**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;

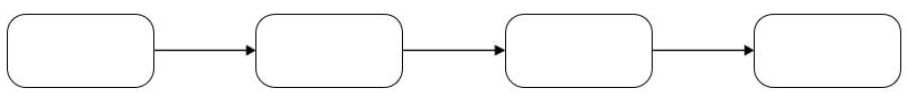
— создание базиса для планирования разработки проекта;

— определение пользовательского интерфейса.

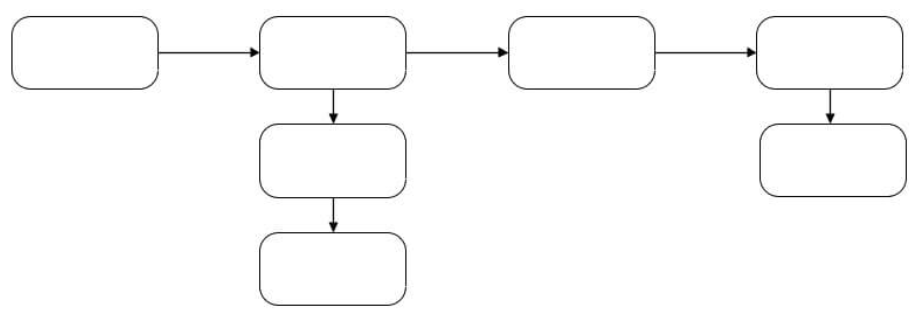
Для достижения этих целей используются Структурная система линейная, линейная с ответвлениями, древовидную, также диаграммы вариантов использования.

Линейная структура сайта является наиболее простой и представляет собой последовательное расположение веб-страниц, которые пользователь просматривает подобно слайд-шоу. На таких сайтах отсутствует иерархия контента и все страницы имеют равный вес, призванный привлечь внимание каждого посетителя. Несмотря на простоту реализации, линейная структура имеет множество недостатков, что сильно ограничивает ее область применения. Обычно линейная структура используется на сайтах, ориентированных на создание имиджа, презентации продуктов и услуг, а также в онлайн-учебниках.

На рисунке 2.1 представлена схема линейной структуры сайта и линейная структура с ответвлениями



а)



б)

Рисунок 2.1 – Схема линейной структуры (а) и структуры линейной с ответвлением (б)

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 2.2 представлена схема древовидной структуры сайта [1].

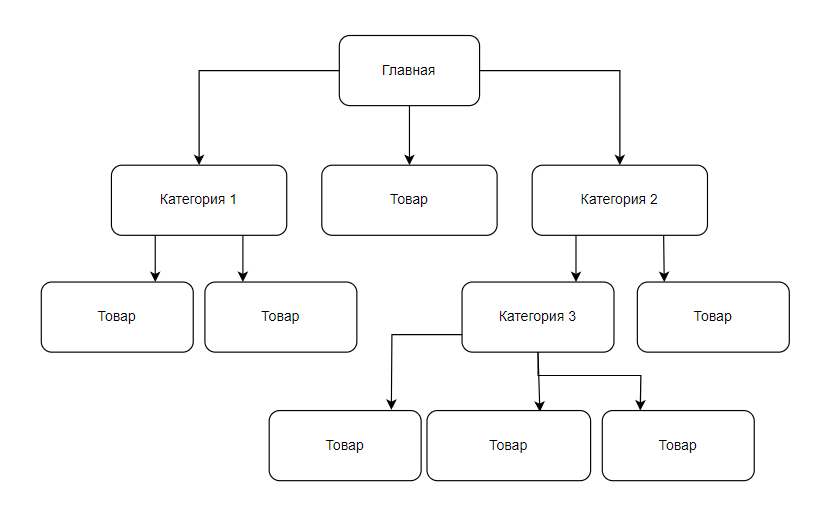


Рисунок 2.2 – Схема древовидной структуры

UML (Use case diagrams). 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 рисунках 2.3

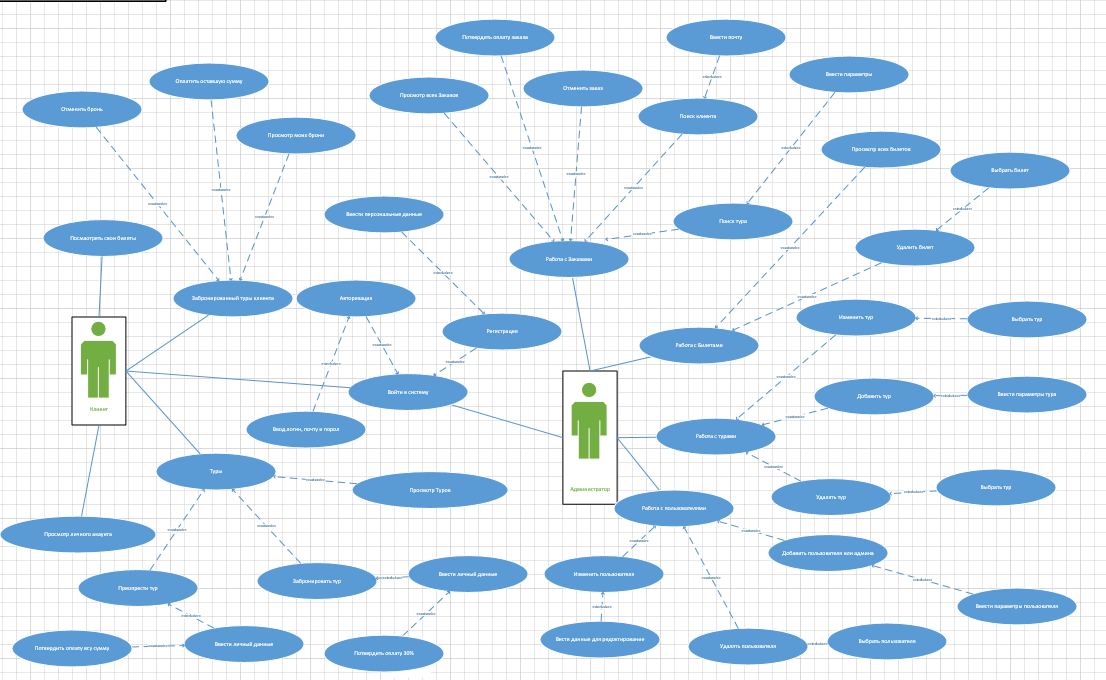


Рисунок 2.3 вариантов использования.

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

**Диаграмма последовательности**

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 2.4

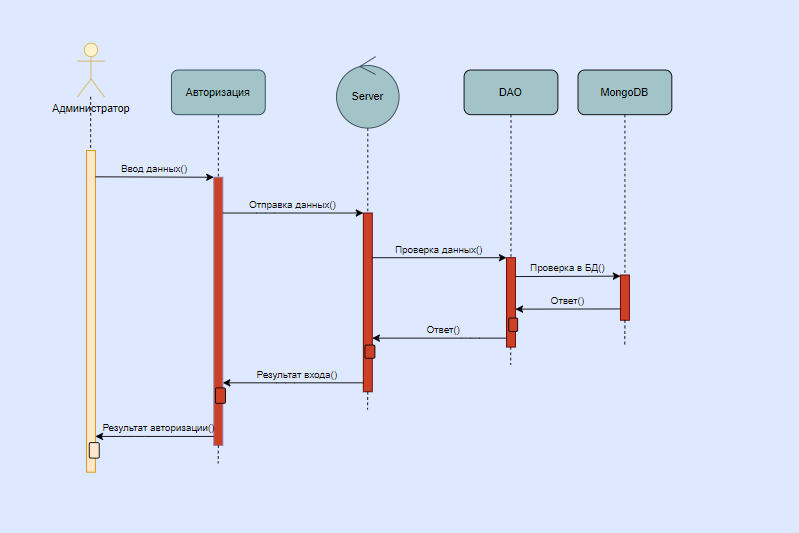


Рисунок 2.4 Диаграмма вариантов последовательности

**Диаграмма состояния**

Объекты 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.

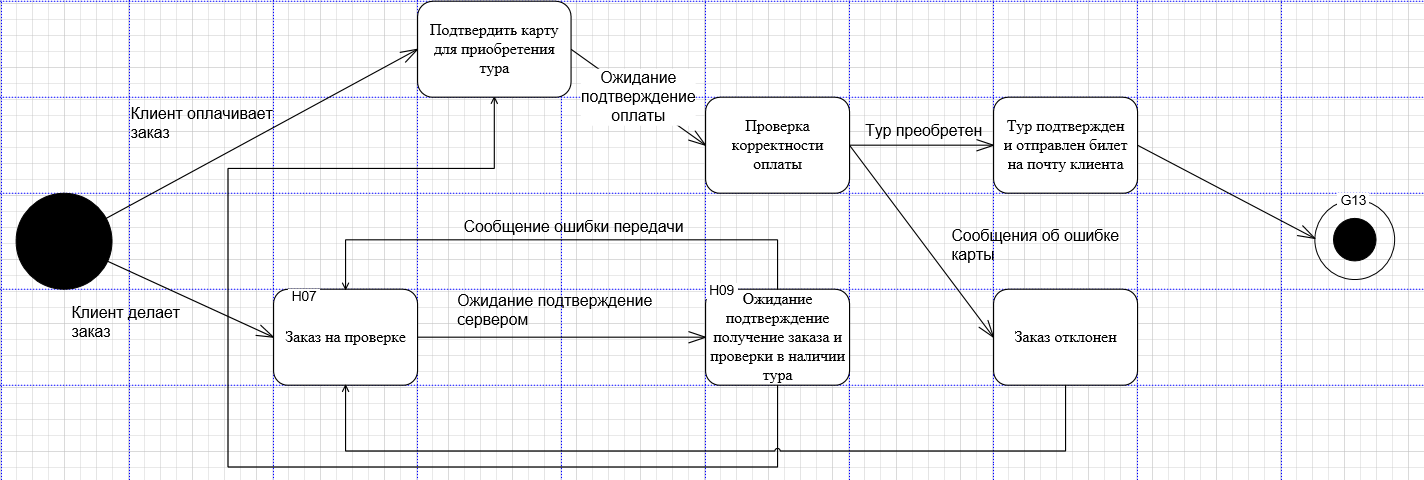
В отличие от других диаграмм диаграмма состояний описывает процесс изменения состояний только одного класса, а точнее – одного экземпляра определенного класса, т. е. моделирует все возможные изменения в состоянии конкретного объекта. Диаграмма состояния данного проекта представлена на рисунках 2.5

Рисунок 2.5 Диаграмма состояние

**Применение паттерна проектирования**

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 2.6



Рисунок 2.6 – 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.

**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 2.7

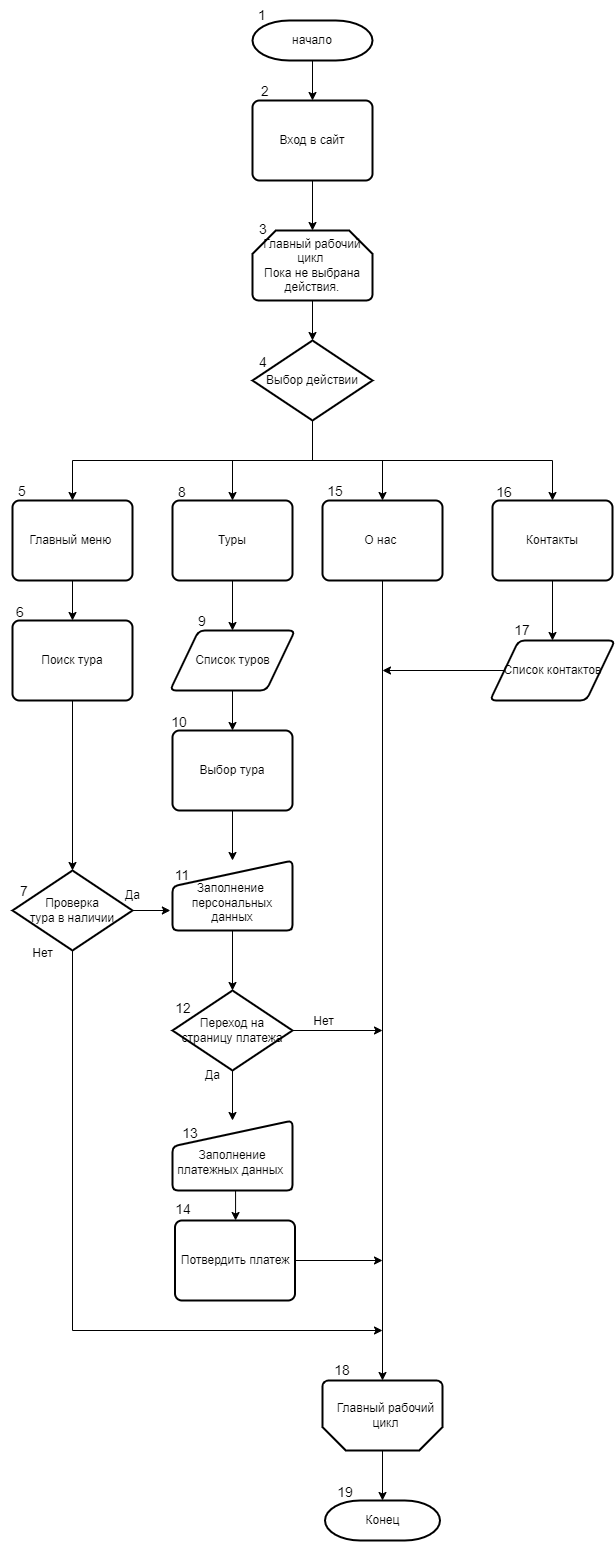


Рисунок алгоритм работы не авторизованного клиента 2.7

## **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, 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 2.8

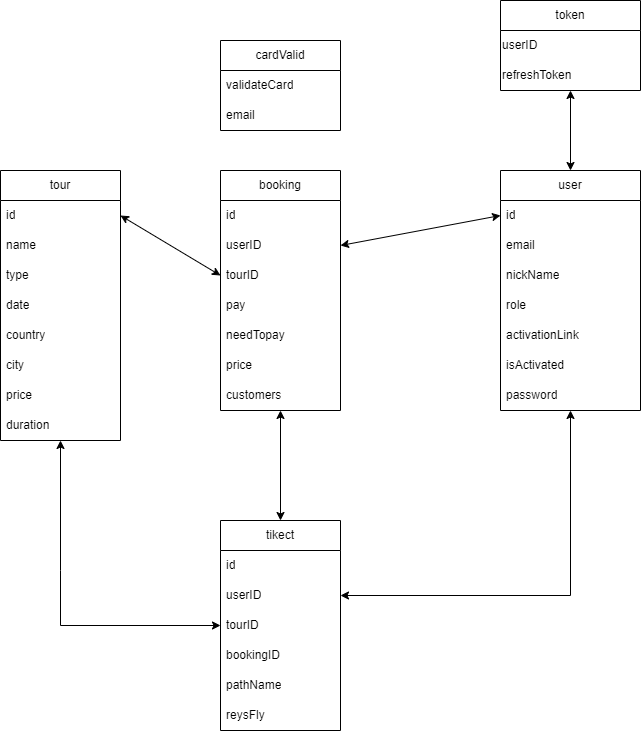


Рисунок Структура базы данных информационной системы торического агентства 2.8

## **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.