и ее возможности, а затем применять их на практике.

Конкретный подход к написанию определяется многими факторами:

* назначением программы и областью ее применения;
* сложностью программы;
* количеством разнообразных вариантов использования;

Принимая во внимание все различия и особенности, сложно привести структуру любого руководства пользователя к одному виду. В документ «Руководство пользователя» обычно включают такие разделы, как:

* введение, где указывают область применения ПО, кратко описывают его возможности, требуемый уровень знаний пользователя и список документов, которые необходимо изучать помимо настоящего руководства.
* назначение и условие применения, где описывают виды деятельности и функции, которые автоматизированы и условия, при соблюдении которых автоматизация используется.
* подготовка к работе, где описывают комплектность дистрибутива, порядок установки и загрузки программы, а также способ проверки ее работоспособности.
* описание операций, представляет собой основной раздел, где описывают функции программы, процессы работы с данными, выполнение конкретных задач пользователя.
* аварийные ситуации, где описывают действия в нештатных ситуациях – сбоях в программе, ошибок в данных и так далее.
* рекомендации по освоению, где приводят методические рекомендации по изучению программы и примеры использования.

Данная структура может меняться и дополняться – например, основной раздел часто разбивают на несколько значимых разделов пои группам функций и задач, также в современных системах нередко добавляют раздел Интерфейс пользователя, где описывают взаимодействие пользователя с программой с примерами и снимками экрана. На этих требованиях было реализовано руководство пользователя разрабатываемого веб-приложения. Руководство пользователя размещено в Приложении В.

1. **Охрана труда, техника безопасности, противопожарные мероприятия и охрана окружающей среды**

В Республике Беларусь отмечается бурное внедрение во всех отраслях человеческой деятельности персональных компьютеров. Их применение позволило значительно повысить производительность труда в различных сферах трудовой деятельности, изменить характер и содержание труда.

По определению, охрана труда – это система законодательных актов, социально-экономических, организационных, технических, гигиенических и лечебно-профилактических мероприятий и средств, обеспечивающих безопасность, сохранение здоровья и работоспо­собности человека в процессе труда.

Функциями охраны труда являются исследования санитарии и гигиены труда, прове­дение мероприятий по снижению влияния вредных факторов на организм работников в про­цессе труда. Основным методом охраны труда является использование техники безопасно­сти. При этом решаются две основные задачи: создание машин и инструментов, при работе с которыми исключена опасность для человека, и разработка специальных средств защиты, обеспечивающих безопасность человека в процессе труда, а также проводится обучение ра­ботающих безопасным приемам труда и использования средств защиты, создаются условия для безопасной работы.

Научно-технический прогресс существенно изменил характер жизненной деятельно­сти человека. С трудовой деятельностью людей связана особая группа психофизиологиче­ских факторов, создающих высокие уровни физических и нервно-психических нагрузок и обусловленную ими тяжесть и напряженность труда.

Работа с персональным компьютером и программирование связано с необходимостью длительно находиться в вынужденной рабочей позе, что ведет к различным формам заболе­ваний опорно-двигательного аппарата человека. Работа преподавателей информатики связа­на с восприятием изображения на экране, необходимостью постоянного слежения за дина­микой изображения, различением текста рукописных или печатных материалов, выполнени­ем машинописных, графических работ и других операций. В большинстве случаев работа с дисплеем требует высокой степени сосредоточенности, звуковые раздражения, вызываемые посторонними шумами, должны быть сведены к минимуму.

Персональные электронные вычислительные машины являются электроустановками, в помещении с ними могут возникнуть аварийные ситуации: короткое замыкание, возгора­ние проводки и оборудования, поражение операторов электротоком. Таким образом, условия труда операторов электронной вычислительной системы, несмотря на отсутствие явных вредностей, нуждаются в оптимизации [12, с. 93].

* 1. **Организация рабочего места для работающих на компьютере**

Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, желательно слева.

Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м.

Рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы. Целесообразно применение столов, имеющих отдельную от основной столешницы специальную рабочую поверхность для размещения клавиатуры. Используются рабочие столы с регулируемой и нерегулируемой высотой рабочей поверхности. При отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.

Глубина рабочей поверхности стола должна составлять 800 мм (допускаемая не менее 600 мм), ширина – соответственно 1 600 мм и 1 200 мм. Рабочая поверхность стола не должна иметь острых углов и краев, иметь матовую или полуматовую фактору.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали).

Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю.

Для удобства считывания информации с документов применяются подвижные подставки (пюпитры), размеры которых по длине и ширине соответствуют размерам устанавливаемых на них документов. Пюпитр размещается в одной плоскости и на одной высоте с экраном.

Для обеспечения физиологически рациональной рабочей позы, создания условий для ее изменения в течение рабочего дня применяются подъемно-поворотные рабочие стулья с сиденьем и спинкой, регулируемыми по высоте и углам наклона, а также расстоянию спинки от переднего края сидения.

Конструкция стула должна обеспечивать:

* ширину и глубину поверхности сиденья не менее 400 мм;
* поверхность сиденья с закругленным передним краем;
* регулировку высоты поверхности сиденья в пределах 400-550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов;
* высоту опорной поверхности спинки 300±20 мм, ширину — не менее 380 мм и радиус кривизны горизонтальной плоскости 400 мм;
* угол наклона спинки в вертикальной плоскости в пределах 0±30 градусов;
* регулировку расстояния спинки от переднего края сидения в пределах 260-400 мм;
* стационарные или съемные подлокотники длиной не менее 250 мм и шириной 50-70 мм;
* регулировку подлокотников по высоте над сиденьем в пределах 230±30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм;
* поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим неэлектризующимся, воздухонепроницаемым покрытием, легко очищаемым от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 град. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

* 1. **Производительность труда и рабочее место**

Рабочее место, хорошо приспособленное к трудовой деятельности, правильно и целе­сообразно организованное, в отношении пространства, формы, размера обеспечивает удоб­ное положение при работе и высокую производительность труда при наименьшем физиче­ском и психическом напряжении.

При правильной организации рабочего места производительность труда работника возрастает от 8 до 20%.

Согласно ГОСТ 12.2.032-78 конструкция рабочего места и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологи­ческим требованиям. Большое значение имеет также характер работы. В частности, при ор­ганизации рабочего места учащегося должны быть соблюдены следующие основные усло­вия:

* оптимальное размещение оборудования, входящего в состав рабочего места;
* достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения;
* необходимо естественное и искусственное освещение для выполнения поставлен­ных задач;
* уровень акустического шума не должен превышать допустимого значения.

Главными элементами рабочего места являются письменный стол и кресло. Основ­ным рабочим положением является положение сидя. Рабочее место для выполнения работ в положении сидя организуется в соответствии с ГОСТ 12.2.032-78.

Негативное влияние компьютеров на здоровье пользователей выражается в повы­шенном зрительном напряжении, психологической нагрузке, длительном неизменном поло­жении тела в процессе работы, а также воздействии некоторых физических факторов (элек­тромагнитных излучений, статического электричества, ультрафиолетового и рентгеновского излучения).

Важнейшее значение в возникновении зрительного перенапряжения имеет качество более двадцати визуальных параметров изображения на дисплее. Поэтому выполнение тре­бований, установленных действующими стандартами к ним, имеет первостепенное значение в профилактике ухудшения зрения пользователей компьютера.

Компьютеры с жидкокристаллическим экраном не имеют источников мощного элек­тромагнитного излучения и не наводят статического электричества. Однако при использова­нии блока питания возникает некоторое превышение уровня на частоте 50 Гц, поэтому рекомендуется работать больше с использованием аккумулятора.

Эффективным средством защиты от излучений персональных компьютеров с элек­тронно-лучевой трубкой является применение дополнительного металлического внутреннего корпуса, замыкающегося на встроенный закрытый экран. Такая конструкция позволяет уменьшить электрическое и электростатическое поля на расстоянии 7-8 см от корпуса до фоновых значений.

Во всех случаях для снижения уровня облучения монитор рекомендуется располагать на расстоянии не ближе 50 см от пользователя.

* 1. **Зоны досягаемости рук в горизонтальной плоскости**

Все оборудование, средства труда и документация должны иметь постоянные места и быть упорядоченными. Все, что используется чаще всего, должно быть расположено в зоне легкой досягаемости рабочего пространства.

Зоны досягаемости располагаются в рамках моторного поля, т.е. того пространства, в котором могут осуществляться движения рук человека (для пользователя ПЭВМ – в положении сидя). Моторные зоны классифицируют по-разному: оптимальные, допустимые, предельно допустимые; зона максимальной досягаемости; зона досягаемости пальцев при вытянутой руке; зона легкой досягаемости ладони; оптимальное пространство для грубой ручной работы; оптимальное пространство для тонкой ручной работы, изображено на рисунке 10.

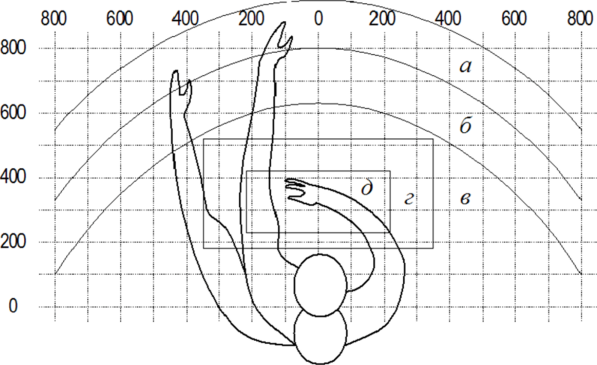


Рисунок 10 – Зоны досягаемости рук в горизонтальной плоскости

а – зона максимальной досягаемости;

б – зона досягаемости пальцев при вытянутой руке;

в – зона легкой досягаемости ладони;

г – оптимальное пространство для грубой ручной работы;

д – оптимальное пространство для тонкой ручной работы.

Максимальной зоной досягаемости рук называется та часть моторного поля, которая охватывается максимально вытянутыми руками, а оптимальной зоной – та часть, которая охватывается предплечьями при относительно неподвижном плече. Исходя из классификации моторных зон, рекомендуется следующее рациональное расположение оборудования и документов в моторном поле пользователя ПЭВМ:

* монитор размещается в зоне а (в центре);
* принтер находится в зоне а (справа);
* системный блок размещается в зоне б (слева);
* клавиатура – в зоне г/д;
* документация (находящаяся в работе) – в зоне в (слева).

При этом необходимо учитывать, что документация может располагаться не только слева, но и между клавиатурой и монитором и т.д. Если изображение на экране монитора имеет невысокое качество, то рекомендуют расстояние от глаз до монитора делать больше (примерно 70 см), а до документации меньше (30 - 45 см). Если же качество изображения на экране монитора хорошее, то эти расстояния могут быть равными.

К конструкции рабочего стола не предъявляется единых требований. Возможно использовать разные конструктивные решения, но все они должны отвечать эргономическим требованиям. Желательно, чтобы высота стола могла регулироваться. Для взрослого пользователя этот диапазон рекомендован в пределах 68 - 80 см. Фиксированная высота стола должна составлять 72,5 см. Модули рабочей поверхности должны учитывать возможную ширину стола в 80,100,120 и 140 см и его глубину 80 - 100 см. Края стола не должны иметь острых углов. Коэффициент отражения поверхности рабочего стола должен составлять 0,5 - 0,7. Поверхность стола должна быть матовой или полуматовой, для того чтобы исключить отраженную бликовость.

* 1. **Требования безопасности при работе на персональных электронно-вычислительных машинах**

Каждое трудовое место оператора оборудовано компьютером (монитором, системным блоком, клавиатурой, мышью). К каждой рабочей поверхности подведены электропитание и сетевой кабель.

Правильное местоположение трудовых мест операторов в рабочих кабинетах обеспечивает свободный доступ операторов к рабочему месту, при изоляции электромагнитных излучений.

Работа за компьютером осуществляется при искусственном освещении и при плотно затянутых окнах, что обеспечивает постоянный уровень освещенности на рабочих местах. Поверхность пола должна быть ровной и без выбоин и щелей. В кабинетах установлены люминесцентные светильники, которые обеспечивают естественное и искусственное освещение.

Каждый кабинет должен регулярно проветриваться для поддержания циркуляции свежего воздуха или быть оборудованным работающей системой приточно-вытяжной вентиляции. Кроме того, правильное проектирование оконных проемов и люминесцентных ламп способствует естественному и искусственному освещению.

Работа за компьютером может неблагоприятно повлиять на здоровье пользователя в виде нагрузки на глаза, тело от продолжительного неизменного положения тела в процессе работы и стресса.

Для снижения напряжения на зрение следует установить комфортный для глаз фон с избеганием токсично-ярких цветов, установить максимальную частоту обновления экрана и комфортное разрешение экрана. Рекомендуется делать перерывы между работой за компьютером каждые 1-2 часа и заниматься гимнастикой для глаз для профилактики зрения. Для снятия напряжения с тела, вызванного сидячей работой, рекомендуется заниматься физическими упражнениями или прогулками на свежем воздухе, что, кроме того, положительно влияет на психическое здоровье. Компьютеры с жидкокристаллическим экраном не источают мощных электромагнитных излучений и не наводят статического электричества. Однако при использовании блока питания возникает некоторое превышение уровня на частоте 50 Гц, поэтому рекомендуется работать с использованием аккумулятора.

Эффективным средством защиты от излучений персональных компьютеров с электронно-лучевой трубкой является применение дополнительного металлического внутреннего корпуса, замыкающегося на встроенный закрытый экран. Такая конструкция помогает уменьшить электрическое и электростатическое поля на расстоянии 7-8 см от корпуса до фоновых значений.

* 1. **Гигиенические требования к правильной посадке при работе на компьютере**

К моментам, влияющим на работоспособность как устройства, так и оператора, являются гигиенические правила, такие как:

* приступая к работе на компьютере, необходимо осмотреть рабочее место и убедиться в отсутствии видимых повреждений оборудования;
* монитор должен быть установлен прямо перед пользователем, на расстоянии 60-70 см. при работе недопустимы повороты головы или корпуса тела пользователя;
* не рекомендуется располагать монитор около яркого источника света; на мониторе не должно быть бликов, сильного контраста с внешним освещением;
* рабочий стол и посадочное место должны иметь такую высоту, чтобы уровень глаз пользователя находился на уровне или чуть выше центра монитора;
* высота стула должна быть такой, чтобы при правильной установке монитора относительно уровня глаз, ноги пользователя были на полу; при необходимости следует установить подставку. Сидеть следует свободно, без напряжения, не сутулясь и не облокачиваясь на спинку стула;
* туловище пользователя должно находиться от стола на расстоянии 15-16 см;
* клавиатура должна быть расположена прямо перед пользователем и на такой высоте, чтобы пальцы рук располагались на ней свободно, без напряжения, а угол между плечом и предплечьем составлял 100-110 градусов;
* во избежание чрезмерных нагрузок на кисть при длительной работе с клавиатурой желательно оборудовать рабочее место креслом с подлокотниками, уровень высоты которых, замеренный от пола, совпадает с уровнем высоты расположения клавиатуры;
* при работе с мышью рука не должна находиться на весу. Локоть или запястье должны иметь твердую опору. Провод мыши должен лежать свободно;
* все питающие провода и соединительные кабели должны располагаться с задней стороны компьютера и периферийных устройств. Их размещение в рабочей зоне пользователя недопустимо;
* в помещении, где находится компьютер, нельзя находиться в верхней и влажной одежде;
* в целях пожарной безопасности нельзя кушать и пить рядом с компьютером.

На основании вышеизложенного можно сказать, что правильно организованное рабо­чее место пользователя ПК в максимальной степени содействует повышению работоспособ­ности и производительности труда пользователей компьютерной техникой.

1. **Экономическая часть**
   1. **Характеристика проекта**

В нынешнее время иметь свой сайт – значит соответствовать высоким стандартам современного подхода к ведению дел организацией. Для современного общества все больше важна доступность интересуемой их информации в сети. Поэтому представление информации в сети Интернет является первостепенной задачей на пути развития любого рода проектов и планов их развития.

Веб-приложение «Виртуальный музей минского метрополитена» предназначено для ознакомления пользователя с историческими датами, событиями, фактами минского метрополитена и обеспечением возможности удобного администрирования через панель администратора. Пользователи могут использовать веб-приложение «Виртуальный музей минского метрополитена» для изучения истории минского метрополитена, просмотра фотографий и видео материалов.

Веб-приложение «Виртуальный музей минского метрополитена» позволяет пользователям получить обширную информацию о минском метрополитене – ознакомиться со станциями, ветками и историей развития метрополитена. Приложение также предоставляет возможность оставлять отзывы и комментарии о посещенных объектах, что позволяет пользователям делиться своим мнением и опытом.

Наличие новостей также является важным функционалом приложения, которое позволяет пользователям быть в курсе последних событий, изменений и нововведений по теме метрополитена города Минска.

Возможность администрирования этого приложения также является важным функционалом. Администраторы могут контролировать содержимое.

Для реализации любого проекта требуется его экономическое обоснование. Таким обоснованием является оценка экономической эффективности проекта. Для проведения такой оценки необходимо сопоставить средства, требующиеся для разработки, и эффект, ожидаемый от внедрения соответствующего проекта.

Главной целью данного дипломного проекта стало разработка веб-приложения «Виртуальный музей минского метрополитена».

Для достижения поставленной цели необходимо решить следующие задачи:

* произвести расчет расходов на оплату труда и отчислений на социальные нужды;
* по проектируемому варианту;
* рассчитать материальные затраты по проектируемому варианту;
* трудоемкость создания программного продукта;
* экономический эффект от разработки и внедрения программного продукта.

Все расчеты выполнены на 03.05.2023 года с учетом цен, ставки первого разряда, существующих налогов и ценовых отчислений.

* 1. **Определение трудоемкости создания программного продукта**

Создание программного продукта очень сложный процесс, поэтому для определения трудоемкости необходимо составить перечень всех видов и этапов работ в соответствие с таблицей 1, выполняемых при исследовании. Особое внимание было уделено упорядочению выполнения работ, для каждой работы определяется ее продолжительность и квалификация исполнителя.

Таблица 1 – План проведения научно-исследовательской работы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов и видов работ | Исполнитель (должность, квалификация) | Количество исполнителей | Трудоемкость, человеко-дни, |
| 1 Составление и согласование технического задания | Техник-программист | 1 | 2 |
| 2 Сбор и изучение научно-технической литературы | Техник-программист | 1 | 1 |
| 3 Формулирование возможных направлений решения задач | Техник-программист | 1 | 2 |
| 4 Разработка методики проведения исследований | Техник-программист | 1 | 2 |
| 5 Исследование задач наблюдения и возможность их сведения к задачам | Техник-  программист | 1 | 4 |
| 6 Разработка и реализация алгоритма решения задачи с большим количеством ограничений | Техник-  программист | 1 | 5 |
| 7 Оценка некоторого количества задач и решение их | Техник-  программист | 1 | 4 |
| 8 Решение ряда задач наблюдения | Техник-  программист | 1 | 4 |
| 10 Составление плана мероприятий по техники безопасности | Техник-  программист | 1 | 2 |
| 11 Оформление документации по выполненному программному продукту | Техник-  программист | 1 | 4 |
| 12 Оформление графического материала | Техник-  программист | 1 | 3 |
| 13 Обобщение результатов исследований | Техник-  программист | 1 | 2 |
| 14 Составление и оформление отчета | Техник-  программист | 1 | 3 |
| Всего: | – | – | 38 |

* 1. **Определение полной себестоимости и отпускной цены программного продукта**

Целью планирования себестоимости программного продукта является экономически обоснованное определение величины затрат на ее выполнение. В плановую себестоимость программного средства включаются все затраты, связанные с ее выполнением, независимо от источника их финансирования. Определение затрат на создание программного средства производятся путем составления калькуляции плановой себестоимости.

Все расчеты выполнены на 03.05.2023 года с учетом цен, ставки первого разряда, существующих налогов и ценовых отчислений.

Калькуляцию плановой себестоимости программного средства составим по следующим статьям затрат:

1. основная заработная плата;
2. дополнительная заработная плата;
3. отчисления на социальные нужды;
4. материальные затраты;
5. содержание и эксплуатация основных фондов;
6. налоги и сборы, включаемые в себестоимость;
7. накладные расходы;
8. прочие затраты.

Основная заработная плата техника-программиста рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (1) |

где  – часовая тарифная ставка i-го разряда;

 – количество часов работы в день ( ч);

 – фонд рабочего времени i-го исполнителя.

Часовая тарифная ставка i-го разряда определяется следующим образом:

|  |  |
| --- | --- |
|  | (2) |

где  – базовая ставка 4 разряда, установленная законодательно – 228 руб.;

 – нормативное количество рабочих часов в месяц. Для пятидневной рабочей недели с выходными днями в субботу и воскресенье расчетная норма рабочего времени в 2023 году составляет 2 011 часов (167,6 часов в месяц);

 – тарифный коэффициент i-того разряда.

Работа техника-программиста тарифицируется 4 тарифным разрядом ( = 1,21). Определим часовую тарифную ставку техника-программиста 4 разряда:

Рассчитаем на основе часовой тарифной ставки основную заработную плату исполнителя:

Дополнительная заработная плата определяется по нормативу в процентах к основной заработной плате:

|  |  |
| --- | --- |
|  | (3) |

где  – норматив дополнительной заработной платы в целом по организации (15%).

Определим размер дополнительной заработной платы:

Отчисления в Фонд социальной защиты населения определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной заработной платы исполнителей. Сумма отчислений в фонд социальной защиты населения () определяется по формуле:

|  |  |
| --- | --- |
|  | (4) |

где  – норматив отчислений в Фонд социальной защиты населения  (НЗСЗ =34%).

Кроме отчислений в Фонд социальной защиты населения в учреждения образования производятся отчисления в Белгосстрах. Определим их следующим образом:

|  |  |
| --- | --- |
|  | (5) |

где  – норматив отчислений в Белгосстрах ( = 0,15 %).

Расходы по статье «Материальные затраты» включают стоимость израсходованных материалов и затраты на электроэнергию.

Стоимость израсходованных материалов была определена исходя из фактического расхода и цены приобретения и размещена в таблице 2.

Таблица 2 – Расчет стоимости израсходованных материалов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование материалов | Ед. изм. | Расход | Цена за единицу, руб. | Стоимость, руб. |
| Бумага | уп. (500 листов) | 1 | 14,00 | 14,00 |
| Краска для картриджей | шприц | 1 | 9,00 | 9,00 |

Продолжение таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CD-диски | шт. | 2 | 2,50 | 5,00 |
| Итого: | − | − | − | 28,00 |

Затраты на электроэнергию () определим следующим образом:

|  |  |
| --- | --- |
|  | (6) |

где  – тариф за 1 кВт/ч электроэнергии, руб.;

,  – время работы персонального компьютера и принтера соответственно, час;

,  – номинальная потребляемая мощность персонального компьютера и принтера соответственно, кВт/ч.



Расходы на содержание и эксплуатацию основных фондов включают суммы амортизационных отчислений от стоимости ЭВМ и от стоимости площадей, занятых ЭВМ.

Годовая величина амортизационных отчислений от стоимости ЭВМ определяется по формуле:

|  |  |
| --- | --- |
|  | (7) |

где  – цена ЭВМ на момент приобретения (руб.);

 – коэффициент удорожания ЭВМ ();

 – норма амортизационных отчислений от стоимости ЭВМ

();

 – балансовая стоимость ЭВМ.



Величина амортизационных отчислений за период разработки программного продукта (38 дней):

Величина амортизационных отчислений от стоимости производственных площадей, занятых ЭВМ определяются по формуле:

|  |  |
| --- | --- |
|  | (8) |

где  – площадь, занимаемая ЭВМ (м2);

 – цена 1 м2 производственной площади ( = 713,95 руб.);

 – норма амортизационных отчислений на производственные площади ();

 – балансовая стоимость площадей, занимаемых ЭВМ.



Величина амортизационных отчислений от стоимости производственных площадей занятых ЭВМ за период разработки программного продукта (38 дней):

К налогам и сборам, включаемым в себестоимость, относится налог на надвижимость. Величина налога на недвижимость определяется по формуле:

 (9)

где  – ставка налога на недвижимость ().

Накладные расходы () – затраты, связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств, а также с расходами на общехозяйственные нужды, вычисляются по формуле:

 (10)

где  – норматив накладных расходов в целом ().

Прочие затраты на конкретное программное средство () включают в себя затраты на приобретение и подготовку специальной литературы и специальной научно-технической информации, и определяются по формуле:

|  |  |
| --- | --- |
|  | (11) |

где  – норматив прочих затрат ().

Общую сумму расходов по всем статьям определим в таблице 3.

Таблица 3 – Калькуляция плановой себестоимости программного продукта

|  |  |
| --- | --- |
| Статья затрат | Сумма, руб. |
| Основная заработная плата |  |
| Дополнительная заработная плата | 75,24 |
| Отчисления на социальные нужды – всего |  |
| - в государственный фонд социальной защиты населения |  |
| - в Белгосстрах |  |
| Материальные затраты – всего | 39,50 |

Продолжение таблицы 3

|  |  |
| --- | --- |
| Статья затрат | Сумма, руб. |
| в том числе:  - материалы | 28,00 |
| - электроэнергия | 11,50 |
| Содержание и эксплуатация основных фондов – всего | 23,79 |
| в том числе:  - амортизационные отчисления от стоимости ЭВМ |  |
| - амортизационные отчисления от стоимости производственных площадей, занимаемых ЭВМ |  |
| Налоги и сборы, включаемые в себестоимость | 28,56 |
| Накладные расходы |  |
| Прочие затраты |  |
| ИТОГО () | 981,06 |

Расчет прогнозируемой прибыли () по разрабатываемому ПС по формуле:

|  |  |
| --- | --- |
|  | (12) |

где  – уровень рентабельности ПС ().

Прогнозируемая цена без налога на добавленную стоимость () складывается из полной себестоимости программного средства и прогнозируемой прибыли. Рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (13) |

Таким образом, подставив значение в данную формулу получается:

На реализацию создаваемого программного средства льготы по налогу на добавленную стоимость (НДС) не распространяются. В связи с этим НДС рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (14) |

где  – ставка налога на добавленную стоимость ().

|  |  |
| --- | --- |
|  |  |

Прогнозируемая отпускная цена () определяется по формуле:

|  |  |
| --- | --- |
|  | (15) |

Таким образом, подставив значение в данную формулу получается:

|  |
| --- |
|  |

Прибыль, остающаяся в распоряжении организации-разработчика, определяется по формуле:

|  |  |
| --- | --- |
|  | (16) |

где – норматив налога на прибыль ().

|  |
| --- |
| 160,98руб. |

Чистая прибыль от реализации ПС остается разработчику и представляет собой экономический эффект от создания нового программного средства.

* 1. **Оценка экономического эффекта от внедрения программного продукта**

Определение экономии времени отражено в таблице 23.

Таблица 13 – Определение экономии времени

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Перечень услуг программного продукта | Время до автоматизации, час. | Время до автоматизации, час. | Экономия времени, час. | Количество операций в год | Годовая экономия времени, час. |
| 1 Ознакомление с историей создания Минского метрополитена |  |  |  |  |  |
| 2 Ознакомление со станциями Минского метрополитена |  |  |  |  |  |
| 3 Возможность оставлять комментарии и отзывы об объектах метрополитена |  |  |  |  |  |
| 4 Ознакомление с новостями о событиях и изменениях в Минском метрополитене |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| Итого: | – | – | – | – |  |

В дипломном проекте было создано, протестировано и подготовлено к эксплуатации веб-приложение «Виртуальный музей минского метрополитена». Данное веб-приложение предоставляет всю необходимую информацию о создании минского метрополитена, его станциях, новостях позволяет просматривать информацию.

Внедрение разработанного программного средства приведёт к условной годовой экономии фонда рабочего времени исполнителя в размере **1513** часа, т.е. снизит трудоёмкость отдельных операций исполнителя. Сэкономить можно приблизительно **0,75** тарифной ставки исполнителя. Использовать данное веб-приложение будут жители Республики Беларусь.

Тогда годовой фонд заработной платы равен:

|  |  |
| --- | --- |
|  | (17) |

Условная экономия по заработной плате составит:

|  |  |
| --- | --- |
|  | (18) |

Вывод: экономический эффект от внедрения веб-приложение «Виртуальный музей минского метрополитена» составит:

|  |  |
| --- | --- |
|  | (19) |

Подводя итог можно сказать, что разработка веб-приложение «Виртуальный музей минского метрополитена» позволит получить экономический эффект от его реализации в сумме **4523,53** рублей.

* 1. **Выводы**

В дипломном проекте было создано, протестировано и подготовлено к эксплуатации веб-приложение «Виртуальный музей минского метрополитена». Данное веб-приложение предоставляет пользователям получить обширную информацию о минском метрополитене – ознакомиться со станциями, ветками и историей развития метрополитена, также приложение предоставляет возможность оставлять отзывы и комментарии о посещенных объектах. Наличие новостей также является важным функционалом приложения, которое позволяет пользователям быть в курсе последних событий, изменений и нововведений по теме метрополитена города Минска. Возможность администрирования этого приложения также является важным функционалом. Администраторы могут контролировать содержимое.

Экономический эффект от реализации веб-приложения «Виртуальный музей минского метрополитена» составляет **4012,45** рублей.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения дипломного проекта была достигнута цель – разработано веб-приложение «Виртуальный музей минского метрополитена», которое представляет собой удобный и интерактивный инструмент для путешественников, желающих более подробно узнать об истории возникновения метрополитена города Минска и новинках станций. В веб-приложении представлены фотографии, видео и информация о станциях минского метрополитена.

Одним из основных приоритетов в разработке приложения было формализация и документирование требований к созданию культурно-исторического веб-приложения. Была проведена работа по определению функциональных и нефункциональных требований и созданию соответствующей документации.

Для достижения задач были применены различные методы исследования, включая анализ литературы, функциональное и нефункциональное тестирование приложения. Была разработана спецификация функциональных требований к приложению и определены нефункциональные требования, такие как интерфейс, масштабируемость и т.д.

Также, результаты выполненных этапов работы приведены в Приложении Г, представленные в виде аннотации.

По результатам выполнения проекта были достигнуты поставленные цели, а именно:

1. обеспечение контроля над данными со стороны администратора;
2. повышение качества и релевантности поиска информации об основании и развитии минского метрополитена;
3. обеспечение бесперебойного доступа к веб-приложению;
4. сокращение времени, необходимого для администрирования по сравнению с стандартными аналогами панелей администратора;
5. обеспечение удобства работы с интерфейсом как с клиентского, так и с администраторского доступа.

Приложение имеет большие перспективы на развитие и может стать ключевым инструментом для туристов, желающих познакомиться с историческими достопримечательностями города.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Веб-форум разработчиков [Электронный ресурс] / Проектирование и разработка корпоративных веб-приложений. – Режим доступа: https://habr.com/ru/post/249863/. – Дата доступа: 11.05.2023
2. Веб сайт дизайнеров интерфейсов [Электронный ресурс] / Проектирование дизайна интерфейсов. – Режим доступа: https://designpub.ru/dive-into-ui-design-d94f011f0ff. – Дата доступа: 11.05.2023
3. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Никсон Робин; Россия. – Санкт-Петербург, 2019. – 212 с.
4. Официальный сайт веб-фреймворка Node.js [Электронный ресурс] / Документация по использованию Node.js. – Режим доступа: https://nodejs.org/ru/docs/. – Дата доступа: 11.05.2023.
5. Официальный сайт веб-библиотеки Express.js [Электронный ресурс] / Документация по использованию пакета Express.js. – Режим доступа: https://expressjs.com/. – Дата доступа: 11.05.2023
6. Официальный сайт веб-библиотеки React [Электронный ресурс] / Документация по использованию React. – Режим доступа: https://ru.reactjs.org/docs/getting-started.html. – Дата доступа: 11.05.2023
7. Официальный сайт веб-библиотеки JWT [Электронный ресурс] / Документация по использованию пакета JWT. – Режим доступа: https://jwt.io/introduction. – Дата доступа: 11.05.2023
8. Официальный сайт веб-библиотеки React [Электронный ресурс] / Документация по использованию React. – Режим доступа: https://ru.reactjs.org/docs/getting-started.html. – Дата доступа: 11.05.2023
9. Сайт Республиканской научно-технической библиотеки [Электронный ресурс] / Типовая инструкция по охране труда при работе с компьютером. – Режим доступа: https://rlst.org.by/informational-resources/izdania/tipovaya-instruktsiya-po-ohrane-truda-pri-rabote-s-kompyuterom/. – Дата доступа: 11.05.2023
10. Официальный сайт министерства труда и социальной защиты Республики Беларусь библиотеки [Электронный ресурс] / Постановление об утверждении типовой инструкции по охране труда при работе с персональными электронно-вычислительными машинами. – Режим доступа: https://www.mintrud.gov.by/system/extensions/spaw/uploads/files/ POSTANOVLENIE-130.pdf. – Дата доступа: 11.05.2023
11. Охрана труда в учреждениях образования: пособие для слушателей учреждений, обеспечивающих дополнительное образование взрослых: методические рекомендации / В. Н. Петраков; Беларусь. – Минск, 2022. – 133 с.
12. ТКП 181-2009 (02230) Правила технической эксплуатации электроустановок потребителей / Постановление Совета Министров Респ. Беларусь; Беларусь. – Минск, 2014. – 317 с.
13. Портал статей и рефератов [Электронный ресурс] / Методика технико-экономического обоснования разработки программного продукта. – Режим доступа: https://studref.com/641706/ekonomika/metodika\_tehniko\_ekonomicheskogo\_obosnovaniya\_razrabotki\_programmnogo\_produkta. – Дата доступа: 11.05.2023.
14. Экономическая теория для бакалавров / Носова С.С., Новичкова В.И.; Россия. – Москва, 2015. – 243 с.
15. Основы экономики: учебное пособие / Борисов Е.Ф.; Россия. – Москва, 2010. – 183 с.