СОДЕРЖАНИЕ

[**ВЕДЕНИЕ** 6](#_Toc134498789)

[**1** **Информационная система туристического агентства** 7](#_Toc134498790)

[**1.1** **Анализ предметной области** 7](#_Toc134498791)

[**1.2 Аналоги информационной системы туристического агентства** 9](#_Toc134498792)

[**1.3 Выводы и постановка задач на дипломное проектирование** 12](#_Toc134498793)

[**2** **Разработка информационной системы туристического агентства** 14](#_Toc134498794)

[**2.1 Структура информационной системы** 14](#_Toc134498795)

[**2.2 Алгоритм работы информационной системы** 18](#_Toc134498796)

[**2.3 Структура базы данных информационной системы** 21](#_Toc134498797)

[**2.4 Выводы и оценка результатов разработки** 25](#_Toc134498798)

[**3 Анализ расчета надежности информационной системы туристического агентства** 26](#_Toc134498799)

[**3.1 Расчет надежности информационной системы по модели сложности** 26](#_Toc134498800)

[**3.2 Расчет надежности информационной системы по модели Джелинского - Моранды** 30](#_Toc134498801)

[**3.3 Расчет надежности информационной системы по модели Муса и выводы** 32](#_Toc134498802)

[**4 Тестирование информационной системы туристического агентства** 34](#_Toc134498803)

[**5 Технико-экономическое обоснование разработки информационной системы туристического агентства** 36](#_Toc134498804)

[**5.1 Характеристика программного продукта** 36](#_Toc134498805)

[**5.2 Расчет инвестиций в разработку программного средства** 36](#_Toc134498806)

[**5.3 Расчет экономического эффекта от использования программного средства** 38](#_Toc134498807)

[**5.4 Расчет показателей экономической эффективности разработки и использования программного средства** 40](#_Toc134498808)

[**6.** **Охрана труда. разработка мероприятий по повышению работоспособности и производительности труда разработчика информационной системы поддержки деятельности туристического агентства** 42](#_Toc134498809)

[**ЗАКЛЮЧЕНИЕ** 47](#_Toc134498810)

[**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ** 48](#_Toc134498811)

[**Приложения A** 49](#_Toc134498812)

[**Приложения В** 49](#_Toc134498813)

[**Приложения Д** 49](#_Toc134498814)

[**Приложения Е** 49](#_Toc134498815)

[**Приложения Б** 50](#_Toc134498816)

[**Приложения Г** 51](#_Toc134498817)

**ВЕДЕНИЕ**

Туризм является одной из крупнейших и наиболее перспективных отраслей мировой экономики. Он играет значительную роль в создании рабочих мест, экспортных поступлений большого ряда стран. Туристская отрасль подвержена тенденциям мировой экономики. Важнейшими трендами, позволяющими компаниям осваивать новые туристские рынки и бизнес-модели, являются автоматизация и цифровизация процессов. Цифровизация предлагает технологии, подходы и инструменты, которые позволяют повысить ценность туристского продукта [1], [2]. Использование возможностей цифровых технологий в туристской деятельности существенно увеличивает производительность, что приводит к значимой экономии временных, финансовых и человеческих ресурсов [3], [4].

Благодаря информационным технологиям современные туристические услуги становятся более гибкими и индивидуальными, более привлекательными и доступными для широкого потребителя. Туризм более чем другие отрасли неразделим с информацией, которая несет в себе сведения о предлагаемых турах, туроператорах, услугах, условиях проживания, перемещения, оздоровления и т. д. Возможность анализа информации позволяет принять на ее основе единственно правильное решение, обеспечить наилучший выбор услуги согласно индивидуальным требованиям и возможностям.

Современные автоматизированные информационные системы для организации работы в туристическом бизнесе не только обеспечивают руководителей и менеджеров информацией о различных аспектах деятельности организации для ее своевременной оценки и анализа, но и предоставляют информацию о конкурентоспособности туров в простой и удобной форме.

Целью проекта является разработка информационной системы поддержки деятельности туристического агентства, выполняющая функции создания и управления заказами, учета финансовых операций и учета клиентов.

Для разработки информационной системы туристического агентства был выбран стек технологий MERN, который включает в себя MongoDB для хранения данных, Express.js для создания сервера, React для разработки клиентской части и Node.js для выполнения кода на сервере.

Использование стека технологий MERN[16] позволяет создавать полноценные веб-приложения, которые могут быть легко масштабированы и обеспечивать высокую производительность информационной системы. Кроме того, данные технологии позволяют использование единого языка программирования (JavaScript), что упрощает процесс разработки и поддержки кода, обеспечивая высокую производительность.

1. **Информационная система туристического агентства** 
   1. **Анализ предметной области**

В индустрии туризма применяются самые разнообразные информационные технологии, начиная от широко распространенных технологий работы с текстом, электронными таблицами и базами данных до использования специализированных программных продуктов, обеспечивающих автоматизацию работы отдельной туристической фирмы или отеля, и глобальных компьютерных сетей и спутниковых систем навигации [5], [6].

Многообразие применяемых в туризме информационных технологий и систем приведено на рисунке 1.1

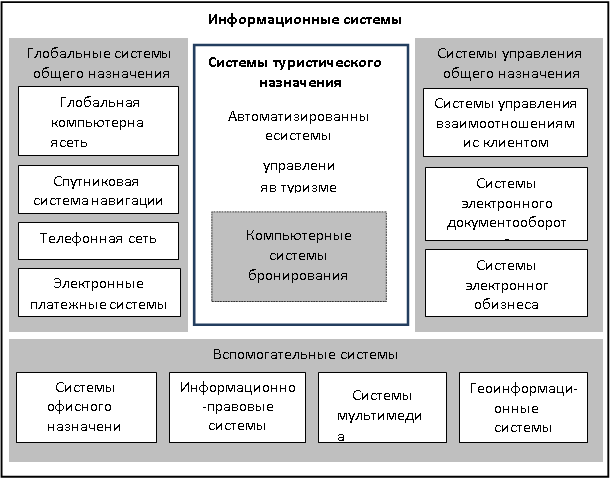


Рисунок 1.1 - Информационные системы в туристической отрасли

Автоматизированные системы управления в туризме – системы управления производственно-обслуживающим процессом в туристических предприятиях и организациях. Они служат для создания эффективной туристической структуры, позволяющей обеспечить комфортные условия труда персонала за счет его профессионального развития и управления его деловой карьерой. Функциональные возможности этих систем должны обеспечивать ввод, редактирование и хранение информации о турах, клиентах, о состоянии заявок; предусматривать вывод информации в форме различных документов; расчет стоимости туров с учетом курса валют, скидок, контроль оплаты туров, формирование билетов; перевод экспорт данных в другие программные продукты (Pdf, Word) и прочие функции [5].

Информационные технологии в туризме представляют собой взаимосвязь компьютерных и коммуникационных технологий. Это позволяет различать несколько уровней автоматизации работы предприятий турбизнеса. Необходимый уровень автоматизации определяется объемами, с которыми сталкивается компания.

Автоматизация деятельности туристического агентства включает в себя следующий набор функций:

получение и обработка информации от разных туроператоров;

ведение внутреннего документооборота и бухгалтерии;

выстраивание взаимоотношений с туроператорами;

анализ данных и получение статистических отчетов.

В зависимости от используемых средств для реализации этих функций можно рассматривать следующие уровни автоматизации туристического предприятия:

1. Применение стандартного программного обеспечения. Может быть достаточным для небольших объемов клиентского обслуживания, т.к ограничен использованием компьютера с общедоступными офисными программами, которые позволяют отчасти автоматизировать рутинную повседневную работу организации. Безусловным является и наличие возможности осуществления электронных коммуникаций посредством сервисов сети Интернет.

2. Применение специального программного обеспечения туристических программ. На данном уровне автоматизации требуется более высокая профессиональная подготовка персонала компании.

3. Использование глобальных компьютерных систем бронирования. В этом случае туроператоры рассматривают свою работу в едином информационном пространстве, что позволяет в значительной степени расширить свое представительство в сети Интернет.

4.Участие в электронном бизнесе. Данный уровень автоматизации предполагает расширение собственного присутствия турфирмы в сети Интернет. Это может быть представлено следующими средствами:

- сайт-визитка со списком услуг, реквизитами, прайс-листом, фотографиями и другой информацией, дублирующей рекламные проспекты фирмы;

- интерактивный сайт для быстрой связи клиентов с фирмой через Интернет, способный выдавать информацию по запросу пользователя, отвечать на его вопросы, иметь средства обратной связи с фирмой (гостевую книгу, форум, ответы на часто задаваемые вопросы, форму запроса информации и др.). Такой сайт способствует увеличению числа потенциальных клиентов;

- интернет-магазин, способный принимать платежи за туристические услуги, бронировать путевки, билеты, заключать договора с клиентами, оповещать их о свободных местах, путевках и др. Такой сайт выполняет функцию полноценного параллельного механизма реализации туристических услуг, позволяет разгрузить менеджеров, ускорить выполнение рутинных операций (прием платежей, подготовка и подпись бумаг, приезд клиента в офис и др.);

- подключение туристического агентства на основе абонентской платы к посредническим бизнес-системам, реализующим вышеупомянутые возможности и берущим на себя ответственность за безопасность ведения бизнеса.

Обобщая возможности автоматизации предприятия турбизнеса, можно назвать основные решаемые при этом задачи:

- мониторинг состояния рынка – для поиска и бронирования туров, в том числе в режиме онлайн;

- автоматизация внутреннего документооборота – выписка необходимых туристу документов (путевка, приходный/расходный кассовый ордер, договор, ваучер и др.), отслеживание жизненного цикла заявки клиента;

- автоматизация взаимоотношений с туроператорами – создание и печать бланка заявки, автоматическое отслеживание прохождения заявки от момента ее формирования до момента отправки в архив;

- автоматизация бухгалтерии – использование специализированных бухгалтерских программ. В ряде случаев турагентства, работающие по упрощенной системе налогообложения, пользуются услугами аудиторских компаний, сдавая им лишь первичную документацию. В этом случае функций внутриофисных программ, касающихся учета финансов, оказывается достаточно. Возможно сопряжение специализированных внутриофисных туристических программ с бухгалтерскими на уровне обмена файлами;

- автоматизация анализа данных и получение статистики – формирование статистических отчетов, показывающих рентабельность работы компании за промежуток времени, среднюю доходность заявок по направлениям и туроператорам и т. д. [5]. Это позволяет агентству правильно ориентироваться на рынке и разрабатывать нужные направления деятельности, выстраивая взаимоотношения с туроператорами, в нужное время давать нужную рекламу и оценивать, как она работает.

Чтобы воспользоваться преимуществами цифровых технологий в полной мере, туристские компании должны интегрировать их в повседневные процессы [7].

## **1.2 Аналоги информационной системы туристического агентства**

Для проведения сравнительного анализа веб-приложения, необходимо понять какими чаще всего пользуются пользователи, принадлежащие к целевой аудитории.

Проведен обзор наиболее популярных веб-приложения Республики Беларусь. Портал Toursoyuz.by подвел итоги опроса, который проводился в ноябре-декабре среди туристических веб-ресурсов Беларуси. Список веб-приложения по направлению туризма в Беларуси с результатами опроса представлен в таблице 1.1 и на рисунке 1.2 [8].

Таблица 1.1 – Рейтинг веб- приложения туризма Беларуси

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название службы | Нaзвание веб-ресурсов | Известность |
| 1 | Виаполь | *Viapol.by* | 46,4 % |
| 2 | Тез Тур | *Teztour.by* | 35,3 % |
| 3 | Тайм Вояж | [*Time-voyage.relax.by*](https://time-voyage.relax.by/) | 26,1 % |
| 4 | Травеллаб | *Travellab.by* | 22,2 % |
| 5 | ЦентрКурорт | *Otpusk.by* | 15,9 % |
| 6 | Внешинтурист | *321.by* | 9,2 % |
| 7 | Бел-Ориентир | *Bel-orientir.by* | 8,2 % |
| 8 | АТТ | *Att.by* | 5,3 % |
| 9 | Топ-Тур | *Toptour.by* | 4,8 % |

Первую строчку в предпочтениях агентов занимает «Виаполь» [9]. В тройке лидеров также – «Тез Тур» и «Тайм Вояж». Далее по популярности идут такие компании как «Травеллаб», «Внешинтурист», «АТТ», «Бел-Ориентир», «ЦентрКурорт» и «Топ-Тур».



Рисунок 1.2 – Рейтинг веб-ресурсов внутреннего туризма Беларуси

По результатам опроса веб-приложения «Тез Тур», «Тайм Вояж» являются наиболее известными в Республике Беларусь. Эти веб-приложении рассмотрим в качестве аналогов.

«TezTour» имеет заслуженную репутацию одной из самых высокотехнологичных и надежных компаний на рынке туризма, которая напрямую сотрудничает с такими известными сетями отелей как Hilton, Iberostar, Four Seasons, Radisson, Aldemar, предлагает пакетные туры на основе чартеров, лоукостов и регулярных рейсов. При бронировании туров, туристы получают необходимую визовую поддержку, услуги страхования, трансферы, различные дополнительные услуги и круглосуточную помощь от специалистов в решении любых вопросов касательно путешествия [10].

Бронируя туры на сайте, клиент не только приобретает качественный туристический продукт, но и экономит время. Подбор и бронирование туров онлайн представлен на рисунке 1.3

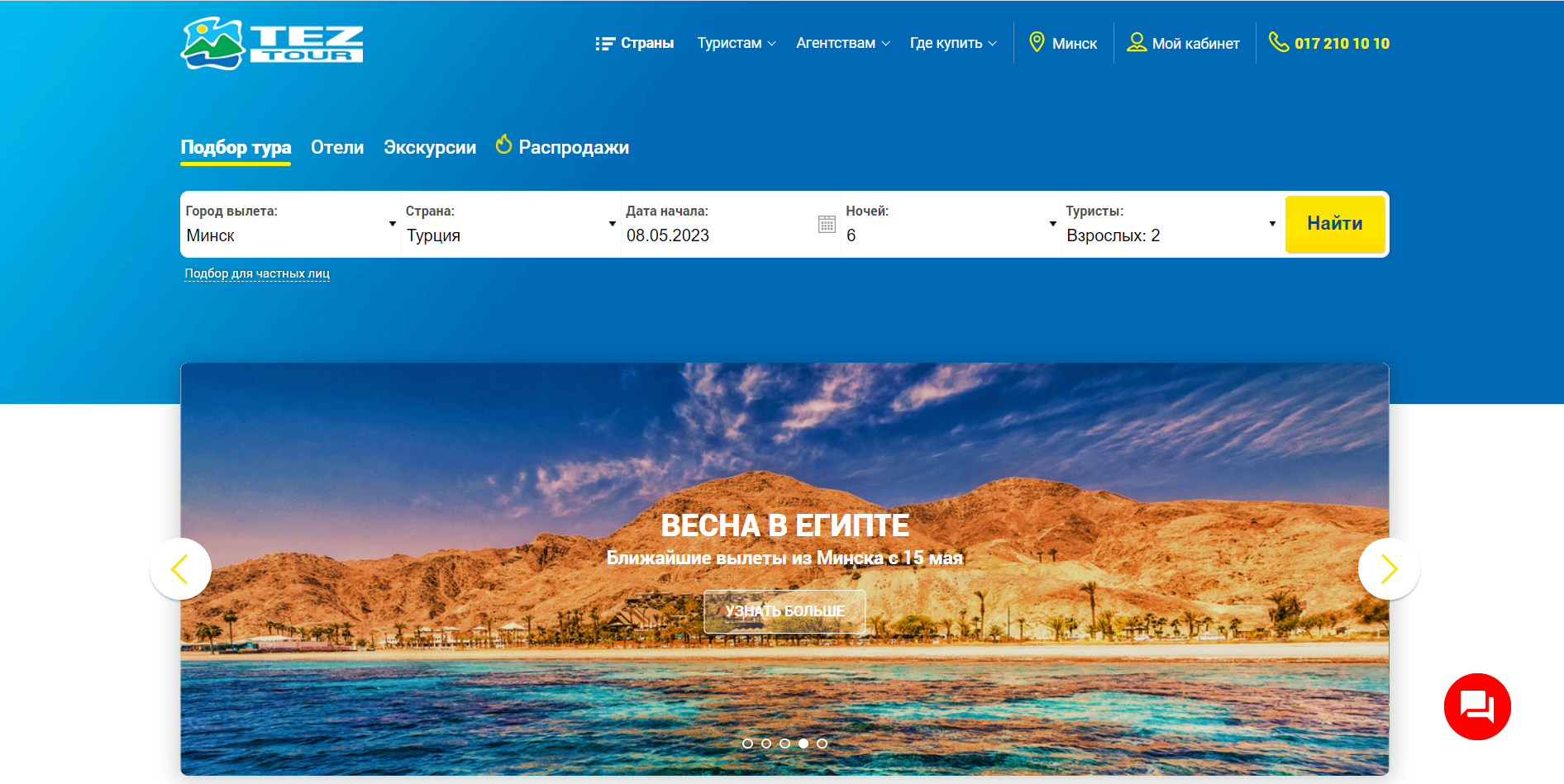


Рисунок 1.3 – Подбор и бронирование туров онлайн веб-приложения «TezTour»

Основными достоинствами веб-ресурса «TezTour» являются:

Туристическая компания «Тайм Вояж» [11] работает на рынке туристических услуг Беларуси с 1992 года. За это время компания стала представителем популярной сети услуг туров горящих путевок, членом республиканского союза туристических организаций и уполномоченным агентом крупнейшего туроператора «TezTour». Основная информация о веб-ресурсе туров «Тайм Вояж» представлена в таблице 1.2.

Таблица 1.2 – Основная информация о веб- приложения туров «Тайм Вояж»

|  |  |
| --- | --- |
| Название | «Тайм Вояж» |
| Веб-ресурс | t-v.by |
| Описание | «Тайм Вояж» – внесено Минским городским исполнительным комитетом в справочник Единого государственного регистра юридических лиц и индивидуальных предпринимателей от 14.04.1998. |
| Основная функция | – поиск по названию [страны/курорты/города](https://t-v.by/countries-resorts-cities/)  – поиск по названию все типы туров  – [горячие туры](https://t-v.by/tourists/hot-tours/)  – [акции и скидки](https://t-v.by/tourists/promotions-and-discounts/)  – [бронирование тура](https://t-v.by/tourists/payment-order/bronirovanie-i-dostavka.php)  – [порядок оплаты](https://t-v.by/tourists/payment-order/)  – [бонусы](https://t-v.by/tourists/bonuses/)  – контакты |

На веб-приложения представлен каталог служб с возможностью сортировки по рейтинг, типу туров. В краткое описание службы входят название, типы туров. Представлена информация о рейтинговой оценке службы и возможность ознакомиться с отзывами других клиентов. Для поиска определенной службы или тура предусмотрена специальная форма «Поиск».

Скриншот главная страница представлен на рисунке 1.4 [11].

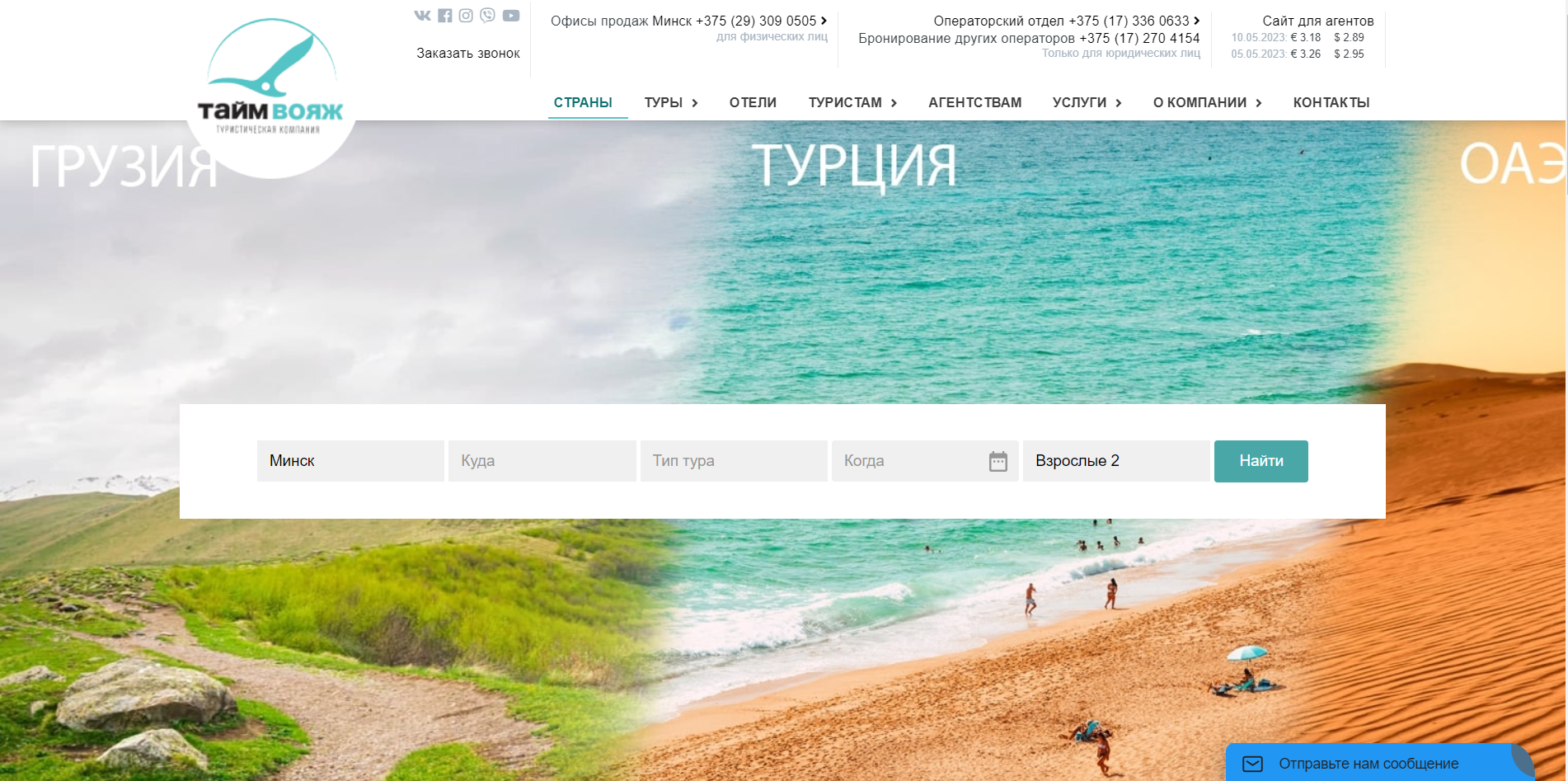
****

Рисунок 1.4 – Главная страница веб-приложения «Тайм Вояж»

## **1.3 Выводы и постановка задач на дипломное проектирование**

Анализ предметной области турагентства включает в себя определение концептуальных требований и информационных потребностей пользователей ИС, а также определение основных модулей, входящих в состав ИС, и связей между ними.

Концептуальные требования и информационные потребности пользователей ИС могут быть следующими:

– Учет клиентов турагентства;

– Возможность создания профиля клиента;

– Хранение данных о контактах клиента;

– Возможность просмотра истории бронирований и путешествий клиента;

– Учет туров и предложений;

– Возможность добавления и редактирования туров;

– Хранение данных о дате, месте, длительности, стоимости и других характеристиках туров;

– Возможность просмотра доступных туров и предложений, а также их фильтрация по различным параметрам;

– Возможность бронирование туров и управление заказами;

– Хранение информации о статусе заказа и его истории изменений;

– Возможность оплаты заказа и выставления счетов.

Таким образом, ИС для турагентства должна обеспечивать возможность учета клиентов и доступных туров, создания и управления заказами, учета финансовых операций. Модули ИС должны взаимодействовать между собой, передавая необходимую информацию и обеспечивая циклический процесс работы турагентства. ИС для турагентства должна обеспечивать комплексную автоматизацию бизнес-процессов, связанных с работой турагентства, от учета клиентов и доступных туров до управления заказами, финансовым учетом и эффективности работы. Важно учитывать специфику работы турагентства и потребности его пользователей при разработке ИС, чтобы обеспечить ее эффективность и удобство использования.

Цель дипломного проекта – создание функционирующего приложения в архитектуре клиент-сервер.

Назначение и цели разработки ИС туристического агенаства на основе стека технологий MERN[16]: с организацией взаимодействия с базой данных на объектно-ориентированном языке JavaScript. MERN – это стек технологий[16], который используется для разработки полноценных веб-приложений. Он состоит из MongoDB[17], Express.js[18], React[19] и Node.js[20].

Серверное приложение может быть реализовано в виде консольного приложения или GUI-приложения. Который в ключает в себе стек технологи MongoDB, Node JS и ExpressJS. Клиентское приложение: веб-приложение с использованием стандартных библиотек пользовательского интерфейса (React JS, Redux[21], Bootstrap[22]).

Для успешной реализации системы необходимо, в первую очередь, выделить основные задачи, которые будет решать система «Информационная система поддержки деятельности туристического агентства» а также те задачи, которые необходимо выполнить для правильной работы системы.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* выполнить обзор предметной области;
* разработать структурную схему информационной системы туристического агентства;
* разработать алгоритм работы информационной системы, базу данных, диаграмму вариантов использования;
* выполнить расчет надежности и производственных рисков;
* выполнить технико-экономическое обоснование эффективности разработки и реализации информационной системы;
* дать рекомендации по разработке мероприятий по повышению работоспособности и производительности труда разработчика информационной системы поддержки деятельности туристического агентства.

**2 Разработка информационной системы туристического агентства**

## **2.1 Структура информационной системы**

Разработка программного средства информационной системы для туристического агентства имеет целью создание продукта, который облегчит процесс управления туристической деятельностью, автоматизирует рутинные процессы и повысит эффективность работы агентства.

Структура и функционирование информационной систем: информационной систем должна содержать подсистемы для бронирования туров, управления клиентской информацией и.т.д. Режимы функционирования информационной систем работа в онлайн и офлайн режимах.

Надежность информационной систем: ИС должна быть высоконадежной и обеспечивать безопасность информации о клиентах и турах.

Плюсы разрабатываемой информационной систем: информационной систем должна обеспечивать быстрый доступ к информации и удобный интерфейс для пользователей, повышение эффективности работы сотрудников турагентства и увеличение количества продаж туров.

Первым шагом при описании функциональности системы является моделирование требований к ней. ­­­

Целями анализа и моделирования требований являются:

— достижение соглашения между разработчиками, заказчиками и пользователями о том, что должна делать ИС;

— достижение лучшего понимания разработчиками поведения ИС;

— создание базиса для планирования разработки проекта;

— определение пользовательского интерфейса.

Для достижения этих целей используются Структурная система линейная, линейная с ответвлениями, древовидную, также диаграммы вариантов использования.

Древовидная структура сайта предполагает иерархическое расположение категорий и подкатегорий, где каждая последующая ветвь вложена в предыдущую. Такая структура используется для удобной категоризации товаров или услуг по брендам, категориям и типам продукции. Большинство современных сайтов используют именно эту структуру, где пользователи могут легко найти нужный раздел, подраздел или конкретный товар, или услугу.

На рисунке 2.1 представлена схема древовидной структуры сайта [1].

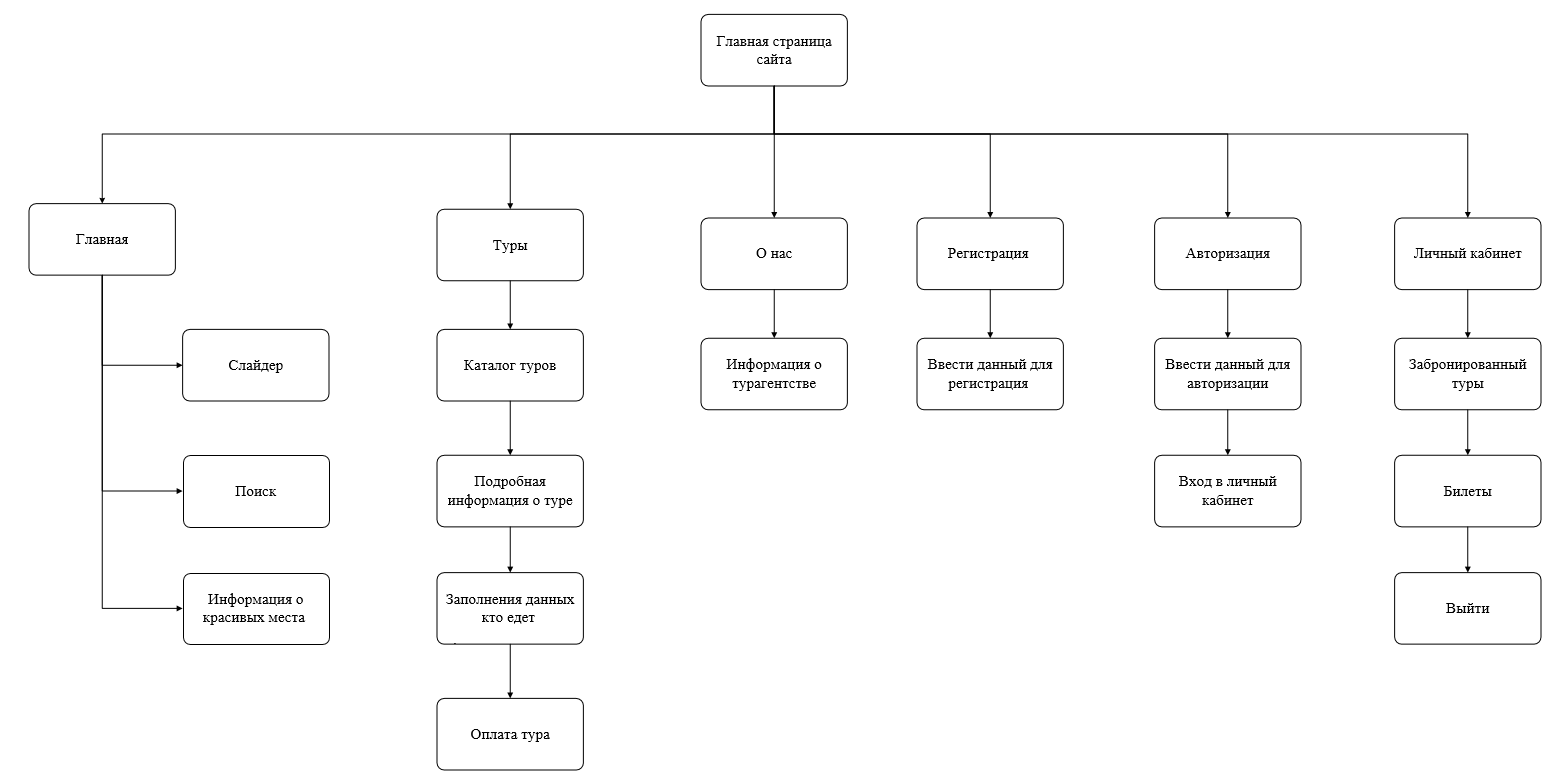


Рисунок 2.1 – Схема древовидной структуры

UML (Unified Modeling Language). На диаграммах вариантов использования (ВИ) изображаются актеры и варианты использования, между которыми существуют отношения. Актером будем называть внешнюю по отношению к ИС сущность, которая может взаимодействовать с системой. Актерами могут быть как люди, так и внешние системы или устройства. Следует всегда помнить, что актер – это не конкретный человек или устройство, а роль (должностная обязанность), в которой он выступает по отношению к программной системе. Диаграмма вариантов использования данного проекта представлена на Приложении Г

Итак, смоделировав диаграмму вариантов использования, мы получили наглядное представление о нашей системе. Были определены действующие лица модели, и именно сотрудник и клиент.

Диаграммы последовательности визуально моделируют поток логики в системе, позволяя документировать и проверять свою логику, и обычно используются как для анализа, так и для целей проектирования. Диаграммы последовательности являются наиболее популярным артефактом UML для динамического моделирования, которое фокусируется на определении поведения в данной системе. Другие методы динамического моделирования включают в себя диаграмму активности, диаграмму связи, временную диаграмму и диаграмму обзора взаимодействия. Диаграммы последовательности, а также физические модели данных, на мой взгляд, являются наиболее важными моделями уровня проектирования для разработки современных Турагенства. Диаграмма вариантов последовательности данного проекта представлена на рисунках - 2.2

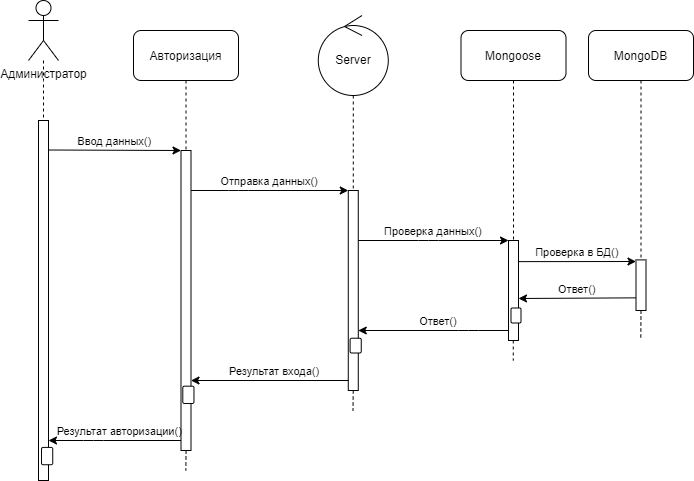


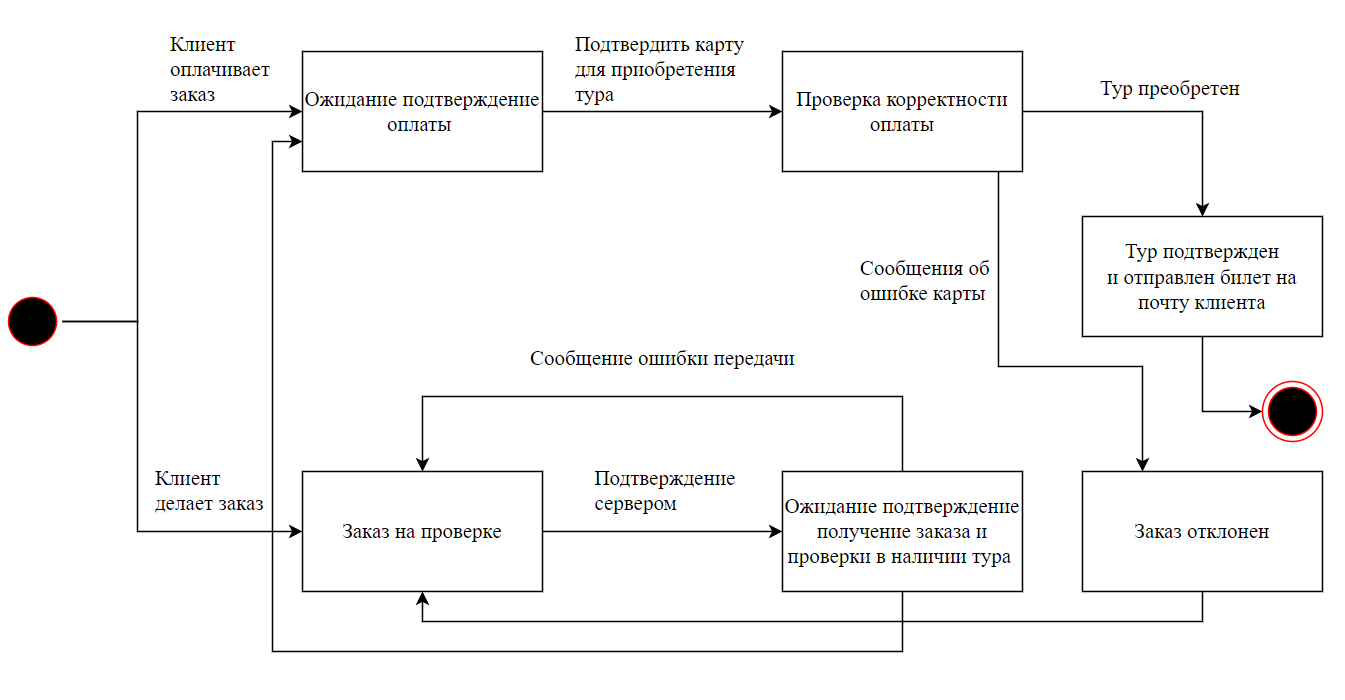
Рисунок - 2.2 Диаграмма вариантов последовательности.

Диаграмма состоянияОбъекты меняют своё состояние в ответ на происходящие события и с течением времени. Диаграмма состояний представляет состояния объекта и переходы между ними, а также начальное и конечное состояние объекта.

Основными элементами диаграммы состояний являются «Состояние» и «Переход». Диаграмма состояний имеет схожую семантику с диаграммой деятельности, только деятельность здесь заменена состоянием, переходы символизируют действия. Таким образом, если для диаграммы деятельности отличие между понятиями «Деятельность» и «Действие» заключается в возможности дальнейшей декомпозиции, то на диаграмме состояний деятельность символизирует состояние, в котором объект находится продолжительное количество времени, в то время как действие моментально.

Цель данной диаграммы является показать поведение одного объекта в течение его жизни, начиная от создания объекта и заканчивая его уничтожением.

В отличие от других диаграмм диаграмма состояний описывает процесс изменения состояний только одного класса, а точнее – одного экземпляра определенного класса, т. е. моделирует все возможные изменения в состоянии конкретного объекта. Диаграмма состояния данного проекта представлена на рисунках - 2.3

Рисунок - 2.3 Диаграмма состояние

Паттерн проектирования – повторимаяархитектурная конструкция, представляющая собой решение проблемыпроектированияв рамках некоторого часто возникающего контекста, иными словами, паттерны проектирования – это шаблоны необходимых шагов для решения каких-либо определенных задач.

Паттерн MVC (Model-View-Controller) состоит из объектов трех видов. Модель – это объект приложения, а вид – экранное представление. Контроллер описывает, как интерфейс реагирует на управляющие воздействия пользователя.

Первым с клиентским запросом взаимодействует контроллер, контроллер разбивает данный запрос на элементы, далее он инициализирует объекты модели. После обработки данные отправляются на уровень представления.

Паттерн MVC – простой способ построения структуры приложения, целью которого является отделение бизнес-логики от пользовательского интерфейса.

Диаграмма взаимодействия компонентов для паттерна «MVC» изображена на рисунке - 2.4

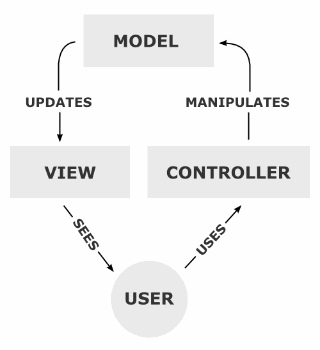


Рисунок - 2.4 – Диаграмма для паттерна «MVC»

## **2.2 Алгоритм работы информационной системы**

**Алгоритм** — представляет собой пошаговую процедуру, которая определяет набор инструкций, которые должны быть выполнены в определенном порядке, чтобы получить желаемый результат. Алгоритмы, как правило, создаются независимо от базовых языков, то есть алгоритм может быть реализован на нескольких языках программирования.

Алгоритмы широко используются в различных областях, включая информационные системы, науку о данных, искусственный интеллект, математику, криптографию и многие другие. Они позволяют автоматизировать процессы, упрощать решение задач, повышать эффективность и точность вычислений.

Важным преимуществом алгоритмов является их независимость от конкретных языков программирования. Алгоритм может быть описан на естественном языке или в форме блок-схемы, и затем реализован на различных языках программирования, таких как C++, Python, Java, JavaScript и других. Это позволяет разработчикам выбирать наиболее подходящий язык для конкретного проекта или задачи.

Однако создание хорошего алгоритма требует определенных навыков и знаний. Хороший алгоритм должен быть понятным, эффективным, легко поддерживаемым и устойчивым к ошибкам. Кроме того, в процессе разработки алгоритма необходимо учитывать особенности конкретной задачи, области применения и доступных ресурсов.

Основные модули, входящие в состав ИС, могут быть следующими:

Модуль учета клиентов и предложений содержит информацию о клиентах турагентства и доступных предложениях. Взаимодействует с модулем бронирования туров и управления заказами

Модуль бронирования туров и управления заказами содержит информацию о созданных заказах и их статусах

Взаимодействует с модулем учета клиентов и предложений и модулем учета финансовых операций

Связи между модулями можно представить в виде цикла, где модуль учета клиентов и предложений связан с модулем бронирования туров и управления заказами, который в свою очередь взаимодействует с модулем учета финансовых операций, который в свою очередь связан с модулем учета клиентов и предложений. Таким образом, при создании заказа на тур, информация о клиенте и выбранном туре передается из модуля учета клиентов и предложений в модуль бронирования туров и управления заказами. После создания заказа, информация о нем передается в модуль учета финансовых операций, где происходит учет финансовых операций, связанных с заказом.

Алгоритм работы функции не авторизованного пользователя**,** При запуске открывается главная веб-страница, в котором пользователь может просмотреть туры и приобрести их не авторизоваясь в системе**,** алгоритм покупки тура, клиент выбирает тур открывается страница тура, где пользователю нужно вести паспортные данные для биллета и ввести почту для оплаты и для отправки электронного билета, после того как пользователь ведет данные, то на странице покупки открывается меню для оплаты по карте, для подтверждения карты система сама отправит на почту четырёх значный строку для подтверждение а пользователь должен ввести этот код подтверждения в форме для карты и отправить, если код подтверждения и почта совпадает, то система моментально генерируют онлайн билет и отправляет на почту. Функция работы не авторизованного клиента представлена на рисунке - 2.5

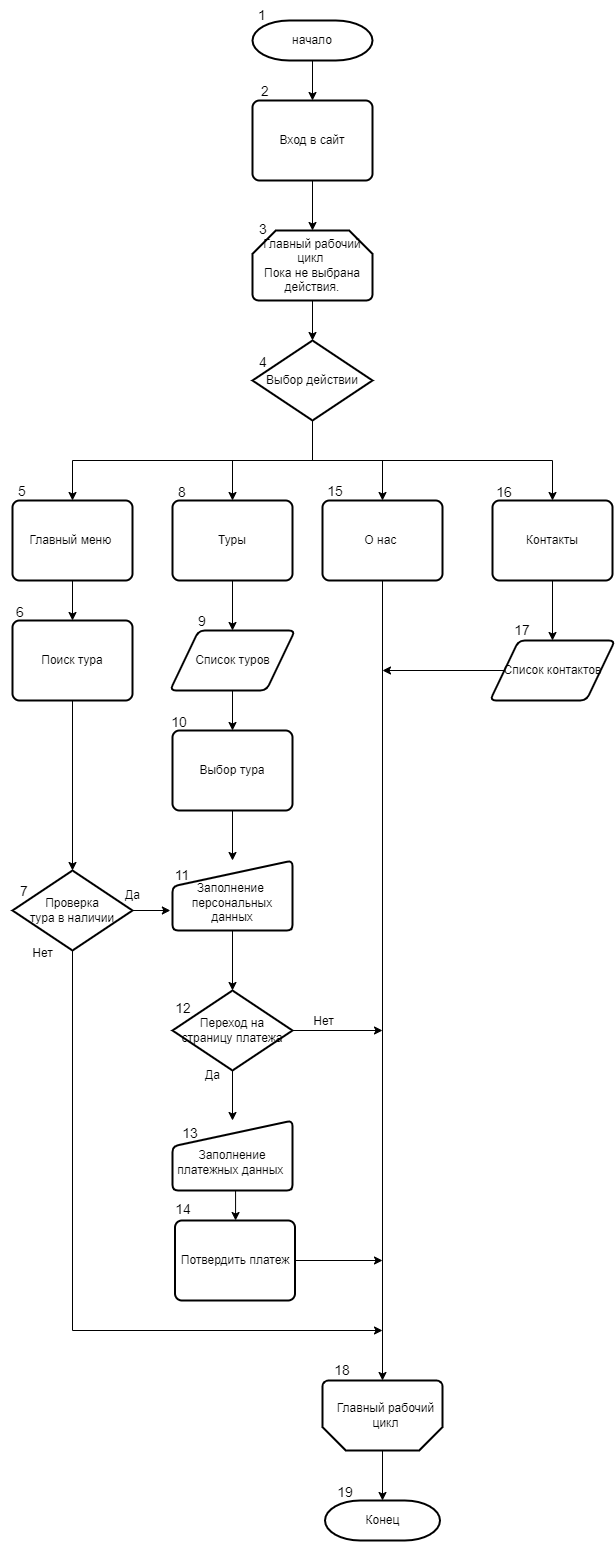


Рисунок - 2.5 алгоритм работы не авторизованного клиента

## **2.3 Структура базы данных информационной системы**

База данных является основой любой информационной системы. Она представляет собой структурированную коллекцию данных, которые используются для хранения, организации и управления информацией. Базы данных могут быть различными по своей структуре, в зависимости от требований конкретной системы.

Структура базы данных определяет, каким образом данные будут храниться, как они будут организованы и как они будут связаны друг с другом. В структуре базы данных могут быть определены таблицы, поля, индексы, отношения и другие элементы, которые необходимы для эффективной работы системы.

Процесс разработки (проектирования) базы данных включает два этапа: разработку логической организации базы данных и создание ее на носителе. Логическая организация базы данных - это предоставление пользователя о предметной области, информация о которой должна храниться в базе данных.

Под физической организацией базы данных понимается совокупность средств и методов размещения данных во внешней памяти и на их основе внутренняя модель данных. Внутренняя модель является средством отображения логической модели данных, показывает, каким образом записи размещаются в базе данных, как они упорядочиваются, как организуются связи, каким путем можно осуществить выборку и так далее.

Проектирование базы данных системы управления Туристической агентством начиналось с создания всех нужных таблиц в базе, всех полей, входящих в каждую таблицу, взаимодействия таблиц между собой с помощью специальных отношений и создание в соответствии с этими параметрами первичных и вторичных ключей.

В данной информационной системе используется база данных mongoDB.

MongoDB - это документ-ориентированная база данных[44], которая использует формат BSON (Binary JSON) для хранения и организации данных. Она является распространенной NoSQL базой данных и используется для множества приложений и сервисов, таких как системы управления контентом, аналитические приложения, онлайн-магазины и многое другое.

MongoDB позволяет хранить данные в формате JSON-подобных документов, которые могут содержать любое количество полей и поддокументов. Это дает возможность хранить сложные иерархические данные, которые не могут быть представлены в привычных таблицах и строках. MongoDB также поддерживает масштабирование горизонтально (scale-out), что позволяет обрабатывать большие объемы данных и высокую нагрузку. Она также имеет множество функций, таких как автоматическое шардирование данных, репликацию и индексацию, что позволяет создавать высокопроизводительные и отказоустойчивые приложения.

Структура базы данных информационной системы туристического агентства представлен на рисунке 2.6.

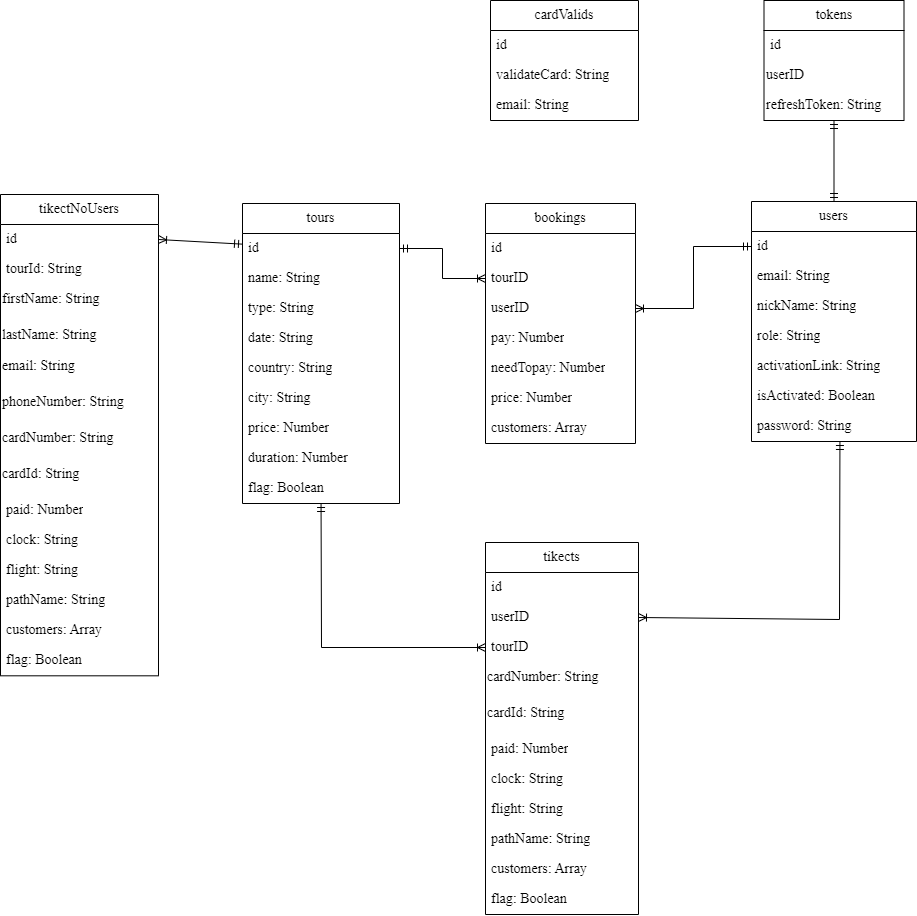


Рисунок – 2.6. Структура базы данных информационной системы туристического агентства

Коллекция "users" содержит информацию о пользователях, которая используется приложением в процессе авторизации пользователей в системе. Если пользователь вводит верные данные, то система дает ему доступ к системе. Структура представлена в таблице 2.1.

Таблица 2.1 – Структура коллекций «users»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| id | String | Идентификатор пользователя |
| email | String | Эл. почта пользователя |
| nickName | String | Имя пользователя |
| role | String | Роль пользователя |
| activationLink | String | Аккцивационная сылка |
| isActivated | Boolean | Активирован ли пользоаватель |
| password | String | Пароль пользователя |

Коллекция "tokens" содержит информацию о токене пользователя, которая используется приложением в процессе сеанса пользователей в системе. Структура представлена в таблице 2.2.

Таблица 2.2 – Структура коллекций «tokens»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| id | String | Идентификатор токена |
| userID | String | Идентификатор пользователя |
| refreshToken | String | Token пользователя |

Коллекция «tours» предназначена для добавления туров. В коллекции будет содержаться информация о названии тура, стране, город, длительность, цена, тип тура, а также о дате тура. Структура приведена в таблице 2.3.

Таблица 2.3 – Структура коллекций «tours»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| id | String | Идентификатор тура |
| name | String | Название тура |
| type | String | Тип тура |
| date | String | Дата тура |
| country | String | Страна тура |
| city | String | Город тура |
| price | Number | Цена за тур |
| duration | Number | Длительность тура |
| flag | Boolean | Состояния тура |

Коллекция «bookings» предназначена для бронирования тура, в коллекции будет содержаться информация о стоимости тура сколько оплачено взнос, сколько нужно будет оплатить и кто едет. коллекция приведена в таблице 2.4.

Таблица 2.4 – Структура коллекций «bookings»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| id | String | Идентификатор броня |
| tourId | String | Идентификатор тура |
| userId | String | Идентификатор пользователя |
| pay | Number | Оплачено за тур |
| needToPay | Number | Нужно оплатить |
| price | Number | Общая стоимость тура |
| customers | Array | Кто едет |

Коллекция «tickets» предназначена для генерирования билета, в коллекции будет содержаться информация о рейсе, кто едет, с какой карты был оплачено тур, и время. коллекция приведена в таблице 2.5.

Таблица 2.5 – Структура коллекций «tickets»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| id | String | Идентификатор билета |
| tourId | String | Идентификатор тура |
| userId | String | Идентификатор пользователя |
| cardNumber | String | С какой карты был оплачен тур |
| cardId | Number | Идентификатор подтверждения карты |
| clock | String | Время полета |
| flight | String | Рейс тура |
| pathName | String | Название билета |
| customers | Array | Кто едет |
| flag | Boolean | Состояние билет |

Коллекция «cardValids» предназначена для подтверждения оплаты картой, в коллекции будет содержаться информация о почте покупателя и идентификатор подтверждавший личность данного покупателя. коллекция приведена в таблице 2.6.

Таблица 2.6 – Структура коллекций «cardValids»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| id | String | Идентификатор cardValids |
| validateCard | String | Идентификатор подтверждения карты |
| email | String | Почта пользователя |

Коллекция «ticketNoUser» предназначена для не зарегистрированных пользователей при приобретении тура, в коллекции будет содержаться информация о фамилии, имени, почте, номер телефона, с какой карты был оплачен, покупателя и идентификатор подтверждавший личность данного покупателя, также рейс, кто едет состояние билет. коллекция приведена в таблице 2.7.

Таблица 2.7 – Структура коллекций «ticketNoUser»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Описание |
| id | String | Идентификатор билета |
| firstName | String | Фамилия пользователя |
| lastName | String | Имя пользователя |
| email | String | Почта пользователя |
| phoneNumber | String | Номер телефона пользователя |
| cardNumber | String | Номер карты пользователя |
| cardId | String | Номер подтверждения пользователя |
| paid | String | Сколько оплаченного тура |
| clock | String | Время полета |
| flight | String | Рейс |
| pathName | String | Название билета |
| customers | Array | Кто едет |
| flag | Boolean | Состояние билет |

Туристическое агентство, в котором используется модель нереляционной базы данных, имеет все коллекции данных, связанные между собой с помощью соответствующих структур, что обеспечивает логичную и удобную работу. Кроме того, такая модель данных позволяет более эффективно обрабатывать большие объемы информации, связанные с заказами туров и клиентской базой агентства.

## **2.4 Выводы и оценка результатов разработки**

В процессе проектирование информационной системы было предложена реализация структуры информационной систем, алгоритм работы и структура базы данных, В разделе структуры информационной систем были проведены примеры видов линейной структуры, линейной с ответвлением, древовидной структуры, также были реализованы диаграммы вариантов использования, последовательности, состояние и диаграмма паттерного проектирвоание MVC.

В разделе Алгоритм работы представлен пошаговый процедура, которая определяет набор инструкций. Было реализована алгоритм работы функции не авторизованного пользователя, также реализована структура базы данных

Выше описанные структуры значительно облегчают процесс создания информационной системы и выевления недочетов и ошибок на сдати реализации информационной системы.

**3 Анализ расчета надежности информационной системы туристического агентства**

**3.1 Расчет надежности информационной системы по модели сложности**

В ходе расчета модели сложности рассчитываются следующие метрики:

* метрики размера: объем программы *V*; потенциальный объем программы ;
* метрики сложности потока управления (по Джиблу): абсолютная сложность программы *CL*; относительная сложность программы *cl*; максимальный уровень вложенности оператора *CLI[1]*;
* метрики сложности потока данных: метрика Чепина *Q*;
* метрики стилистики и понятности программ (по Холстеду): теоретическая длина программы ; метрика корректности программы *L*; метрика корректности реальной программы ; число элементарных решений, принятых при написании программы *E*;
* объектно-ориентированные метрики: суммарная сложность всех методов класса *WMC*; глубина дерева наследований *DIT*; количество потомков *NOC*; сцепление между классами *CBO*; мощность множества классов *RFC*; недостаток сцепления методов *LCOM*.

Метрики размера программного средства (ПС). В данные метрики включают следующее[7]:

* объем программы *V*;
* потенциальный объем программы *V\**.

Объем V определяется по формуле:

|  |  |
| --- | --- |
| , | (3.1) |

где (*n*1 +*n*2) – словарь ПС; *n*1 – число уникальных операторов программы, включая символы-разделители, имена процедур и знаки операций (словарь операторов); *n*2 – число уникальных операндов программы (словарь операндов); (*N*1 + *N*2) – длина ПС *N*; *N*1 – общее число операторов в программе; *N*2 – общее число операндов в программе.

Подставив значения в формулу (3.1) получим выражение

*V* = (5463+7812) ∙ *log*2(4658 + 6647) = 13275 ∙ 13,465 = 178747,875

Потенциальный объем *V*\* рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (3.2) |

где *n*\* – теоретический словарь ПС.

Подставив значения в формулу (3.2) получим выражение

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

Метрики сложности потока управления. Используются следующие метрики предложенные Джиблом:

* абсолютная сложность управления *CL*, характеризующаяся количеством операторов условия;
* относительная сложность *cl*;
* насыщенность ПС операторами условия определяется отношением *CL* к общему числу операторов;
* максимальный уровень вложенности оператора условия *CLI*.

Суть метода состоит в оценке информационной прочности отдельно взятого программного модуля с помощью анализа характера использования переменных из списка ввода-вывода.

Для этого все переменные, составляющие список ввода-вывода, разбивается на четыре функциональные группы:

1. *Р* – вводимые переменные для расчетов и обеспечения вывода.
2. *М* – модифицируемые, или создаваемые внутри программы переменные.
3. *С* – переменные, участвующие в управлении работой программного модуля (управляющие переменные).
4. *Т* – не используемые в программе («паразитные») переменные.

Поскольку каждая переменная может выполнять одновременно несколько функций, необходимо учитывать ее в каждой соответствующей функциональной группе.

Сложность потока данных *Q* определяется выражением:

|  |  |
| --- | --- |
|  | (3.3) |

Подставив значения в формулу (3.3) получим следующее значение:

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

Метрики стилистики и понятности ПС. В данной группе используются три метрики:

* отклонение реальной *N* от теоретической *N*^ длины ПС ∆*N* (в %);
* уровень качества программирования *L* (уровень ПС);
* оценка интеллектуальных усилий на разработку ПС *E*.

Отклонение *N* от *N^* определяется выражением:

|  |  |
| --- | --- |
| , | (3.4) |

где *N*^ определяется выражением:

|  |  |
| --- | --- |
|  | (3.5) |

где *n*1 – словарь операторов; *n*2–словарь операндов ПС.

Таким образом, теоретическая длина *N*^ информационной системы передачи информации составит:

|  |  |
| --- | --- |
| 141161.336 |  |

Отклонение *N* от *N*^ определим по формуле (3.4):

Уровень качества программирования *L* расчитывается по формуле:

|  |  |
| --- | --- |
|  | (3.6) |

Подставив численные значения получим:

|  |
| --- |
|  |

Исходным для введения этой характеристики является предположение о том, что при снижении стилистического качества программирования уменьшается содержательная нагрузка на каждый компонент программы и, как следствие, расширяется объем реализации исходного алгоритма.

Интеллектуальные усилия на разработку ПС *Е* рассчитывается по формуле:

|  |  |
| --- | --- |
| , | (3.7) |

где *V* – объем программы; *V*\* – потенциальный объем программы.

Подставив численные значения в формулу (3.7) получим:

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

Объектно-ориентированные метрики.Рассчитаем метрики Мартина:

* центростремительное сцепление *Ca* – количество классов вне этой категории, которые зависят от классов внутри этой категории;
* центробежное сцепление *Ce* – количество классов внутри этой категории, которые зависят от классов вне этой категории;
* нестабильность *I*.

Нестабильность определяется по формуле:

|  |  |
| --- | --- |
|  | (3.8) |

Подставив численные значения получим:

Далее рассчитаем нормативный *xmin* и фактический *xфi* уровни для каждой из метрик по формулам:

|  |  |
| --- | --- |
| *,* | (3.9) |
|  | (3.10) | |

где *ai* – значение метрики, рассчитанное для конкретного ПС; *amin* – минимально возможное значение этой метрики для данного типа ПС; *amax* – максимально возможное значение этой метрики для данного ПС.

Также рассчитываем дискриминант *di* для каждой метрики по формуле:

|  |  |
| --- | --- |
| , | (3.11) |

Результаты вычислений приведены в таблице 3.1.

Таблица 3.1 – нормативные пределы метрик, результаты вычислений нормативных и фактических уровней метрик, дискриминантов метрик

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Метрики | Значение *ai* | *amin* | *amax* | *xmin* | *xϕi* | *di* |
| *V* | 178747,875 | 31 000 | 240 000 | 0.129167 | 0.744 | 0.0508 |
| *V\** | 49637,716 | 508 | 69 000 | 0.007362 | 0.71 | 0.0028 |
| *Q* | 1070 | 57 | 3425 | 0.016642 | 0.312 | 0.0372 | |
| *ΔN* | 9,4 | 4 | 35 | 0.114286 | 0.2685 | 0.3514 | |
| *L* | 0,277 | 0,003 | 3,1 | 0.000968 | 0.089 | 0.00987 | |
| *E* | 643679,95 | 23000 | 991000 | 0.023209 | 0.6495 | 0.0128 | |
| *Ca* | 23 | 11 | 141 | 0.078014 | 0.1631 | 0.434 | |
| *Ce* | 25 | 11 | 210 | 0.052381 | 0.1190 | 0.409 | |
| *I* | 0,52 | 0 | 2 | 0 | 0.26 | 0 | |

Рассчитываем риск снижения надежности работы программного средства *R* по формуле:

|  |  |
| --- | --- |
|  | (3.12) |

где *λi* – весовые коэффициенты для конкретных метрик. Отражает на сколько та или иная метрика имеет больший вес для надежности ПС и должна удовлетворять условию:

|  |  |
| --- | --- |
|  | (3.13) |

где *N* – количество метрик, используемое при расчете риска снижения надежности.

Для упрощения задачи примем равнозначным вклад каждой метрики в результат расчета риска снижения надежности.

Таким образом, по формуле (3.12) для данного программного продукта риск снижения надежности равен:

Следовательно, надежность равна:

N = 10,056 = 0,9432

## **3.2 Расчет надежности информационной системы по модели Джелинского - Моранды**

Модель Джелинского – Моранды строится на основе следующих допущений[6]:

1 Интенсивность обнаружения ошибок λ(t) пропорциональна текущему числу ошибок в программном средстве, т. е. числу оставшихся ошибок.

2 Все ошибки одинаково вероятны, и их появления независимы.

3 Каждая ошибка имеет один и тот же порядок серьезности.

4 Время до следующего отказа (ошибки) распределено экспоненциально.

5 Программное средство функционирует в среде, близкой к реальной.

6 Ошибки постоянно корректируются без внесения в программное средство новых.

7 λ(t) = const в интервале между двумя соседними ошибками.

В соответствии с этими допущениями интенсивность возникновения ошибок в программном средстве можно представить в виде

|  |  |
| --- | --- |
|  | (3.14) |

где *t* – произвольное время между обнаружением (*i*-1) и *i*-й ошибок; *K* – неизвестный коэффициент; *B* – неизвестное общее число ошибок в программном средстве.

А вероятность безотказной работы программного средства рассчитывается согласно выражению

|  |  |
| --- | --- |
|  | (3.15) |

Для нахождения *B* используют выражения

|  |  |
| --- | --- |
|  | (3.16) |

|  |  |
| --- | --- |
|  | (3.17) |

|  |  |
| --- | --- |
|  | (3.18) |

|  |  |
| --- | --- |
|  | (3.19) |

|  |  |
| --- | --- |
|  | (3.20) |

где *m* ≥ *n*+1 – число прогнозируемых ошибок; *n –* количество выявленных ошибок.

Неизвестный коэффициент *K* рассчитывают по формуле

|  |  |
| --- | --- |
|  | (3.21) |

На этапе отладки программного средства за 179 день было выявлено 10 ошибок. Исходные данные сведены в таблицу 3.2 в виде интервалов времени *Хi* (дни) между соседними ошибками (*i* – номер ошибки). Необходимо найти вероятность безотказной работы программного средства.

Таблица 3.2 – Исходные данные для расчетов

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *i* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| *Xi,* день | 2 | 9 | 11 | 14 | 16 | 18 | 22 | 27 | 33 | 39 |

Подставив численные значения в формулу (3.19) получим *A=*7,1712

Осуществим расчеты по формулам (3.17), (3.18) и сведем их в таблицу 3.3.

Таблица 3.3 – Результаты вычислений

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *i* | *Xi* | *i∙Xi* | *m* | *gn*(*m, A*) | *fn*(*m*) | *|fn*(*m*) – *gn*(*m, A*)| |
| 1 | 2 | 2 | 11 | 2.553191 | 2.928968254 | 0.375777 |
| 2 | 9 | 18 | 12 | 2.033898 | 2.019877345 | 0.01402 |
| 3 | 11 | 33 | 13 | 1.690141 | 1.603210678 | 0.08693 |
| 4 | 14 | 56 | 14 | 1.445783 | 1.346800422 | 0.09898 |
| 5 | 16 | 80 | 15 | 1.263158 | 1.168228993 | 0.09493 |
| 6 | 19 | 114 | 16 | 1.121495 | 1.03489566 | 0.0866 |
| 7 | 22 | 154 | 17 | 1.008403 | 0.930728993 | 0.07767 |
| 8 | 27 | 216 | 18 | 0.916031 | 0.84669538 | 0.06934 |
| 9 | 33 | 297 | 19 | 0.839161 | 0.777250935 | 0.06191 |
| 10 | 39 | 390 | 20 | 0.774194 | 0.718771403 | 0.05542 |

Из таблицы 3.3 видно, что наилучшим решением для уравнения (3.16) является *m=*12, в таком случае согласно выражению (3.20) *B* = 14.

Подставив в формулу (1.29) рассчитанные значения получим

Далее согласно с формулой (1.22) рассчитаем интенсивность возникновения ошибок после того, как обнаружена ( *i –* 1) ошибка.

По формуле (3.14) вероятность безотказной работы программного средства *P*(*t*) = 0,94.

## **3.3 Расчет надежности информационной системы по модели Муса и выводы**

В модели Муса надежность ПС на этапе эксплуатации оценивается по результатам тестирования [7].

Средняя наработка до отказа после тестирования определяется по формуле:

|  |  |
| --- | --- |
| , | (3.22 ) |

где τ0 – средняя наработка до отказа до начала тестирования; *С* – коэффициент, учитывающий уплотнение тестового времени по сравнению с временем реальной эксплуатации.

Неизвестный параметр τ0 можно оценить из следующего соотношения:

|  |  |
| --- | --- |
|  | (3.23) |

Средняя скорость исполнения одного оператора ПС рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (3.24) |

Общее время тестирования *Т + ΔТ* должно удовлетворять соотношению:

|  |  |
| --- | --- |
|  | (3.25) |

Надежность ПС для периода эксплуатации *t* определяются по формуле:

|  |  |
| --- | --- |
|  | (3.26) |
|  |  |

Для данного программного модуля длительности этапов тестирования составляют *t1* = 19 часов, *t2* = 23 часов, *t3* = 28 часов. Число отказов на первом этапе *m*1 = 3, на втором – *m*2 = 5, на третьем – *m*3 = 1. Средняя скорость исполнения ПС *A* = 104 операторов/час, количество операторов в ПС *B* = 740. Период эксплуатации *t* = 189 часов.

Найдем среднюю частоту выполнения одного оператора с помощью формулы (3.24):

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

Первоначальное количество ошибок в программном средстве (из модели Шумана) *N* = 4. Коэффициент проявления ошибок *K* для данного программного средства 1,5 ∙ 10-6, в таком случае τ0 можно рассчитать по формуле (3.23):

Один час тестирования соответствует 6 часам работы, следовательно, значение коэффициента *С* = 5. Тогда средняя наработка до отказа после тестирования на этапе эксплуатации ПО согласно с формулой (3.22) равна:

Найдем надежность программного средства для периода эксплуатации *t* = 189 часов по формуле (3.26):

**4 Тестирование информационной системы туристического агентства**

Тестирование - это процесс проверки программного обеспечения, приложения или системы на соответствие заданным требованиям и ожиданиям. Это важный этап в разработке любого программного продукта, который помогает обнаруживать ошибки и дефекты, а также улучшать качество и производительность системы. Тестирование включает в себя создание тестовых сценариев, выполнение тестовых случаев, анализ результатов и создание отчетов о тестировании. В зависимости от типа тестируемой системы и целей тестирования, могут использоваться различные методы и подходы к тестированию, такие как функциональное тестирование, интеграционное тестирование, регрессионное тестирование, тестирование производительности и другие.

Далее, основная цель тестирования - убедиться, что программа или система работают корректно и соответствуют требованиям. Это также позволяет выявить и устранить ошибки и недочеты, которые могут привести к нежелательным последствиям при использовании системы в реальных условиях.

Виды тестирования в зависимости от глубины тестового покрытия: Smoke, MAT, AT.

Тестовое покрытие – одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода.

Smoke Test – поверхностное тестирование для определения пригодности

сборки для дальнейшего тестирования, должно покрывать базовые функции программного обеспечения; уровень качества: Acceptable/Unacceptable.

Minimal Acceptance Test (MAT, Positive Test) – тестирование системы или ее части только на корректных данных/сценариях; уровень качества: High/Medium/Low.

Acceptance Test (AT) – полное тестирование системы или ее части как на корректных (Positive Test), так и на некорректных данных/сценариях (Negative Test); уровень качества: High/Medium / Low. Тест на этом уровне покрывает все возможные сценарии тестирования: проверку работоспособности модулей при вводе корректных значений; проверку при вводе некорректных значений; использование форматов данных, отличных от тех, которые указаны в требованиях; проверку исключительных ситуаций, сообщений об ошибках; тестирование на различных комбинациях входных параметров; проверку всех классов эквивалентности; тестирование гран

Функциональное тестирование – тестирование, основанное на сравнительном анализе спецификации и функциональности компонента или системы.

Тестирование взаимодействия с графическим интерфейсом

Для тестирования данного раздела был разработан ряд тест-кейсов, приведённый в таблице 4.1. Тестирование производилось путем выполнения данных кейсов и сравнения полученных результатов с ожидаемыми.

Таблица - 4.1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Приоритет | Требование | Модуль | Подмодуль | Описание  Тест- кейса | Ожидаемые результаты | Тест пройден |
| 1 | Smoke | Корректной отображения главной страницы | Главная страница | Основная часть | Графический пользовательский интерфейс | Корректно отображает Графически  интерфейс Главной страницы | Да |
| 2 | АТ | Регистрация пользователя | Главная страница | Форма регистрация полей для ввода данных | Ввести в  форму  регистрации почту пользователя, содержащее некорректные данные | Получено оповещение о некорректном вводе почты | Да |
| 3 | МAT | Авторизация в системе | Главная | Форма авторизация полей для ввода данных | Ввести в  форму  авторизации данный пользователя, содержащее корректные данные | В ход в личный кабинет | Да |
| 4 | Smoke | Корректное отображения страницы туров | Страница туров | Список всех туров | Отображения туров на странице | Отображения список туров на странице | Да |
|  |  |  |  |  |  |  |  |

Хорошо спланированное и хорошо выполненное тестирование позволяет увеличить качество и надежность программного обеспечения, ускорить процесс разработки и снизить риски при использовании системы в реальных условиях. В данной тестовом разделе были выполнены все поставленные задач.

## **5 Технико-экономическое обоснование разработки информационной системы туристического агентства**

### **5.1 Характеристика программного продукта**

Целью разработки является создание информационной системы (ИС) для туристического агентства, которое позволит автоматизировать процессы продажи туров, управления бронирования туров. Разработанное программное средство будет использоваться в организации для улучшения эффективности работы туристического агентства и удовлетворения потребностей клиентов.

Актуальная потребность в разработке программного средства обусловлена увеличением объемов продаж и необходимостью улучшения качества обслуживания клиентов. Введение информационной системы позволит сократить время на обработку, уменьшить количество ошибок при бронировании тура. Также информационной системы позволит клиентам совершать онлайн покупки тура, что значительно расширит аудиторию информационной системы и повысит его конкурентоспособность.

Дополнительным фактором, обусловившим необходимость разработки программного средства, является рост конкуренции в отрасли отдыха и туризма. Согласно статистике, на 2021 год, более 2 миллиардов человек совершают покупки онлайн, что составляет около 27% населения мира. Также отмечается рост количества покупателей онлайн на 10-15% ежегодно, что свидетельствует о необходимости наличия у турагентство эффективной онлайн-платформы.

В результате внедрения программного средства, турагентство сможет значительно повысить эффективность работы и удовлетворить потребности клиентов. Ожидается увеличение объемов продаж на 20% в первый год использования системы и увеличение числа клиентов на 30%.

Экономическая целесообразность инвестиций в разработку программного продукта определяется на основе расчета и оценки следующих показателей:

– чистая дисконтированная стоимость (ЧДД);

– срок окупаемости инвестиций (ТОК);

– рентабельность инвестиций (Р**и**).

### **5.2 Расчет инвестиций в разработку программного средства**

Часовая заработная плата определяется путем деления месячной заработной платы на количество рабочих часов в месяце. Среднемесячная расчетная норма рабочего времени на 202 год составляет 21 дня.

Расчет основной заработной платы () исполнителей:

(5.1)

где n – количество исполнителей, занятых разработкой программных

средств (ПС);

– коэффициент премий;

– часовая заработная плата исполнителя (руб);

– трудоемкость работ, выполняемых исполнителем *i*-й категории, определяется исходя из сложности разработки программного обеспечения и объема выполняемых им функций (час).

Список категорий исполнителей, их месячная и часовая заработная плата, а также трудоемкость работ представлены в таблице 5.1.

Согласно формуле 5.1 был произведен расчет основной заработной платы с учетом коэффициента премий, равного 1,5.

Таблица 5.1 – Расчет затрат на основную заработную плату команды

разработчиков

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория исполнителя | Месячная заработная плата, р. | Часовая заработная плата, р. | Трудоемкость работ, ч. | Итого, р. |
| Руководитель  проекта | 1650 | 9,82 | 65 | 638,3 |
| Программист | 1850 | 11,01 | 120 | 1321,2 |
| Тестировщик | 1000 | 5,95 | 35 | 208,25 |
| Дизайнер | 1200 | 7,14 | 57 | 406.98 |
| Итого |  | | | 2574,73 |
| Премия |  | | | 1287,36 |
| Всего затраты на основную заработную плату разработчиков | | | | 3862,09 |

Дополнительная заработная плата разработчиков определяется по формуле:

(5.2)

где ‒ норматив дополнительной заработной платы, (10%).

После подстановки значений в формулу (5.2) дополнительная заработная плата составит:

386,20 р.

Для определения отчислений на социальные нужды воспользуемся формулой:

(5.3)

где ‒ ставка отчислений в ФСЗН и Белгосстрах.

На момент написания проведения расчетов по формуле (5.3), в Республике Беларусь ставка отчислений в ФСЗН и Белгосстрах составляла 34,6%. Согласно формуле (5.3) размер отчислений в фонд социальной защиты населения и на обязательное страхование составит:

1469,90

Затраты по статье «Прочие расходы» (Pпр), связанные с необходимостью содержания аппарата управления, а также расходами на общехозяйственные нужды, рассчитываются по формуле:

, (5.4)

где Нпр – норматив накладных расходов (30%).

Расходы по статье «Накладные расходы» равны:

= 1158,62 р.

Общая сумма инвестиций (затрат) на разработку программного продукта рассчитывается по формуле:

(5.5)

Общая сумма инвестиций (затрат) на разработку:

3862,09+386,20+1469,90+1158,62 = 6876,81 р.

|  |  |  |
| --- | --- | --- |
| Наименование статьи затрат | Формула/таблица для расчета | Сумма, р. |
| 1. Основная заработная плата разработчиков | Табл. 5.1 | 3862,09 |
| 2. Дополнительная заработная плата разработчиков | Формула (5.2) | 386,20 |
| 3. Отчисления на социальные нужды | Формула (5.3) | 1469,90 |
| 4. Прочие расходы | Формула (5.4) | 1158,62 |
| 5. Общая сумма инвестиций (затрат) на разработку | Формула (5.5) | 6876,81 |

### **5.3 Расчет экономического эффекта от использования программного средства**

Экономическим эффектом в результате использования программного средства является прирост чистой прибыли, полученный за счет:

‒ экономии затрат на заработную плату с начислениями на заработную плату служащих в связи с сокращением их численности;

‒ экономии материальных затрат, электроэнергии, затрат на оплату труда и пр. в результате снижения брака, технологических потерь;

‒ снижения себестоимости продукции (работ, услуг) в результате роста производительности труда;

‒ снижения затрат на заработную плату с начислениями на заработную плату основных производственных рабочих;

‒ снижения материальных затрат на производство продукции (работ, услуг) и т. п.

Экономия на заработной плате и начислениях на заработную плату сотрудников за счет снижения трудоемкости работ:

(5.6)

где – коэффициент премий;

, ‒ трудоемкость выполнения работ сотрудниками до и после внедрения программного средства, ч;

‒ часовой оклад (часовая тарифная ставка) сотрудника, использующего программное средство, р.;

– плановый объем работ, выполняемых сотрудником;

– норматив дополнительной заработной платы (10 %);

– ставка отчислений от заработной платы, включаемых в себестоимость (34,6 %).

Экономия на заработной плате и начислениях на заработную плату сотрудников за счет снижения трудоемкости работ:

Экономическим эффектом при использовании программного средства является прирост чистой прибыли, полученной за счет экономии на текущих затратах предприятия, который рассчитывается по формуле:

(5.7)

где = – экономия на текущих затратах при использовании программного средства, р.;

– прирост текущих затрат, связанных с использованием программного средства (затраты на сопровождение программного средства, затраты на интернет-трафик и т. п.), р.;

‒ ставка налога на прибыль согласно действующему законодательству (18 %).

Экономическим эффектом при использовании программного средства

### **5.4 Расчет показателей экономической эффективности разработки и использования программного средства**

Приведение доходов и затрат к настоящему моменту времени осуществляется посредством так называемого дисконтирования, т. е. путем их умножения на коэффициент дисконтирования, который определяется по формуле:

(5.8)

где 𝑑 – требуемая норма дисконта, которая по своему смыслу соответствует устанавливаемому инвестором желаемому уровню рентабельности инвестиций, доли единицы; t – порядковый номер года, доходы и затраты которого приводятся к расчетному году; – расчетный год, к которому приводятся доходы и инвестиционные затраты (=1).

Чистый дисконтированный доход определяется по формуле:

(5.9)

Результаты расчета показателей эффективности сведены в таблице 5.2.

Таблица 5.2 – Расчет эффективности инвестиций (затрат) в реализацию проектного решения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Показатель | Расчетный период | | | | |
| 2024 | 2025 | 2026 | 2027 |
| 1. Прирост чистой прибыли, р. | 2584,08 | 2868,34 | 3183,85 | 3534,08 |
| 2. Дисконтированный результат, р. | 2584,08 | 2581,50 | 3534,07 | 3922,82 |
| 3. Инвестиции (затраты) в реализацию проектного решения, р. | 6876,81 |  |  |  |
| 4. Дисконтированные инвестиции, р. | 6876,81 |  |  |  |
| 5. Чистый дисконтированный доход по годам, р. | -4292,73 | -1711,23 | 1822,84 | 3922,82 |
| 6. Чистый дисконтированный доход нарастающим итогом, р. | -4290,93 | -2579,7 | -756,86 | 3165,96 |
| 7. Коэффициент дисконтирования, доли единицы | 1 | 0,90 | 0,81 | 0,73 |

Для определения экономической целесообразности инвестиций в разработку и использование необходимо определить чистый дисконтированный доход, срок окупаемости инвестиций и их рентабельность.

Рассчитаем среднюю норму прибыли/рентабельности инвестиций () :

(5.10)

где n – расчетный период, лет;

– затраты (инвестиции) в году t, так как затраты уходят сразу на разработку то независимо затраты будут равны ;

– прирост чистой прибыли в году t в результате реализации проекта, р.

Средняя норма прибыли/рентабельности инвестиций:

Рассчитаем оценку экономической эффективности разработки :

(5.11)

Время окупаемости инвестиции при внедрении программного средства:

В результате технико-экономического обоснования разработки программного продукта были получены следующие значения показателей эффективности:

– чистый дисконтированный доход (ЧДД) за четыре года составил 3165,96 р.;

– окупаемость инвестиций наступает на четвёртый год;

– рентабельность инвестиций по проекту составляет .

Таким образом, разработка и продажа программного продукта является эффективным и инвестиции в его разработку целесообразно осуществлять.

1. **Охрана труда. разработка мероприятий по повышению работоспособности и производительности труда разработчика информационной системы поддержки деятельности туристического агентства**

Целью дипломного проекта явилась разработка информационной системы (ИС) [7], для туристического агентства. Данная система представляет собой веб-приложение, предназначенный для автоматизации определенных задач в заданной области информационной системы. Она включает в себя различные компоненты, такие как серверная часть, клиентская часть и базы данных, которые взаимодействуют друг с другом для реализации необходимых функций.

Актуальностью разработки данной информационной системы заключается в упрощении процесса оформления туристических путевок и повышении эффективности работы турагентства. Для достижения этой цели были учтены основные принципы удобства использования и быстроты доступа к необходимой информации, что обеспечивает удобное и продуктивное использование данной информационной системы.

Во время разработки ИС [7], условия труда играют важную роль в производительности разработчика ИС. Рабочее место, здоровье и настроение работника, доступность необходимых ресурсов и технологий - все эти факторы могут повлиять на качество и скорость работы разработчика.

Физические условия рабочего места могут включать такие факторы, как, неудобное расположение оборудования, шум и плохое освещение. Неудобная мебель и оборудование могут вызвать напряжение в мышцах, боли в шее, спине и руках, а также усталость глаз. Шум может вызвать раздражение и отвлечение, что может снизить концентрацию и эффективность работы. Плохое освещение может привести к ухудшению зрения и быстрой усталости глаз. Работающие в таких условиях, могут столкнуться с уменьшением производительности, ухудшением здоровья и настроения.

В целом, проблемы, связанные с условиями труда [8], могут значительно повлиять на производительность и качество работы разработчика информационной системы. Решение этих проблем может повысить производительность и эффективность работы, а также улучшить качество жизни разработчика и снизить риск заболеваний и стресса.

В процессе разработки ИС часто работал в кабинете, где расположено мое рабочее место за компьютером. Большую часть времени я проводил на небольшой площади, где находится мой стол и кресло. Работа в основном связана с использованием монитор, компьютер, мыши и клавиатуры, поэтому эти инструменты являются основными элементами моего рабочего места.

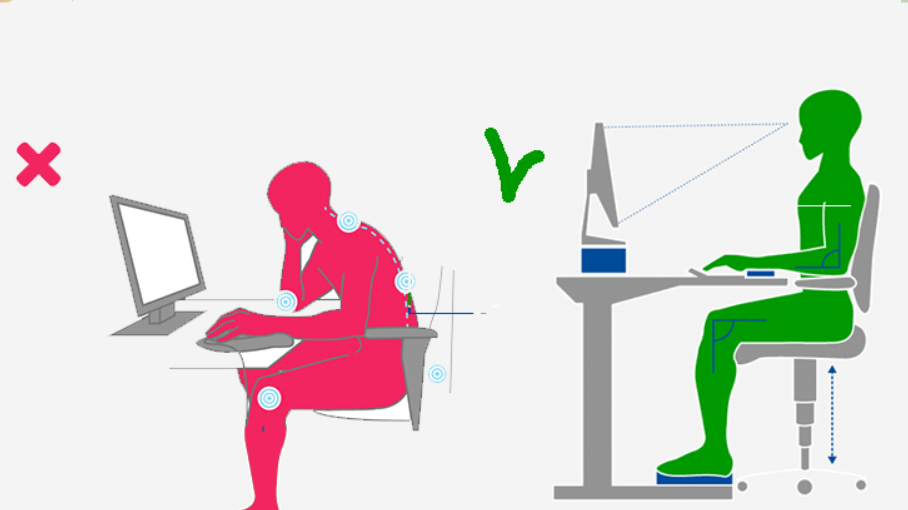
Уделено большое внимание комфорту рабочего места. При разработке ИС использовал компьютер - это мощная машина, оснащенная достаточным количеством оперативной памяти и дискового пространства для обработки больших объемов данных и запуска различных инструментов разработки.

Монитор имеет высокое разрешение и отличную цветопередачу, что обеспечивает максимально комфортное визуальное восприятие информации. Клавиатура и мышь эргономичны и приспособлены для интенсивной работы, чтобы максимально снизить риск заболеваний, связанных с работой за компьютером.

А также кресло имеет регулируемую высоту и угол наклона спинки, а стол достаточно просторный, чтобы обеспечить комфортную работу с компьютером.

Время разработки ИС велась три месяца, все рабочие дни недели. Начала рабочего дня начиналась с 9:00 до 18:00 обеденное перерыв входит в это время, работа за компьютером составляло 8 часов в день.

Условия труда [5,8] на конкретном рабочем месте могут оказывать существенное влияние на производительность работника, особенно если речь идет о разработчике информационной системы. Важным фактором является положение тела и рабочая поза человека. Неудобная поза может привести к чрезмерным статическим нагрузкам на позвоночник, невротическим состояниям и патологическим нарушениям со стороны опорно-двигательного аппарата и внутренних органов. Также важно организовать рабочую зону таким образом, чтобы центр тяжести тела человека был расположен в пределах площади его опоры. Неустойчивое положение тела может привести к быстрому утомлению, травмам и заболеваниям опорно-двигательного аппарата, что также может отрицательно сказаться на производительности работника. Поэтому важно учитывать эти факторы при организации рабочего места для разработчика информационной системы.



а) б)

Рисунок 6.1 – Схема биомеханического анализа рабочей позы при неустойчивом (а), (б) и устойчивом положениях.

Выбор оптимальной рабочей позы (сидя, стоя, сидя-стоя) зависит от физической нагрузки на работника. При работе с небольшими усилиями (не более 50 Н) рекомендуется использовать рабочую позу сидя. При работе с усилиями от 50 до 100 Н, рабочую позу можно выбрать на свое усмотрение, так как эффект на организм при сидении или стоянии будет примерно одинаковым. Однако, при работе с усилиями, превышающими 100 Н, рекомендуется работать стоя. [5]

Работа стоя может быть эффективной для наладки оборудования, но увеличивает нагрузку на нижние конечности и требует больше энергии. Работа в позе сидя более рациональна и менее утомительна, но может приводить к застою в органах таза. Смена позы помогает перераспределить нагрузку и улучшить кровообращение [6], поэтому рекомендуется предусматривать возможность для работников менять положение тела. Для профессионалов в области разработки ИС важно оптимизировать условия труда, чтобы избежать заболеваний и повысить продуктивность.

Кресло является неотъемлемой частью рабочего места, и его удобство имеет большое значение. Для комфортной работы сотрудник должен иметь кресло с удобным сиденьем, поддержкой спины и подлокотниками, который позволит ему сидеть в течение длительного времени.

Существует множество различных типов кресла, которые могут помочь уменьшить нагрузку на спину и шею. Некоторые кресла позволяют регулировать высоту, наклон спинки и сиденья, что позволяет сотруднику настроить их под свои индивидуальные потребности.



Рисунок 6.2 – Кресло

Одним из ключевых элементов правильного рабочего места является стол. Стол должен иметь достаточную площадь, чтобы разместить компьютер, клавиатуру, мышь, документы и другие необходимые материалы. Он должен быть достаточно прочным и стабильным, чтобы не дрожал при нажатии на клавиши клавиатуры или движении мыши. Высота стола должна соответствовать росту сотрудника, чтобы предотвратить излишнее напряжение в шее, плечах и спине. Регулируемая высота стола может помочь удовлетворить потребности сотрудников с разным ростом.

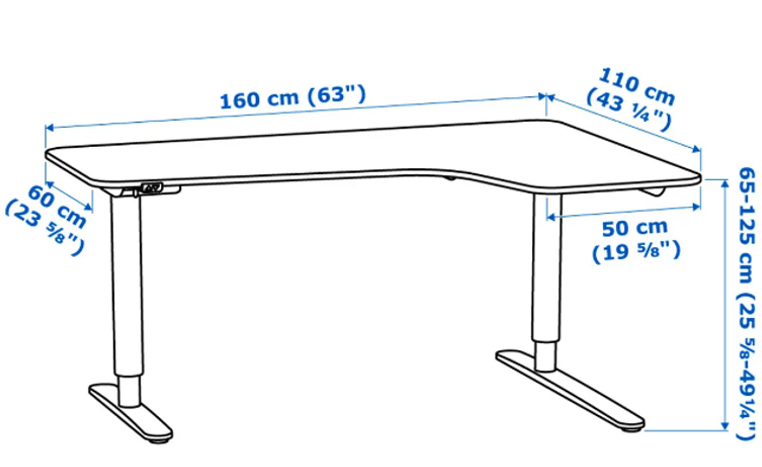


Рисунок 6.3 – Пример стола для организации рабочего места

Для улучшения условий труда и повышения работоспособности разработчика информационной системы на его рабочем месте, можно применить следующие пути:

1. Эргономичная организация рабочего места. Рабочее место должно быть организовано таким образом, чтобы соответствовать физиологическим особенностям разработчика. Например, стол и кресла должны иметь оптимальные размеры, высота стола должна соответствовать росту разработчика, кресла должен иметь удобное сиденье с поддержкой спины и подлокотниками.

2. Регулярные перерывы. Длительное время в одной и той же позе может привести к усталости и боли в мышцах и суставах. Поэтому необходимо делать регулярные перерывы в течение рабочего дня, чтобы размяться и расслабиться. Рекомендуется делать перерыв каждые 45-60 минут.

3. Установка правильной освещенности. Освещение должно быть достаточным, равномерным и не вызывать ослепления. Разработчик должен иметь возможность регулировать яркость света на своем рабочем месте.

4. Регулярные упражнения для разминки. Регулярные упражнения для разминки помогут снять напряжение с мышц и напряженности суставов, уменьшить усталость и улучшить кровообращение.

5. Использование специализированного оборудования. Существует большое количество специального оборудования, такого как эргономические клавиатуры и мыши, которые могут помочь уменьшить нагрузку на суставы и мышцы рук и предотвратить синдром запястного канала.

В результате работы даны предложения по повышению работоспособности и производительности труда разработчика информационной системы, оптимизация условий труда для разработчиков ИС является важным фактором для повышения эффективности работы и предотвращения заболеваний. Для улучшения условий труда на конкретном рабочем месте необходимо уделить внимание выбору эргономичной мебели и оборудования, оптимизации освещения, а также организации рабочего пространства и разработке эффективных рабочих процессов. Регулярные перерывы и упражнения, направленные на разгрузку мышц и снятие напряжения, также могут помочь снизить риск возникновения заболеваний.

# **ЗАКЛЮЧЕНИЕ**

В результате проделанной работы был проведен анализ предметной области, а именно системы управление туристической агентством.

В программе реализованы такие задачи, как:

* в программе реализован множественный функционал, в разной степени доступный пользователям в зависимости от их прав;
* программный продукт позволяет сотрудникам записать данные;
* данный продукт может быть использован, как сотрудниками клиники, так и простыми пользователями;
* предусмотрена обработка ошибок, которые могут возникнуть в процессе работы;
* применение базы данных в качестве хранилища информации позволяет оптимально и эффективно хранить информацию, ее структурировать;
* безопасность обеспечивается за счет разграничения прав доступа, системы авторизации;
* для удобства пользователя, программа располагает достаточно понятным, наглядным и удобным интерфейсом взаимодействия пользователя с программой.

Помимо бизнес-логики также были определены требования к работе, внешнему виду и содержанию системы, а также были определены основные сущности системы для организации моделей. В процессе разработки, приложение было протестировано, в результате не было выявлено серьезных ошибок, а все выявленные незначительные ошибки были устранены.

Разработанная информационная система позволит повысить производительность и качество работы туристической агентства, поможет работникам туристической агентства в решении практических задач, позволит снизить трудозатраты, улучшит качество и повысит скорость обслуживания клиентов.

В будущем приложение может быть усовершенствовано посредством улучшения и расширения информационной модели системы, оптимизации имеющихся функций и добавления новых.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. [1] Национальная стратегия устойчивого развития Республики Беларусь до 2035 года [Элекронный ресурс]. – Режим доступа: [NSUR-2035-1.pdf (economy.gov.by)](https://economy.gov.by/uploads/files/ObsugdaemNPA/NSUR-2035-1.pdf). – Дата доступа: 02.04.2023.
2. [2] Миронова Н.А. Туристская отрасль в контексте цифровой экономики [Элекронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/turistskaya-otrasl-v-kontekste-tsifrovoy-ekonomiki>. – Дата доступа: 05.03.2023.
3. [3] Костюченко Е.П. Возможности цифровизации в туриндустрии // Интеллектуальные ресурсы – региональному развитию. 2022. № 1. С. 249-255.
4. [4] Григорьева Н.С., Колычева Ж.Я., Дынник Д.И. Цифровая трансформация системы управления туристским комплексом // Наука и образование: хозяйство и экономика; предпринимательство; право и управление. – 2021. – № 1 (128). – С. 40-44.
5. [5] Иконников, В. Ф.Информационные технологии в индустрии туризма : учеб.-метод. пособие / В. Ф. Иконников, М. Н. Садовская. – Минск : РИПО, 2014. – 78 с.
6. [6] Туризм и туристические ресурсы в Республике Беларусь : буклет / под ред. И. В. Медведевой. ‒ Минск : Национальный статистический комитет Республики Беларусь, 2021. – 31 с.
7. [7] Козлов М.В., Брыксин В.Е., Немчинова Е.Е. Факторы влияния цифровизации на туристский бизнес // Вестник Алтайской академии экономики и права. – 2023. – № 2-1. – С. 52-56.
8. [8]Toursoyuz.by [Электронный ресурс]. – Режим доступа: <https://www.toursoyuz.by/2021/12/20/rejtingi-belorusskih-turoperatorov-s-kem-sotrudnichali-agenty/>. – Дата доступа: 01.04.2023.
9. [9]Viapol.by [Электронный ресурс]. – Режим доступа: https://viapol.by/index.htm. – Дата доступа: 01.04.2023.
10. [10]Tez-tour [Электронный ресурс]. – Режим доступа: https://www.tez-tour.com/– Дата доступа: 01.04.2023.
11. [11]T-v.by [Электронный ресурс]. – Режим доступа: https://t-v.by/about-company/about-company/. – Дата доступа: 01.04.2023.
12. [webpay.by](http://www.webpay.by) [Электронный ресурс]. – Режим доступа: https://webpay.by/ – Дата доступа: 01.04.2023.
13. Михнюк Т.Ф. Охрана труда. Учебное пособие для вузов. – Минск: ИВЦ Минфин, 2009.
14. Аттестация рабочих мест по условиям труда //Библиотека журнала «Ахова працы». 2003. № 10.
15. Понятие информационных систем <https://www.vsavm.by/knigi/kniga3/220.html>
16. MERN https://blog.logrocket.com/mern-stack-tutorial/
17. База данных mongodb https://www.mongodb.com/
18. Express фреймворк https://developer.mozilla.org/ru/docs/Learn/Server-side/Express\_Nodejs
19. React https://ru.legacy.reactjs.org/
20. Nodejs https://metanit.com/web/nodejs/1.1.php
21. Redux https://redux.js.org/
22. Bootstrap https://getbootstrap.com/

# **Приложения A**

Структурная схема

**Приложения В**

Блок-схема алгоритма оформления заказа

# **Приложения Д**

Эскизы рабочих окон программы

# **Приложения Е**

Результаты расчета надежности информационной системы

**Приложения Б**

Структура базы данных информационной системы туристического агентства

**Приложения Г**

Диаграмма вариантов использование