МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломной работе на тему**:**

Веб-приложение «Туристическое агентство»

Дипломник Тихонович Максим Александрович

(Ф.И.О.)

Руководитель работы ст.преп. Панченко О.Л.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Заведующий кафедрой к.т.н., доц., Пацей Н.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Консультанты: асс. Романенко О.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

ст.преп. Козловский В.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Нормоконтролер асс. Жигаровская С.А.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Дипломная работа защищена с оценкой

Председатель ГЭК к.т.н., доц. Дюбков В.К.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Минск 2020

Учреждение образования «БЕЛОРУССКИЙ

ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра1 Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-приложений)

УТВЕРЖДАЮ

Заведующий кафедрой ПИ

\_\_\_\_\_\_\_\_\_\_\_\_\_ Пацей Н.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**ЗАДАНИЕ**

**на дипломный проект (работу) студенту**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Тихоновичу Максиму Александровичу\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

1. Тема проекта (работы):Веб-приложение «Туристическое агентство»

утверждена приказом по университету от 27 февраля 2020 г. № 45-С

2. Срок сдачи законченного дипломного проекта: 5 июня 2020 г.

3. Исходные данные к проекту (работе)

*– клиент имеет возможность оформить заказ тура;*

*– клиент имеет возможность просмотра заказанных им туров;*

*– клиент может оставить отзыв о приобретенном им ранее туре;*

*– администратор имеет возможность создания и удаления туров и отелей;*

*– администратор может принять или удалить комментарий пользователя.*

4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов):

*1) реферат;*

*2) содержание;*

*3) введение;*

*4) раздел 1: аналитический обзор литературы и постановка задачи;*

*5) раздел 2: проектирование веб-приложения;*

*6) раздел 3: разработка веб-приложения;*

*7) раздел 4: тестирование веб-приложения;*

*8) раздел 5: методика установки и использования;*

*9) раздел 6: технико-экономическое обоснование проекта;*

*10) заключение;*

*11) список использованных источников;*

*12) приложения и графическая часть;*

*13) справка о внедрении.*

5. Перечень графического материала (с точным указанием обязательных чертежей):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*1) скриншоты работы программы;*

*2) блок-схема создания тура;*

*3) диаграмма вариантов использования;*

*4) логическая схема базы данных;*

**Реферат**

Пояснительная записка содержит N страниц, N рисунков, N таблиц,   
N источников, N приложений.

ВЕБ-ПРИЛОЖЕНИЕ «ТУРИСТИЧЕСКОЕ АГЕТСТВО», ЯЗЫК ПРОГРАММИРОВАНИЯ C#, ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ VISUAL STUDIO, ТЕКСТОВЫЙ РЕДАКТОР КОДА VISUAL STUDIO CODE

Основной целью дипломного проекта является разработка веб-приложения «Туристическое агентство».

Пояснительная записка состоит из введения, пяти разделов и заключения.

* первом разделе описаны цель и задачи дипломного проекта, обзор аналогов и обзор средств разработки.

Во втором разделе представлены архитектура приложения, проектирование и структура таблиц базы данных.

Третий раздел посвящен разработке программного средства.

* четвертом разделе приведено руководство пользователя.
* пятом разделе приводится расчёт экономических параметров и себестоимость программного продукта.
* заключении представлены итоги дипломного проекта и задачи, которые были решены в ходе разработки программного средства.

У

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*8*

*ДП 00.00.ПЗ*

*Разраб.*

*Тихонович М.А.*

*Провер.*

*.*

*Панченко О.Л.*

Консульт.

*Панченко О.Л.*

*Н. Контр.*

*Утверд.*

*Пацей Н.В.*

Реферат

*Лит.*

*Листов*

*1*

БГТУ 74420054, 2018

СОДЕРЖАНИЕ

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*8*

*ДП 00.00.ПЗ*

*Разраб.*

*Тихонович М.А.*

*Провер.*

*.*

*Панченко О.Л.*

Консульт.

*Панченко О.Л.*

*Н. Контр.*

*Утверд.*

*Пацей Н.В.*

Содержание

*Лит.*

*Листов*

*2*

БГТУ 74420054, 2018

[**ВВЕДЕНИЕ** 7](#_Toc36835318)

[**1 Постановка задачи и обзор аналогов** 8](#_Toc36835319)

[1.1 Основные технические требования 9](#_Toc36835322)

[1.2 Обзор аналогов 9](#_Toc36835323)

[1.3 Обзор средств разработки 10](#_Toc36835324)

[1.3.1 Выбор языка программирования 10](#_Toc36835325)

[1.3.2 Выбор средства программирования 10](#_Toc36835326)

[1.3.3 Выбор платформы разработки 12](#_Toc36835327)

[1.4 Обзор используемых библиотек 13](#_Toc36835328)

[1.4.1 Библиотека Entity Framework Core для решения задач объектно-реляционного отображения 13](#_Toc36835329)

[1.4.2 Библиотека React для разработки пользовательских интерфейсов 14](#_Toc36835330)

[1.5 Вывод по разделу 15](#_Toc36835331)

[**2 Проектирование** 16](#_Toc36835332)

[2.1 Архитектура приложения 16](#_Toc36835333)

[2.2 Проектирование базы данных 16](#_Toc36835334)

[2.3 Вывод по разделу 17](#_Toc36835335)

[**2.4 Вывод по разделу** 18](#_Toc36835336)

[**3 Разработка программного средства** 19](#_Toc36835337)

[3.1 Реализация серверной части приложения 19](#_Toc36835338)

[3.2 Разработка диаграммы прецедентов 19](#_Toc36835339)

[**4 Руководство пользователя** 21](#_Toc36835340)

[4.1 Демонстрационное мобильное приложение 21](#_Toc36835341)

[**5 Экономический раздел** 22](#_Toc36835342)

[**5.1 Общая характеристика разрабатываемого программного средства** 22](#_Toc36835343)

[5.2 Исходные данные для проведения расчетов 22](#_Toc36835344)

[5.3 Методика обоснования цены 23](#_Toc36835345)

[5.4 Определение объема программного средства 23](#_Toc36835346)

[5.5 Расчет трудоемкости выполняемой работы 24](#_Toc36835347)

[5.6 Расчет основной заработной платы 25](#_Toc36835348)

[5.7 Расчет дополнительной заработной платы 25](#_Toc36835349)

[5.8 Расчет отчислений на социальные цели 26](#_Toc36835350)

[5.9 Расчет расходов на материалы 26](#_Toc36835351)

[5.10 Расчет расходов на оплату машинного времени 27](#_Toc36835352)

[5.11 Расчет прочих прямых затрат 27](#_Toc36835353)

[5.12 Расчет общепроизводственных и общехозяйственных расходов 27](#_Toc36835354)

[5.13 Расчет суммы расходов на разработку ПС 27](#_Toc36835355)

[5.14 Расчет расходов на сопровождение и адаптацию 28](#_Toc36835356)

[5.15 Расчет общей суммы расходов 28](#_Toc36835357)

[5.16 Определение цены и оценка эффективности 28](#_Toc36835358)

[**5.17 Вывод по разделу** 29](#_Toc36835359)

[ЗАКЛЮЧЕНИЕ 30](#_Toc36835360)

[**Список использованных источников** 31](#_Toc36835361)

# **ВВЕДЕНИЕ**

Туризм – одна из самых перспективных и быстроразвивающихся индустрий. По данным Всемирной Туристической организации в настоящие время туристская индустрия является одним из самых динамичных секторов мировой экономики. По уровню доходности туризм уступает лишь индустрии добычи нефти и автостроению.

Важнейшим направлением деятельности фирм, работающих в туристической сфере, являются продвижение турпродукта на рынок, реклама и реализация подготовленного турпакета.

В настоящее время средства массовой информации, специальные издания, рекламные проспекты буквально переполнены разнообразными туристскими предложениями, и путь турфирмы к успеху заключается в том, чтобы довести до потенциального клиента соответствующую информацию и вызвать его ответные действия.

Туризм сегодня – это глобальный компьютеризированный бизнес, в котором участвуют крупнейшие авиакомпании, гостиничные цепочки и туристические корпорации всего мира. Современный турпродукт становится более гибким и индивидуальным, более привлекательным и доступным для потребителя.

Целью данного дипломного проекта является повышение эффективности работы туристического агентства. Для достижения поставленной цели необходимо:

* исследовать преимущества и недостатки аналогичных приложений;
* разработать структуру базы данных;
* реализовать веб-приложение;
* рассчитать экономические показатели.
* сделать выводы об экономической целесообразности использования программного продукта.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*8*

*ДП 00.00.ПЗ*

*Разраб.*

*Тихонович М.А.*

*Провер.*

*.*

*Панченко О.Л.*

Консульт.

*Панченко О.Л.*

*Н. Контр.*

*Утверд.*

*Пацей Н.В.*

ВВЕДЕНИЕ

*Лит.*

*Листов*

*1*

БГТУ 74420054, 2018

# **1 Постановка задачи и обзор аналогов**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*8*

*ДП 01.00.ПЗ*

*Разраб.*

*Тихонович М.А.*

*Провер.*

*.*

*Панченко О.Л.*

Консульт.

*Панченко О.Л.*

*Н. Контр.*

*Утверд.*

*Пацей Н.В.*

Постановка задачи и обзор аналогов

*Лит.*

*Листов*

*9*

БГТУ 74420054, 2018

## Роль туризма в мировой практике постоянно возрастает. Будучи одной из крупнейших, высокодоходных и наиболее динамичных отраслей, уступая по доходности лишь добыче и переработке нефти, туризм, по данным ЮНВТО, обеспечивает 10% оборота производственно-сервисного рынка планеты. На сферу туризма приходится 6% мирового ВНП, 7% мировых инвестиций, каждое 16-е рабочее место, 11% мировых потребительских расходов, 5% всех налоговых поступлений.

## Следовательно, туризм, испытывающий на себе влияние ряда внешних факторов, сам способен оказывать значительное воздействие на экономическую, политическую, экологическую, социально-культурную ситуацию тех стран и регионов, где он развивается. Причем это воздействие может носить как позитивный, так и негативный характер.

В последнее время в нашей стране стал необычайно актуален вопрос туризма. Связано это с улучшением благосостояния граждан, с расширением открытости границ и стремительным развитием туристического бизнеса в целом. В любой момент человек может обратиться в соответствующую фирму или компанию и заказать тур для поездки в далекие экзотические страны.

Выбор курортов сегодня просто огромен – от далеких таиландских и арабских, до украинских, российских, чешских и других. Вопрос, куда же отправиться, сейчас стоит очень остро, так как в большом количестве предложений легко можно запутаться.

Моё приложение поможет Вам забронировать наиболее подходящие место для туризма. На сайте собрано достаточное количество информации, касающейся различных видов туризма, мест, где можно прекрасно отдохнуть, а также другие данные, обладание которыми сильно поможет Вам, если Вы соберетесь куда-либо отправиться.

Основной задачей дипломного проекта является разработка веб-приложения «Туристическое агентство». Данное приложение должно предоставлять удобный интерфейс пользователю для осуществления online-бронирования туристических билетов.

## 1.1 Основные технические требования

Целью данного дипломного проекта является разработка программного средства, представляющего собой web-приложение. Обеспечивающие удобное бронирование билетов для пассажиров и работу для администраторов, необходимо реализовать серверную и клиентскую часть приложения.

Frontend часть приложения должна:

* обладать удобным интерфейсом;
* предоставлять информацию о турах;
* быть легко поддерживаемым.

Backend часть приложения должна:

* обеспечивать хранение данных;
* основываться на REST-архитектуре;
* обеспечивать легкую поддержку.

## Обзор аналогов

### 1.2.1 Приложение LoveTour.by

В качестве одного из прототипов приложения был выбран сайт LoveTour.by.

Интерфейс приложения представлен на рисунке 1.1 [1].

Данный сайт предназначен для ознакомления с возможными турами для последующего бронирования посредством мобильной связи. У него есть возможность выбора автобусного тура или авиатура. В категории спец. предложения можно ознакомиться с уникальными предложениями от туроператора.

Возможности онлайн бронирования данный сайт не предоставляет.

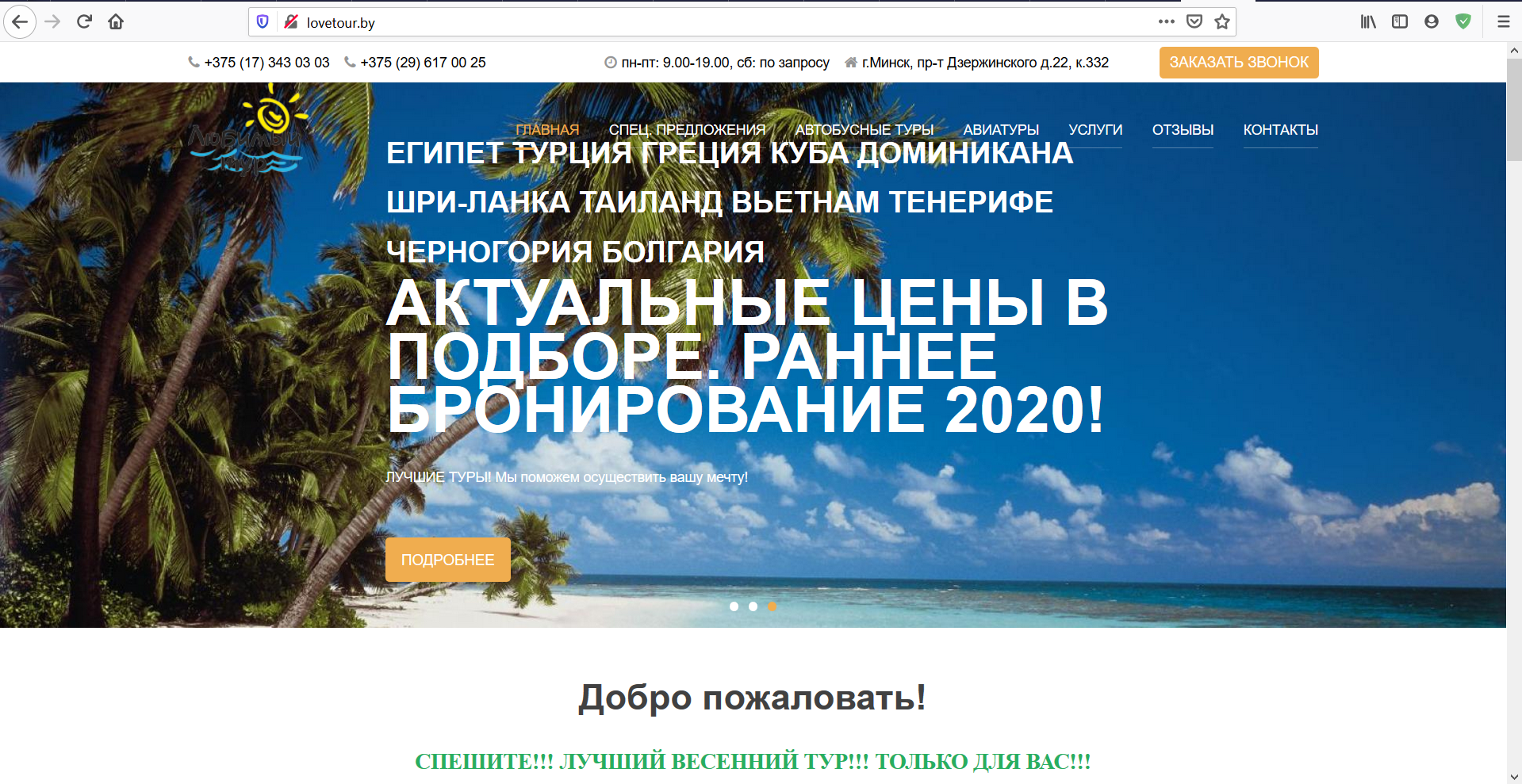


Рисунок 1.1 – Сайт LoveTour.by

Сайт предоставляет удобный интерфейс для взаимодействия с пользователем.

Возможность регистрации аккаунтов не предусмотрена.

### 1.2.2 Приложение Magellan.by

Интерфейс приложения представлен на рисунке 1.2 [2]

Возможности:

* поиск тура между выбранными странами с выбором периода вылета, количеством взрослых и детей, периодом ночей;
* возможность добавления в избранное понравившихся туров;
* наличие online-консультанта;
* услуга «Перезвони мне». Вы указываете в специальной форме свое имя и номер телефона.

Существенным недостатком является не адаптивность данного сайта. Так как в современном мире количество запросов с мобильных устройств преобладает над количеством запросов с персональных компьютеров, очень важно правильное отображение сайта на различных устройствах, динамически подстраивающийся под заданные размеры окна браузера.

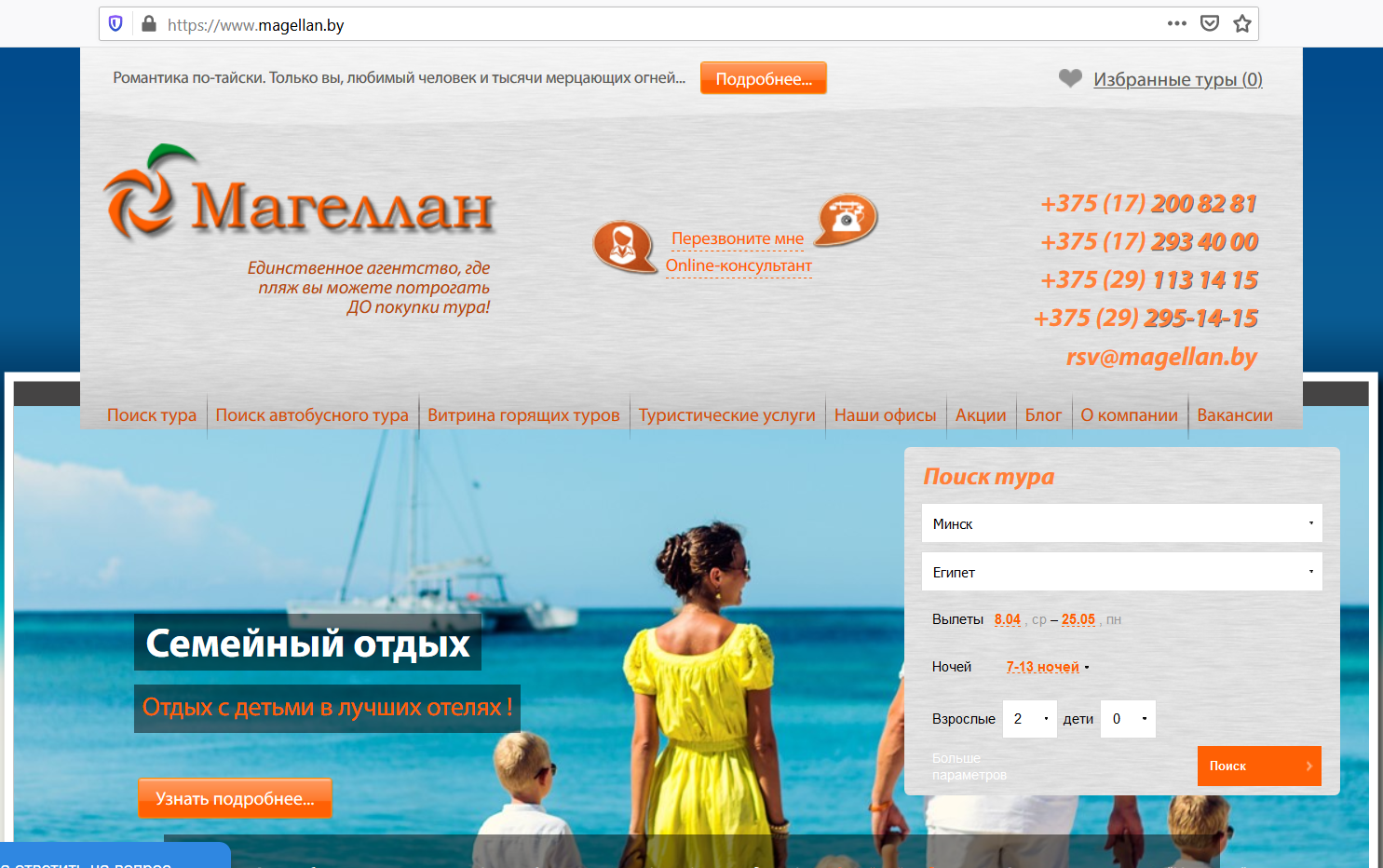


Рисунок 1.2 – Сайт Magellan.by

Приложение имеет ряд ошибок и недостатков.

### 1.2.3 Приложение PeopleTravel.by

Судя по отзывам на сайте, данный сервис имеет хорошую репутацию среди пользователей. Данный сайт предоставляет удобный пользовательский интерфейс и представлен на рисунке 1.3 [3].



Рисунок 1.3 – Сайт PeopleTravel.by

Возможности:

* оставить заявку на подбор тура;
* возможность выбора экскурсионного, авиа или круизного тура;
* наличие туристического календаря;
* возможность получения рассрочки без посещения офиса;
* есть возможность сравнения туров;
* найти ответы на часто задаваемые вопросы.

Существенным недостатком данного приложения является отсутствие возможности бронирования тура онлайн. Присутствуют незначительные баги с отображением информации.

### Обзор средств разработки

В данном разделе будут рассмотрены различные библиотеки, технологии и подходы, которые будут использованы в разработке приложения.

### Выбор языка программирования

Для разработки программного средства был выбран язык программирования C#. В этом подразделе указаны основные возможности, достоинства и недостатки языка, а также его сравнение с другими языками программирования.

C# – объектно-ориентированный язык программирования. Разработан в 1998-2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[4] как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает FCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем. (Однако эта закономерность была нарушена с выходом C# 3.0, представляющим собой расширения языка, не опирающиеся на расширения платформы .NET.) CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

Свой синтаксис C# во многом унаследовал от C++ и Java. Но вместе с тем он является во многом новаторским – атрибуты, делегаты и события, вписанные в общую идеологию языка, прочно заняли место в сердцах .NET - разработчиков. Их введение позволило применять принципиально новые приемы программирования

### Выбор средства программирования

Интегрированные среды разработки, среды разработки (Integrated Development Environment – IDE) – совокупность программных средств, с помощью которых программистами производится написание, корректировка, преобразование в машинные коды, наладка и запуск различного программного обеспечения. Интегрированные среды разработки обычно содержат:

* текстовый редактор, предназначенный для ввода, подсвечивания и корректировки текста программы;
* компилятор, с помощью которого программа переводится с языка, на котором она написана, в машинные коды;
* средства отладки и запуска программ;
* общие библиотеки, которые обычно содержат многократно используемые элементы программ;
* справочную систему и другие элементы.

Интегрированная среда разработки Microsoft Visual Studio 2019 – это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию. Эти инструменты предназначены для максимально эффективной совместной работы. Все они доступны в интегрированной среде разработки Visual Studio.

Visual Studio можно использовать для создания различных типов приложений, от простых приложений для магазина и игр для мобильных клиентов до больших и сложных систем, обслуживающих предприятия и центры обработки данных. Вы можете создавать:

* приложения и игры, которые выполняются не только на платформе Windows, но и на Android и iOS;
* веб-сайты и веб-службы на основе ASP.NET, JQuery, AngularJS и других популярных платформ;
* приложения для самых разных платформ и устройств, включая, но не ограничиваясь: Office, Sharepoint, Hololens, Kinect и "Интернета вещей";
* игры и графические приложения для разных устройств Windows, включая Xbox, с поддержкой DirectX.

По умолчанию Visual Studio обеспечивает поддержку C#, C и C++, JavaScript, F# и Visual Basic. Visual Studio хорошо работает и интегрируется со сторонними приложениями, например, Unity и Apache Cordova, с помощью расширений и средств Visual Studio для Unity и инструментов Visual Studio для Apache Cordova соответственно. Вы также можете самостоятельно расширить Visual Studio, создав собственные инструменты для выполнения специализированных задач.

Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых компанией Microsoft.

Интерфейс программы состоит:

* меню файл, правка, поиск и тд;
* область выбора файлов проекта для редактирования;
* окна, появляющиеся в режиме отладки: Call Stack, Output, Breakpoints, Watch windows.
* список открытых файлов;
* область выбора файлов проекта для редактирования;
* область написания кода.

Изначально, без плагинов и тем оформлений, интерфейс программы довольно прост и стандартен. Ничего лишнего для начинающего разработчика: подсветка синтаксиса кода в области редактирования, каталог директорий проекта.

Вышеперечисленные возможности этой среды – плагины, удобный интерфейс, хороший отладчик, удобный и быстрый редактор кода, делают Visual Studio одним из лучших средств для разработки программного обеспечения. При этом стоит отметить, что я не перечислил и сотой доли тех возможностей, которые Visual Studio в себя включает.

### Выбор платформы разработки

.NET Core – это открытая универсальная модульная платформа разработки, которая поддерживается корпорацией Майкрософт и сообществом .NET на сайте GitHub. Она является кроссплатформенной, поддерживает Windows, Mac OS и Linux и может использоваться на устройствах, в облаке, во внедренных системах и в сценариях IoT (Интернета вещей). В её основе лежат технологии .NET Framework и Silverlight. Она оптимизирована для мобильных и серверных рабочих нагрузок, поскольку обеспечивает поддержку самодостаточных развёртываний XCOPY[5].

.NET Core обладает следующими характеристиками:

* **кроссплатформенность;**
* согласованность между архитектурами;
* **открытый исходный код;**
* **гибкая разработка;**
* **программы командной строки;**
* **совместимость;**
* **поддержка от Microsoft.**

.NET Core состоит из перечисленных ниже компонентов:

* среда выполнения .NET Core – предоставляет систему типов, функции загрузки сборок, сборщик мусора, собственные функции взаимодействия и другие базовые службы. Библиотеки платформы .NET Core предоставляют примитивные типы данных, типы компоновки приложений и базовые служебные программы;
* среда выполнения ASP.NET Core – это платформа для создания современных облачных приложений, подключенных к сети Интернет: веб-приложений, приложений IoT, серверной части мобильных решений и многого другого;
* .NET Core CLI и компиляторы языков (Roslyn и F#) реализуют возможности разработки .NET Core.

## 1.4 Обзор используемых технологий и библиотек

### 1.4.1 Технология Entity Framework Core для решения задач объектно-реляционного отображения

Entity Framework Core (EF Core) представляет собой объектно-ориентированную, легковесную и расширяемую технологию от компании Microsoft для доступа к данным. EF Core является ORM-инструментом (object-relational mapping – отображения данных на реальные объекты). То есть EF Core позволяет работать c базами данных, но представляет собой более высокий уровень абстракции: EF Core позволяет абстрагироваться от самой базы данных и ее таблиц и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работаем с объектами.

Entity Framework Core поддерживает множество различных систем баз данных. Таким образом, мы можем через EF Core работать с любой СУБД, если для нее имеется нужный провайдер.

По умолчанию на данный момент Microsoft предоставляет ряд встроенных провайдеров: для работы с MS SQL Server, для SQLite, для PostgreSQL. Также имеются провайдеры от сторонних поставщиков, например, для MySQL.

Также стоит отметить, что EF Core предоставляет универсальный API для работы с данными. И если, к примеру, мы решим сменить целевую СУБД, то основные изменения в проекте будут касаться прежде всего конфигурации и настройки подключения к соответствующим провайдерам. А код, который непосредственно работает с данными, получает данные, добавляет их в БД и т.д., останется прежним.

Как технология доступа к данным Entity Framework Core может использоваться на различных платформах стека .NET. Это и стандартные платформы типа Windows Forms, консольные приложения, WPF, UWP и ASP.NET Core. При этом кроссплатформенная природа EF Core позволяет задействовать ее не только на ОС Windows, но и на Linux и Mac OS X.

Центральной концепцией Entity Framework является понятие сущности или entity. Сущность определяет набор данных, которые связаны с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их коллекциями.

Любая сущность, как и любой объект из реального мира, обладает рядом свойств. Например, если сущность описывает человека, то мы можем выделить такие свойства, как имя, фамилия, рост, возраст. Свойства необязательно представляют простые данные типа int или string, но могут также представлять и более комплексные типы данных. И у каждой сущности может быть одно или несколько свойств, которые будут отличать эту сущность от других и будут уникально определять эту сущность. Подобные свойства называют ключами.

При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Отличительной чертой Entity Framework Core, как технологии ORM, является использование запросов LINQ для выборки данных из БД. С помощью LINQ мы можем создавать различные запросы на выборку объектов, в том числе связанных различными ассоциативными связями. А Entity Framework при выполнении запроса транслирует выражения LINQ в выражения, понятные для конкретной СУБД (как правило, в выражения SQL).

### 1.4.2 Библиотека React для разработки пользовательских интерфейсов

React – популярная библиотека, используемая для создания пользовательских интерфейсов. Она была создана в Facebook с целью решить ряд проблем, связанных с крупномасштабными сайтами, управляемыми данными.

Главная задача React – обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты. Данные фрагменты называются компонентами.

Компонент React – это участок кода, который представляет часть веб-страницы. Каждый компонент – это JavaScript-функция, которая возвращает кусок кода, представляющего фрагмент страницы.

Для формирования страницы мы вызываем эти функции в определённом порядке, собираем вместе результаты вызовов и показываем их пользователю.

Когда мы пытаемся манипулировать html-элементами с помощью JavaScript, то мы можем столкнуться со снижением производительности, особенно при изменении большого количества элементов. А операции над элементами могут занять некоторое время, что неизбежно скажется на пользовательском опыте. Для решения проблемы производительности появилась концепция виртуального DOM.

Виртуальный DOM (VDOM) – это концепция программирования, в которой идеальное или «виртуальное» представление пользовательского интерфейса хранится в памяти и синхронизируется с «реальным» DOM с помощью библиотеки, такой как ReactDOM.

Данный процесс называется согласованием. Данный подход обеспечивается декларативным React API: вы говорите React, в каком состоянии должен находиться UI, и он гарантирует, что DOM соответствует этому состоянию. Это абстрагирует манипуляции с атрибутами, обработку событий и ручное обновление DOM, которые в противном случае вам пришлось бы использовать для создания приложения.

Еще одна особенность React – использование JSX. JSX представляет комбинацию кода JavaScript и XML и предоставляет простой и интуитивно понятный способ для определения кода визуального интерфейса.

Стоит отметить, что React развивается как открытый проект, и весь исходный код библиотеки доступен на веб-сервисе GitHub, который предназначен для хостинга IT-проектов.

## 1.5 Вывод по разделу

Развитие туризма является одним из приоритетных направлений социально-экономического развития Республики Беларусь.

В связи с этим было решено разработать веб-приложение, предоставляющие удобное бронирование билетов для пользователей и работу с клиентами для администраторов.

Необходимо реализовать серверную и клиентскую часть приложения.

Клиентская часть приложения должна: обладать удобным интерфейсом, предоставлять информацию о турах, быть легко поддерживаемым.

Серверная часть приложения должна: обеспечивать хранение данных, основываться на REST-архитектуре, обеспечивать легкую поддержку [16].

В качестве аналогов разрабатываемого приложения были рассмотрены следующие веб-приложения: LoveTour.by, Magellan.by и PeopleTravel.by. Были выбраны следующие средства разработки: Visual Studio 2019 для разработки серверной части приложения, Visual Studio Code для клиетской части приложения.

В процессе разработки были использованы:

* свободно-распространяемый кроссплатформенный фреймворк для создания веб-приложений ASP.NET Core;
* объектно-ориентированная технология доступа к данным Entity Framework Core;
* JavaScript-библиотека для разработки пользовательских интерфейсов React.

# **2 Проектирование**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*16*

*ДП 02.00.ПЗ*

*Разраб.*

*Тихонович М.А.*

*Провер.*

*.*

*Панченко О.Л*

Консульт.

*Панченко О.Л*

*Н. Контр.*

*Утверд.*

*Пацей Н..В.*

Проектирование

*Лит.*

*Листов*

*9*

БГТУ 74420054, 2018

## 2.1 Архитектура приложения

Архитектура программного средства – это его строение как оно видно (или должно быть видно) извне его, т.е. представление программного средства как системы, состоящей из некоторой совокупности взаимодействующих подсистем. В качестве таких подсистем выступают обычно отдельные программы. Разработка архитектуры является первым этапом борьбы со сложностью программного средства, на котором реализуется принцип выделения относительно независимых компонент.

Основные задачи разработки архитектуры программного средства (ПС):

* выделение программных подсистем и отображение на них внешних функций (заданных во внешнем описании) ПС;
* определение способов взаимодействия между выделенными программными подсистемами.

С учетом принимаемых на этом этапе решений производится дальнейшая конкретизация функциональных спецификаций.

Архитектура программного средства представлена на рисунке 2.1.

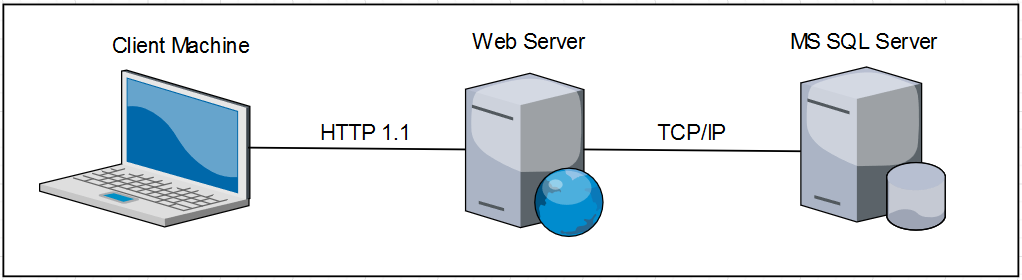


Рисунок 2.1 – Архитектура программного средства «Туристическое агентство»

Как видно на рисунке в главной роли, по обработке всех данных, а также в формировании и передаче их по удаленным каналам отвечает Web Server. Клиент посылает запросы на сервер по HTTP протоколу версии 1.1 [17]. Все процессы, отвечающие за хранение данных возложены на MS SQL Server [18].

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Конечно, если разрабатывается простая программа, которая может выполняться локально на компьютере пользователя, не задействуя никаких периферийных устройств и ресурсов, то в этом случае нет необходимости в разработке дополнительных диаграмм.

Во-первых, сложные программные системы могут реализовываться в сетевом варианте на различных вычислительных платформах и технологиях доступа к распределенным базам данных.

Во-вторых, интеграция программной системы с Интернетом определяет необходимость решения дополнительных вопросов при проектировании системы, таких как обеспечение безопасности, криптозащищенности и устойчивости доступа к информации для корпоративных клиентов. Эти аспекты в немалой степени зависят от реализации проекта в форме физически существующих узлов системы, таких как серверы, рабочие станции, брандмауэры, каналы связи и хранилища данных.

Наконец, технологии доступа и манипулирования данными в рамках общей схемы «клиент-сервер» также требуют размещения больших баз данных в различных сегментах корпоративной сети, их резервного копирования, архивирования, кэширования для обеспечения необходимой производительности системы в целом. Эти аспекты также требуют визуального представления с целью спецификации программных и технологических особенностей реализации распределенных архитектур.

На рисунке 2.2 представлена диаграмма развертывания.

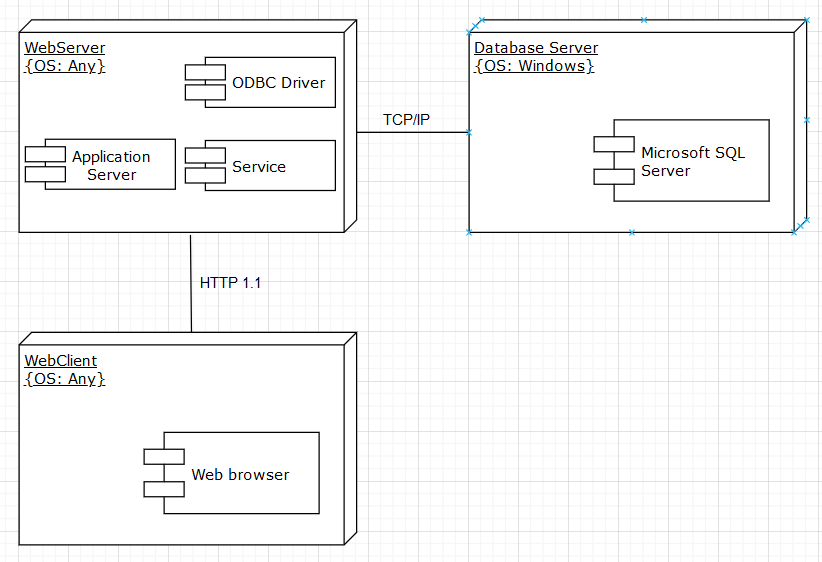


Рисунок 2.2 – Диаграмма развертывания «CoTravel»

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения. При этом представляются только компоненты-экземпляры программы, являющиеся исполнимыми файлами или динамическими библиотеками.

Данная диаграмма более детально описывает архитектуру программного средства «Trav», указывая какие технологии и библиотеки, в каких модулях используются.

## 2.2 Проектирование базы данных

База данных является важнейшей частью приложения, так как хранит в себе все данные. К проектированию базы данных требуется подходить с умом, так как в дальнейшем вносить изменения в структуру, будет сложно. Для взаимодействия с базой данных приложению необходим создавать подключение к базе данных.

Использование ORM позволит ускорить процесс разработки, улучшить читаемость кода и избавиться от «чистых» SQL-запросов в коде приложения. Исключение использования «чистых» SQL-запросов из приложения позволяет повысить безопасность приложения. Это достигается за счёт того, что исключает возможность внедрения SQL-инъекции в запущенное приложение. Под SQL-инъекцией понимается возможность получения сервером «чистого» SQL-запроса, который, например, может удалить все данные из таблицы.

Для взаимодействия с базой данных было решено выбрать ORM Entity Framework Core.

Целью Entity Framework Core является освобождение разработчика от значительного объёма сравнительно низкоуровневого программирования при работе в объектно-ориентированных средствах в реляционной базе данных. Разработчик может использовать Entity Framework Core как в процессе проектирования системы классов и таблиц «с нуля», так и для работы с уже существующей [базой данных.](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)

Изначально таблицы проектируемой базы данных будут пустыми, что не позволит пользователям незамедлительно начать работу. Чтобы этого избежать, требуется произвести первоначальную инициализацию.

Для взаимодействия с базой данных для ORM требуется описать сущности базы данных. Каждая сущность представляет собой одну таблицу базы данных, является её отражением на приложение, тем самым позволяя взаимодействовать с данными, хранимыми в базе данных.

На рисунке 2.3 представлен список сущностей приложения, которые располагаются в пакете «Entities» проекта TravelAgency.DAL.

Все сущности, представленные ниже, будут задействованы в приложении.

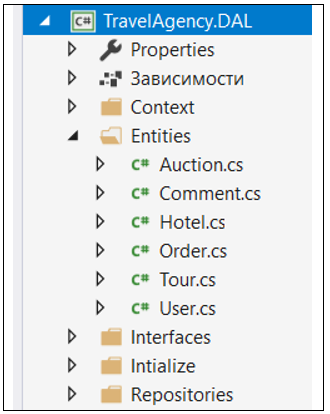


Рисунок 2.3 – Список моделей

На рисунке 2.4 приведена схема базы данных, где видны все связи между различными таблицами.

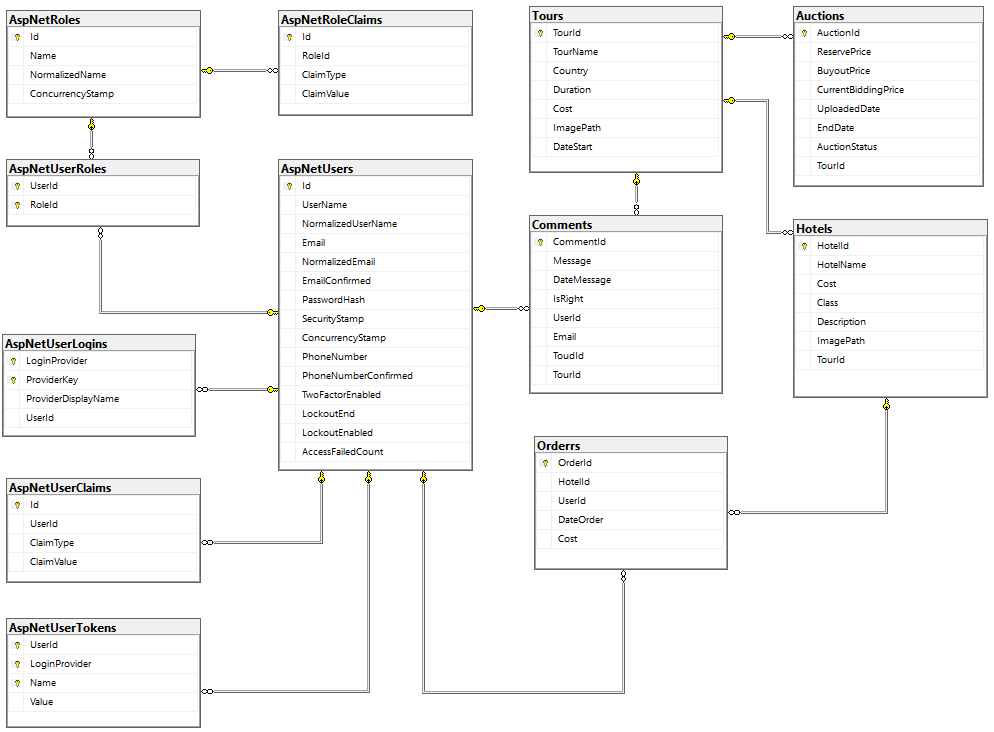


Рисунок 2.4 – Схема базы данных

Таблица Tours содержит информацию о турах, созданных администратором. Перечень полей таблицы Tours приведен в таблице 2.1.

Таблица 2.1 – Описание полей таблицы Tours

|  |  |
| --- | --- |
| Поле | Типы данных, ограничение целостности |
| TourId | Int, identity primary key |
| TourName | Nvarchar(70) not null |
| Country | Nvarchar(70) not null |
| Duration | Int not null |
| Cost | Int not null |
| ImagePath | Nvarchar(max) null |
| DateStart | Datetime2(7) not null |

Свойство TourId отвечает за идентификацию тура.

Свойство TourName хранит наименование тура.

Свойство Country хранит название страны тура.

Свойство Duration хранит информацию о количестве дней тура.

Свойство Cost хранит в себе информацию о цене предоставляемого тура.

Свойство ImagePath может содержать закодированное изображение тура в формате base64.

Свойство DateStart содержит информацию о начале тура.

Таблица Hotels содержит информацию о отелях, созданных администратором. Для одного тура у администратора имеется возможность создания нескольких отелей. Перечень полей таблицы Hotel приведен в таблице 2.2.

Таблица 2.2 – Описание полей таблицы Hotels

|  |  |
| --- | --- |
| Поле | Типы данных, ограничение целостности |
| HotelId | Int, identity primary key |
| HotelName | Nvarchar(50) not null |
| Cost | Int not null |
| Class | Int not null |
| Description | Nvarchar(500) not null |
| ImagePath | Nvarchar(max) null |
| TourId | Int, foreign key not null |

Свойство HotelId отвечает за идентификацию отеля.

Свойство HotelName отвечает хранит наименование отеля.

Свойство Cost хранит в себе информацию о цене отеля.

Свойство Class определяет количество звезд отеля.

Свойство Description позволяет дать развернутое описание отеля.

Свойство ImagePath может содержать закодированное изображение отеля в формате base64.

Свойство TourId внешний ключ, указывающий на тур.

Таблица Orderrs содержит информацию о заказах пользователей. Перечень полей таблицы приведен ниже в таблице 2.3.

Таблица 2.3 – Описание полей таблицы Orderrs

|  |  |
| --- | --- |
| Поле | Типы данных, ограничение целостности |
| OrderId | Int, identity primary key |
| HotelId | Int, foreign key not null |
| UserId | Nvarchar(450), foreign key not null |
| DateOrder | Datetime2(7) not null |
| Cost | Int not null |

Свойство OrderId отвечает за идентификацию заказа.

Свойство HotelId внешний ключ, указывающий на Hotels.

Свойство UserId внешний ключ, указывающий на AspNetUsers.

Свойство DateOrder хранит дату бронирования тура.

Свойство Cost содержит итоговую сумму стоимости отеля и тура.

Таблица Comments содержит комментарии к туру, оставленные пользователями. Перечень полей таблицы приведен ниже в таблице 2.4.

Таблица 2.4 – Описание полей таблицы Comments

|  |  |
| --- | --- |
| Поле | Типы данных, ограничение целостности |
| CommentId |  |
| Message | Nvarchar(max) not null |
| DateMessage | Datetime2(7) not null |
| IsRight | Bit not null |
| UserId | Nvarchar(450), foreign key |
| Email | Nvarchar(max) |
| TourId | Int, foreign key |

Свойство CommentId отвечает за идентификацию комментария.

Свойство Message хранит комментарий, оставленный пользователем.

Свойство DateMessage хранит дату создания комментария.

Свойство IsRight определяет будут ли пользователи видеть комментарий. Для отображения комментария администратор должен его подтвердить.

UserId внешний ключ, указывающий на пользователя.

Свойство Email – хранит логин пользователя, оставившего комментарий.

Свойство TourId внешний ключ, указывающий на тур.

Таблица Auctions содержит информацию о проводимых аукционах на тур. Перечень полей таблицы приведен ниже в таблице 2.4.

Таблица 2.5 –Описание полей таблицы Auctions

|  |  |
| --- | --- |
| Поле | Типы данных, ограничение целостности |
| AuctionId | Int, identity primary key |
| ReservePrice | Int not null |
| BuyoutPrice | Int not null |
| CurrentBiddingPrice | Int not null |
| UploadedDate | Datetime2(7) not null |
| AuctionStatus | Int not null |
| EndDate | Datetime2(7) not null |
| TourId | Int, foreign key |

Свойство AuctionId отвечает за идентификацию аукциона.

Свойство ReservePrice хранит стартовую цену.

Свойство BuyoutPrice хранит цену выкупа

Свойство CurrentBiddingPrice хранит цену на текущий момент.

Свойство UploadedDate хранит дату объявления аукциона.

Свойство AuctionStatus хранит текущее состояние аукциона.

Свойство EndDate хранит дату завершения аукциона.

Свойство TourId внешний ключ, указывающий на тур.

## 2.3 Вывод по разделу

Программный продукт включает в себя следующие компоненты:

* Microsoft SQL Server;
* сервер приложения на базе ASP.NET Core;
* клиентская часть приложения, разработанная с помощью библиотеки React.
* данной главе были представлены архитектура программного средства и схема базы данных с подробным описанием. Для серверной части приложения спроектировано N таблиц базы данных под управлением Microsoft SQL Server.

# **3 Разработка программного средства**

Разрабатываемое приложение включает в себя две части: серверную и клиентскую. Для разработки каждой из них будет использован свой набор технологий, включающий различные библиотеки, языки программирования.

## 3.1 Реализация серверной части приложения

Для разработки серверной части приложения использовалась среда разработки Visual Studio 2019. Для взаимодействия с сервером базы данных использовалась технология Entity Framework Core. В качестве системы контроля версиями использовался сервис GitHub [21].

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*16*

*ДП 03.00.ПЗ*

*Разраб.*

*Провер.*

*.*

*.*

Консульт.

*Н. Контр.*

*.*

*Утверд.*

*Лит.*

*Листов*

*14*

БГТУ 74420054, 2018

## 3.2 Реализация клиентской части приложения

## 3.3 Вывод по разделу

У

# **4 Руководство пользователя**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*16*

*ДП 04.00.ПЗ*

*Разраб.*

*Провер.*

*.*

*.*

Консульт.

*Н. Контр.*

*.*

*Утверд.*

Руководство пользователя

*Лит.*

*Листов*

*7*

БГТУ 74420054, 2018

## 4.1 Роль «Пользователь»

У

# **5 Экономический раздел**

Изм

№ докум.

Подпись

Дата

Лист

1

ДП 06.00.ПЗ

Разраб.

*Горевой Д.А.*

Провер.

*Козловский В.В.*

Консульт.

*Романенко О.В.*

Н. Котр.

*Жигаровская С.А.*

Утверд.

Смелов В.В.

Экономический раздел

Лит.

Листов

9

БГТУ 74420054, 2018

У

# **5.1 Общая характеристика разрабатываемого программного средства**

В данном дипломном проекте был разработан веб-сервис онлайн редактор кода. Пользователь, находясь в интернете с любого устройства может получить доступ для совместной работы над проектом. Возможно взаимодействие с другими пользователями через голосовой чат. Актуальность сервиса подтверждается развитием информационных технологий, все виды сервисов и приложений, вплоть до редактирования кода в браузере, становятся доступными в интернете.

Данное программное средство можно отнести к типу 1.0 (ПС общего назначения). В качестве среды разработки была выбрана Visual Studio. Программное средство относится к группе сложности 3, к группе новизны В.

## 5.2 Исходные данные для проведения расчетов

Исходные данные для расчета стоимости программного продукта представлены в таблице 5.1.

Таблица 5.1 — Исходные данные для расчета

|  |  |  |  |
| --- | --- | --- | --- |
| Показатель | Единица измерения | Условное обозначение | Норматив |
| Коэффициент изменения скорости обработки информации | ед. | Кск | 0,6 |
| Тарифная ставка 1 разряда в организации за месяц | руб. | ОТ1 | 150 |
| Тарифный коэффициент | ед. | Кт | 3,04 |
| Коэффициент естественных потерь рабочего времени | ед. | Кп | 1,3 |
| Коэффициент премирования | ед. | Кпр | 1 |
| Норматив дополнительной заработной платы | % | Ндз | 150 |
| Ставка отчислений в Фонд социальной защиты населения | % | Нфсзн | 34 |
| Стоимость одного машино-часа | руб. | Смч | 0,5 |
| Норматив прочих прямых затрат | % | Нпз | 10 |
| Норматив общепроизводственных и общехозяйственных расходов | % | Нобп,обх | 60 |

## 

Окончание таблицы 5.1

|  |  |  |  |
| --- | --- | --- | --- |
| Норматив расходов на сопровождение и адаптацию | % | Нрса | 10 |
| Уровень рентабельности | % | Урент | 30 |
| Ставка НДС | % | Нндс | 20 |

## 5.3 Методика обоснования цены

В современных рыночных экономических условиях ПС выступает преимущественно в виде продукции организаций, представляющей собой функционально завершенные и имеющие товарный вид ПС, реализуемые покупателями по рыночным отпускным ценам.

Проектирование и разработка программных средств требует значительных затрат объемов ресурсов. В связи с этим, разработка и реализация каждого проекта обосновывается, как технически, так и экономически. Проект стоит разрабатывать, если он дает определенные преимущества по сравнению с известными передовыми аналогами или, в крайнем случае, по сравнению с существующей практикой. Поэтому, до того, как приступить к разработке проекта программного средства, специалисты, используя соответствующие методы находят наиболее рациональные программные решения, обеспечивающее высокий технический уровень программы и дающие существенную экономию ресурсов, как при разработке проекта, так и при его эксплуатации у пользователя (покупателя, заказчика).

У разработчика экономический эффект выступает в виде чистой прибыли от реализации ПС, остающейся в распоряжении организации, а у пользователя – в виде экономии трудовых, материальных и финансовых ресурсов.

Стоимостная оценка ПС предполагает определение затрат, что включает следующие статьи:

* заработная плата исполнителей — основная и дополнительная;
* отчисления в фонд социальной защиты населения;
* отчисления по обязательному страхованию от несчастных случаев и профессиональных заболеваний;
* расходы на материалы и комплектующие;
* расходы на спецоборудование;
* расходы на оплату машинного времени;
* прочие прямые затраты;
* общехозяйственные и общепроизводственные расходы.

На основании вышеперечисленных затрат рассчитывается себестоимость и отпускная цена ПС.

## 5.4 Определение объема программного средства

В сумме объем функций на разрабатываемое ПС составил 29820 условных машинных команд. Основные функции, которые реализованы в проекте, отображены в таблице 5.2.

Таблица 5.2 — Содержание и объем функций на разрабатываемое ПС

|  |  |  |
| --- | --- | --- |
| Номер функции | Содержание функции | Объем, условных машино-команд |
| 101 | Организация ввода информации | 150 |
| 109 | Организация ввода/вывода в интерактивном режиме | 320 |
| 201 | Генерация структуры базы данных | 4300 |
| 203 | Формирование баз данных | 2180 |
| 206 | Обслуживание базы данных в интерактивном режиме | 6950 |
| 207 | Манипулирование данными | 9550 |
| 208 | Организация поиска и поиск в базе данных | 5480 |
| 506 | Обработка ошибочных и сбойных ситуаций | 410 |
| 707 | Графический вывод результатов | 480 |

Объем ПС определяется путем подбора аналогов на основании классификации ПС, каталогов функций ПС и аналогов ПС в разрезе функций, которые постоянно обновляются и утверждаются в установленном порядке. Общий объем функций рассчитывается по формуле 5.1 [12].

(5.1)

(условных машинных команд),

где  — объем *i*-й функции ПС, условных машинных команд.

Скорректированный объем функций вычисляется по формуле 5.2

(условных машинных команд), (5.2)

где **—** коэффициент изменения скорости обработки информации, берется из таблицы 5.1.

## 5.5 Расчет трудоемкости выполняемой работы

Определение трудоемкости необходимо для дальнейшего расчета суммы основной заработной платы.

Сумма основной заработной платы рассчитывается на основе численности специалистов, соответствующих тарифных ставок и фондов рабочего времени. Численность специалистов, календарные сроки разработки программы и фонд рабочего времени определяется по укрупненным нормам времени на разработку, сопровождение и адаптацию ПС или экспериментальным путем.

При определении трудоемкости ПС учитывается объем ПС (в тысячах строк условного кода), объем документации (в тысячах строк), новизна и сложность ПС, язык программирования. Степень использования типов (стандартных) программ.

На основании общего объема ПС определяется нормативная трудоемкость ). Кроме того, устанавливаются дополнительные коэффициенты сложности ПС.

Проанализировав необходимые таблицы было установлено, что = 362 чел.-дней, так как объем ПС = 17892 строк условных машинных команд и наше ПС относится к третьей группе сложности. Проанализировав таблицу дополнительных коэффициентов сложности ед., так как одной из характеристик ПС является обеспечение хранения, введения и работа с данными в сложных структурах.

Общая трудоемкость ПС рассчитывается по формуле (5.3)

(чел.- дней), (5.3)

где  — нормативная трудоемкость ПС, чел.-дней;

 — дополнительный коэффициент сложности ПС, ед.

## 5.6 Расчет основной заработной платы

Месячная тарифная ставка каждого из исполнителей (программистов) определяется путем умножения действующей тарифной ставки I разряда на тарифный коэффициент, соответствующий установленному тарифному разряду, рассчитывается по формуле (5.4)

(руб.), (5.4)

где  — тарифная ставка I разряда в организации за месяц, руб., табл.1;

 — тарифный коэффициент, раз, по исходным данным, табл.5.1.

Тарифная ставка за день вычисляется по формуле (5.5)

(руб.) (5.5)

Основная заработная плата исполнителей на конкретное ПС рассчитывается по формуле (5.6)

(руб.), (5.6)

где  — тарифная ставка за день, руб.;

 — общая трудоемкость ПС, чел.-дней, рассчитано по формуле 5.5;

 — коэффициент естественных потерь рабочего времени, процентов, выраженных в долях, табл. 5.1;

*—* коэффициент премирования, процентов, выраженных в долях, табл. 5.1.

## 5.7 Расчет дополнительной заработной платы

Дополнительная заработная плата на конкретное ПС включает выплаты, предусмотренные законодательством о труде (отплата отпусков, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей) и определяется по нормативу в процентах к основной заработной плате, вычисляется по формуле 5.7

(руб.), (5.7)

где  — основная заработная плата исполнителей на конкретное ПС, руб., формула 6.6;

 — норматив дополнительной заработной платы, %, табл.5.1.

## 5.8 Расчет отчислений на социальные цели

Отчисления в Фонд социальной защиты населения (ФСЗН) и по обязательному страхованию от несчастных случаем на производстве и профессиональных заболеваний определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной заработной платы исполнителей, и рассчитывается по формуле 5.8.

, (5.8)

где  — сумма отчислений в Фонд социальной защиты населения и по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний, руб.;

 — основная заработная плата исполнителей на конкретное ПС, руб., формула 5.6;

 — основная и дополнительная заработная плата исполнителей, руб., формула 5.7;

 — норматив отчислений в Фонд социальной защиты населения и по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний, %, табл. 5.1 [12].

(руб.)

## 5.9 Расчет расходов на материалы

Расходы на материалы определяется с учетом действующих нормативов. Нормы расхода материалов в суммарном выражении определяются в расчете на 100 строк исходного кода. Сумма затрат на материалы рассчитывается по формуле 5.9.

(руб.), (5.9)

где = 0,04 руб. — норма расхода материалов в расчете на 100 строк исходного кода ПС, руб.;

скорректированный объем функций, условных машинно-команд, формула 5.2.

## 5.10 Расчет расходов на оплату машинного времени

Расходы включают оплату машинного времени, необходимого для разработки и отладки ПС, которое определяется по нормативам (в машино-часах) на 100 строк исходного кода машинного времени в зависимости от характера решаемых задач и типа ПЭВМ. Сумма расходов на оплату машинного времени рассчитывается по формуле 5.10

(руб.), (5.10)

где  — стоимость одного машинного часа, руб., табл.5.1;

= условных машинно-команд — скорректированный объем функций, подсчитанный по формуле 5.2;

 — норматив расхода машинного времени на отладку 100 строк исходного кода, машино-часов.

## 5.11 Расчет прочих прямых затрат

Расходы на конкретное ПС включают затраты на приобретение и подготовку специальной научно-технической информации, и специальной литературы, определяются по нормативу в процентах к основной заработной плате, вычисляются по формуле 5.11

(руб.), (5.11)

где  — основная заработная плата, руб., формула 5.6;

– норматив прочих затрат в целом по организации, %, табл.5.1.

## 5.12 Расчет общепроизводственных и общехозяйственных расходов

Данные затраты, связаны с необходимостью содержания аппарата управления разных уровней (отделов и предприятий в целом), а также с расходами на общехозяйственные нужды, относятся на конкретное ПС по нормативу в процентном отношении к основной заработной плате исполнителей, формула 5.12.

(руб.), (5.12)

где — сумма общепроизводственных, общехозяйственных расходов, руб.;

 — основная заработная плата, руб., рассчитано по формуле 6.6;

 — норма общепроизводственных, общехозяйственных расходов в целом по организации, %, табл.5.1.

## 5.13 Расчет суммы расходов на разработку ПС

Общая сумма расходов на разработку ПС находится по формуле 5.13

, (5.13)

где  — основная заработная плата, руб., формула 5.6;

 — основная и дополнительная заработная плата исполнителей, руб., формула 5.7;

 — сумма отчислений в Фонд социальной защиты населения и по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний, руб., формула 5.8;

 — сумма расходов на оплату машинного времени, руб., формула 5.10;

**—** сумма общепроизводственных, общехозяйственных расходов, руб., формула 5.12;

 — сумма затрат на материалы, руб., формула 5.9;

 — прочие прямые затраты для данной разработки, руб., формула 5.11.

(руб.)

## 5.14 Расчет расходов на сопровождение и адаптацию

Кроме того, организация – разработчик осуществляет затраты на сопровождение и адаптацию ПС, которые определяются по нормативу и формула для расчета следующая 5.14

(руб.), (5.14)

где  — сумма расходов на сопровождена и адаптацию ПС, руб.;

 — общая сумма расходов на разработку ПС, руб., формула 5.13;

= 10% – нормативный расход на сопровождение и адаптацию, %, табл.5.1.

## 5.15 Расчет общей суммы расходов

Общая сумма расходов на разработку с затратами на сопровождение и адаптацию ПС находится по формуле 5.15

(руб.), (5.15)

где  — общая сумма расходов на разработку ПС, руб., формула 5.13;

 — сумма расходов на сопровождена и адаптацию ПС, руб., формула 5.14.

## 5.16 Определение цены и оценка эффективности

Рентабельность и прибыль по создаваемому ПС определяются исходы из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость. Прибыль от реализации ПС вычисляется по формуле 5.16

(руб.), (5.16)

где — полная себестоимость ПС, руб., формула 5.15;

 — уровень рентабельности ПС, %, табл.5.1.

Цена разработчика ПС без налогов находится по формуле 5.17

(руб.), (5.17)

где  — полная себестоимость ПС, руб., формула 5.15;

 — прибыль от реализации ПС, руб., формула 5.16.

Сумма налога на добавленную стоимость находится по формуле 5.18

(руб.), (5.18)

где — цена разработчика ПС без налогов, руб., формула 5.17;

 — ставка НДС, %, табл.6.1.

Планируемая отпускная цена с НДС находится по формуле 5.19

(руб.), (5.19)

где — цена разработчика ПС без налогов, руб., формула 5.17;

 — сумма налога на добавленную стоимость, руб., формула 5.18.

Результаты расчетов приведены в сводной табл. 5.3.

Таблица 5.3 — Расчет затрат на создание ПС и отпускной цены

|  |  |
| --- | --- |
| Статья затрат | Сумма,  руб. |
| Основная заработная плата |  |
| Дополнительная заработная плата |  |
| Отчисления в Фонд социальной защиты населения и по обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний |  |
| Расходы на материалы |  |
| Прочие затраты |  |
| Общепроизводственные расходы |  |
| Производственная себестоимость (расходы на разработку ПС) |  |
| Расходы на сопровождение и адаптацию ПС |  |
| Полная себестоимость ПС |  |
| Плановая прибыль |  |
| Цена без НДС |  |
| Отпускная цена с НДС |  |

# **5.17 Вывод по разделу**

Таким образом, из таблицы 5.3 видно, что большую часть отпускной цены разрабатываемого программного средства составляют расходы на его разработку. Планируемая прибыль от реализации разработанного в ходе дипломного проекта ПС составляет руб. Согласно проведенным расчетам цена программного средства с учетом НДС составила руб.

ЗАКЛЮЧЕНИЕ

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*8*

*ДП 00.00.ПЗ*

*Разраб.*

*Провер.*

*.*

*.*

*Консултант*

*.*

*Н. Контр.*

*Утверд.*

*Смелов В.В.*

ЗАКЛЮЧЕНИЕ

*Лит.*

*Листов*

*1*

БГТУ 74420054, 2018

В рамках дипломного проекта был разработан программный комплекс криптографической защиты информации. Произведен обзор теоретических материалов по теме дипломного проекта, произведен обзор аналогов.

Были представлены основные технические средства, которые использовались при разработке. Дана их краткая характеристика, возможности и достоинства.

В результате проделанной работы разработана библиотека, которая позволяет использовать, изменять, встраивать, тестировать программный код, реализующий государственные стандарты криптографии.

Целью данного дипломного проекта являлось разработка и создание программного комплекса криптографической защиты информации. Для достижения поставленной цели были реализованы следующие задачи:

* разработана реализация государственных стандартов криптографии РБ;
* реализовано демонстрационное консольное приложение;
* реализовано демонстрационное мобильное приложение.

Так же было проведено экономическое обоснование и рассмотрены мероприятия по охране труда.

Была проделана работа по ручному функциональному тестированию. Результаты соответствуют задуманным требованиям. Также был разработан модуль самотестирования приложения. Модуль самотестирования выполняет тестирование всех разработанных компонентов на контрольных значениях, а так же проверяет скорость выполнения алгоритмов.

Данный программный комплекс должен заинтересовать программистов и заказчиков своей простотой, скоростью, легкостью встраивания и удобством пользования.

Разработанный программный комплекс имеет сертефикат соответствия государственным стандартам РБ. Сертефикат был выдан Оперативно-аналитическим центром при Президенте Республики Беларусь.

У

*Пустовалова Н.Н.*

# **Список использованных источников**

1. LoveTour // LoveTour [Электронный ресурс]. – 2020. – Режим доступа: http://lovetour.by/. – Дата доступа: 11.04.2020.
2. Magellan // Magellan [Электронный ресурс]. – 2020. – Режим доступа: https://www.magellan.by/. – Дата доступа: 11.04.2020.
3. PeopleTravel // PeopleTravel [Электронный ресурс]. – 2020. – Режим доступа https://peopletravel.by/. – Дата доступа: 11.04.2020.
4. Либерти Д. Язык программирования C# // Программирование на C#. — Санкт-Петербург. — 2003: Символ-Плюс. — С. 26. — 688 с. — ISBN 5-93286-038-3.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*1*

*8*

*ДП 00.00.ПЗ*

*Разраб.*

*Тихонович М.А.*

*Провер.*

*.*

*Панченко О.Л.*

*Консултант*

*Панченко О.Л.*

*Н. Контр.*

*Утверд.*

*Пацей Н.В.*

Список использованных источников

*Лит.*

*Листов*

*1*

БГТУ 74420054, 2018

1. Gosha20777.github.io / Gosha20777 [Электронный ресурс] – Режим доступа: https://gosha20777.github.io/code/2018/02/22/dotnetcore/ – Дата доступа: 18.04.2020.

4 Gosha20777.github.io [Электронный ресурс] / Gosha20777 – Режим доступа: https://gosha20777.github.io/code/2018/02/22/dotnetcore/ – Дата доступа: 18.04.2020.

17 Dev.by ИТ в Беларуси [Электронный ресурс] / dev.by – Режим доступа: [https://dev.by/lenta/main/top-12-yazykov-programmirovaniya-dlya-novichkov-v-2015-godu – Дата доступа 27.03.2018](https://dev.by/lenta/main/top-12-yazykov-programmirovaniya-dlya-novichkov-v-2015-godu%20–%20Дата%20доступа%2027.03.2018).

18 GeekBrains [Электронный ресурс] / GeekBrains – Режим доступа: https://geekbrains.ru/posts/6\_php\_frameworks – Дата доступа 20.03.2018.

19 PracticalEcommerce [Электронный ресурс] / PracticalEcommerce – Режим доступа: http://www.practicalecommerce.com/articles/464-Basic-Definitions-Web-1-0-Web-2-0-Web-3-0 – Дата доступа: 24.03.2018.

20 Урбанович П.П. Информационная безопасность и надежность систем: учебно-методическое пособие для студентов специальности 1–40 01 02 «Информационные системы и технологии» / П. П. Урбанович, Д. М. Романенко, Е. В. Романцевич. – Минск: БГТУ, 2007. – 100 с.

21 Метод Гаусса: описание алгоритма решения системы линейных уравнений, примеры, решения [Электронный ресурс] – Режим доступа: http://www.cleverstudents.ru/systems/solving\_systems\_Gauss\_method.html/. Дата доступа: 05.05.2018.

22 Несобственные интегралы. Примеры решений. [Электронный ресурс] – Режим доступа: http://mathprofi.ru/nesobstvennye\_integraly.html/. Дата доступа: 06.05.2018.

23 Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник для вузов / Т.А. Павловская. – СПб.: Питер, 2007. – 432 с.

Общие санитарно–гигиенические требования к воздуху рабочей зоны: ГОСТ 12.1.005-88 ССБТ. – Введ. 01.01.89. – М.: Государственный комитет по стандартам, 1989. – 52 с.

25 Естественное и искусственное освещение. Строительные нормы проектирования: ТКП 45-2.04-153-2009. – Введ. 01.01.10. – Минск: Министерство архитектуры и строительства Республики Беларусь, 2010. – 106 с.

26 Босак В.Н. Дипломное проектирование: мероприятия по охране труда и безопасности жизнедеятельности / В. Н. Босак, А. К. Гармаза, И. Т. Ермак, Б. Р. Ладик. – Минск: БГТУ, 2013. – 130 с.

27 Сокол Т.С. Охрана труда. Учебное пособие / Т. С. Сокол – Москва: Дизайн ПРО, 2005. – 304 с.