

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Высшая школа бизнеса

Управление требованиями и проектирование ИС.
Проект «НаКурорт».

Москва, 2024 год

Содержание

1. Сбор требований. Вопросы заказчику.....	7
1.1. Описание высокоуровневых требований для проекта "НаКурорт".....	7
1.1.1. Основные функции системы.....	8
1.1.2. Ключевые требования.....	8
1.2. Вопросы заказчику для сбора высокоуровневых требований в целях реализации проекта.....	8
1.2.1. Бизнес-цели и ключевые показатели успеха.....	8
1.2.2. Целевая аудитория.....	8
1.2.3. Поставщики услуг.....	9
1.2.4. Модели монетизации.....	9
1.2.5. Подбор турпакетов и интерфейс клиента.....	9
1.2.6. Платежи и безопасность.....	10
1.2.7. Юридические и налоговые требования.....	10
1.2.8. Технические и нефункциональные требования.....	10
2. Список требований для ПО в форме технического образования.....	11
Техническое задание.....	11
1. Общие сведения.....	11
1.1. Наименование системы.....	11
1.1.a. Полное наименование системы.....	11
1.1.b. Краткое наименование системы.....	11
1.2. Основания для проведения работ.....	11
1.3. Наименование организаций – Заказчика и Разработчика.....	11
1.3.a. Заказчик.....	11
1.3.b. Разработчик.....	12
1.4. Плановые сроки начала и окончания работы.....	12
1.5. Источники и порядок финансирования.....	13
2. Назначение и цели создания системы.....	13
2.1. Назначение системы.....	13
2.2. Цели создания системы.....	15
3. Характеристика объектов автоматизации.....	16
4. Требования к системе.....	16
4.1. Требования к системе в целом.....	16
4.1.1. Требования к структуре и функционированию системы.....	16
4.1.2. Требования к численности и квалификации персонала системы и режиму его работы.....	19
4.1.2.a. Требования к численности персонала.....	19
4.1.2.b. Требования к квалификации персонала.....	20
4.1.2.c. Требования к режимам работы персонала.....	20

4.1.3. Показатели назначения.....	21
4.1.3.a. Параметры, характеризующие степень соответствия системы назначению.....	21
4.1.3.b. Требования к приспособляемости системы к изменениям.....	21
4.1.3.c. Требования к сохранению работоспособности системы в различных вероятных условиях.....	22
4.1.4. Требования к надежности.....	23
4.1.4.a. Состав показателей надежности для системы в целом.....	23
4.1.4.b. Перечень аварийных ситуаций, по которым регламентируются требования к надежности.....	24
4.1.4.c. Требования к надежности технических средств и программного обеспечения.....	25
4.1.4.d. Требования к методам оценки и контроля показателей надежности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами.....	28
4.1.5. Требования к эргономике и технической эстетике.....	29
4.1.6. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы.....	30
4.1.7. Требования к защите информации от несанкционированного доступа.....	31
4.1.7.a. Требования к информационной безопасности.....	31
4.1.7.b. Требования к антивирусной защите.....	34
4.1.7.c. Разграничение ответственности ролей при доступе к системе..	35
4.1.8. Требования по сохранности информации при авариях.....	35
4.1.9. Требования к защите от влияния внешних воздействий.....	36
4.1.10. Требования по стандартизации и унификации.....	37
4.1.11. Дополнительные требования.....	38
4.1.12. Требования безопасности.....	38
4.2. Требования к функциям, выполняемым системой.....	40
4.2.1. Подсистема сбора, обработки и загрузки данных.....	40
4.2.1.a. Временной регламент реализации каждой функции, задачи..	40
4.2.1.b. Требования к качеству реализации функций, задач.....	40
4.2.1.c. Перечень критериев отказа для каждой функции.....	41
4.3. Требования к видам обеспечения.....	42
4.3.1. Требования к математическому обеспечению.....	42
4.3.2. Требования к информационному обеспечению.....	45
4.3.2.a. Требования к составу, структуре и способам организации данных в системе.....	45
4.3.2.b. Требования к информационному обмену между компонентами системы.....	47
4.3.2.c. Требования к информационной совместимости со смежными	

системами.....	47
4.3.2.d. Требования по использованию классификаторов и унифицированных документов.....	47
4.3.2.e. Требования по применению систем управления базами данных.....	48
4.3.2.f. Требования к защите данных от разрушений при авариях и сбоях в электропитании системы.....	48
4.3.2.g. Требования к контролю, хранению, обновлению и восстановлению данных.....	48
4.3.2.h. Требования к процедуре придания юридической силы документам, продуцируемым техническими средствами системы....	50
4.3.3. Требования к лингвистическому обеспечению.....	50
4.3.4. Требования к программному обеспечению.....	51
4.3.4.a. Функциональные требования.....	51
4.3.4.b. Нефункциональные требования.....	52
4.3.4.c. Перечень покупных программных средств.....	52
4.3.4.d. Требования к качеству ПС.....	52
4.3.5. Требования к техническому обеспечению.....	52
4.3.6. Требования к метрологическому обеспечению.....	53
4.3.7. Требования к организационному обеспечению.....	53
4.3.7.a. Требования к структуре и функциям подразделений.....	53
4.3.7.b. Требования к организации функционирования системы:.....	53
4.3.7.c. Требования к защите от ошибочных действий персонала.....	53
4.3.8. Требования к методическому обеспечению.....	54
4.3.9. Требования к патентной чистоте.....	54
5. Состав и содержание работ по созданию системы.....	54
6. Порядок контроля и приемки системы.....	57
6.1. Виды и объем испытаний системы.....	57
7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.....	58
7.1. Технические мероприятия.....	58
7.2. Организационные мероприятия.....	58
7.3. Изменения в информационном обеспечении.....	59
8. Требования к документированию.....	59
8.1. Обязательные и опционные требования к ПО.....	60
8.1.1. Must Have (Должно быть).....	60
8.1.2. Should Have (Желательно).....	61
8.1.3. Could Have (Могло бы быть).....	62
8.1.4. Won't Have (Не будет включено).....	62
3. Модель жизненного цикла и методология разработки ПО.....	62
3.1. Модель жизненного цикла программного обеспечения.....	62

3.1.1. Основные преимущества инкрементной модели для проекта «НаКурорт»..	63
3.1.2. Основные преимущества agile-подхода.....	63
3.2. Методология разработки программного обеспечения.....	64
3.3. Сравнение с другими моделями жизненного цикла и методологиями разработки ПО.....	65
3.3.1. Модели жизненного цикла ПО.....	65
3.3.1.a. Водопадная модель (Waterfall).....	65
3.3.1.b. Спиральная модель.....	66
3.3.1.c. V-модель (V-Model).....	66
3.3.2. Методологии разработки ПО.....	66
3.3.2.a. Kanban.....	66
3.3.2.b. DevOps.....	67
4. Описание и разработка диаграммы прецедентов.....	68
4.1. Моделирование диаграммы прецедентов.....	68
4.1.1. Инструмент PlantUML.....	68
4.2. Описание диаграммы.....	68
4.2.1. Роли взаимодействия с системой.....	69
4.2.2. Описание прецедентов.....	69
4.2.2.a. Модуль бронирования.....	69
4.2.2.b. Модуль выбора турпакета.....	69
4.2.2.c. Модуль администрирования пользователей.....	70
4.2.2.d. Модуль управления турпакетами.....	70
4.2.2.e. Модуль управления платежами.....	70
4.2.2.f. Модуль личный кабинет пользователя.....	70
4.2.2.g. Модуль службы поддержки.....	70
4.3. Обоснование выбора функций.....	71
4.4. Описание аномальных сценариев.....	72
4.4.1. Модуль бронирования.....	72
4.4.2. Модуль управления платежами.....	73
4.4.3. Модуль личный кабинет пользователя.....	73
4.4.4. Модуль выбора турпакета.....	74
4.5. Формулирование основных Use-case.....	75
5. Первичное проектирование. Выявление первичных классов.....	91
5.1. Первичное проектирование.....	91
5.1.1. Выявление первичных классов.....	91
5.1.2. Разработка иерархии классов с детализацией абстрактных/полиморфных методов.....	92
5.1.2.a. Пользователь (User) (абстрактный класс).....	92
5.1.2.b. Клиент (Client) (наследник класса User).....	92

5.1.2.c. Менеджер турагентства (Manager) (наследник класса User).....	93
5.1.2.d. Администратор (Administrator) (наследник класса User).....	94
5.1.2.e. Турпакет (Tour_Package) (абстрактный класс).....	94
5.1.2.f. Поставщик (Supplier) (абстрактный класс).....	95
5.1.2.g. Отель (Hotel) (наследник класса Supplier).....	95
5.1.2.h. Авиакомпания (Airline) (наследник класса Supplier).....	96
5.1.2.i. Экскурсия (Excursion) (наследник класса Supplier).....	96
5.1.2.j. Бронирование (Booking).....	97
5.1.2.k. Платеж (Payment) (абстрактный класс).....	98
5.1.2.l. Мастер выбора турпакетов (Tour_Selection_Wizard).....	98
5.1.2.m. Управление пользователями (User_Management).....	99
5.1.2.n. Роль (Role).....	99
5.1.2.o. Служба поддержки (Support_Service).....	100
5.1.2.p. Личный кабинет пользователя (User_Account).....	100
5.1.2.q. Интеграция с поставщиками (Supplier_Integration).....	101
5.1.3. Первичная диаграмма классов.....	102
5.1.4. Описание отношений между классами.....	102
6. Детальная диаграмма классов. Выбор архитектуры.....	105
6.1. Детальная диаграмма классов.....	105
6.1.1. Диаграмма классов.....	105
6.1.2. Описание отношений классов.....	105
6.2. Выбор архитектуры.....	108
6.2.1. Обоснование выбора Многоуровневой архитектуры.....	108
6.2.1.a. Четкое разделение обязанностей и модульность.....	108
6.2.1.b. Масштабируемость и гибкость.....	109
6.2.1.c. Гибкость и легкость интеграции.....	110
6.2.1.d. Упрощенная поддержка и обслуживание.....	110
6.2.1.e. Поддержка высоких стандартов безопасности.....	110
6.2.1.f. Эффективное управление изменениями.....	111
6.2.1.g. Заключение.....	111
6.2.2. Сравнение с другими шаблонами архитектур.....	111
6.2.2.a. SOA (Service-Oriented Architecture).....	111
6.2.2.b. Микросервисная архитектура.....	112
6.2.2.c. Клиент-серверная архитектура.....	113
6.2.2.d. Монолитная архитектура.....	114
6.2.2.e. Событийно-ориентированная архитектура.....	115
7. Динамика системы.....	116
7.1. Диаграмма последовательности (Sequence diagram).....	116
7.2. Описание диаграммы последовательности.....	116
7.2.1. Акторы (внешние участники).....	116

7.2.2. Классы системы.....	117
7.2.3. Типы связей (взаимодействий).....	118
7.2.3.a. Синхронные вызовы.....	118
7.2.3.b. Асинхронные вызовы.....	118
7.2.3.c. Асинхронные вызовы.....	118
7.2.3.d. Завершение объекта.....	119
7.2.3.e. Завершение объекта.....	119
8. Диаграмма состояния.....	120
8.1 Класс Клиент (Client).....	120
a. Гостевой_Режим — клиент находится на сайте без авторизации и может просматривать турпакеты.....	121
• Переход в Зарегистрирован: клиент завершает регистрацию для получения полного доступа к системе.....	121
b. Зарегистрирован — клиент успешно завершил регистрацию.....	121
• Переход в Просмотр_Туров: клиент авторизуется и может начать поиск туров.....	121
c. Просмотр_Туров — клиент ищет доступные туры по предпочтениям.....	121
• Переход в Оформление_бронирования: клиент выбрал тур и приступает к его бронированию.....	121
d. Оформление_бронирования — клиент оформляет бронирование выбранного тура.....	121
• Переход в Завершение_бронирования: бронирование подтверждено и успешно завершено.....	121
e. Завершение_бронирования — клиент завершил бронирование тура.....	121
• Переход в Авторизован: клиент возвращается к главному меню и может продолжать использование системы.....	121
f. Авторизован — клиент авторизован и имеет доступ ко всем функциям системы, включая просмотр туров.....	121
• Переход в Просмотр_Туров: клиент продолжает поиск туров.....	121
• Переход в Неактивен: клиент неактивен в течение длительного времени.....	121
g. Неактивен — клиент был неактивен в течение определенного времени.....	121
• Переход в Авторизован: клиент повторно авторизуется и возвращается к функционалу системы.....	121
8.2. Класс Турпакет (Tour_Package).....	122
8.3. Класс Booking.....	124
8.4. Класс платежная система.....	125
9. Матрица трассировки.....	127
9.1. Описание тест-кейсов.....	132
9.2. Выводы.....	139
10. План реализации, тестирования и сопровождения. Бизнес-план.....	141
1. План реализации системы.....	141
I. Введение.....	141
II. Разработка системы.....	142
III. Тестирование и отладка (план тестирования).....	143

IV. Запуск системы.....	143
V. Переход к эксплуатации системы.....	143
2. План тестирования.....	143
1. Введение.....	143
1.1 Цель.....	143
1.2 Область тестирования.....	143
2. Типы тестирования.....	143
2.1 Функциональное тестирование.....	143
2.2 Интеграционное тестирование.....	144
2.3 Производительность.....	144
2.4 Безопасность.....	144
3. Сценарии тестирования.....	144
3.1 Сценарий: Регистрация и авторизация пользователя.....	144
3.2 Сценарий: Выбор и бронирование турпакета.....	144
3.3 Сценарий: Интеграция с поставщиками услуг.....	144
3.4 Сценарий: Оплата турпакета.....	145
3.5 Сценарий: Отчеты по продажам и бронированиям.....	145
4. Критерии приемки.....	145
5. Риски.....	145
6. План выполнения тестирования.....	145
7. Отчетность.....	146
8. Заключение.....	146
3. План эксплуатации.....	146
1. Введение.....	146
2. Административное управление.....	147
3. Управление турпакетами и предложениями.....	147
4. Управление пользователями.....	147
5. Оплата и финансовый учет.....	147
6. Безопасность и доступ.....	148
7. Обучение и поддержка пользователей.....	148
8. План обновления.....	148
9. Резервное копирование и восстановление.....	148
10. Аудит и оценка.....	149
11. Заключение.....	149
4. Бизнес-план.....	149
1. Объем инвестиций.....	149
2. Суть бизнеса.....	150
3. Конкурентные преимущества проекта.....	150
4. Период окупаемости и возврат инвестиций.....	150
5. Рентабельность.....	150

11.Ревизия выявленных ошибок в ходе выполнения проекта.....	151
12. Итоговые результаты работы.....	151
13. Дальнейшие планы развития.....	152
1. Расширение функциональности:.....	152
2. Улучшение взаимодействия с пользователями:.....	152
3. Развитие аналитики и отчетности:.....	153
4. Оптимизация интерфейса:.....	153
5. Международное расширение:.....	153

1. Сбор требований. Вопросы заказчику

1.1. Описание высокоуровневых требований для проекта "НаКурорт"

Проект "НаКурорт" направлен на создание системы, которая предоставляет клиентам возможность подбирать турпакеты, бронировать гостиницы и заказывать экскурсии. Основные функциональные возможности системы включают автоматическую генерацию турпакетов на основе предпочтений клиента и взаимодействие с поставщиками услуг (перелеты, трансферы, размещение и экскурсионные программы). Система также предоставляет клиентам выбор готовых типажей туров для упрощения процесса.

1.1.1. Основные функции системы

1. Подбор турпакетов, включающих перелет, трансфер, проживание и экскурсии.
2. Взаимодействие с компаниями-поставщиками для получения предложений и классов обслуживания.
3. Генерация турпакетов на основе запросов клиента с возможностью выбора готовых шаблонов.
4. Бесплатное использование системы для клиентов, с оплатой услуг через банковские транзакции.

1.1.2. Ключевые требования

1. Система должна поддерживать автоматическую сборку турпакетов с учетом пожеланий клиента.
2. Необходимо реализовать мастер для выбора турпакетов, который позволяет клиентам последовательно выбирать услуги.
3. Важно обеспечить интеграцию с системами поставщиков для получения актуальной информации об услугах и ценах.

1.2. Вопросы заказчику для сбора высокоуровневых требований в целях реализации проекта

1.2.1. Бизнес-цели и ключевые показатели успеха

1. Какие ключевые бизнес-цели преследует проект "НаКурорт"? Как система должна способствовать их достижению?
2. Какие конкурентные преимущества система должна иметь по сравнению с аналогичными решениями на рынке (удобство, скорость, стоимость услуг и т.д.)?
3. Какие ключевые показатели эффективности будут отслеживаться для оценки успеха проекта? (например, количество проданных турпакетов, процент возвратов, средний чек, количество активных пользователей).

1.2.2. Целевая аудитория

4. Какие категории клиентов будут использовать систему? (индивидуальные туристы, группы, корпоративные клиенты)
5. Есть ли сегментация клиентов по регионам или другим критериям? Как это влияет на формирование предложений?

1.2.3. Поставщики услуг

6. Какие требования предъявляются к поставщикам услуг? Есть ли ограничения по типам поставщиков (например, только сертифицированные агентства)?

7. Какую информацию о предлагаемых услугах должны предоставлять поставщики (минимальный и рекомендуемый набор данных: название, описание, цена, классы обслуживания, доступность и т.д.)?
8. Как будет организован процесс загрузки и обновления данных поставщиками?
9. Как система должна обрабатывать изменения в ценах и доступности услуг поставщиков?

1.2.4. Модели монетизации

10. Какой процент комиссии будет взиматься с поставщиков за использование системы?
11. Какие дополнительные источники дохода предусмотрены для системы (например, реклама, платные дополнительные функции для поставщиков)?

1.2.5. Подбор турпакетов и интерфейс клиента

12. Как будет организован процесс подбора турпакетов для клиентов? Какие параметры они смогут выбирать (бюджет, даты, тип тура, предпочитаемый класс обслуживания и т.д.)?
13. Какие готовые типажи (комплекты пожеланий) будут предложены клиентам для упрощения выбора?
14. Какие возможности по управлению заказами будут у клиентов (изменение бронирования, отмена, изменение услуг)?
15. Какой интерфейс и каналы взаимодействия с системой предпочтительны для клиентов (веб-сайт, мобильное приложение)? Требуется ли поддержка мультиязычности?

1.2.6. Платежи и безопасность

16. Какие системы оплаты должны поддерживаться (банковские карты, электронные платежи, криптовалюта)? Есть ли специфические предпочтения для разных регионов?

17. Какой уровень безопасности требуется для защиты данных клиентов и финансовых транзакций?

1.2.7. Юридические и налоговые требования

18. Есть ли юридические или налоговые требования, которые могут повлиять на работу системы (например, правила предоставления туристических услуг, налоговые ставки в разных странах)?
19. Какие правовые требования необходимо учитывать в отношении защиты персональных данных?

1.2.8. Технические и нефункциональные требования

20. Какие показатели производительности критичны для системы (например, время ответа на запросы, скорость обработки заказов)?
21. Какой объем данных система должна обрабатывать и хранить (например, количество активных предложений, клиентов)?
22. Требуется ли поддержка масштабируемости системы для роста числа клиентов и поставщиков услуг?

2. Список требований для ПО в форме технического образования

Техническое задание

1. Общие сведения

1.1. Наименование системы

1.1.а. Полное наименование системы

Полное наименование системы: "НаКурорт" — система подбора и бронирования турпакетов.

1.1.b. Краткое наименование системы

"НаКурорт".

1.2. Основания для проведения работ

Система выполняется на основании договора №1 от 10.09.2024 между туристическая компания "ГлобалТур" и ООО "Системы Путешествий".

1.3. Наименование организаций – Заказчика и Разработчика

1.3.a. Заказчик

Название организации	Туристическая компания "ГлобалТур"
Тип организации	Туристическое агентство
Адрес	789 ул. Путешественников, Горный Город, Республика Альтая
Контактное лицо	Петров Александр Сергеевич
Электронная почта	petrov.alexander@globaltour.al
Телефон	+7 (444) 444-4444

1.3.b. Разработчик

Название организации	ООО "Системы Путешествий"
Тип организации	IT-компания
Адрес	123 ул. Технологов, Техноград, Страна Цифр
Контактное лицо	Иванова Марина Викторовна

Должность контактного лица	Руководитель проектов
Электронная почта	ivanova.marina@travelsystems.com
Телефон	+7 (999) 999-9999

1.4. Плановые сроки начала и окончания работы

Вся работа над системой "НаКурорт" разделена на несколько стадий. После завершения каждого этапа исполнитель обязан предоставить отчет о результатах работы:

1. Подготовительный этап (инициация проекта): 01.10.2024 – 31.10.2024
2. Проектирование (архитектура и MVP): 01.11.2024 – 15.12.2024
3. Разработка MVP: 15.12.2024 – 15.02.2025
4. Полная разработка системы: 15.02.2025 – 15.05.2025
5. Тестирование системы: 15.05.2025 – 30.06.2025
6. Внедрение и ввод в эксплуатацию: 01.07.2025 – 15.08.2025
7. Поддержка и сопровождение.

Плановые сроки завершения разработки и внедрения проекта: 30.09.2025.

1.5. Источники и порядок финансирования

Проект финансируется за счет внутренних ресурсов компании "ГлобалТур". Средства компании будут использованы для начальной стадии разработки системы "НаКурорт". Это включает оплату услуг разработчиков, аренду серверов и программного обеспечения, а также другие операционные расходы, связанные с реализацией проекта.

Финансирование будет разделено на несколько этапов:

1. *Этап анализа и проектирования:* Оплата за услуги по сбору и анализу требований.

2. *Этап разработки:* Финансирование программирования и тестирования системы.
3. *Этап внедрения и обучения:* Расходы на обучение сотрудников компании "ГлобалТур" и внедрение системы в эксплуатацию.
4. *Этап сопровождения:* Средства будут выделены на поддержку и обновление системы после её запуска.

1.6. Порядок оформления и предъявления заказчику результатов работ

Работы по созданию системы "НаКурорт" сдаются Разработчиком поэтапно в соответствии с календарным планом Проекта. По завершении каждого этапа Разработчик предоставляет Заказчику соответствующие отчетные документы, состав которых определяется Договором. Отчёты включают результаты анализа требований, проектирования, разработки, тестирования и внедрения системы. Все этапы работы фиксируются и принимаются на основе согласованных критериев оценки.

2. Назначение и цели создания системы

2.1. Назначение системы

Система "НаКурорт" предназначена для автоматизации процесса подбора, бронирования и управления туристическими услугами. Она объединяет в себе комплекс инструментов для поиска, конфигурирования и бронирования турпакетов, которые включают перелеты, трансферы, проживание и экскурсионные программы. Основная цель системы — упростить процесс выбора и бронирования туров для пользователей, сделать его интуитивно понятным, быстрым и доступным с любой платформы.

Ключевые задачи системы:

1. *Автоматический подбор турпакетов:* На основе предпочтений клиентов (бюджет, даты, местоположение, тип тура, желаемый класс обслуживания), система автоматически формирует турпакеты, которые включают перелеты, трансферы, проживание и экскурсии.

Это позволяет пользователям избежать ручного поиска и сопоставления отдельных элементов поездки.

2. *Гибкая настройка и персонализация*: Клиенты могут выбирать как готовые пакеты, так и индивидуализировать их в зависимости от своих предпочтений, корректируя маршруты, классы обслуживания или добавляя дополнительные услуги (например, аренду автомобилей, страхование).
3. *Взаимодействие с поставщиками услуг*: Система интегрируется с базами данных поставщиков туристических услуг (авиакомпаний, гостиницы, транспортные компании, экскурсионные агентства), позволяя получать актуальные предложения по ценам и наличию услуг в реальном времени. Это гарантирует пользователям доступ к самым свежим и выгодным предложениям.
4. *Процесс бронирования и оплаты*: Система должна предоставлять клиентам возможность безопасного бронирования выбранных туров и оплаты через различные методы (банковские карты, электронные платежи). Этот процесс максимально оптимизирован для упрощения, что позволяет пользователю завершить бронирование в несколько шагов.
5. *Обеспечение прозрачности и контроля*: Пользователи могут отслеживать статус своих бронирований, получать уведомления об изменениях (например, изменениях времени вылета или наличии номеров в гостиницах), а также иметь доступ к истории своих заказов. Это повышает доверие клиентов и помогает им планировать свои поездки с большей уверенностью.

2.2. Цели создания системы

Целью проекта является разработка программного обеспечения для платформы автоматизации подбора и бронирования туристических пакетов. Каждый турпакет должен включать предложения по перелету, трансферу, проживанию и экскурсионным программам, формируемым на основе предпочтений клиента. Пользователи системы должны иметь возможность самостоятельно выбирать туры, настраивать их по своим требованиям, оплачивать и бронировать услуги.

При разработке системы должны быть учтены следующие факторы: количество пользователей платформы, интеграция с различными поставщиками туристических услуг, высокие нагрузки на систему в периоды сезонного спроса, а также обеспечение безопасности персональных данных клиентов и защиты информации о транзакциях.

Конечными потребителями системы являются:

1. *Индивидуальные туристы и группы:* Пользователи будут использовать систему для подбора турпакетов, бронирования услуг и оплаты поездок. Они смогут настраивать свои путешествия в соответствии с личными предпочтениями и требованиями.
2. *Поставщики услуг:* Авиакомпании, гостиницы, транспортные компании и агентства по организации экскурсий также будут взаимодействовать с системой, предоставляя актуальные данные о своих услугах, доступности и ценах.
3. *Администраторы компании "ГлобалТур":* Администрация компании сможет управлять интеграцией с поставщиками, отслеживать и анализировать продажи туров, а также проводить маркетинговые кампании и формировать отчеты по различным показателям, таким как популярность направлений и уровень клиентской удовлетворенности.

3. Характеристика объектов автоматизации

Заказчик, компания "ГлобалТур", занимается подбором и бронированием туристических пакетов для своих клиентов. Основные процессы включают взаимодействие с поставщиками услуг (авиакомпаниями, гостиницами, транспортными компаниями и экскурсионными агентствами), подбор туров на основе предпочтений клиентов, а также обработку платежей и управление бронированиями.

Выделены следующие процессы в деятельности компании "ГлобалТур", для которых возможна автоматизация:

Структурное подразделение	Наименование процесса	Возможность автоматизации	Решение об автоматизации в ходе проекта
Отдел продаж	Подбор турпакетов	Возможна	Будет автоматизирован
Отдел продаж	Бронирование услуг	Возможна	Будет автоматизирован
Финансовый отдел	Прием и обработка платежей	Возможна	Будет автоматизирован
Отдел поддержки клиентов	Управление изменениями бронирований	Возможна	Будет автоматизирован
Отдел маркетинга	Формирование отчетов по продажам	Возможна	Будет автоматизирован

4. Требования к системе

4.1. Требования к системе в целом

4.1.1. Требования к структуре и функционированию системы

Архитектура системы "НаКурорт":

1. Централизованное хранилище данных:

Все ключевые данные о клиентах, турпакетах, поставщиках и транзакциях хранятся в защищенной базе данных, которая обеспечивает быстрый доступ к информации и защиту от несанкционированного доступа. Доступ к данным предоставляется в реальном времени.

2. Многоуровневая архитектура:

2.1. Уровень 1 — Подбор туров: Система анализирует запросы пользователей и моментально генерирует доступные турпакеты на основе предпочтений и актуальных предложений поставщиков.

2.2. Уровень 2 — Хранилище данных: Здесь система собирает и организует информацию, которую можно использовать для принятия решений — бронирования, отчетности и аналитики.

- 2.3. Уровень 3 — Аналитика и отчетность: Административный интерфейс предоставляет аналитические инструменты для создания отчетов по продажам, предпочтениям клиентов и другим бизнес-показателям.

Функциональные подсистемы:

1. *Подсистема обработки данных:* Система собирает данные от поставщиков (авиабилеты, отели, трансферы) и гарантирует, что все предложения актуальны и легко доступны для клиентов.
2. *Подсистема хранения:* Надежное хранение всех данных о клиентах, бронированиях и оплатах. Эта подсистема интегрирована с механизмами безопасности и защиты данных.
3. *Подсистема аналитики:* Визуализация данных о популярности туров, предпочтениях клиентов и доходах компании. Пользователи могут получать готовые отчеты или настраивать собственные аналитические панели.

Механизмы обмена информацией:

1. *Транспортные протоколы:* Для обмена данными система использует безопасные протоколы передачи, такие как TCP/IP, что обеспечивает надежную и быструю передачу информации.
2. *Прикладные протоколы:* HTTP/HTTPS для обмена данными с внешними системами и REST API для интеграции с поставщиками услуг.
3. *Доступ к отчетности:* Вся аналитика и отчеты доступны через веб-интерфейс с поддержкой HTTPS для защиты данных.

Интеграция с внешними системами:

1. *Партнерские системы:*
 - a. Системы бронирования отелей, авиабилетов и экскурсий.
 - b. Платежные системы для обработки транзакций.
2. *Источники данных:*
 - a. СУБД (например, PostgreSQL) для хранения данных о клиентах и турпакетах.

- б. Внешние системы поставщиков для получения актуальных данных о доступности услуг.
- 3. *Взаимодействие:*
 - а. API-интерфейсы для обмена данными с партнерами.
 - б. Автоматическое обновление информации о турах и ценах через синхронизацию с поставщиками.

Режимы работы системы:

1. *Основной режим:*

Все подсистемы работают круглосуточно, обеспечивая постоянный доступ пользователей к бронированию и обработке заказов. Для клиентов система должна быть всегда доступна, независимо от времени суток или местоположения.
2. *Технический режим:*

В периоды минимальной активности система может кратковременно переходить в режим обслуживания для обновлений, технического осмотра и улучшений. В этот период клиенты заранее получают уведомления о возможных перерывах в работе.

Поддержка и диагностика:

1. *Мониторинг системы:*

Постоянный контроль работоспособности всех компонентов системы. В случае возникновения проблем система автоматически уведомляет администраторов, фиксирует инциденты и предлагает возможные пути решения.
2. *Плановое обслуживание:*

Регулярная проверка системы на предмет ошибок, обновление баз данных и устранение сбоев. Эта активность помогает поддерживать стабильную работу и защищает данные от потерь.

4.1.2. Требования к численности и квалификации персонала системы и режиму его работы

4.1.2.a. Требования к численности персонала

Для обеспечения работы системы "НаКурорт" в подразделениях Заказчика, необходимо выделить следующие ключевые роли:

1. Менеджеры по туризму — n человек (где n зависит от объема клиентской базы и количества предложений от поставщиков). Число менеджеров определяется из расчета: $\text{Количество менеджеров по туризму} = (\text{Количество клиентов} / 500) + (\text{Количество предложений от поставщиков} / 1000)$.
2. Операторы службы поддержки — n человек. Число операторов службы поддержки определяется из расчета: $\text{Количество операторов службы поддержки} = (\text{Количество клиентов} / 1000) + (\text{Среднее количество запросов в день} / 500)$.
3. Технические специалисты по поддержке системы — n человек. Число технических специалистов определяется из расчета: $\text{Количество технических специалистов} = (\text{Количество пользователей системы} / 200) + (\text{Количество технических интеграций} / 100)$. Техническая интеграция представляет собой процесс соединения системы "НаКурорт" с другими программными решениями, сервисами или платформами для обеспечения обмена данными, функциональной совместимости и расширения возможностей системы. Каждая такая интеграция требует настройки, разработки, тестирования и поддержки, что влияет на рабочую нагрузку технических специалистов.

4.1.2.b. Требования к квалификации персонала

1. Менеджер по туризму — должен обладать навыками работы с онлайн-платформами бронирования, иметь опыт взаимодействия с клиентами и понимание основ туристического бизнеса. Также необходимо знание специфики работы с туроператорами и поставщиками услуг.
2. Оператор службы поддержки — навыки общения с клиентами, умение быстро решать их запросы, знание работы с системами

тикетов для перенаправления запросов на техническую поддержку. Требуется высокая цифровая грамотность и умение работать с различными коммуникационными платформами.

3. Технический специалист по поддержке системы — глубокие знания в области работы с системами управления базами данных (СУБД), опыт работы с API для интеграции с внешними поставщиками, понимание сетевых протоколов и безопасность данных. Необходимо владение SQL и навыки программирования для анализа и устранения технических проблем системы.

4.1.2.с. Требования к режимам работы персонала

Персонал, ответственный за сопровождение и обслуживание системы "НаКурорт", должен работать в следующих режимах:

1. Менеджер по туризму — работает в соответствии с основным графиком работы компании "ГлобалТур", установленным Заказчиком. График может корректироваться в зависимости от сезонного спроса.
2. Оператор службы поддержки — работает в двухсменном режиме, чтобы обеспечить круглосуточную поддержку клиентов. Это гарантирует, что пользователи всегда могут получить помощь, независимо от времени суток.
3. Технический специалист по поддержке системы — работает посменно в соответствии с установленным графиком, чтобы обеспечивать бесперебойную работу системы и оперативно реагировать на технические сбои или внештатные ситуации.

4.1.3. Показатели назначения

4.1.3.а. Параметры, характеризующие степень соответствия системы назначению

Система "НаКурорт" должна обеспечивать следующие количественные показатели, которые характеризуют ее соответствие назначению:

1. Количество обрабатываемых бронирований — до 500 бронирований в день.
2. Количество подключенных поставщиков услуг — не менее 50.
3. Количество генерируемых отчетов — до 20 аналитических отчетов в день по продажам, предпочтениям клиентов и активности поставщиков.
4. Среднее время отклика системы — не более 2 секунд для каждого запроса пользователя.

4.1.3.b. Требования к приспособляемости системы к изменениям

1. Гибкое администрирование — система должна легко адаптироваться к изменениям в структуре данных и добавлению новых функциональностей.
2. Модернизация — система должна поддерживать возможность обновления процессов сбора и обработки данных от поставщиков услуг.
3. Модификация доступа — возможность быстрой настройки и модификации прав доступа для различных категорий пользователей.
4. Конфигурируемость — поддержка настроечных и конфигурационных файлов для оперативного изменения ключевых параметров работы системы.
5. Соответствие стандартам — система должна поддерживать современные стандарты безопасности, обмена данными и протоколов связи.
6. Документация и обучение — должна быть обеспечена актуальная документация и инструкции для администраторов и пользователей системы.
7. Тестирование изменений — любая новая функциональность или обновления должны проходить тщательное тестирование перед развертыванием.
8. Обратная совместимость — при обновлении системы необходимо сохранять поддержку предыдущих версий для минимизации рисков отказов и сбоев.

4.1.3.с. Требования к сохранению работоспособности системы в различных вероятных условиях

Вероятное условие	Требование
Сбой серверов из-за перегрузки или сетевых проблем	Автоматическое обнаружение сбоев и мгновенное переключение на резервные сервера, если они доступны.
Выход из строя сервера хранения данных	Автоматическое обнаружение сбоя, переключение на резервные сервера и уведомление администратора системы для решения проблемы.
Потеря связи с поставщиками услуг	Хранение локальных буферных копий данных и автоматическая синхронизация с внешними источниками после восстановления соединения.
Обработка больших объемов данных от поставщиков	Возможность масштабирования системы для обработки больших объемов данных без снижения производительности.
Нарушение целостности данных или утечка информации	Регулярное резервное копирование данных, использование шифрования и аутентификации, мониторинг системы безопасности.
Плановые технические работы	Уведомление пользователей заранее о проведении работ, выбор времени с минимальной активностью для проведения обслуживания.
Увеличение числа пользователей или объема данных	Поддержка горизонтального и вертикального масштабирования системы для обеспечения ее производительности и доступности.
Обновление операционных систем и компонентов	Тестирование обновлений в тестовой среде перед развертыванием, регулярное обновление компонентов с минимальными простоями.

Сбой серверов хранения данных	Постоянный мониторинг состояния серверов и автоматическое переключение на резервные серверы в случае сбоя.
Завершение поддержки внешних компонентов	Замена устаревших компонентов или интеграция с новыми совместимыми решениями.
Пиковая активность пользователей	Масштабирование вычислительных ресурсов для обеспечения бесперебойной работы системы в периоды повышенной активности.
Изменения в бизнес-процессах и требованиях пользователей	Система должна быть гибкой для изменения бизнес-логики и интерфейсов в соответствии с новыми требованиями.
Нецелевые атаки и вирусные угрозы	Использование современных систем защиты информации, регулярное обновление антивирусных баз и постоянный мониторинг безопасности.

4.1.4. Требования к надежности

4.1.4.a. Состав показателей надежности для системы в целом

Надежность системы "НаКурорт" должна быть обеспечена за счет:

1. *Использования высокопроизводительных технологий:* Применение современных серверов и программного обеспечения, которые способны обрабатывать большие объемы данных и справляться с пиковой нагрузкой.
2. *Мгновенной реакции на сбой:* Сотрудники технической поддержки должны устранять ошибки в кратчайшие сроки, не допуская долгих простоев.
3. *Эффективного управления ресурсами:* Грамотное распределение нагрузки на серверы, чтобы избежать перегрузки системы, особенно в периоды повышенной активности пользователей.
4. *Обучения персонала:* Пользователи и операторы должны быть обучены правильной эксплуатации системы, что позволит минимизировать человеческие ошибки.

5. *Планового обслуживания*: Регулярные профилактические работы по поддержанию стабильной работы серверов и программных компонентов, чтобы избежать сбоев.

Показатели надежности:

1. Среднее время восстановления работоспособности (MTTR) — не более 4 часов за период одного месяца.
2. Доступность системы — не менее 99,9% времени: Система должна быть доступна практически круглосуточно, за исключением краткосрочных периодов на техническое обслуживание.
3. Частота сбоев — не более 1 раза в месяц: Количество критических сбоев должно быть сведено к минимуму за счет использования отказоустойчивой инфраструктуры.

4.1.4.b. Перечень аварийных ситуаций, по которым регламентируются требования к надежности

1. Сбой в электроснабжении серверов системы — Отказ внешнего источника питания может привести к недоступности серверов и сервисов системы.
2. Сбой электропитания пользовательских рабочих станций — Пользователи могут столкнуться с отключениями, что нарушит процесс бронирования.
3. Сбой в работе локальной сети — Потеря соединения между компонентами системы из-за проблем с сетевым оборудованием.
4. Ошибки в системе "НаКурорт", не выявленные при тестировании — Неожидаемые баги, возникающие в процессе эксплуатации системы.
5. Сбой серверного программного обеспечения — Отказ программного обеспечения серверов, что может повлиять на доступ к данным и функциональности системы.
6. Потеря или некорректная модификация данных — Ошибки, приводящие к изменению или удалению данных, что может нарушить целостность системы.

7. Отказ оборудования или серверов хранения данных — Сбой физического оборудования, влияющий на доступ к данным пользователей и заказов.
8. Сбои при обновлении или модернизации системы — Проблемы, возникающие в процессе обновления или внедрения новых функций.
9. Кибератаки и вирусные угрозы — Нарушения, вызванные хакерскими атаками или вредоносными программами.
10. Потеря связи с внешними источниками данных — Потеря подключения к системам поставщиков услуг, что может временно остановить бронирование.

4.1.4.с. Требования к надежности технических средств и программного обеспечения

Надежность оборудования:

1. Использование серверов и сетевого оборудования, соответствующих высоким стандартам надежности, способных выдерживать высокие нагрузки без снижения производительности. Оборудование должно соответствовать стандартам ГОСТ Р 57580-2017, ISO 9001 и ISO 27001.
 - 1.1. Uptime: Не менее 99.99% в год.
 - 1.2. MTBF: Не менее 100,000 часов.
 - 1.3. MTTR: Не более 2 часов.
 - 1.4. Нагрузочная способность: Серверы — не менее 200 одновременных пользователей; сетевое оборудование — пропускная способность не менее 10 Гбит/с.
2. Возможность восстановления аппаратно-программного комплекса после сбоев должна быть обеспечена автоматическим переключением на резервные системы. Конфигурация N+1 для всех критически важных компонентов.

- 2.1. Время переключения — не более 60 секунд.
- 2.2. Ежеквартальное тестирование резервных систем.
- 2.3. Внедрение систем мониторинга и автоматических оповещений.

Надежность электроснабжения:

1. Все серверы должны быть оснащены источниками бесперебойного питания (ИБП) с автономной работой не менее 15 минут, что позволяет завершить критические операции и предотвратить потерю данных. ИБП должны быть оборудованы системой автоматического оповещения администраторов при переходе на резервное питание.
2. Наличие системы автоматического оповещения администраторов при переходе на резервное питание. Встроенные агенты должны автоматически завершать работу операционных систем при перебое электропитания, превышающем 10 минут.
3. Бесперебойное питание должно быть обеспечено для активного сетевого оборудования, чтобы не допустить перебоев в соединениях.

Организационные мероприятия для надежности аппаратных и программных средств:

1. Проведение предварительного обучения всех пользователей и обслуживающего персонала для снижения риска ошибок из-за человеческого фактора.
2. Регулярное администрирование и техническое обслуживание всех программно-аппаратных средств для предотвращения аварий.
3. Строгое соблюдение регламентов по эксплуатации и обслуживанию оборудования.
4. Регулярное резервное копирование данных для их защиты и возможности быстрого восстановления в случае сбоев.

Надежность программного обеспечения подсистем:

1. Использование проверенного системного ПО и надёжных библиотек для разработки ПО системы "НаКурорт".
 - 1.1. Системное программное обеспечение: Все компоненты системы "НаКурорт" должны использовать операционные

системы и серверные решения, сертифицированные по стандарту ГОСТ Р 56939-2016 «Информационная технология. Защита информации. Требования к системе защиты информации».

- 1.2. Библиотеки и фреймворки: При разработке программного обеспечения необходимо использовать библиотеки и фреймворки, прошедшие независимую проверку на безопасность и соответствие стандартам ГОСТ Р 57580-2017 «Информационная технология. Термины и определения».
- 1.3. Сертификация ПО: Все используемые сторонние компоненты должны иметь актуальные сертификаты соответствия ГОСТ Р ИСО/МЭК 15408-2008 «Критическая оценка безопасности информационных технологий. Методика оценки безопасности».

2. Проведение комплексных тестов и мероприятий по отладке системы для выявления и устранения потенциальных ошибок.

- 2.1. *Модульное тестирование*: Проведение автоматизированных модульных тестов для каждого компонента системы, обеспечивающих покрытие не менее 90% кода тестами.
- 2.2. *Интеграционное тестирование*: Проверка взаимодействия между модулями и подсистемами с использованием тестовых сценариев, охватывающих основные бизнес-процессы.
- 2.3. *Системное тестирование*: Полное тестирование системы "НаКурорт" в условиях, максимально приближенных к боевым, включая нагрузочное тестирование с использованием инструментов JMeter или LoadRunner.
- 2.4. *Приемочное тестирование*: Проведение тестов совместно с представителями Заказчика для подтверждения соответствия системы требованиям.

3. Ведение системных журналов и логов для анализа ошибок и корректировок конфигурации системы.

- 3.1. *Системные журналы*: Все ключевые события системы должны фиксироваться в системных журналах (логах) с использованием стандартизированных форматов (например,

JSON, XML). Логи должны включать информацию о времени события, уровне важности (INFO, WARN, ERROR), источнике события и подробном описании.

- 3.2. *Хранение логов*: Логи должны храниться не менее 90 дней на защищенных серверах с регулярными резервными копиями. Доступ к логам должен быть ограничен только уполномоченными лицами с использованием методов аутентификации и авторизации согласно ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита.».
- 3.3. *Анализ и мониторинг*: Внедрение системы централизованного сбора и анализа логов (например, ELK Stack — Elasticsearch, Logstash, Kibana). Настройка автоматических оповещений при обнаружении критических ошибок или аномалий в логах.
- 3.4. *Обеспечение целостности*: Использование механизмов хеширования и цифровой подписи для обеспечения неизменности логов.

4.1.4.d. Требования к методам оценки и контроля показателей надежности на разных стадиях создания системы в соответствии с действующими нормативно-техническими документами

Оценка надежности системы "НаКурорт" должна проводиться на всех этапах её создания, с использованием следующих методов:

- 1. На этапе проектирования — проверка надежности осуществляется посредством моделирования и расчетов, чтобы оценить потенциальные риски сбоев и нагрузки на систему.
- 2. На этапе разработки — производится тестирование системы на реальных задачах с моделированием различных аварийных ситуаций для выявления слабых мест и корректировки архитектуры.
- 3. На этапе внедрения — система предоставляется Заказчику для испытаний в реальных условиях эксплуатации с возможностью оценки её надежности при различных нагрузках и условиях.
- 4. Документирование — каждая стадия создания системы должна быть зафиксирована в технической документации, включая результаты тестов и рекомендации по улучшению.

4.1.5. Требования к эргономике и технической эстетике

Разрабатываемая система "НаКурорт" должна обладать следующими свойствами, касающимися эргономики и технической эстетики:

1. *Фирменный стиль:* На всех страницах системы должен быть логотип компании "ГлобалТур", что поддерживает единый брендовый стиль.
2. *Уведомления об ошибках:* При возникновении ошибки пользователю должно отображаться информативное окно с описанием проблемы и контактами службы поддержки для оперативного решения.
3. *Интерфейс:* Все интерфейсы системы должны быть выполнены в едином минималистичном стиле, с использованием фирменных цветов компании "ГлобалТур" (например, сине-золотая палитра), что создает профессиональный и привлекательный внешний вид.
4. *Шрифты и размеры текста:*
 - 4.1. Заголовки на главных страницах: 22 полужирный.
 - 4.2. Подзаголовки: 18 полужирный.
 - 4.3. Основной текст: 14 обычный.
5. *Тип шрифта:* Использовать шрифт Roboto, который обеспечивает читабельность и современный внешний вид.
6. *Безопасность паролей:* При вводе пароля на страницах авторизации, он должен скрываться для безопасности. По желанию пользователя возможен показ символов пароля.
7. *Типизированные интерфейсы:* Все элементы интерфейса должны быть стандартизированы для обеспечения единообразия и удобства навигации.
8. *Мобильная адаптация:* Интерфейсы системы должны корректно отображаться на мобильных устройствах и планшетах, предоставляя пользователям возможность бронировать туры в любое время и в любом месте.

4.1.6. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

1. *Условия эксплуатации и обслуживание:* Все условия эксплуатации и периодичность обслуживания технических средств системы

"НаКурорт" должны соответствовать документации производителя. Это включает регулярное техническое обслуживание серверов, сетевого оборудования и программного обеспечения, чтобы обеспечить их бесперебойную работу и безопасность данных.

2. *Климатические условия для оборудования:* Технические средства и рабочие места персонала должны находиться в помещениях Заказчика, соответствующих климатическим стандартам, указанным в ГОСТ 15150-69. Температурный режим эксплуатации: от 5 до 40 °С, с относительной влажностью воздуха от 40% до 80% при температуре 25 °С и атмосферным давлением от 630 до 800 мм ртутного столба.
3. *Организация рабочих мест:* Размещение технических средств и автоматизированных рабочих мест должно соответствовать требованиям ГОСТ 21958-76. Все рабочие места операторов и технического персонала должны быть организованы с учетом эргономики, чтобы обеспечить комфортное взаимодействие с системой и минимизировать нагрузку на сотрудников.
4. *Требования к электропитанию:* Для стабильной работы технических средств системы "НаКурорт" необходимо наличие трёхфазной четырёхпроводной сети с заземленной нейтралью на 380/220 В (+10-15)% и частотой 50 Гц (+1-1) Гц. Каждое устройство должно быть подключено через однофазные розетки на 220 В с заземляющим контактом, что обеспечит защиту оборудования от перепадов напряжения и аварий.
5. *Запасные компоненты и ремонт:* Для обеспечения надежности работы системы должен быть предусмотрен комплект запасных изделий и приборов (ЗИП), который позволит оперативно заменить вышедшие из строя компоненты. Состав и условия хранения ЗИП будут определены на этапе технического проектирования. Ремонт и замена компонентов должны выполняться квалифицированным персоналом с учетом рекомендаций производителя.

4.1.7. Требования к защите информации от несанкционированного доступа

4.1.7.а. Требования к информационной безопасности

1. *Комплексная защита:* Система должна использовать комбинацию программных и технических средств, а также организационных мер для обеспечения полной защиты данных и ресурсов.
 - 1.1. *Программные средства:* Антивирусное ПО, системы обнаружения вторжений (IDS/IPS), фаерволы.
 - 1.2. *Технические средства:* Аппаратные средства защиты, такие как фаерволы, системы резервного питания, системы контроля доступа.
 - 1.3. *Организационные меры:* Политики безопасности, процедуры управления доступом, регулярные аудиты безопасности.
2. *Защита на всех этапах:* Все стадии обработки информации, от ввода данных до их хранения и передачи, должны находиться под постоянной защитой. Это также касается периодов технического обслуживания, модернизации и ремонта.
 - 2.1. Шифрование данных при передаче и хранении.
 - 2.2. Контроль доступа во время обслуживания и модернизации.
 - 2.3. Использование защищенных каналов связи (например, VPN) при удалённом доступе.
3. *Минимальное влияние на производительность:* Программные средства безопасности не должны существенно снижать скорость работы системы или ограничивать её гибкость в конфигурировании и эксплуатации.
 - 3.1. Программные средства безопасности не должны снижать скорость работы системы более чем на 5% и не должны ограничивать её гибкость в конфигурировании и эксплуатации.
4. *Разграничение доступа:* Права доступа пользователей и администраторов должны быть четко разграничены на основе принципа "что не разрешено, то запрещено".
 - 4.1. Использование ролевой модели доступа (RBAC).
 - 4.2. Применение принципа наименьших привилегий.
 - 4.3. Регулярный аудит и пересмотр прав доступа.

5. *Предотвращение несанкционированного доступа:* В системе должны быть встроены механизмы для мониторинга, обнаружения и предотвращения попыток несанкционированного доступа к данным и ресурсам.
 - 5.1. Системы обнаружения и предотвращения вторжений (IDS/IPS).
 - 5.2. Многофакторная аутентификация (MFA).
 - 5.3. Логирование и анализ подозрительной активности.
6. *Шифрование данных:* Важная информация и конфиденциальные данные (например, персональные данные клиентов и детали транзакций) должны быть зашифрованы для предотвращения утечек и несанкционированного доступа.
 - 6.1. Хранение ключей шифрования в защищенных хранилищах (например, HSM — Hardware Security Module).
 - 6.2. Регулярная ротация ключей шифрования.
 - 6.3. Использование алгоритмов AES-256 для данных в покое.
 - 6.4. Использовать TLS 1.2 или выше для данных в транзите.
7. *Мониторинг безопасности:* Система должна регулярно анализироваться на предмет уязвимостей и потенциальных угроз. Необходимо проводить постоянный мониторинг для выявления и устранения слабых мест.
 - 7.1. Проведение ежеквартальных аудитов безопасности.
 - 7.2. Использование инструментов для сканирования уязвимостей (например, Nessus, Qualys).
 - 7.3. Внедрение системы управления информационной безопасностью (ISMS) по стандарту ГОСТ Р ИСО/МЭК 27001-2012.
8. *Резервное копирование:* Должны быть установлены четкие процедуры резервного копирования данных, а также способы быстрого восстановления информации в случае её утери или повреждения.
 - 8.1. Ежедневное резервное копирование критически важных данных.
 - 8.2. Хранение резервных копий в географически разнесенных местах.

- 8.3. Регулярное тестирование восстановления данных не реже одного раза в квартал.
- 8.4. Время восстановления: Восстановление данных должно быть осуществлено в течение 4 часов после обнаружения утраты или повреждения.
- 9. *Физическая безопасность*: Серверные помещения и оборудование системы "НаКурорт" должны быть защищены физически, включая контроль доступа к таким помещениям и оборудование.
- 10. *Обучение персонала*: Все сотрудники, имеющие доступ к системе, должны проходить регулярное обучение по вопросам информационной безопасности, чтобы следовать установленным правилам и политикам безопасности.
- 11. *Управление инцидентами безопасности*: В системе должны быть предусмотрены процедуры управления инцидентами и реагирования на потенциальные угрозы, включая уведомление ответственных лиц о возможных нарушениях безопасности и анализ их последствий.
- 12. *Соответствие нормативным требованиям*: Система должна соответствовать действующим законодательным и регуляторным требованиям в области защиты информации и персональных данных, включая стандарты по защите информации, такие как GDPR и другие международные нормы.
 - 12.1. GDPR (General Data Protection Regulation) для обработки персональных данных граждан ЕС.
 - 12.2. ФЗ-152 «О персональных данных» (для российских организаций).
 - 12.3. Международные стандарты безопасности информации, такие как ISO/IEC 27001.

4.1.7.b. Требования к антивирусной защите

- 1. *Всеобъемлющая защита*: Антивирусное ПО должно быть установлено на всех компьютерах, серверах и устройствах, связанных с системой "НаКурорт", включая рабочие места пользователей, администраторов и технических специалистов.
- 2. *Централизованное управление*: Антивирусная система должна обеспечивать централизованное управление всеми процессами защиты, включая автоматическое сканирование файлов, удаление

- угроз и подробное протоколирование действий, связанных с вирусной активностью.
3. *Автоматическая инсталляция:* Антивирусное ПО должно устанавливаться автоматически и централизованно на всех рабочих местах сотрудников и серверных системах, исключая необходимость ручной установки.
 4. *Автоматическое обновление:* Обновления вирусных сигнатур и компонентов антивирусного ПО должны происходить автоматически и синхронно для всех устройств в системе, чтобы защитить их от новейших угроз.
 5. *Протоколирование действий:* Все антивирусные действия, включая сканирование, обнаруженные угрозы и меры, принятые администраторами, должны быть подробно записаны в журналы вирусной активности для последующего анализа.
 6. *Гибкое администрирование:* Администраторам системы должны быть предоставлены полные права для настройки и мониторинга антивирусной защиты на всех устройствах. Важно, чтобы администраторы могли оперативно реагировать на угрозы и настраивать правила защиты.
 7. *Регулярное сканирование:* В системе должна быть настроена регулярная проверка компьютеров и серверов на наличие вирусов и вредоносных программ. Сканирование должно проводиться как в автоматическом режиме, так и по запросу администратора.
 8. *Защита в реальном времени:* Антивирусное ПО должно работать в режиме реального времени, отслеживая создание, открытие и выполнение файлов, чтобы предотвратить распространение угроз еще до их активации.
 9. *Карантин и удаление:* При обнаружении вирусов или вредоносных программ система должна автоматически изолировать зараженные файлы в карантин и предоставлять возможность их удаления без угрозы для других данных.
 10. *Автоматические уведомления:* В случае обнаружения критических угроз, антивирусная система должна незамедлительно уведомлять администраторов через систему оповещений для быстрого реагирования на инциденты.

11. *Актуальность защиты*: Антивирусная система должна регулярно обновляться и соответствовать последним требованиям и стандартам в области кибербезопасности, чтобы гарантировать защиту от современных вирусов и угроз.

4.1.7.с. Разграничение ответственности ролей при доступе к системе

Код ответственности	Объект доступа	Роль/Сотрудник/Единица оргструктуры
Ф	Формирование турпакетов	Менеджер по туризму
О	Ответственность за бронирования	Оператор службы поддержки
И	Использование системы для выбора туров	Клиент
О	Отслеживание статуса бронирования	Оператор службы поддержки, Клиент
Ф	Формирование отчетов по продажам и бронированиям	Менеджер по аналитике

4.1.8. Требования по сохранности информации при авариях

Система "НаКурорт" должна обеспечивать сохранность информации при следующих аварийных событиях:

1. *Сбой в электроснабжении*: При кратковременных сбоях электроснабжения система должна автоматически переключаться на бесперебойное питание, продолжая работу без потери данных. В случае длительных сбоев должна быть возможность безопасного автоматического завершения работы системы с сохранением всех данных.
2. *Отказ накопителей данных*: При выходе из строя одного или нескольких жестких дисков или элементов хранилища данные не должны теряться. Система должна иметь возможность автоматически восстанавливать данные из резервных копий и продолжать работу без простоев.

3. *Аварийное завершение работы серверов:* В случае неожиданного отключения серверов, система "НаКурорт" должна обеспечивать целостность данных и иметь механизмы для быстрого восстановления работоспособности.
4. *Сбой в сетевой инфраструктуре:* При сбоях в сетевой инфраструктуре (например, потеря связи или поломка сети), система должна автоматически переключаться на резервные каналы связи или альтернативные способы передачи данных для поддержания работы.
5. *Аварийное завершение работы программных компонентов:* Если программные компоненты системы аварийно завершили работу, система должна автоматически перезапускать их, сохраняя данные и обеспечивая целостность информации.
6. *Потеря связи с базой данных:* При потере связи с базой данных система должна автоматически восстанавливать соединение и возобновлять работу с данными без потери информации.
7. *Сбой в программном обеспечении:* Система должна иметь встроенные механизмы для автоматического обнаружения и исправления сбоев программного обеспечения, что предотвратит потерю данных и обеспечит стабильную работу.
8. *Аварийное завершение работы операционной системы:* В случае аварийного завершения работы операционной системы система должна автоматически перезагружаться, восстанавливая все рабочие процессы и обеспечивая сохранность данных.

4.1.9. Требования к защите от влияния внешних воздействий

Для программно-аппаратного окружения системы "НаКурорт" предъявляются следующие требования по защите от внешних воздействий.

Требования к радиоэлектронной защите:

1. Электромагнитное излучение в радиодиапазоне, возникающее при работе бытовых приборов, электрических установок или передающих устройств, расположенных вблизи аппаратных средств

системы, не должно нарушать работу подсистем системы "НаКурорт".

Требования по стойкости, устойчивости и прочности к внешним воздействиям:

1. Функционирование при колебаниях напряжения: Система должна сохранять работоспособность при колебаниях напряжения электропитания в диапазоне от 155 до 265 В ($220 \pm 20\%$ - 30 %), что обеспечит стабильность работы даже при изменениях в электросети.
2. Температурные условия: Система должна сохранять полную функциональность в диапазоне температур окружающей среды, установленных изготовителем аппаратного обеспечения. Это гарантирует стабильную работу системы в стандартных климатических условиях.
3. Влажность: Аппаратные средства системы "НаКурорт" должны сохранять работоспособность в пределах допустимых значений влажности, рекомендованных производителем оборудования.
4. Устойчивость к вибрациям: Система должна быть способна функционировать в условиях допустимых вибраций, установленных производителем аппаратных средств, что защитит оборудование от повреждений при незначительных механических воздействиях.

4.1.10. Требования по стандартизации и унификации

Разработка системы "НаКурорт" должна проводиться с применением следующих методологий и стандартов:

1. *Методологии моделирования:* Для функционального моделирования используется IDEF0, для информационного моделирования — IE/IDEF1X и UML-диаграммы. Это позволяет структурировать систему и обеспечить её согласованное и стандартизированное развитие.
2. *Стандарты разработки:* Проектирование и разработка системы должны соответствовать рекомендациям по стандартизации Р50.1.028-2001 «Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования».

3. *Средства моделирования*: Моделирование бизнес-процессов системы осуществляется в соответствии с международными стандартами, поддерживаемыми ПО ERWin 4.X.
4. *Языки запросов*: Для работы с базами данных системы "НаКурорт" должен применяться язык SQL, что обеспечит унификацию запросов к различным системам хранения данных.
5. *Языки программирования*: Для разработки пользовательских интерфейсов используются Java (Java 11) и C++. Эти языки программирования обеспечивают кросс-платформенность и стабильную работу системы на различных устройствах.
6. *Использование классификаторов*: При необходимости в системе могут быть использованы единые классификаторы и словари для обработки алфавитно-цифровой и текстовой информации, что повышает унификацию данных в системе.

4.1.11. Дополнительные требования

Для тестирования системы "НаКурорт" на этапе разработки должны предоставляться beta-версии Заказчику для тестирования в отдельных подразделениях компании. Разработчик обязан предоставить полные инструкции для тестирования, а также возможность оперативной обратной связи для устранения выявленных проблем.

4.1.12. Требования безопасности

При внедрении, эксплуатации и обслуживании технических средств системы "НаКурорт" должны выполняться следующие меры безопасности:

1. *Электробезопасность*: Все работы по установке и обслуживанию оборудования должны соответствовать «Правилам устройства электроустановок» и «Правилам техники безопасности при эксплуатации электроустановок потребителей». Это гарантирует защиту персонала и оборудования от электрических повреждений.
2. *Пожарная безопасность*: Аппаратное обеспечение системы должно соответствовать требованиям пожарной безопасности производственных помещений по ГОСТ 12.1.004-91 «ССБТ. Пожарная безопасность. Общие требования». Это позволяет

минимизировать риски возгорания в местах установки серверов и другого оборудования.

3. *Общие требования безопасности:* Обслуживание системы должно соответствовать ГОСТ 12.2.003-91 «ССБТ. Оборудование производственное. Общие требования безопасности». Это касается всех аспектов эксплуатации, включая техническое обслуживание и взаимодействие с аппаратной частью системы.
4. *Заземление оборудования:* Аппаратная часть системы должна быть заземлена в соответствии с требованиями ГОСТ Р 50571.22-2000 «Электроустановки зданий. Часть 7. Требования к специальным электроустановкам. Раздел 707. Заземление оборудования обработки информации». Это обеспечит защиту от статического электричества и снижает риск повреждения оборудования.
5. *Акустический шум:* Уровень шума, создаваемого аппаратурой системы "НаКурорт", должен соответствовать ГОСТ 21552-84 «Средства вычислительной техники. Общие технические требования, приемка, методы испытаний, маркировка, упаковка, транспортирование и хранение». Допустимые значения уровня шума:
 - 5.1. Не более 50 дБ — при работе оборудования без печатающих устройств.
 - 5.2. Не более 60 дБ — при работе оборудования с использованием печатающих устройств.

4.2. Требования к функциям, выполняемым системой

4.2.1. Подсистема сбора, обработки и загрузки данных

4.2.1.а. Временной регламент реализации каждой функции, задачи

Задача	Требования к временному регламенту
Создание, редактирование и удаление турпакетов	Весь период функционирования системы, при возникновении необходимости обновления или изменения турпакетов
Формирование предложений от поставщиков	Весь период функционирования системы, при необходимости

	модификации и обновления предложений поставщиков
Определение и изменение расписания туров	Весь период функционирования системы, при необходимости изменения расписания туров
Запуск процедур бронирования и оплаты	Ежедневно, в любое время суток, по запросу пользователя
Выставление счетов и подтверждение оплат	В течение 24 часов после завершения бронирования, в рабочее время с 09:00 до 18:00
Мониторинг доступности туров и формирование отчетов	Регулярно, при работе подсистемы мониторинга и отчетности
Оперативное уведомление пользователей и менеджеров о нештатных ситуациях	Регулярно, при возникновении нештатных ситуаций в процессе работы системы

4.2.1.b. Требования к качеству реализации функций, задач

Задача	Форма представления выходной информации	Характеристики точности и времени выполнения
Создание, редактирование и удаление турпакетов	Изменение записей в базе данных, отображение туров в интерфейсе платформы	Не более 3 часов для передачи информации в БД
Формирование предложений от поставщиков	Изменение записей в базе данных, обновление предложений в интерфейсе платформы	Не более 3 часов для передачи информации в БД
Определение и изменение расписания туров	Изменение записей в базе данных, обновление расписания в интерфейсе платформы	Не более 3 часов для передачи информации в БД

Запуск процедур бронирования и оплаты	Изменение записей в базе данных, отправка уведомлений пользователям	Не более 3 часов для передачи информации в БД
Выставление счетов клиентам	Изменение записей в базе данных, создание счетов в интерфейсе платформы, текстовый файл	Не более 3 часов для передачи информации в БД
Мониторинг продаж и формирование отчетов для менеджеров	Изменение записей в базе данных, создание отчетов в текстовом файле	Не более 3 часов для передачи информации в БД
Оперативное уведомление пользователей и менеджеров о нештатных ситуациях	Изменение записей в базе данных, рассылка уведомлений	Не более 3 часов для передачи информации в БД. Уведомление о нештатной ситуации — не более 1 часа

4.2.1.с. Перечень критериев отказа для каждой функции

Функция	Критерии отказа	Время восстановления	Коэффициент готовности
Создание, редактирование и удаление турпакетов	Не обновляется информация о турпакете	4 часа	0.95
Формирование предложений от поставщиков	Не выполняется одна из подзадач	8 часов	0.75
Определение и изменение расписания туров	Не выполняется одна из задач функции	12 часов	0.7
Запуск процедур бронирования и оплаты	Не выполняется одна из задач функции	2 часа	0.5
Выставление счетов клиентам	Не обновляется счет	4 часа	0.8
Мониторинг продаж и	Не формируется	12 часов	0.75

формирование отчетов	текстовый файл для менеджеров		
Оперативное уведомление пользователей и менеджеров о нештатных ситуациях	Не рассылаются уведомления	8 часов	0.6

4.3. Требования к видам обеспечения

4.3.1. Требования к математическому обеспечению

1. Применение оптимизированных алгоритмов для сортировки туров и предложений поставщиков с целью обеспечения быстрого поиска и фильтрации информации пользователями системы "НаКурорт".
 - 1.1. *Время отклика:* Алгоритмы сортировки и фильтрации должны обеспечивать время отклика не более 200 миллисекунд при обработке запросов с объемом данных до 10,000 туров и 5,000 предложений от поставщиков.
 - 1.2. *Параллельная обработка:* Поддержка параллельных вычислений для обработки больших объемов данных без значительного увеличения времени отклика.
 - 1.3. *Сложность алгоритмов:* Использование алгоритмов с оптимальной временной сложностью, предпочтительно $O(n \log n)$ для сортировки и $O(n)$ для фильтрации.
 - 1.4. *Кэширование:* Реализация механизмов кэширования для часто используемых запросов и фильтров, снижая нагрузку на серверы и ускоряя обработку данных.
 - 1.5. *Горизонтальная масштабируемость:* Алгоритмы должны быть способны эффективно работать в распределенной среде, поддерживая горизонтальное масштабирование при увеличении объема данных.
 - 1.6. *Обработка пиковых нагрузок:* Способность системы справляться с пиковыми нагрузками (например, при запуске рекламных акций) без ухудшения производительности.

- 1.7. *Настраиваемые параметры:* Возможность настройки критериев сортировки и фильтрации в соответствии с требованиями бизнеса и предпочтениями пользователей.
- 1.8. *Поддержка различных типов данных:* Обработка структурированных и неструктурированных данных, включая текстовые описания, рейтинги, цены и географическое расположение туров.
- 1.9. *Оптимизация по ГОСТам:* Алгоритмы должны соответствовать требованиям ГОСТ Р 57580-2017 «Информационная технология. Термины и определения» и ГОСТ Р ИСО/МЭК 12207-2008 «Процессы жизненного цикла программных средств».
2. Использование алгоритмов предсказания для анализа предпочтений клиентов и формирования персонализированных рекомендаций по турам.
 - 2.1. *Метрики точности:* Достижение не менее 85% точности в рекомендациях, измеряемой через показатели Precision и Recall.
 - 2.2. *Адаптивность моделей:* Возможность обновления и адаптации моделей машинного обучения на основе новых данных и изменений в поведении клиентов.
 - 2.3. *Время генерации рекомендаций:* Генерация персонализированных рекомендаций для пользователя должна происходить не более чем за 500 миллисекунд.
 - 2.4. *Обработка данных в реальном времени:* Поддержка анализа и формирования рекомендаций на основе данных в реальном времени, включая последние действия пользователя.
 - 2.5. *Машинное обучение:* Применение современных алгоритмов машинного обучения и глубинного обучения, таких как Collaborative Filtering, Content-Based Filtering, Matrix Factorization, Neural Networks.
 - 2.6. *Big Data технологии:* Использование технологий обработки больших данных (например, Hadoop, Spark) для анализа и обработки больших объемов данных о пользователях и предложениях.

- 2.7. *Индивидуальные профили пользователей*: Создание и поддержка детализированных профилей пользователей, включая историю просмотров, бронирований, оценок и предпочтений.
- 2.8. *Адаптивные рекомендации*: Рекомендации должны учитывать контекстные факторы, такие как местоположение пользователя, сезонность, текущие тренды.
- 2.9. *API для рекомендаций*: Предоставление API для интеграции рекомендаций в пользовательский интерфейс системы.
- 2.10. *Совместимость*: Обеспечение совместимости с существующими модулями системы "НаКурорт", такими как поисковая система, фильтры и интерфейсы бронирования.
- 2.11. *Соблюдение GDPR и других нормативов*: Обработка и хранение персональных данных клиентов должна соответствовать требованиям GDPR, ФЗ-152 «О персональных данных» и другим применимым нормативным актам.
- 2.12. *Анонимизация данных*: Применение методов анонимизации и псевдонимизации данных при анализе и хранении информации о клиентах.
- 2.13. *Оптимизация по ГОСТам*: Алгоритмы должны соответствовать требованиям ГОСТ Р 57580-2017 и ГОСТ Р ИСО/МЭК 25010-2011 «Системы и программные продукты. Требования к качеству».

4.3.2. Требования к информационному обеспечению

- 1. *Защита данных при сбоях электропитания*: Должна быть предусмотрена возможность автоматического сохранения данных и их безопасного восстановления после сбоев в подаче электропитания.
- 2. *Информационная совместимость с внешними системами*: Система должна быть совместима с информационными системами поставщиков услуг и смежными системами организации-заказчика для обеспечения корректного обмена данными.
- 3. *Применение СУБД*: Для работы с данными системы должна использоваться мощная и гибкая СУБД (например, PostgreSQL или

MySQL), обеспечивающая надежное хранение и быстрое извлечение данных.

4. Восстановление данных: Система должна поддерживать возможность восстановления данных в случае сбоев, повреждений или утрат информации, используя резервные копии и автоматическое дублирование.
5. Хранение и обновление данных: Все данные о турпакетах, пользователях и бронированиях должны храниться в базе данных с возможностью их регулярного обновления по запросу пользователей и администраторов.

4.3.2.а. Требования к составу, структуре и способам организации данных в системе

1. *Структура данных:*
 - 1.1. *Клиенты:* Хранение информации о пользователях, включая личные данные, идентификаторы, историю бронирований, предпочтения по турам и т.д.
 - 1.2. *Туры:* Информация о доступных турпакетах, включая название, описание, стоимость, доступные даты, требования к бронированию и т.д.
 - 1.3. *Поставщики услуг:* Информация о поставщиках, таких как авиакомпании, отели, транспортные компании, включая их контактные данные и доступные услуги.
 - 1.4. *Бронирования:* Данные о текущих и прошлых бронированиях клиентов, статусах заказов, платежах и возвратах.
 - 1.5. *Аналитические данные:* Исторические данные о продажах, популярности туров и пользовательской активности.
2. *Организация данных:*
 - 2.1. База данных должна поддерживать реляционную модель для хранения информации о клиентах, турах и бронированиях, с возможностью связи между различными сущностями.
 - 2.2. Использование гибкой схемы данных для работы с предложениями от поставщиков, включая хранение

неструктурированных данных в формате JSON для легкой интеграции и обновлений.

3. *Способы организации данных:*

- 3.1. Данные о клиентах и турах должны быть структурированы в таблицах с соответствующими полями для быстрого поиска и анализа.
- 3.2. Хранение данных о бронированиях и платежах должно быть организовано с возможностью детального отслеживания статусов заказов и истории платежей.
- 3.3. Реализация индексов для оптимизации поиска данных о пользователях, бронированиях и статусах туров.

4. *Система резервного копирования:*

- 4.1. Регулярное резервное копирование всех данных системы с возможностью восстановления информации в случае сбоя или потерь.
- 4.2. Хранение резервных копий в безопасных и изолированных хранилищах для предотвращения потерь данных.

5. *Защита данных:*

- 5.1. Обеспечить шифрование данных клиентов и бронирований для защиты конфиденциальной информации.
- 5.2. Аутентификация пользователей и строгие меры контроля доступа для предотвращения несанкционированного доступа к системе.

6. *Производительность и масштабируемость:*

- 6.1. Учесть возможность масштабирования системы при увеличении числа клиентов, туров и бронирований.
- 6.2. Оптимизировать запросы и индексацию для обеспечения высокой производительности системы, даже при больших объемах данных.

7. *Регистрация и журналирование:*

- 7.1. Вести журнал событий, действий пользователей и изменений данных для обеспечения отслеживаемости и анализа работы системы.
- 7.2. Обеспечить возможность мониторинга работы системы для выявления и предотвращения нештатных ситуаций.

4.3.2.b. Требования к информационному обмену между компонентами системы

Компоненты системы	Система управления клиентами	Система управления турами	Система управления поставщиками
Система управления клиентами	X	X	
Система управления турами	X		X
Система управления поставщиками		X	X

4.3.2.c. Требования к информационной совместимости со смежными системами

Система "НаКурорт" должна поддерживать передачу и прием данных от смежных систем Заказчика через интерфейсные файлы или таблицы, обеспечивая бесперебойный обмен информацией между внутренними и внешними системами.

4.3.2.d. Требования по использованию классификаторов и унифицированных документов

Система должна использовать единую базу данных клиентов и поставщиков услуг, что позволит централизованно управлять информацией о пользователях, партнерах и других участниках системы.

4.3.2.e. Требования по применению систем управления базами данных

Для реализации подсистемы хранения данных системы "НаКурорт" должна использоваться промышленная СУБД PostgreSQL, которая обеспечит надежное и эффективное управление базами данных.

4.3.2.f. Требования к защите данных от разрушений при авариях и сбоях в электропитании системы

1. *Надежное хранение данных:* Информация, хранящаяся в базе данных системы "НаКурорт", должна надежно сохраняться в случае аварийных ситуаций, связанных с сбоями в электропитании.
2. *Бесперебойное электропитание:* Система должна быть оборудована источником бесперебойного питания (ИБП), который способен поддерживать работу системы в течение не менее 30 минут в случае отключения внешнего энергоснабжения.
3. *Автоматическое резервное копирование:* Регулярное автоматическое резервное копирование данных должно осуществляться с заданной периодичностью. Объем резервных копий должен быть достаточным для полного восстановления системы в случае сбоя.
4. *Физическая защита резервных копий:* Резервные копии данных должны храниться в надежном и физически отдельном от основной системы месте для обеспечения сохранности информации в случае повреждения основной системы.
5. *Автоматическое восстановление системы:* После восстановления электропитания система должна автоматически возобновить свою работоспособность, используя последние созданные резервные копии для восстановления данных.

4.3.2.g. Требования к контролю, хранению, обновлению и восстановлению данных

Требования к контролю данных:

1. Система "НаКурорт" должна отслеживать все действия пользователей: создание бронирований, выбор туров, изменение личных данных и использование бонусов.
2. Все изменения в профилях клиентов, информации о турах и предложениях поставщиков должны записываться с указанием времени и идентификатора пользователя.
3. Должна быть возможность просмотра и проверки всех изменений, внесенных в систему, для проведения аудита и анализа.

4. Система должна поддерживать различные уровни доступа к данным, в зависимости от роли пользователя (клиент, менеджер, администратор).

Требования к хранению данных:

1. Все данные о клиентах, бронированиях, платежах и турах должны храниться в зашифрованном виде, защищенные от несанкционированного доступа.
2. Информация о бронированиях и платежах должна храниться на протяжении всего периода обслуживания клиента.
3. После завершения активного использования данные клиентов и их бронирования должны быть перемещены в архивное хранилище и сохраняться там в течение 10 лет.

Требования к обновлению данных:

1. Информация о турах и предложениях от поставщиков должна регулярно обновляться, чтобы отражать актуальные изменения и предложения.
2. Администраторам и менеджерам должна быть предоставлена возможность обновления данных о турах, бронированиях и клиентах.

Требования к восстановлению данных:

1. Система должна автоматически создавать резервные копии всех данных ежедневно на отдельный физический носитель.
2. Хранилище резервных копий должно располагаться в безопасном и удалённом месте относительно основного сервера для обеспечения сохранности данных.
3. В случае неполадок или потери информации система должна обеспечивать возможность быстрого восстановления данных из резервных копий.

4. Проверка работоспособности системы восстановления данных должна проводиться каждые три месяца для предотвращения потерь информации.

4.3.2.h. Требования к процедуре придания юридической силы документам, продуцируемым техническими средствами системы

Требования не предъявляются. Система "НаКурорт" не предусматривает генерацию юридически значимых документов.

4.3.3. Требования к лингвистическому обеспечению

Для лингвистического обеспечения системы "НаКурорт" предъявляются следующие требования к используемым языкам программирования, взаимодействия с пользователем и техническими средствами системы, а также требования к кодированию данных и организации диалога:

1. *Основные языки разработки и программирования:* Для логики приложения следует использовать язык Java, а для обработки и хранения данных — SQL. Это обеспечит высокую производительность и надежность системы.
2. *Интерфейс и взаимодействие с пользователем:* Для создания графического интерфейса и взаимодействия с пользователем применяются языки HTML, JavaScript и CSS. Java и JavaScript должны использоваться для обеспечения интерактивности и динамического контента, позволяя пользователям эффективно взаимодействовать с системой.
3. *Стандарты и форматы для API и интеграции:* Для интеграции с другими системами и обмена данными следует использовать форматы JSON или XML, которые обеспечат гибкость и совместимость при передаче данных между различными компонентами системы.
4. *Кодировка данных:* Все данные в системе должны кодироваться в формате UTF-8, что гарантирует поддержку многоязычных интерфейсов и универсальность работы с различными наборами символов.

5. *Запросы к базе данных:* Для работы с базами данных применяется SQL, а для более сложных операций можно использовать процедурные расширения, такие как PL/SQL для управления данными и запросами.
6. *Средства для описания предметной области:* Для описания и моделирования предметной области (турагентства и его процессов) следует использовать инструменты типа Erwin, что обеспечивает структурированное представление бизнес-процессов.
7. *Пользовательский интерфейс:* Интерфейс системы должен быть ориентирован на графическое и оконное представление, с акцентом на удобство и интуитивное использование для всех категорий пользователей. Система должна быть адаптивной, чтобы корректно отображаться на мобильных устройствах, обеспечивая пользователям гибкий доступ к функциональности в любом месте.

4.3.4. Требования к программному обеспечению

4.3.4.a. Функциональные требования

1. Регистрация и учет профилей клиентов, менеджеров и поставщиков.
2. Возможность добавления, редактирования и удаления туров с указанием их характеристик, стоимости и доступных услуг.
3. Механизм выбора туров клиентами с учетом их предпочтений и истории бронирований.
4. Отслеживание статусов бронирований и транзакций. Менеджеры должны иметь возможность подтверждать и изменять статусы заказов.
5. Возможность для менеджеров генерировать типовые предложения для клиентов.
6. Генерация отчетов по продажам и бронированиям для аналитического отдела.
7. Возможность проведения платежных транзакций и учета бонусных систем.

4.3.4.b. Нефункциональные требования

1. Программное обеспечение должно поддерживать работу на операционных системах Windows, Linux, macOS.

2. Интерфейс системы должен быть интуитивно понятным для всех категорий пользователей: клиентов, менеджеров и поставщиков.
3. Высокий уровень безопасности для защиты персональных данных клиентов и конфиденциальной информации поставщиков.
4. Возможность масштабирования и модернизации системы с увеличением объема данных и числа пользователей.

4.3.4.с. Перечень покупных программных средств

1. СУБД: (выбор заказчика, например, PostgreSQL, MySQL).
2. ETL-средство: (выбор заказчика, например, Talend, Apache Nifi).
3. BI-приложение: (выбор заказчика, например, Power BI, Tableau).

4.3.4.d. Требования к качеству ПС

1. Функциональность, надежность, легкость применения, эффективность, и сопровождаемость должны соответствовать вышеуказанным требованиям.
2. Контроль качества на всех этапах разработки, включая тестирование и ревизии.

4.3.5. Требования к техническому обеспечению

Техническое обеспечение системы "НаКурорт" должно включать серверные мощности, сетевое оборудование и клиентские устройства, совместимые с операционными системами Windows, macOS и Linux. Все компоненты должны обеспечивать стабильную работу системы при высоких нагрузках, связанных с бронированиями и транзакциями. Оборудование должно поддерживать работу с бесперебойным питанием и обеспечивать резервное копирование данных.

4.3.6. Требования к метрологическому обеспечению

Требования не предъявляются. Проект "НаКурорт" не включает задачи, связанные с измерением физических величин или калибровкой оборудования, что характерно для систем, работающих с метрологией.

4.3.7. Требования к организационному обеспечению

4.3.7.a. Требования к структуре и функциям подразделений

1. *Отдел продаж*: Ответственен за создание, обновление и модификацию предложений туров, а также за анализ отчетов по продажам и бронированиям.
2. *Поставщики услуг*: Обеспечивают предоставление актуальных данных о доступных услугах (отели, перелеты, экскурсии) и могут предлагать изменения в своих предложениях.
3. *Клиенты*: Выбирают туры, бронируют услуги и могут оставлять отзывы на основании своих впечатлений о путешествии.
4. *IT-подразделение*: Обеспечивает эксплуатацию, поддержку и модернизацию системы, обрабатывает запросы пользователей.

4.3.7.b. Требования к организации функционирования системы:

1. Все запросы на изменение функциональности или доработку системы должны проходить через IT-подразделение, где они регистрируются в электронной системе тикетов.
2. IT-подразделение регулярно проводит консультации и обучение пользователей для повышения эффективности использования системы.
3. Плановые технические работы и обновления системы должны быть анонсированы заранее с четким указанием времени начала и окончания работ.

4.3.7.c. Требования к защите от ошибочных действий персонала

1. Разграничение доступа к системе основано на ролевой модели: клиенты, поставщики, менеджеры и IT-подразделение имеют разные уровни доступа.
2. Все действия пользователей регистрируются в системе, что позволяет восстановить исходное состояние данных в случае ошибочных действий.

3. Критические действия (например, удаление туров) требуют двухфакторной аутентификации или подтверждения другим пользователем с соответствующими правами.
4. Руководство пользователя должно регулярно обновляться после изменений в функциональности системы.

4.3.8. Требования к методическому обеспечению

Требования включают в себя создание документации и руководств для пользователей, которые будут содержать детальные инструкции по использованию системы "НаКурорт". Методическое обеспечение должно быть обновляемым, чтобы учитывать любые изменения в системе.

4.3.9. Требования к патентной чистоте

Система "НаКурорт" должна соответствовать требованиям патентной чистоты на территории России. Все программные и аппаратные компоненты системы должны быть лицензированы и не нарушать существующие патенты и лицензии третьих лиц. Ответственное лицо должно провести проверку на предмет соблюдения этих требований до внедрения системы.

5. Состав и содержание работ по созданию системы

Работы по созданию системы "НаКурорт" выполняются в семь этапов:

1. Подготовительный этап (инициация проекта)

1.1. Формирование команды проекта.

1.2. Разработка видения продукта и определение ключевых целей проекта.

1.3. Сбор и анализ требований, определение приоритетов для первой версии продукта (MVP).

1.4. Создание Product Backlog — списка всех требований и функциональности для системы.

Продолжительность этапа: 1 месяц (01.10.2024 – 31.10.2024).

2. Проектирование (архитектура и MVP)

2.1. Разработка концепции системы и архитектуры.

2.2. Проектирование минимально жизнеспособного продукта (MVP) с включением только необходимой функциональности для первых пользователей.

2.3. Определение архитектурных компонентов и приоритизация их реализации по спринтам.

2.4. Планирование спринтов для разработки первой версии системы.

Продолжительность этапа: 1,5 месяца (01.11.2024 – 15.12.2024).

3. Разработка MVP (минимально жизнеспособного продукта)

3.1. Разработка первых инкрементов системы, включающих ключевую функциональность для первоначальной версии.

3.2. Постоянная интеграция и тестирование каждого инкремента.

3.3. Проведение демонстрации по окончании каждого спринта и получение обратной связи от заказчика.

3.4. Внесение корректировок на основе обратной связи и приоритизация новых задач.

Продолжительность этапа: 2 месяца (15.12.2024 – 15.02.2025).

4. Полная разработка системы

4.1. Расширение системы новыми функциями и улучшениями на основе отзывов по MVP.

4.2. Параллельная разработка интерфейсов, баз данных и интеграционных модулей.

4.3. Адаптация системы под запросы заказчика и новые требования.

4.4. Проводится разработка инкрементов с тестированием по окончании каждого спринта.

Продолжительность этапа: 3 месяца (15.02.2025 – 15.05.2025).

5. *Тестирование системы*

5.1. Полное функциональное тестирование всех компонентов системы.

5.2. Тестирование интеграции с внешними сервисами и системами.

5.3. Проведение пользовательского тестирования (User Acceptance Testing, UAT) для проверки соответствия системы ожиданиям заказчика.

5.4. Устранение ошибок и багов, выявленных в ходе тестирования.

Продолжительность этапа: 1,5 месяца (15.05.2025 – 30.06.2025).

6. *Внедрение и ввод в эксплуатацию*

6.1. Внедрение системы на стороне заказчика и интеграция с существующими системами.

6.2. Обучение пользователей и технического персонала работе с системой.

6.3. Проведение пилотного запуска системы и финальная проверка.

6.4. Передача системы в эксплуатацию.

Продолжительность этапа: 1,5 месяца (01.07.2025 – 15.08.2025).

7. *Поддержка и сопровождение*

7.1. Обеспечение технической поддержки в период эксплуатации.

7.2. Реализация доработок и улучшений на основе отзывов пользователей.

7.3. Мониторинг работы системы и устранение выявленных проблем.

Конкретные сроки выполнения стадий и этапов разработки и создания системы "НаКурорт" определяются Планом выполнения работ, который является неотъемлемой частью Договора на выполнение работ по настоящему техническому заданию.

Перечень организаций-исполнителей работ и назначение ответственных за проведение этих работ определяется Договором.

Возможна следующая таблица для укрупненного описания работ по каждому этапу:

Этап	Основные работы	Выходные результаты	Ответственность
Проектирование	Разработка эскизного и технического проекта	Готовый эскизный и технический проект	Разработчик
Разработка рабочей документации	Создание документации, адаптация ПО	Рабочая документация, тестирование системы	Разработчик, участие Заказчика
Ввод в действие	Внедрение системы, обучение пользователей	Система введена в эксплуатацию, обученные пользователи	Разработчик, Заказчик

6. Порядок контроля и приемки системы

6.1. Виды и объем испытаний системы

Система "НаКурорт" подвергается испытаниям следующих видов:

1. Предварительные испытания.
2. Опытная эксплуатация.
3. Приемочные испытания.

Состав, объем и методы проведения предварительных испытаний системы "НаКурорт" определяются документом «Программа и методика испытаний», который разрабатывается на стадии «Разработка рабочей документации».

Состав, объем и методы проведения опытной эксплуатации системы определяются документом «Программа опытной эксплуатации», разрабатываемым на стадии «Ввод в действие».

Состав, объем и методы проведения приемочных испытаний системы определяются документом «Программа и методика испытаний», который разрабатывается на стадии «Ввод в действие» с учетом результатов предварительных испытаний и опытной эксплуатации.

7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

7.1. Технические мероприятия

1. Осуществлена подготовка помещения для размещения серверов и оборудования системы "НаКурорт" в соответствии с требованиями, изложенными в данном техническом задании.
2. Закуплено и установлено необходимое серверное и сетевое оборудование для функционирования системы.
3. Организовано сетевое взаимодействие между всеми компонентами системы, включая каналы передачи данных для интеграции с внешними поставщиками и сервисами.

7.2. Организационные мероприятия

1. Организован доступ к базам данных туристических поставщиков для обмена информацией о турах, услугах и бронированиях.
2. Определен регламент информирования об изменениях в структуре данных или услугах, предоставляемых внешними поставщиками.
3. Назначены ответственные специалисты со стороны Заказчика для взаимодействия с проектной командой по вопросам интеграции и работы с системами поставщиков данных.

7.3. Изменения в информационном обеспечении

Изменения в текущих информационных системах Заказчика не предполагаются. Все обновления в базах данных системы "НаКурорт" будут происходить автоматически, обеспечивая актуальность данных о турах, бронированиях и платежах без необходимости ручного вмешательства.

8. Требования к документированию

Этап	Документ
Проектирование. Разработка концепции и технического проекта	Ведомость концепции проекта
	Пояснительная записка к концепции проекта
	Ведомость технического проекта
	Пояснительная записка к техническому проекту
	Схема архитектуры системы
	Функциональная схема взаимодействия модулей
Разработка общей документации. Адаптация программ	Ведомость машинных носителей данных
	Паспорт системы
	Общее описание системы
	Технологическая инструкция
	Руководство пользователя
	Описание технологического процесса обработки данных
	Инструкция по работе с базами данных и каталог баз данных

	Программа и методика испытаний
	Спецификация компонентов
	Описание программных модулей и текст программ
Ввод в действие	Акт приемки в опытную эксплуатацию
	Протокол испытаний
	Акт приемки системы в промышленную эксплуатацию
	Акт завершения работ

В рамках определения обязательных и опциональных требований для системы "НаКурорт" мы выделили ключевые задачи и сгруппировали их для упрощения понимания приоритетов. Для приоритизации мы применили метод MoSCoW, который позволяет четко разделить требования на те, что должны быть реализованы в первую очередь, и на те, что могут быть выполнены на последующих этапах.

Использование метода MoSCoW помогает как проектной команде, так и заказчику лучше осознать приоритеты, выделить наиболее важные элементы системы, а также сфокусироваться на реализации функциональности, которая имеет наибольшее значение для конечных пользователей. Это также значительно упрощает процесс управления изменениями и позволяет гибко реагировать на корректировки и новые запросы, возникающие в ходе разработки.

8.1. Обязательные и опционные требования к ПО

8.1.1. Must Have (Должно быть)

1. *Подбор турпакетов*: Обязательно должна быть реализована автоматическая сборка турпакетов, включающая перелет, трансфер, проживание и экскурсии, на основе предпочтений клиента. Это основная функция системы.

2. *Мастер выбора турпакетов*: Необходимо предоставить мастера, который позволит клиентам последовательно выбирать и комбинировать услуги по своему запросу.
3. *Интеграция с поставщиками*: Система должна поддерживать интеграцию с системами поставщиков для получения актуальных данных об услугах, классах обслуживания и ценах в режиме реального времени.
4. *Регистрация и аутентификация*: Система должна включать механизм регистрации и аутентификации пользователей для безопасности и персонализации процесса бронирования.
5. *Оплата услуг*: Клиенты должны иметь возможность оплатить выбранные туры через банковские транзакции. Эта функция обязательна для завершения процесса бронирования.
6. *Управление турпакетами*: Администраторы системы должны иметь возможность управлять типовыми шаблонами туров, редактировать предложения и добавлять новые.
7. *Служба поддержки*: Клиенты должны иметь возможность обратиться в службу поддержки для решения технических проблем и получения помощи по вопросам бронирования.

8.1.2. Should Have (Желательно)

1. *Уведомления*: Система должна отправлять клиентам уведомления о статусе их бронирований, изменениях цен или доступности услуг. Это улучшит пользовательский опыт.
2. *Аналитика для администраторов*: Желательно добавить модуль аналитики для отслеживания ключевых показателей эффективности системы (количество проданных туров, средний чек, активные пользователи).
3. *Оценки и отзывы*: Клиенты должны иметь возможность оставлять отзывы и оценки о турах, что позволит повысить доверие и качество услуг.
4. *Интеграция с платежными системами*: Желательно предусмотреть возможность подключения различных платежных систем, чтобы у клиентов был выбор способов оплаты.

8.1.3. Could Have (Могло бы быть)

1. *Персонализированные рекомендации*: Система может предлагать персонализированные турпакеты на основе истории бронирований и предпочтений клиентов. Это улучшит качество взаимодействия с пользователями.
2. *Кросс-продажи*: В процессе бронирования клиентам могут быть предложены дополнительные услуги, такие как страхование или аренда автомобиля, что увеличит доход системы.
3. *Адаптивный интерфейс для мобильных устройств*: Возможность адаптации интерфейса системы для работы на мобильных устройствах через веб-приложение.

8.1.4. Won't Have (Не будет включено)

1. *Разработка отдельного мобильного приложения*: На данном этапе не планируется создание отдельного мобильного приложения для доступа к системе "НаКурорт", поскольку интерфейс будет доступен через веб-версию.

3. Модель жизненного цикла и методология разработки ПО

3.1. Модель жизненного цикла программного обеспечения

Для проекта «НаКурорт» наиболее подходящей моделью жизненного цикла разработки программного обеспечения является инкрементная модель в сочетании с agile-подходом. Основные причины выбора таких моделей обусловлены особенностями проекта и его требований.

Инкрементная модель подходит для «НаКурорт» благодаря своей способности постепенно наращивать функциональность системы и обеспечивать гибкость разработки. Этот подход позволяет разрабатывать и внедрять отдельные части системы, такие как модули подбора турпакетов или интеграции с поставщиками, поэтапно. В результате каждый новый

инкремент системы добавляет новую функциональность, которая уже может использоваться клиентами и получать обратную связь.

3.1.1. Основные преимущества инкрементной модели для проекта «НаКурорт»

1. *Постепенное внедрение и тестирование функциональности:* В каждом инкременте реализуется ограниченный набор функций, что позволяет быстрее выявлять и исправлять ошибки на ранних этапах. Например, первым инкрементом может быть базовая версия системы, обеспечивающая подбор турпакетов без учета всех пожеланий клиента, а следующими инкрементами — добавление персонализированных предложений, интеграций с поставщиками и систем оплаты.
2. *Эффективное управление рисками:* В проекте с множественными интеграциями с внешними системами (авиакомпания, гостиницы и т. д.) инкрементная модель позволяет минимизировать риски, так как каждая интеграция реализуется и тестируется поочередно. Это снижает вероятность возникновения критических ошибок в финальной версии.
3. *Гибкость и возможность адаптации:* По мере изменения требований или обнаружения новых потребностей инкрементная модель позволяет добавлять или изменять функциональные блоки системы без необходимости пересмотра всей архитектуры. Например, если появляется необходимость в новом типе экскурсий или изменении логики подбора туров, это можно легко внедрить в одном из следующих инкрементов.

Использование agile-подхода в сочетании с инкрементной моделью позволит команде разработки эффективно реагировать на изменения требований и оперативно вносить корректировки в план разработки. Это особенно актуально для туристической отрасли, где предпочтения клиентов и рыночные условия могут меняться довольно быстро.

3.1.2. Основные преимущества agile-подхода

1. *Постоянное взаимодействие с заказчиком*: Регулярные встречи и демонстрации текущего функционала позволяют оперативно получать обратную связь от заказчика и вносить изменения в проект на любом этапе, что повышает удовлетворенность клиента и снижает вероятность несоответствия конечного продукта ожиданиям.
2. *Гибкое планирование и приоритизация*: Agile-подход позволяет изменять приоритеты задач и функционала на основе текущих потребностей проекта и заказчика. Например, если интеграция с новым поставщиком услуг становится более приоритетной, ее можно внести в план следующего спринта, не затрагивая другие части проекта.

3.2. Методология разработки программного обеспечения

Для разработки системы «НаКурорт» наиболее подходящей является инкрементная модель жизненного цикла с использованием agile-подхода и методологии Scrum. Это сочетание обеспечивает гибкость, адаптивность и прозрачность, которые необходимы для динамичных проектов в туристической индустрии, где поставщики услуг и предпочтения клиентов могут быстро меняться.

Основное преимущество инкрементной модели заключается в постепенном наращивании функциональности системы. Каждый новый инкремент добавляет новую часть функционала, такую как модули для подбора турпакетов или интеграции с поставщиками, что позволяет предоставлять заказчику работающую версию продукта на каждом этапе разработки. Agile-подход помогает оперативно адаптироваться к изменениям требований и приоритетов, сохраняя высокую скорость и качество разработки.

Методология Scrum, используемая в рамках этой модели, делит процесс разработки на короткие итерации — спринты, которые обычно длятся от двух до четырех недель. В конце каждого спринта команда предоставляет заказчику готовый инкремент продукта с новым функционалом, что позволяет регулярно получать обратную связь и

вносить необходимые корректировки на ранних этапах. Это снижает риски накопления ошибок и несоответствия системы ожиданиям пользователей.

Прозрачность процессов, которую обеспечивает Scrum, является ключевым фактором успешной реализации проекта. Каждый член команды знает свои задачи и ответственность, что упрощает взаимодействие и повышает продуктивность. Ежедневные стендап-встречи, планирование спринтов и ретроспективы помогают команде контролировать ход разработки, своевременно выявлять проблемы и находить решения.

Постоянная обратная связь с заказчиком в процессе проведения демонстраций по завершении каждого спринта позволяет корректировать направление работы и улучшать продукт, ориентируясь на реальные потребности и пожелания пользователей. Такой подход обеспечивает высокое качество системы, гибкость в изменении функционала и эффективное управление проектом.

Таким образом, использование инкрементной модели с agile-подходом и методологией Scrum идеально подходит для разработки системы «НаКурорт». Это сочетание позволяет команде быстро реагировать на изменения, снижать риски и поддерживать высокий уровень прозрачности и контроля на каждом этапе, обеспечивая успешную реализацию проекта и удовлетворенность заказчика.

3.3. Сравнение с другими моделями жизненного цикла и методологиями разработки ПО

3.3.1. Модели жизненного цикла ПО

3.3.1.a. Водопадная модель (Waterfall)

Недостатки для проекта «НаКурорт»:

1. *Низкая гибкость:* Трудно вносить изменения после завершения этапов.

2. *Длиительные циклы разработки*: Отсутствие ранних рабочих версий продукта.
3. *Высокие риски несоответствия ожиданиям заказчика*, так как обратная связь поступает только в конце проекта.

Заключение: Не подходит для динамичного проекта, требующего частых изменений и быстрой адаптации.

3.3.1.b. Спиральная модель

Недостатки для проекта «НаКурорт»:

1. *Высокие затраты на управление рисками*: Требуется значительных ресурсов для постоянного анализа рисков.
2. *Сложность реализации*: Необходимость глубокого понимания рисков и постоянного их мониторинга.

Заключение: Может быть избыточной для проекта с четко определенными требованиями и умеренным уровнем рисков.

3.3.1.c. V-модель (V-Model)

Недостатки для проекта «НаКурорт»:

1. *Жесткая структура*: Трудности при внесении изменений в требования на поздних стадиях.
2. *Ограниченная гибкость*: Не позволяет эффективно реагировать на изменения в требованиях.

Заключение: Не подходит для проектов, требующих гибкости и частых итераций.

3.3.2. Методологии разработки ПО

3.3.2.a. Kanban

Недостатки для проекта «НаКурорт»:

1. *Отсутствие четких временных рамок и итераций* может затруднить планирование.

2. Меньше структурированности по сравнению со Scrum, что может привести к потере фокуса на ключевых целях.

Заключение: Может быть использована в сочетании с Scrum, но сама по себе не обеспечивает необходимую структурированность для данного проекта.

3.3.2.b. DevOps

Недостатки для проекта «НаКурорт»:

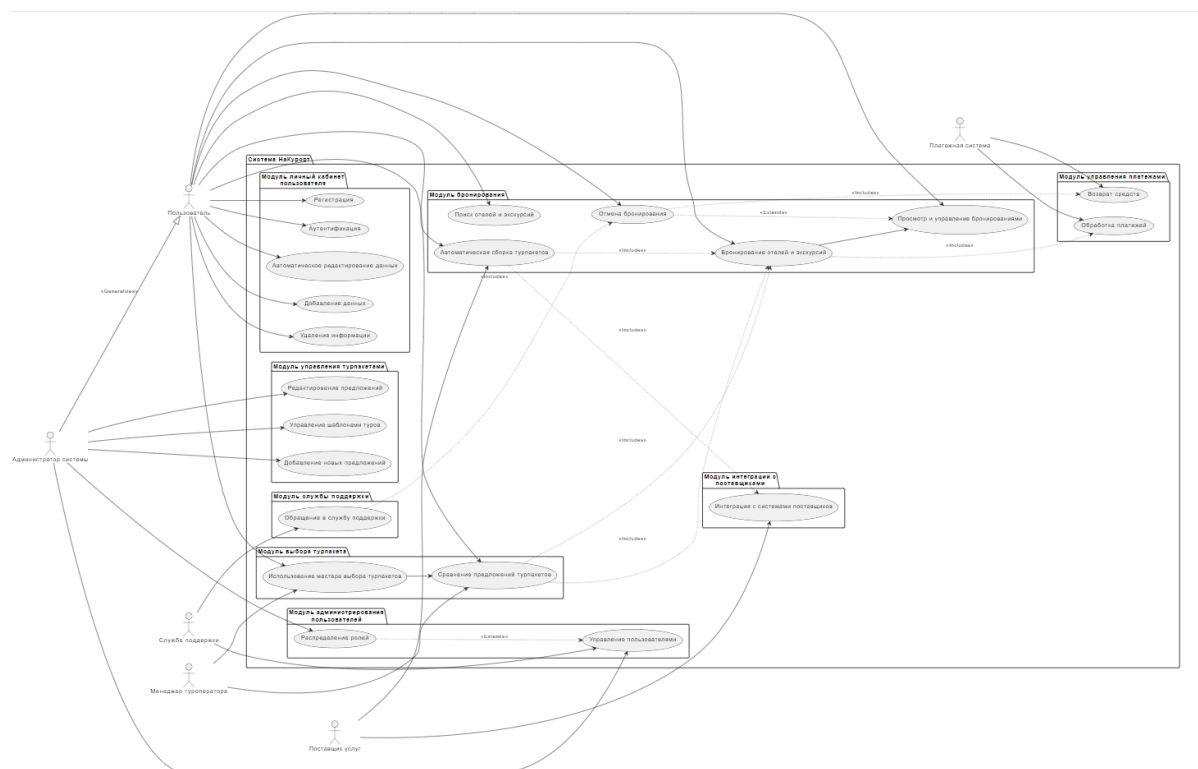
1. Не является полной методологией разработки, а скорее дополняет существующие подходы.
2. Требуется значительных изменений в организационной культуре и процессах.

Заключение: Подходит как дополнение к Scrum для улучшения процессов развертывания и эксплуатации, но не заменяет методологию разработки.

4. Описание и разработка диаграммы прецедентов

4.1. Моделирование диаграммы прецедентов

4.1.1. Инструмент PlantUML



4.2. Описание диаграммы

Диаграмма прецедентов системы "НаКурорт" отражает основные функциональные возможности системы, а также взаимодействие пользователей и администраторов с различными модулями. В систему включены четыре основных модуля: бронирование, выбор турпакетов, администрирование пользователей и управление платежами. Эти модули были выбраны на основе обязательных и желательных требований к функциональности системы.

4.2.1. Роли взаимодействия с системой

1. *Пользователь*: Основной клиент системы, использует функционал поиска, бронирования и выбора туров.
2. *Администратор системы*: Управляет пользователями, выполняет настройки системы, редактирует данные, управляет турпакетами.
3. *Менеджер туроператора*: Управляет турпакетами, обновляет предложения.
4. *Служба поддержки*: Обрабатывает запросы пользователей, помогает в решении проблем с бронированием.
5. *Платежная система*: Обрабатывает все транзакции, связанные с оплатой услуг.

4.2.2. Описание прецедентов

4.2.2.a. Модуль бронирования

1. *Поиск отелей и экскурсий*: Позволяет пользователю найти подходящие варианты на основе заданных параметров (даты, тип жилья, услуги и пр.).
2. *Бронирование отелей и экскурсий*: Позволяет пользователю совершить бронирование и оплатить выбранные услуги.
3. *Просмотр и управление бронированиями*: Позволяет пользователю управлять своими активными бронированиями (просмотр, изменение, отмена).
4. *Отмена бронирования*: Специальный прецедент для отмены существующих бронирований.

4.2.2.b. Модуль выбора турпакета

1. *Использование мастера выбора турпакетов*: Позволяет пользователю последовательно выбрать и комбинировать различные услуги, включая перелет, проживание и экскурсии.
2. *Сравнение предложений турпакетов*: Позволяет сравнить несколько турпакетов по различным критериям и выбрать наиболее подходящий.

4.2.2.с. Модуль администрирования пользователей

1. *Распределение ролей:* Администратор может назначать роли пользователям для разграничения прав доступа.
2. *Управление пользователями:* Управление всей информацией и доступами пользователей системы.

4.2.2.d. Модуль управления турпакетами

1. *Управление шаблонами туров:* Администраторы могут управлять типовыми шаблонами туров.
2. *Редактирование предложений:* Администраторы могут редактировать существующие предложения туров.
3. *Добавление новых предложений:* Администраторы могут добавлять новые предложения в систему.

4.2.2.e. Модуль управления платежами

1. *Обработка платежей:* Обеспечивает проведение платежей через систему.
2. *Возврат средств:* Позволяет пользователям получать возврат средств при отмене бронирования.

4.2.2.f. Модуль личный кабинет пользователя

1. *Автоматическое редактирование данных:* Пользователи могут самостоятельно редактировать свои личные данные.
2. *Добавление данных:* Пользователи могут добавлять новые данные в свой личный кабинет.
3. *Удаление информации:* Пользователи могут удалять ненужную или устаревшую информацию из своего личного кабинета.

4.2.2.g. Модуль службы поддержки

1. *Обращение в службу поддержки:* Пользователи могут отправить запрос в службу поддержки для решения технических проблем или получения помощи по вопросам бронирования.

4.3. Обоснование выбора функций

Функции, представленные на диаграмме прецедентов системы "НаКурорт", были выбраны для моделирования ключевых аспектов работы системы в ее MVP-версии. Эти функции обеспечивают основную ценность для пользователей и решают их ключевые проблемы, связанные с поиском, бронированием и оплатой туров, а также управлением ими.

Модуль	Обоснование выбора
<i>Бронирование</i>	Функции обеспечивают основную задачу клиента — возможность легко и безопасно забронировать тур или услугу, управлять своим бронированием и получать возврат при отмене. Это делает систему удобной и эффективной для пользователей.
<i>Выбор турпакета</i>	Функции помогают пользователям легко выбирать наиболее подходящие туры, что решает одну из ключевых болей клиентов — сложность в выборе и сравнении туров. Это делает процесс бронирования гибким и удобным для пользователя.
<i>Администрирование пользователей</i>	Функции необходимы для обеспечения безопасности и стабильной работы системы. Администраторы могут контролировать доступ пользователей и управлять информацией, поддерживая актуальность данных и безопасность системы.
<i>Управление турпакетом</i>	Функции позволяют системе быть гибкой и актуальной, обеспечивая клиентам доступ к современным предложениям туров. Возможность редактировать и добавлять новые туры позволяет быстро реагировать на изменения рынка и предпочтения пользователей.
<i>Управление платежами</i>	Функции являются обязательными для завершения процесса бронирования. Безопасность и надежность транзакций необходимы для удовлетворения потребностей клиентов, а возможность возврата средств при отмене бронирований делает систему более гибкой и ориентированной на клиента.
<i>Личный кабинет пользователя</i>	Функции позволяют пользователям самостоятельно управлять своей информацией, что упрощает процесс использования системы и снижает нагрузку на администраторов. Возможность редактирования и удаления данных улучшает пользовательский опыт и помогает поддерживать актуальность информации.

4.4. Описание аномальных сценариев

4.4.1. Модуль бронирования

1. Нормальный сценарий: Бронирование отелей и экскурсий

1. Пользователь заходит в раздел бронирования.
2. Вводит необходимые параметры (даты, количество гостей).
3. Находит подходящий отель или экскурсию.
4. Выбирает отель или экскурсию для бронирования.
5. Переходит к оплате и успешно завершает транзакцию.
6. Система подтверждает бронирование и отправляет письмо с подтверждением на электронную почту.

Аномальный сценарий 1: Ошибка при оплате

1. Пользователь заходит в раздел бронирования.
2. Вводит необходимые параметры (даты, количество гостей).
3. Находит подходящий отель или экскурсию.
4. Выбирает отель или экскурсию для бронирования.
5. Переходит к оплате, но система отклоняет платеж (например, из-за недостатка средств на карте).

2. Нормальный сценарий: Просмотр и управление бронированиями

1. Пользователь заходит в личный кабинет.
2. Переходит в раздел "Мои бронирования".
3. Просматривает список своих активных бронирований.
4. Выбирает конкретное бронирование для просмотра или изменения.
5. Вносит изменения (например, изменяет даты или количество гостей) и сохраняет их.
6. Система подтверждает успешное обновление бронирования.

Аномальный сценарий 2: Попытка изменения недоступного бронирования

1. Пользователь заходит в личный кабинет.
2. Переходит в раздел "Мои бронирования".

3. Пытается изменить бронирование, которое уже завершено или находится в периоде, когда изменения невозможны (например, бронирование с истекшим сроком).
4. Система выдает сообщение о невозможности изменения и предлагает выбрать другое бронирование.

4.4.2. Модуль управления платежами

3. Нормальный сценарий: Обработка возврата средств

1. Пользователь заходит в раздел управления бронированиями.
2. Выбирает бронирование для отмены.
3. Система подтверждает отмену и инициирует возврат средств.
4. Возврат проходит успешно, и система уведомляет пользователя о зачислении средств.

Аномальный сценарий 3: Ошибка при возврате средств

1. Пользователь заходит в раздел управления бронированиями.
2. Выбирает бронирование для отмены.
3. Система подтверждает отмену, но возникает ошибка при возврате средств (например, из-за проблем с платежной системой).
4. Система уведомляет пользователя о том, что возврат не может быть произведен в данный момент, и предлагает попробовать позже или связаться с поддержкой.

4.4.3. Модуль личный кабинет пользователя

4. Нормальный сценарий: Редактирование данных

1. Пользователь заходит в личный кабинет.
2. Выбирает раздел редактирования данных.
3. Изменяет контактную информацию (например, обновляет номер телефона или адрес электронной почты).
4. Сохраняет изменения, и система подтверждает успешное обновление данных.

Аномальный сценарий 4: Ошибка при сохранении данных

1. Пользователь заходит в личный кабинет.

2. Выбирает раздел редактирования данных.
3. Пытается ввести некорректные данные (например, вводит недействительный адрес электронной почты).
4. Система уведомляет пользователя об ошибке и не сохраняет данные до тех пор, пока не будут введены корректные значения.

4.4.4. Модуль выбора турпакета

5. Нормальный сценарий: Сравнение предложений турпакетов

1. Пользователь заходит в раздел выбора турпакетов.
2. Выбирает несколько предложений для сравнения (например, разные комбинации перелета, отеля и экскурсий).
3. Система отображает сравнение пакетов по различным критериям, таким как стоимость, дата вылета, включенные услуги.
4. Пользователь выбирает наиболее подходящий пакет и переходит к его бронированию.

Аномальный сценарий 5: Ошибка при сравнении турпакетов

1. Пользователь заходит в раздел выбора турпакетов.
2. Пытается выбрать несколько предложений для сравнения, но один из туров больше не доступен (например, был снят с продажи или изменился).
3. Система выводит сообщение о том, что одно или несколько предложений недоступны для сравнения, и предлагает пересмотреть выбор.

6. Нормальный сценарий: Добавление нового турпакета

1. Администратор заходит в раздел управления турпакетами.
2. Выбирает опцию "Добавить новый турпакет".
3. Вводит информацию о новом пакете (например, даты, услуги, стоимость).
4. Сохраняет пакет, и система добавляет его в список доступных для пользователей предложений.
5. Пользователи могут видеть новый пакет в разделе выбора турпакетов.

Аномальный сценарий 6: Ошибка при добавлении нового турпакета

1. Администратор заходит в раздел управления турпакетами.
2. Пытается добавить новый турпакет, но система обнаруживает дублирование данных (например, идентичный пакет уже существует).
3. Система выдает сообщение о том, что пакет с такими параметрами уже добавлен, и предлагает изменить данные или внести корректировки.

4.5. Формулирование основных Use-case

1. Поиск отелей и экскурсий

<i>Вариант использования</i>	Поиск отелей и экскурсий
<i>Область действия</i>	Модуль бронирования
<i>Уровень</i>	Цели пользователя
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и перешел к модулю поиска
<i>Гарантии успеха</i>	Система отображает доступные отели и экскурсии на основе введенных
<i>Триггер</i>	Пользователь инициирует поиск по заданным параметрам
<i>Описание</i>	<ol style="list-style-type: none">1. Пользователь вводит параметры поиска (даты, тип жилья, услуги и пр.).2. Система отправляет запрос к базе данных отелей и экскурсий.3. Система проверяет доступность по введенным параметрам.4. Пользователь получает список доступных вариантов с возможностью фильтрации и сортировки.

<i>Расширение</i>	3.1. Если нет доступных вариантов по заданным параметрам, система предлагает альтернативные даты или услуги.
-------------------	--

2. Бронирование отелей и экскурсий

<i>Вариант использования</i>	Бронирование отелей и экскурсий
<i>Область действия</i>	Модуль бронирования
<i>Уровень</i>	Цели пользователя
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отель, экскурсионные компании
<i>Предусловие</i>	Пользователь авторизован и выбрал отель или экскурсию для бронирования
<i>Гарантии успеха</i>	Пользователь успешно забронировал отель или экскурсию и оплатил услуги
<i>Триггер</i>	Пользователь нажимает кнопку "Забронировать"
<i>Описание</i>	<ol style="list-style-type: none"> 1. Пользователь выбирает отель или экскурсию для бронирования. 2. Система проверяет доступность услуг на выбранные даты. 3. Пользователь вводит данные для оплаты. 4. Система проводит транзакцию и подтверждает бронирование. 5. Пользователь получает подтверждение о бронировании и уведомление на почту.
<i>Расширение</i>	4.1. Если платеж не проходит, система предлагает пользователю повторить попытку или выбрать другой способ оплаты.

3. Просмотр и управление бронированиями

<i>Вариант использования</i>	Просмотр и управление бронированиями
<i>Область действия</i>	Модуль бронирования
<i>Уровень</i>	Операции пользователя
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и имеет активные бронирования
<i>Гарантии успеха</i>	Пользователь может просматривать, изменять или отменять свои бронирования
<i>Триггер</i>	Пользователь переходит в раздел "Мои бронирования"
<i>Описание</i>	<ol style="list-style-type: none">1. Пользователь заходит в личный кабинет и открывает раздел "Мои бронирования".2. Система отображает список активных и прошлых бронирований.3. Пользователь выбирает бронирование для изменения или отмены.4. Система предоставляет возможность внести изменения (даты, количество людей) или отменить бронирование.
<i>Расширение</i>	<ol style="list-style-type: none">4.1. Если бронирование нельзя изменить, система уведомляет пользователя и предлагает альтернативные действия.

4. Отмена бронирования

<i>Вариант использования</i>	Отмена бронирования
<i>Область действия</i>	Модуль бронирования
<i>Уровень</i>	Операции пользователя
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и имеет активные бронирования
<i>Гарантии успеха</i>	Бронирование отменено, средства возвращены (если применимо)
<i>Триггер</i>	Пользователь инициирует отмену бронирования
<i>Описание</i>	<ol style="list-style-type: none">1. Пользователь выбирает бронирование для отмены.2. Система проверяет возможность отмены и условия возврата средств.3. Пользователь подтверждает отмену бронирования.4. Система отменяет бронирование и производит возврат средств (если применимо).5. Пользователь получает уведомление о завершении процесса отмены.
<i>Расширение</i>	2.1. Если бронирование нельзя отменить, система уведомляет пользователя об условиях.

5. Использование мастера выбора турпакетов

<i>Вариант использования</i>	Использование мастера выбора турпакетов
------------------------------	--

<i>Область действия</i>	Модуль выбора турпакета
<i>Уровень</i>	Цели пользователя
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и хочет создать индивидуальный турпакет
<i>Гарантии успеха</i>	Турпакет успешно собран и отправлен на бронирование
<i>Триггер</i>	Пользователь запускает мастера выбора турпакета
<i>Описание</i>	<ol style="list-style-type: none"> 1. Пользователь запускает мастера выбора турпакета. 2. Система последовательно предлагает выбрать перелет, отель и экскурсии. 3. Пользователь выбирает услуги на каждом этапе. 4. Система собирает турпакет на основе введенных данных. 5. Пользователь подтверждает турпакет и переходит к бронированию.
<i>Расширение</i>	4.1. Если выбранные услуги недоступны, система предлагает альтернативные варианты.

6. Сравнение предложений турпакетов

<i>Вариант использования</i>	Сравнение предложений турпакетов
<i>Область действия</i>	Модуль выбора турпакета
<i>Уровень</i>	Цели пользователя
<i>Основное действующее лицо</i>	Клиент

<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и хочет сравнить несколько турпакетов
<i>Гарантии успеха</i>	Пользователь выбирает наиболее подходящий турпакет
<i>Триггер</i>	Пользователь выбирает несколько турпакетов для сравнения
<i>Описание</i>	<ol style="list-style-type: none"> 1. Пользователь выбирает несколько турпакетов для сравнения. 2. Система отображает сравнение по ключевым параметрам (цена, услуги, продолжительность). 3. Пользователь анализирует и выбирает наиболее подходящий тур. 4. Система сохраняет выбранный тур для дальнейшего бронирования.
<i>Расширение</i>	2.1. Если параметры недостаточны для выбора, система предлагает дополнительные фильтры.

7. Распределение ролей

<i>Вариант использования</i>	Распределение ролей
<i>Область действия</i>	Модуль администрирования пользователей
<i>Уровень</i>	Операции системы
<i>Основное действующее лицо</i>	Администратор
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Администратор авторизован и имеет права для управления пользователями
<i>Гарантии успеха</i>	Роль пользователя успешно назначена

<i>Триггер</i>	Администратор выбирает пользователя для назначения роли
<i>Описание</i>	<ol style="list-style-type: none"> 1. Администратор выбирает пользователя для изменения роли. 2. Система отображает доступные роли. 3. Администратор назначает или изменяет роль пользователя. 4. Система сохраняет изменения и обновляет статус пользователя.
<i>Расширение</i>	2.1. Если роль не может быть изменена, система уведомляет администратора.

8. Управление пользователями

<i>Вариант использования</i>	Управление пользователями
<i>Область действия</i>	Модуль администрирования пользователей
<i>Уровень</i>	Операции системы
<i>Основное действующее лицо</i>	Администратор
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Администратор авторизован и имеет доступ к управлению пользователями
<i>Гарантии успеха</i>	Пользователь успешно создан, изменен или удален
<i>Триггер</i>	Администратор инициирует управление пользователями
<i>Описание</i>	<ol style="list-style-type: none"> 1. Администратор открывает модуль управления пользователями. 2. Система отображает список всех пользователей. 3. Администратор выбирает действие: создать, изменить или

	удалить пользователя. 4. Система применяет выбранные изменения и обновляет информацию о пользователе.
<i>Расширение</i>	3.1. Если действие не может быть выполнено, система уведомляет администратора об ошибке.

9. Управление шаблонами туров

<i>Вариант использования</i>	Управление шаблонами туров
<i>Область действия</i>	Модуль управления турпакетами
<i>Уровень</i>	Операции системы
<i>Основное действующее лицо</i>	Администратор
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Администратор авторизован для управления турпакетами
<i>Гарантии успеха</i>	Шаблон турпакета успешно создан, изменен или удален
<i>Триггер</i>	Администратор выбирает управление шаблонами туров
<i>Описание</i>	<ol style="list-style-type: none"> 1. Администратор выбирает шаблон для редактирования или добавления нового. 2. Система запрашивает информацию о шаблоне (название, описание, цена и т.д.). 3. Администратор вводит или редактирует данные и сохраняет изменения. 4. Система обновляет шаблон в базе данных.

<i>Расширение</i>	2.1. Если введены некорректные данные, система уведомляет администратора.
-------------------	---

10. Редактирование предложений туров

<i>Вариант использования</i>	Редактирование предложений туров
<i>Область действия</i>	Модуль управления турпакетами
<i>Уровень</i>	Операции системы
<i>Основное действующее лицо</i>	Администратор
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Администратор авторизован для редактирования предложений туров
<i>Гарантии успеха</i>	Турпакет успешно изменен и обновлен в системе
<i>Триггер</i>	Администратор инициирует редактирование существующего предложения
<i>Описание</i>	<ol style="list-style-type: none"> 1. Администратор выбирает турпакет для редактирования. 2. Система отображает текущие данные о турпакете (название, описание, цена и т.д.). 3. Администратор вносит необходимые изменения в турпакет. 4. Система сохраняет изменения и обновляет данные в базе.
<i>Расширение</i>	3.1. Если данные введены некорректно, система уведомляет администратора и предлагает исправить ошибку.

11. Добавление новых предложений

<i>Вариант использования</i>	Добавление новых предложений
<i>Область действия</i>	Модуль управления турпакетами
<i>Уровень</i>	Операции системы
<i>Основное действующее лицо</i>	Администратор
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Администратор авторизован для добавления новых турпакетов
<i>Гарантии успеха</i>	Новый турпакет успешно добавлен в систему и доступен для бронирования
<i>Триггер</i>	Администратор инициирует добавление нового турпакета
<i>Описание</i>	<ol style="list-style-type: none">1. Администратор открывает модуль добавления новых предложений.2. Система запрашивает данные о новом турпакете (название, описание, цена, услуги и т.д.).3. Администратор вводит данные и подтверждает создание.4. Система сохраняет новый турпакет в базе данных и делает его доступным для бронирования.
<i>Расширение</i>	<ol style="list-style-type: none">2.1. Если данные не соответствуют требованиям (например, отсутствует обязательное поле), система уведомляет администратора об ошибке.

12. Обработка платежей

<i>Вариант использования</i>	Обработка платежей
<i>Область действия</i>	Модуль управления платежами
<i>Уровень</i>	Операции системы
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Платежная система
<i>Предусловие</i>	Пользователь завершил процесс бронирования и инициировал оплату
<i>Гарантии успеха</i>	Платеж успешно обработан и подтвержден системой
<i>Триггер</i>	Клиент инициирует оплату за бронирование
<i>Описание</i>	<ol style="list-style-type: none"> 1. Клиент выбирает способ оплаты и вводит данные для проведения транзакции. 2. Система отправляет запрос на обработку платежа через платежную систему. 3. Платежная система проверяет данные и проводит транзакцию. 4. Система получает подтверждение успешного платежа и обновляет статус бронирования.
<i>Расширение</i>	3.1. Если платеж не проходит, система уведомляет пользователя и предлагает выбрать другой способ оплаты.

13. Возврат средств

<i>Вариант использования</i>	Возврат средств
------------------------------	------------------------

<i>Область действия</i>	Модуль управления платежами
<i>Уровень</i>	Операции системы
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Платежная система
<i>Предусловие</i>	Пользователь отменил бронирование, и требуется возврат средств
<i>Гарантии успеха</i>	Средства успешно возвращены пользователю
<i>Триггер</i>	Клиент инициирует возврат средств после отмены бронирования
<i>Описание</i>	<ol style="list-style-type: none"> 1. Клиент отменяет бронирование, требующее возврата средств. 2. Система отправляет запрос в платежную систему на возврат. 3. Платежная система обрабатывает запрос и проводит возврат средств. 4. Система обновляет статус бронирования и уведомляет клиента о возврате.
<i>Расширение</i>	2.1. Если возврат средств невозможен (например, из-за истечения срока), система уведомляет пользователя о невозможности возврата.

14. Автоматическое редактирование данных

<i>Вариант использования</i>	Автоматическое редактирование данных
<i>Область действия</i>	Модуль личного кабинета пользователя
<i>Уровень</i>	Операции пользователя

<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и хочет изменить свои личные данные
<i>Гарантии успеха</i>	Личные данные успешно обновлены
<i>Триггер</i>	Пользователь инициирует редактирование данных в личном кабинете
<i>Описание</i>	<ol style="list-style-type: none"> 1. Пользователь заходит в личный кабинет и выбирает опцию редактирования данных. 2. Система отображает текущие данные пользователя. 3. Пользователь вносит изменения и подтверждает обновление данных. 4. Система сохраняет изменения и обновляет информацию в профиле пользователя.
<i>Расширение</i>	<ol style="list-style-type: none"> 3.1. Если введены некорректные данные, система уведомляет пользователя об ошибке и предлагает повторить попытку.

15. Добавление данных

<i>Вариант использования</i>	Добавление данных
<i>Область действия</i>	Модуль личного кабинета пользователя
<i>Уровень</i>	Операции пользователя

<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и хочет добавить новые данные в свой профиль
<i>Гарантии успеха</i>	Новые данные успешно добавлены в профиль
<i>Триггер</i>	Пользователь инициирует добавление данных в личном кабинете
<i>Описание</i>	<ol style="list-style-type: none"> 1. Пользователь выбирает опцию добавления данных в личном кабинете. 2. Система запрашивает информацию для добавления (например, новый контактный номер или адрес). 3. Пользователь вводит данные и подтверждает добавление. 4. Система сохраняет новые данные в профиле пользователя.
<i>Расширение</i>	<ol style="list-style-type: none"> 2.1. Если введены некорректные данные, система уведомляет пользователя и предлагает исправить ошибку.

16. Удаление информации

<i>Вариант использования</i>	Удаление информации
<i>Область действия</i>	Модуль личного кабинета пользователя
<i>Уровень</i>	Операции пользователя

<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Отсутствуют
<i>Предусловие</i>	Пользователь авторизован и хочет удалить ненужную информацию из профиля
<i>Гарантии успеха</i>	Информация успешно удалена
<i>Триггер</i>	Пользователь инициирует удаление данных
<i>Описание</i>	<ol style="list-style-type: none"> 1. Пользователь выбирает опцию удаления информации в личном кабинете. 2. Система отображает данные, которые можно удалить. 3. Пользователь выбирает информацию для удаления и подтверждает действие. 4. Система удаляет выбранные данные из профиля пользователя.
<i>Расширение</i>	<p>2.1. Если данные не могут быть удалены (например, из-за политики безопасности), система уведомляет пользователя.</p>

17. Обращение в службу поддержки

<i>Вариант использования</i>	Обращение в службу поддержки
<i>Область действия</i>	Модуль службы поддержки
<i>Уровень</i>	Операции пользователя
<i>Основное действующее лицо</i>	Клиент
<i>Участники</i>	Сотрудник службы поддержки

<i>Предусловие</i>	Пользователь авторизован в системе и столкнулся с технической проблемой или вопросом по бронированию, который требует помощи.
<i>Гарантии успеха</i>	Запрос успешно обработан, и пользователь получает решение своей проблемы или помощь.
<i>Триггер</i>	Пользователь инициирует запрос в службу поддержки через интерфейс системы.
<i>Описание</i>	<ol style="list-style-type: none"> 1. Пользователь заходит в раздел службы поддержки и выбирает опцию "Отправить запрос". 2. Система отображает форму для ввода запроса, где пользователь описывает проблему или задает вопрос. 3. Пользователь заполняет форму запроса и отправляет ее. 4. Система фиксирует запрос и присваивает ему уникальный идентификатор. 5. Запрос направляется сотруднику службы поддержки для обработки. 6. Сотрудник службы поддержки получает запрос, обрабатывает его и предоставляет ответ пользователю. 7. Система уведомляет пользователя об ответе и обновляет статус запроса (например, "в обработке", "решено").
<i>Расширение</i>	<ol style="list-style-type: none"> 4.1. Если сотрудник не может сразу решить проблему, запрос переводится в статус "ожидание", а пользователю предоставляется информация о времени ожидания и дальнейших действиях.

5. Первичное проектирование. Выявление первичных классов

5.1. Первичное проектирование

5.1.1. Выявление первичных классов

Основные классы:

1. *Пользователь (User)* — абстрактный класс, являющийся общим для всех типов пользователей системы. От этого класса наследуются конкретные роли: *Клиент (Client)*, *Менеджер турагентства (Manager)*, *Администратор (Administrator)*.
2. *Турпакет (Tour_Package)* — абстрактный класс, представляющий туристический пакет, включающий предложения по перелетам, проживанию, экскурсиям и трансферам. От этого класса наследуются классы *Отель (Hotel)*, *Авиакомпания (Airline)*, *Экскурсия (Excursion)*.
3. *Бронирование (Booking)* — класс, отвечающий за процесс бронирования турпакетов.
4. *Платеж (Payment)* — абстрактный класс для управления платежами и возвратами. Конкретные реализации зависят от способа оплаты.
5. *Поставщик (Supplier)* — абстрактный класс, описывающий поставщиков услуг (авиакомпаний, отелей, экскурсионные агентства). От этого класса наследуются классы *Отель (Hotel)*, *Авиакомпания (Airline)*, *Экскурсия (Excursion)*.
6. *Мастер выбора турпакетов (Tour_Selection_Wizard)* — класс, который управляет процессом выбора туристического пакета на основе предпочтений клиента.
7. *Управление пользователями (User_Management)* — класс для управления учетными записями и ролями пользователей.
8. *Роль (Role)* — класс для управления ролями пользователей и правами доступа.
9. *Служба поддержки (Support_Service)* — класс для взаимодействия клиентов с техподдержкой.

10. *Личный кабинет пользователя (User_Account)* — класс, который позволяет пользователям управлять своей личной информацией и бронированиями

5.1.2. Разработка иерархии классов с детализацией абстрактных/полиморфных методов

5.1.2.a. Пользователь (User) (абстрактный класс)

Атрибуты:

1. + имя: String
2. + контакт: String

Методы:

1. + *Авторизоваться()*: void — полиморфный метод: каждый тип пользователя может иметь свои особенности авторизации (например, клиент через стандартную форму, администратор через панель управления).
2. + *Зарегистрироваться()*: void — полиморфный метод: разные типы пользователей могут проходить регистрацию по-разному (например, клиенту нужно меньше данных, а администратору — больше).
3. + *Выйти()*: void — стандартный метод: общий для всех типов пользователей.
4. + *Просмотреть_Туры()*: void — стандартный метод: любой пользователь может просматривать туры без изменения логики.
5. + *Получить_Информацию()*: String — полиморфный метод: возвращает основную информацию о пользователе в зависимости от его роли (например, для клиента — список заказов, для менеджера — список управляемых туров).

5.1.2.b. Клиент (Client) (наследник класса User)

Атрибуты:

1. + clientID: int — уникальный идентификатор клиента.
2. + имя: String — имя клиента.
3. + фамилия: String — фамилия клиента.

4. + email: String — адрес электронной почты клиента.
5. + телефон: String — контактный номер телефона клиента.
6. + список_Бронирований: List<Booking> — список всех бронирований клиента.

Методы:

1. + Поиск_Туров(*предпочтения: String*): List<Tour_Package> — стандартный метод: выполняет поиск туров по заданным предпочтениям.
2. + Просмотреть_Бронирования(): List<Booking> — стандартный метод: возвращает список всех бронирований, связанных с клиентом.
3. + Забронировать_Тур(*тур: Tour_Package*): void — стандартный метод: позволяет клиенту бронировать турпакет.

5.1.2.с. Менеджер турагентства (Manager) (наследник класса User)

Атрибуты:

1. + manager_ID: int — уникальный идентификатор менеджера.
2. + имя: String — имя менеджера.
3. + фамилия: String — фамилия менеджера.
4. + email: String — адрес электронной почты менеджера.
5. + телефон: String — контактный номер телефона менеджера.
6. + список_Туров: List<Tour_Package> — список всех созданных менеджером туров.

Методы:

1. + Создать_Тур(*название: String, описание: String, цена: float*): Tour_Package — стандартный метод: позволяет менеджеру создавать новый турпакет.
2. + Редактировать_Тур(*тур: Tour_Package*): void — стандартный метод: менеджер может вносить изменения в существующий турпакет.
3. + Удалить_Тур(*тур: Tour_Package*): void — стандартный метод: удаляет тур из системы.

4. + *Статистика_Продаж()*: List<String> — стандартный метод: менеджер может просматривать статистику по продажам туров.

5.1.2.d. Администратор (Administrator) (наследник класса User)

Атрибуты:

1. + admin_ID: int — уникальный идентификатор администратора.
2. + имя: String — имя администратора.
3. + фамилия: String — фамилия администратора.
4. + email: String — адрес электронной почты администратора.
5. + телефон: String — контактный номер телефона администратора.
6. + роль: String — роль администратора в системе (например, суперпользователь, менеджер безопасности и т.д.).

Методы:

1. + *Управление_Пользователями()*: List<User> — стандартный метод: администратор управляет всеми пользователями системы.
2. + *Управлять_Ролями(роль: Role)*: void — стандартный метод: администратор может управлять ролями пользователей.

5.1.2.e. Турпакет (Tour_Package) (абстрактный класс)

Атрибуты:

1. + название: String
2. + описание: String
3. + цена: float

Методы:

1. + *Рассчитать_Стоимость()*: float — абстрактный метод: каждый подкласс реализует свою логику расчета стоимости.
2. + *Сравнить(другой_Тур: Tour_Package)*: String — стандартный метод: сравнивает текущий турпакет с другим турпакетом.
3. + *Добавить_Услугу(услуга: Service)*: void — стандартный метод: добавляет услугу в турпакет.

5.1.2.f. Поставщик (Supplier) (абстрактный класс)

Атрибуты:

1. + поставщик_ID: int
2. + название: String
3. + контакт: String

Методы:

1. + *Обновить_Информацию()*: void — стандартный метод: обновляет информацию о поставщике.
2. + *Получить_Данные_О_Поставщике()*: String — стандартный метод: возвращает информацию о поставщике.
3. + *Получить_Отзывы()*: List<String> — стандартный метод: получает отзывы о поставщике.

5.1.2.g. Отель (Hotel) (наследник класса Supplier)

Атрибуты:

1. + hotelID: int — уникальный идентификатор отеля.
2. + название: String — название отеля.
3. + адрес: String — местоположение отеля.
4. + класс_Обслуживания: String — категория отеля (3 звезды, 4 звезды и т.д.).
5. + описание: String — описание отеля.
6. + список_Номеров: List<Room> — список номеров, доступных для бронирования.
7. + доступность_Номеров: Map<Date, bool> — информация о доступности номеров на определенные даты.

Методы:

1. + *Забронировать_Номер*(*дата: Date, количество_Ночей: int*): void — полиморфный метод: реализация бронирования может отличаться в зависимости от условий отеля.
2. + *Изменить_Бронирование*(*новая_Дата_Начала: Date, новая_Дата_Конца: Date*): void — стандартный метод: изменяет дату бронирования.
3. + *Отменить_Бронирование*(): void — стандартный метод: отменяет бронирование.
4. + *Проверить_Доступность*(*дата: Date*): bool — стандартный метод: проверяет доступность номеров на определенные даты.

5.1.2.h. Авиакомпания (Airline) (наследник класса Supplier)

Атрибуты:

1. + *airline_ID*: int — уникальный идентификатор авиакомпании.
2. + *название*: String — название авиакомпании.
3. + *код_Авиакомпании*: String — код авиакомпании (например, Аэрофлот — SU).
4. + *класс_Обслуживания*: List<String> — список доступных классов обслуживания (эконом, бизнес и т.д.).
5. + *список_Рейсов*: List<Flight> — список рейсов, доступных для бронирования.
6. + *доступные_Места*: Map<Flight, int> — информация о количестве доступных мест для каждого рейса.

Методы:

1. + *Забронировать_Рейс*(*рейс: String, дата: Date*): void — полиморфный метод: логика бронирования рейсов может отличаться в зависимости от авиакомпании.
2. + *Изменить_Рейс*(*новый_Рейс: String*): void — стандартный метод: изменяет бронирование на другой рейс.

5.1.2.i. Экскурсия (Excursion) (наследник класса Supplier)

Атрибуты:

1. + *excursion_ID*: int — уникальный идентификатор экскурсии.

2. + название: String — название экскурсии.
3. + описание: String — описание экскурсионной программы.
4. + продолжительность: String — длительность экскурсии (например, 3 часа, 1 день).
5. + список_Доступных_Дат: List<Date> — список доступных для бронирования дат.
6. + количество_Мест: int — максимальное количество участников на экскурсии.

Методы:

1. + *Забронировать_Экскурсию(дата: Date): void* — полиморфный метод: бронирование экскурсий может различаться в зависимости от типа экскурсии.
2. + *Изменить_Экскурсию(новая_Дата: Date): void* — стандартный метод: позволяет изменить дату экскурсии.
3. + *Отменить_Экскурсию(): void* — стандартный метод: отменяет бронированную экскурсию.

5.1.2.j. Бронирование (Booking)

Атрибуты:

1. + booking_ID: int — уникальный идентификатор бронирования.
2. + клиент: Client — клиент, который сделал бронирование.
3. + тур: Tour_Package — туристический пакет, который был забронирован.
4. + дата_Бронирования: Date — дата создания бронирования.
5. + статус_Бронирования: String — статус бронирования (подтверждено, отменено, ожидает подтверждения).
6. + стоимость: float — общая стоимость бронирования.
7. + дата_Отмены: Date — дата отмены бронирования (если применимо).

Методы:

1. + *Создать_Бронирование*(тур: *Tour_Package*, клиент: *Client*): *Booking* — стандартный метод: создает новое бронирование для конкретного клиента и турпакета.
2. + *Подтвердить_Бронирование*(): *void* — стандартный метод: подтверждает бронирование для клиента.
3. + *Отменить_Бронирование*(): *void* — стандартный метод: отменяет бронирование.

5.1.2.k. Платеж (Payment) (абстрактный класс)

Атрибуты:

1. + платеж_ID: *int*
2. + сумма: *float*
3. + метод: *String*
4. + статус: *String*

Методы:

1. + *Обработать_Платеж*(сумма: *float*, метод: *String*): *String* — абстрактный метод: реализация зависит от способа оплаты.
2. + *Сделать_Возврат*(сумма: *float*): *void* — стандартный метод: производит возврат средств за отмененный тур.
3. + *Проверить_Статус_Платежа*(платеж_ID: *int*): *String* — стандартный метод: проверяет текущий статус платежа.

5.1.2.l. Мастер выбора турпакетов (Tour_Selection_Wizard)

Атрибуты:

1. + wizard_ID: *int* — уникальный идентификатор мастера.
2. + список_Предпочтений: *Map<Client, String>* — список предпочтений клиентов, на основе которых происходит подбор туров.
3. + текущий_Этап: *String* — текущий этап выбора (например, выбор отеля, выбор рейса).
4. + выбранный_Пакет: *Tour_Package* — текущий выбранный турпакет.
5. + список_Доступных_Опций: *List<Tour_Package>* — список доступных для выбора туров.

Методы:

1. + *Запустить_Мастер()*: void — стандартный метод: запускает процесс пошагового выбора туристического пакета.
2. + *Получить_Предпочтения()*: String — стандартный метод: получает предпочтения клиента для дальнейшей настройки поиска туров.

5.1.2.m. Управление пользователями (User_Management)

Атрибуты:

1. + user_Management_ID: int — уникальный идентификатор системы управления пользователями.
2. + список_Пользователей: List<User> — список всех зарегистрированных пользователей.
3. + список_Ролей: List<Role> — список доступных ролей пользователей (клиент, менеджер, администратор).
4. + политика_Безопасности: String — политика безопасности и доступов для пользователей.
5. + статус_Аккаунта: Map<User, String> — статус аккаунтов пользователей (активен, заблокирован).

Методы:

1. + *Создать_Аккаунт(пользователь: User)*: void — стандартный метод: создает новый аккаунт пользователя.
2. + *Изменить_Аккаунт(пользователь: User, новые_Данные: String)*: void — стандартный метод: изменяет данные аккаунта.
3. + *Удалить_Аккаунт(пользователь: User)*: void — стандартный метод: удаляет существующий аккаунт пользователя.
4. + *Изменить_Роль(пользователь: User, роль: Role)*: void — стандартный метод: изменяет роль пользователя в системе.

5.1.2.n. Роль (Role)

Атрибуты:

1. + роль_ID: int
2. + название: String
3. + права: String

Методы:

1. + *Добавить_Роль*(название: String): void — стандартный метод: добавляет новую роль в систему.
2. + *Удалить_Роль*(роль: Role): void — стандартный метод: удаляет роль
3. + *Присвоить_Права*(роль: Role, права: String): void — стандартный метод: присваивает права доступа для конкретной роли.

5.1.2.о. Служба поддержки (Support_Service)

Атрибуты:

1. + support_ID: int — уникальный идентификатор службы поддержки.
2. + список_Запросов: List<Request> — список всех обращений в службу поддержки.
3. + статус_Запросов: Map<Request, String> — статус запросов (открыт, в обработке, решен).
4. + ответственные_Сотрудники: List<Support_Agent> — список сотрудников службы поддержки, работающих над запросами.
5. + часы_Работы: String — время работы службы поддержки.

Методы:

1. + *Ответить_На_Запрос*(вопрос: String): String — полиморфный метод: разные типы запросов требуют различных ответов.
2. + *Открыть_Запрос*(вопрос: String, клиент: Client): void — стандартный метод: открывает новый запрос для обработки.
3. + *Закрыть_Запрос*(запрос_ID: int): void — стандартный метод: закрывает запрос после его обработки.

5.1.2.р. Личный кабинет пользователя (User_Account)

Атрибуты:

1. + userID: int
2. + имя: String
3. + контактная_Информация: String
4. + список_Бронирований: List<Booking>

Методы:

1. + *редактировать_Данные(новые_Данные: String): void* — стандартный метод для редактирования личной информации пользователя.
2. + *добавить_Информацию(информация: String): void* — стандартный метод для добавления новой информации в профиль.
3. + *удалить_Информацию(информация: String): void* — стандартный метод для удаления устаревшей или ненужной информации из личного кабинета.
4. + *просмотреть_Бронирования(): List<Booking>* — стандартный метод для просмотра всех активных и завершенных бронирований пользователя.
5. + *отменить_Бронирование(booking_ID: int): void* — стандартный метод для отмены существующего бронирования.

5.1.2.q. Интеграция с поставщиками (Supplier_Integration)

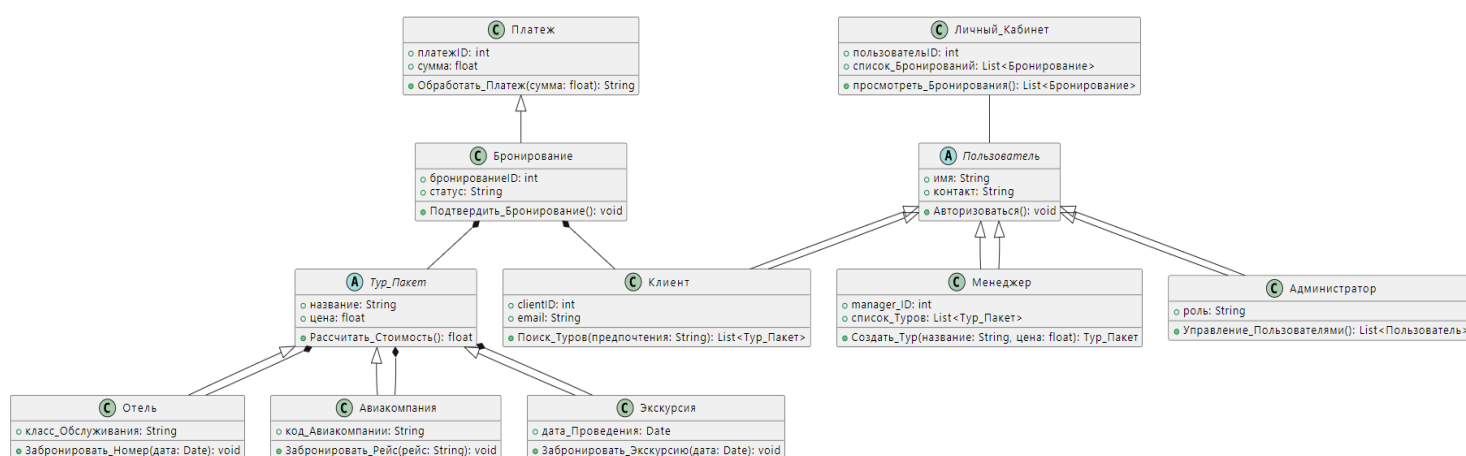
Атрибуты:

1. + *integration_ID: int* — уникальный идентификатор интеграции с поставщиком.
2. + *поставщик: Поставщик* — поставщик, с которым осуществляется интеграция (например, авиакомпания, отель, экскурсионная компания).
3. + *статус_Интеграции: String* — статус интеграции (активна, в процессе, неактивна).
4. + *последнее_Обновление: Date* — дата последнего успешного обновления данных от поставщика.
5. + *URL_API: String* — адрес для подключения к API поставщика.
6. + *метод_Подключения: String* — метод подключения к поставщику (например, REST или SOAP).
7. + *интервал_Обновления: int* — интервал времени в минутах, через который данные обновляются.

Методы:

1. + *установить_Интеграцию*(поставщик: *Поставщик*): void — метод для установления связи с поставщиком и начала интеграции.
2. + *получить_Данные*(): List<Тип_Пакет> — метод для получения данных о доступных предложениях (номерах, рейсах, экскурсиях) от поставщика.
3. + *обновить_Данные*(): void — метод для обновления данных о предложениях от поставщика.
4. + *проверить_Статус*(): String — метод для проверки текущего статуса интеграции (успешно, ошибка, отключено).
5. + *завершить_Интеграцию*(): void — метод для завершения процесса интеграции и отключения от системы поставщика.

5.1.3. Первичная диаграмма классов



5.1.4. Описание отношений между классами

Наследование (Generalization):

1. Пользователь (User) — абстрактный класс, от которого наследуются:

- 1.1. Клиент (Client) — конкретный тип пользователя, который может бронировать туры.
- 1.2. Менеджер (Manager) — конкретный тип пользователя, который управляет турпакетами и бронированиями.
- 1.3. Администратор (Administrator) — пользователь, управляющий системой и ролями.
2. Турпакет (Tour_Package) — абстрактный класс, от которого наследуются:
 - 2.1. Отель (Hotel) — представляет предложения по размещению.
 - 2.2. Авиакомпания (Airline) — представляет предложения по перелетам.
 - 2.3. Экскурсия (Excursion) — представляет экскурсионные программы.
3. Поставщик (Supplier) — абстрактный класс, от которого наследуются:
 - 3.1. Отель (Hotel) — поставщик отельных услуг.
 - 3.2. Авиакомпания (Airline) — поставщик авиаперелетов.
 - 3.3. Экскурсия (Excursion) — поставщик экскурсионных услуг.

Агрегация (Aggregation):

1. Клиент может иметь несколько объектов Бронирование. Клиент может сделать несколько бронирований одновременно, и каждая запись о бронировании существует независимо от клиента.
2. Бронирование агрегирует объект Платеж, так как каждое бронирование связано с одним или несколькими платежами, но платежи могут существовать отдельно от бронирования.
3. Менеджер управляет несколькими объектами Турпакет. Менеджер может создать или редактировать несколько туров, таких как отели, авиакомпании и экскурсии.

Композиция (Composition):

1. Бронирование композитрует объект Платеж, так как платеж не может существовать без бронирования. Если бронирование удаляется, соответствующий платеж также исчезает.

2. Турпакет композитует объекты Поставщик (Отель, Авиакомпания, Экскурсия). Турпакет зависит от данных, предоставленных поставщиками, и без поставщиков турпакет не имеет смысла.

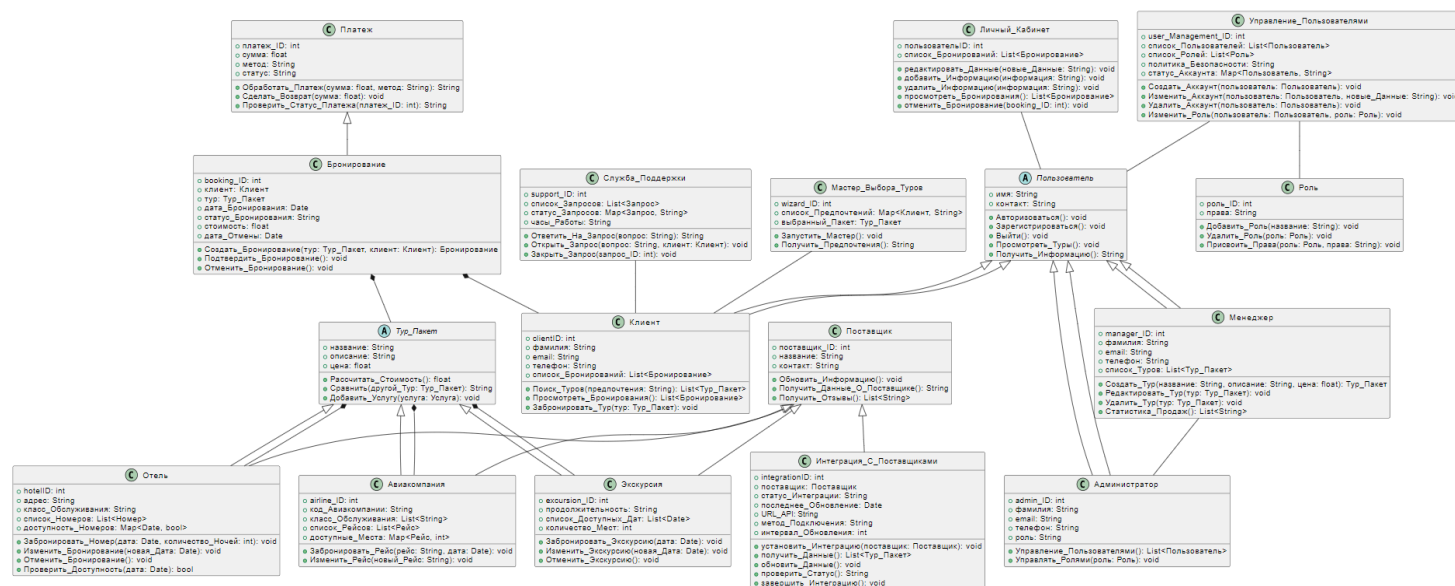
Ассоциация (Association):

1. Клиент взаимодействует с Мастером выбора турпакетов. Мастер помогает клиенту в выборе туров на основе его предпочтений.
2. Менеджер управляет системой Администрирования, выполняя административные задачи и взаимодействуя с Администратором.
3. Служба поддержки связана с Клиентом для решения проблем и обработки запросов. Служба поддержки взаимодействует с клиентами для решения их вопросов и проблем, связанных с бронированием.

6. Детальная диаграмма классов. Выбор архитектуры

6.1. Детальная диаграмма классов

6.1.1. Диаграмма классов



6.1.2. Описание отношений классов

Наследование (Generalization):

1. *Пользователь (User)* — абстрактный класс, от которого наследуются:
 - 1.1. *Клиент (Client)* — это конкретный тип пользователя, который может бронировать туры и взаимодействовать с системой.
 - 1.2. *Менеджер (Manager)* — это пользователь, который управляет созданием и редактированием туров, а также взаимодействует с клиентами.
 - 1.3. *Администратор (Administrator)* — это пользователь, который управляет системой, пользователями и их ролями.

2. *Турпакет (Tour_Package)* — абстрактный класс, от которого наследуются:
 - 2.1. *Отель (Hotel)* — представляет предложение по размещению.
 - 2.2. *Авиакомпания (Airline)* — представляет предложение по авиаперелетам.
 - 2.3. *Экскурсия (Excursion)* — представляет экскурсионные услуги.
3. *Поставщик (Supplier)* — абстрактный класс, от которого наследуются:
 - 3.1. *Отель (Hotel), Авиакомпания (Airline), Экскурсия (Excursion)* — все они являются поставщиками услуг и предоставляют данные для формирования турпакетов.

Агрегация (Aggregation):

1. *Клиент (Client)* агрегирует несколько объектов *Бронирование (Booking)*. Это означает, что один клиент может иметь несколько бронирований одновременно, и каждый объект бронирования связан с клиентом.
2. *Бронирование (Booking)* агрегирует объект *Платеж (Payment)*, так как каждое бронирование связано с соответствующим платежом. Бронирование может существовать без платежа, но не наоборот.
3. *Менеджер (Manager)* агрегирует несколько объектов *Турпакет (Tour_Package)*, включая *Отель, Авиакомпания и Экскурсия*. Это означает, что менеджер может управлять множеством туров.
4. *Управление пользователями (User_Management)* агрегирует несколько объектов *Пользователь (User)*, управляя их учетными записями и ролями.

Композиция (Composition):

1. *Бронирование (Booking)* композитрует объект *Платеж (Payment)*, так как платежи напрямую связаны с бронированием и не могут существовать без него. Если бронирование удаляется, соответствующий платеж также исчезает.
2. *Турпакет (Tour_Package)* композитрует объекты *Поставщик (Supplier)* (например, *Отель, Авиакомпания, Экскурсия*), так как турпакеты формируются на основе данных, предоставленных

поставщиками. Если турпакет удаляется, все связанные с ним предложения от поставщиков также удаляются.

3. *Интеграция с поставщиками (Integration_with_Suppliers)* композирует объекты *Поставщик (Supplier)*, так как интеграция с поставщиками не может существовать без них.

Ассоциация (Association):

1. *Клиент (Client)* взаимодействует с *Мастером выбора турпакетов (Tour_Selection_Wizard)*, который помогает клиенту выбирать туры на основе его предпочтений.
2. *Менеджер (Manager)* связан с *Администратором (Administrator)* через управление административными функциями и взаимодействие с системой.
3. *Служба поддержки (Support_Service)* взаимодействует с *Клиентом (Client)*, обрабатывая их запросы и решая проблемы.

Описание основных связей:

1. Клиент создает и управляет бронированиями, использует мастер выбора туров, взаимодействует со службой поддержки.
2. Менеджер управляет турпакетами и взаимодействует с администратором, выполняя административные функции.
3. Администратор управляет пользователями и их ролями в системе.
4. Турпакет включает предложения от Поставщиков, таких как Отель, Авиакомпания, и Экскурсия, и формирует конечное предложение для клиентов.
5. Интеграция с поставщиками управляет взаимодействием с поставщиками и поддерживает актуальные данные о предложениях.

6.2. Выбор архитектуры

6.2.1. Обоснование выбора Многоуровневой архитектуры

6.2.1.a. Четкое разделение обязанностей и модульность

Многоуровневая архитектура подразумевает разделение системы на несколько уровней (слоев), каждый из которых отвечает за выполнение определенных задач. Для проекта "НаКурорт" это очень важно, так как система содержит множество функциональных блоков, которые требуют четкого разделения:

1. *Уровень представления (Presentation Layer):* Этот уровень взаимодействует с пользователем. В системе "НаКурорт" это будет пользовательский интерфейс для *клиентов, менеджеров и администраторов*. Здесь обрабатываются запросы пользователей, такие как *поиск туров, бронирования и запросы к службе поддержки*. Разделение этого слоя позволяет легко обновлять интерфейс без необходимости затрагивать логику системы.
2. *Уровень бизнес-логики (Business Logic Layer):* Здесь выполняется основная логика приложения: *бронирование туров, работа с поставщиками, обработка платежей, а также взаимодействие с мастер-выбора турпакетов*. Этот уровень связывает взаимодействие между данными и пользователем, и позволяет изолировать логику от пользовательского интерфейса. Благодаря такому подходу можно легко добавлять новые функции, не затрагивая работу остальных уровней.
3. *Уровень данных (Data Layer):* Этот слой отвечает за доступ и управление данными. В проекте "НаКурорт" это будет база данных, содержащая информацию о *пользователях, бронированиях, платежах и турпакетах*. Разделение уровня данных упрощает администрирование базы данных, масштабирование и добавление новых источников данных, таких как внешние API поставщиков услуг.

Преимущества:

1. *Модульность:* Каждый уровень можно разрабатывать и поддерживать независимо. Например, при добавлении нового функционала для работы с турпакетами или интеграции с новыми поставщиками можно изменить только бизнес-логику, не затрагивая интерфейс и базу данных.
2. *Изоляция компонентов:* Обновления или ошибки в одном слое не влияют на другие. Это значительно улучшает стабильность системы и её управляемость.

6.2.1.b. Масштабируемость и гибкость

Многоуровневая архитектура поддерживает независимое масштабирование каждого уровня, что позволяет системе эффективно справляться с увеличением нагрузки. Это особенно важно для проекта "НаКурорт", так как ожидается рост количества пользователей и интеграций с новыми поставщиками.

Пример масштабируемости:

1. *Масштабирование бизнес-логики:* Например, если система сталкивается с увеличением количества бронирований или взаимодействий с поставщиками, можно масштабировать только бизнес-логику, добавляя серверы, которые обрабатывают запросы. Это позволяет снизить нагрузку на другие компоненты системы.
2. *Масштабирование базы данных:* В случае роста объема данных (увеличение записей о бронированиях, платежах, пользователях) можно масштабировать уровень данных, добавляя новые серверы базы данных или переходя на распределенные хранилища данных.
3. *Масштабирование интерфейса:* Если нагрузка возрастает из-за большого числа пользователей, можно масштабировать только слой представления, увеличивая количество веб-серверов или применяя технологии балансировки нагрузки.

6.2.1.с. Гибкость и легкость интеграции

Многоуровневая архитектура очень удобна для интеграции с внешними системами, что является одним из основных требований для проекта "НаКурорт". Она позволяет разделить внутреннюю логику системы от внешних поставщиков (отели, авиакомпании, экскурсии). Это делает интеграцию более управляемой и безопасной.

Пример интеграции с поставщиками:

1. *Интеграция с отелями и авиакомпаниями:* В проекте "НаКурорт" важно получать актуальные данные от поставщиков услуг в реальном времени (цены, доступность мест). Благодаря разделению на слои, интеграция с API поставщиков может быть организована на уровне бизнес-логики, при этом данные могут быть переданы на уровень представления для отображения пользователю без изменения логики работы базы данных.

6.2.1.d. Упрощенная поддержка и обслуживание

Разделение системы на слои упрощает её поддержку и развитие:

1. *Упрощенное тестирование:* Каждый уровень можно тестировать отдельно. Например, можно протестировать только уровень бизнес-логики (функции бронирования и работы с турпакетами) без необходимости трогать интерфейс или уровень данных.
2. *Простота обновления:* Если нужно внести изменения в интерфейс (например, улучшить дизайн для удобства пользователя), это не потребует изменений в логике работы приложения или базы данных. То же самое верно для любого другого уровня системы.

6.2.1.e. Поддержка высоких стандартов безопасности

Так как многоуровневая архитектура изолирует компоненты системы, она позволяет внедрить лучшие практики безопасности на каждом уровне. Например, можно внедрить шифрование данных на уровне хранения (данные пользователей, платежные данные), внедрить систему аутентификации и авторизации на уровне бизнес-логики и предусмотреть защиту пользовательских данных на уровне представления.

6.2.1.f. Эффективное управление изменениями

Многоуровневая архитектура позволяет более гибко реагировать на изменения в проекте. Если в будущем потребуются изменить поставщика услуг или добавить новую интеграцию, это можно сделать, изменив только бизнес-логику, не затрагивая пользовательский интерфейс и базу данных.

6.2.1.g. Заключение

Многоуровневая архитектура идеально подходит для проекта "НаКурорт", так как она:

1. Обеспечивает четкое разделение ответственности между компонентами системы.
2. Позволяет легко масштабировать, обновлять и поддерживать каждый уровень независимо.
3. Обеспечивает гибкость при интеграции с внешними системами, такими как поставщики услуг.
4. Упрощает тестирование, поддержку и обслуживание системы.
5. Поддерживает высокие стандарты безопасности и управления изменениями.

6.2.2. Сравнение с другими шаблонами архитектур

6.2.2.a. SOA (Service-Oriented Architecture)

SOA (архитектура, ориентированная на сервисы) предполагает разбиение системы на отдельные автономные сервисы, которые взаимодействуют через стандартизированные интерфейсы. Каждый сервис решает отдельную задачу и может быть повторно использован.

Основные недостатки архитектуры для проекта "НаКурорт":

1. *Сложность внедрения:* SOA требует сложной инфраструктуры для управления сервисами, их взаимодействием и мониторингом. Это может значительно усложнить проект "НаКурорт" на этапе

внедрения. Например, каждый отдельный сервис (интеграция с отелями, бронирование, платежи) нужно будет внедрять и поддерживать отдельно, что потребует дополнительных ресурсов.

2. *Высокие затраты на поддержку:* Для реализации SOA потребуется внедрение специальных инструментов для управления сервисами, таких как сервисные шины данных (ESB) или брокеры сообщений. Это приводит к увеличению затрат на инфраструктуру и поддержку. Для системы "НаКурорт", которая на начальном этапе не требует столь сложной архитектуры, такие затраты будут неоправданны.
3. *Излишняя модульность:* На текущем этапе развития проекта SOA может оказаться излишней. Многоуровневая архитектура может обеспечить необходимое разделение обязанностей и гибкость без необходимости разрабатывать независимые сервисы для каждой функции.

6.2.2.b. Микросервисная архитектура

Микросервисы предполагают разбиение приложения на множество мелких независимых сервисов, каждый из которых выполняет одну задачу. Эти сервисы работают независимо друг от друга и могут масштабироваться отдельно.

Основные недостатки архитектуры для проекта "НаКурорт":

1. *Сложность разработки и управления:* Микросервисная архитектура требует большого количества независимых сервисов, каждый из которых нуждается в своем API, мониторинге, управлении версиями и взаимодействии с другими сервисами. Это значительно увеличивает сложность системы и требует больших ресурсов на разработку и поддержку. Для "НаКурорт" на начальном этапе это будет слишком трудозатратно.
2. *Оверхед инфраструктуры:* Поддержка микросервисов требует серьезной инфраструктуры — таких инструментов, как контейнеризация (Docker), оркестрация (Kubernetes), мониторинг (Prometheus) и логирование (ELK). Эти инструменты добавляют дополнительные сложности для разработчиков и системных

администраторов, что не оправдано для системы, которая пока не нуждается в таких высоких уровнях независимости компонентов.

3. *Избыточная гранулярность*: Разделение на множество мелких сервисов может быть излишним для "НаКурорт". В текущем виде система не требует столь глубокого разделения логики на микросервисы, так как многие функции взаимосвязаны и не требуют независимого развертывания.

Основные недостатки архитектуры для проекта "НаКурорт":

1. *Ограниченная масштабируемость*: Хотя клиент-серверная архитектура относительно проста в реализации, она плохо справляется с масштабируемостью. Все запросы обрабатываются на сервере, и при увеличении числа пользователей и запросов сервер становится узким местом, что требует масштабирования всего сервера целиком. В "НаКурорт" ожидается рост числа пользователей и поставщиков, поэтому возможность масштабирования отдельных компонентов важна.
2. *Зависимость от серверных ресурсов*: Вся бизнес-логика сосредоточена на сервере, и при увеличении нагрузки (например, большое количество бронирований в пик сезона) сервер может не справляться, что ухудшит производительность. Для системы, работающей с большим количеством пользователей и данных от поставщиков, это может стать проблемой.
3. *Сложность интеграции*: Клиент-серверная архитектура не обеспечивает гибкости в интеграции с внешними сервисами. В проекте "НаКурорт" важны интеграции с системами поставщиков (отели, авиакомпании), а клиент-серверная архитектура не дает нужной гибкости для реализации таких взаимодействий.

6.2.2.с. Клиент-серверная архитектура

Клиент-серверная архитектура подразумевает наличие клиентов (например, веб- или мобильных приложений), которые взаимодействуют с сервером, обрабатывающим запросы и возвращающим результаты.

Основные недостатки архитектуры для проекта “НаКурорт”:

1. *Ограниченная масштабируемость:* Хотя клиент-серверная архитектура относительно проста в реализации, она плохо справляется с масштабируемостью. Все запросы обрабатываются на сервере, и при увеличении числа пользователей и запросов сервер становится узким местом, что требует масштабирования всего сервера целиком. В "НаКурорт" ожидается рост числа пользователей и поставщиков, поэтому возможность масштабирования отдельных компонентов важна.
2. *Зависимость от серверных ресурсов:* Вся бизнес-логика сосредоточена на сервере, и при увеличении нагрузки (например, большое количество бронирований в пик сезона) сервер может не справляться, что ухудшит производительность. Для системы, работающей с большим количеством пользователей и данных от поставщиков, это может стать проблемой.
3. *Сложность интеграции:* Клиент-серверная архитектура не обеспечивает гибкости в интеграции с внешними сервисами. В проекте "НаКурорт" важны интеграции с системами поставщиков (отели, авиакомпании), а клиент-серверная архитектура не дает нужной гибкости для реализации таких взаимодействий.

6.2.2.d. Монолитная архитектура

Монолитная архитектура предполагает, что все компоненты системы разрабатываются и развертываются как единое целое. Вся логика и данные находятся в одном приложении.

Основные недостатки архитектуры для проекта “НаКурорт”:

1. *Сложность в изменениях и поддержке:* В монолитной архитектуре любое изменение в системе требует пересмотра всего приложения. Это приводит к тому, что даже небольшие изменения могут нарушить работу других компонентов. Для проекта "НаКурорт", где планируется развитие функционала и добавление новых модулей (например, интеграции с новыми поставщиками), монолитная архитектура создаст проблемы с поддержкой.

2. *Ограниченная масштабируемость:* Монолитные приложения сложно масштабировать, так как необходимо увеличивать ресурсы всего приложения, даже если нагрузка возрастает только на один модуль (например, на бронирование). Это неэффективно для системы, которая может развиваться поэтапно.
3. *Сложность интеграции:* Для проекта "НаКурорт", где важна интеграция с внешними системами, монолитная архитектура окажется слишком негибкой, так как изменения в одном модуле могут повлиять на всю систему, и интеграции будет сложно поддерживать.

6.2.2.е. Событийно-ориентированная архитектура

В событийно-ориентированной архитектуре система реагирует на события, которые инициируют выполнение определенных операций. Компоненты системы обмениваются событиями и выполняют нужные действия по мере их возникновения.

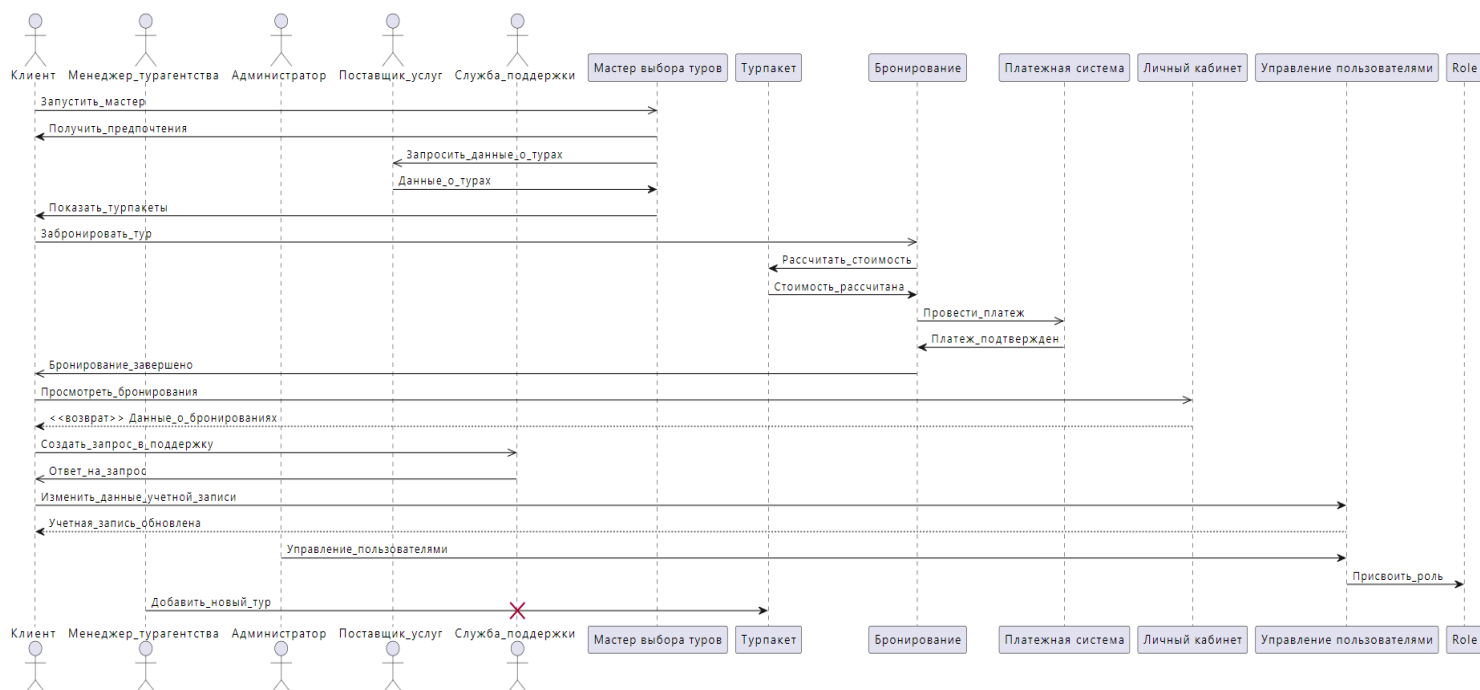
Основные недостатки архитектуры для проекта "НаКурорт":

Сложность реализации: Событийно-ориентированная архитектура требует сложной настройки обработки событий и очередей сообщений. Для проекта "НаКурорт", где основной фокус на бронированиях и взаимодействии с поставщиками, это добавит ненужную сложность.

Необходимость в асинхронности: Хотя асинхронность полезна для обработки большого количества событий, система "НаКурорт" не нуждается в постоянных событиях, которые должны запускать другие процессы. Многоуровневая архитектура с синхронной обработкой запросов будет более эффективной.

7. Динамика системы

7.1. Диаграмма последовательности (Sequence diagram)



7.2. Описание диаграммы последовательности

7.2.1. Акторы (внешние участники)

Актеры представляют собой сущности, которые находятся вне системы, но взаимодействуют с ней. В проекте "НаКурорт" следующие акторы играют ключевые роли:

1. *Клиент (Client)* — основной пользователь системы, который ищет туры, бронирует их и совершает оплату.
2. *Менеджер турагентства (Manager)* — администратор системы, который управляет турпакетами, создаёт и редактирует их.
3. *Администратор (Admin)* — лицо, управляющее системой, ролями и пользователями, обеспечивающее корректную работу системы.

4. *Поставщик услуг (Supplier)* — внешний поставщик, который предоставляет данные о доступности туров, номеров в отелях, рейсов авиакомпаний и экскурсий.
5. *Служба поддержки (Support)* — команда, которая взаимодействует с клиентами для решения возникающих вопросов и проблем.

7.2.2. Классы системы

1. *Мастер выбора туров (MW)* — класс, который управляет процессом выбора туров на основе предпочтений клиента. Он обрабатывает запросы клиента, запрашивает информацию от поставщиков и отображает доступные варианты туров.
2. *Турпакет (TP)* — абстрактный класс, который представляет собой туристический пакет, включающий такие компоненты, как отели, авиабилеты, экскурсии и трансферы. Этот класс отвечает за расчёт стоимости пакетов и управление данными о турпакетах.
3. *Бронирование (B)* — класс, который обрабатывает запросы на бронирование, включая создание бронирований, расчёт стоимости тура и управление статусами бронирования.
4. *Платежная система (PS)* — класс, который занимается проведением платежей и проверкой их статусов. После успешной оплаты подтверждает завершение бронирования.
5. *Личный кабинет (LK)* — класс, который позволяет клиенту просматривать свои бронирования, управлять ими и отслеживать статус заказов.
6. *Управление пользователями (UM)* — класс, отвечающий за управление учётными записями пользователей. Он занимается созданием, редактированием и удалением учётных записей, а также присвоением ролей.
7. *Роль (Role)* — класс, который представляет собой различные роли пользователей в системе, такие как "Клиент", "Менеджер", "Администратор". Он используется для разграничения прав доступа в системе.

7.2.3. Типы связей (взаимодействий)

7.2.3.a. Синхронные вызовы

1. Client ->> MW: *Запустить_Мастер*: Клиент инициирует мастер выбора туров. Это синхронный вызов, так как клиент должен дождаться завершения операции.
2. MW -> Supplier: *Запросить_данные_о_турах*: Мастер выбора туров запрашивает данные о доступных турах у поставщика, и система ожидает завершения операции, прежде чем продолжить.
3. В -> TP: *Рассчитать_стоимость*: Бронирование отправляет запрос на расчёт стоимости турпакета. Вызов синхронный, поскольку расчёт должен завершиться до продолжения процесса.
4. Client -> LK: *Просмотреть_бронирования*: Клиент запрашивает свои бронирования, и система должна предоставить ответ перед продолжением.

7.2.3.b. Асинхронные вызовы

1. Client ->> Support: *Создать_запрос_в_поддержку*: Клиент отправляет запрос в службу поддержки. Это асинхронный вызов, так как система не блокируется в ожидании немедленного ответа.
2. Manager -> TP: *Добавить_новый_тур*: Менеджер турагентства отправляет запрос на добавление нового тура. Это может быть асинхронный вызов, так как система не требует немедленного ответа и продолжает работать.

7.2.3.c. Асинхронные вызовы

1. Supplier -> MW: *Данные_о_турах*: После запроса на получение данных о турах поставщик возвращает данные мастеру выбора туров.
2. TP -> В: *Стоимость_рассчитана*: После расчета стоимости турпакета турпакет возвращает рассчитанную сумму бронированию.
3. LK --> Client: <<возврат>> *Данные_о_бронированиях*: Личный кабинет возвращает клиенту данные о его бронированиях после запроса.

7.2.3.d. Завершение объекта

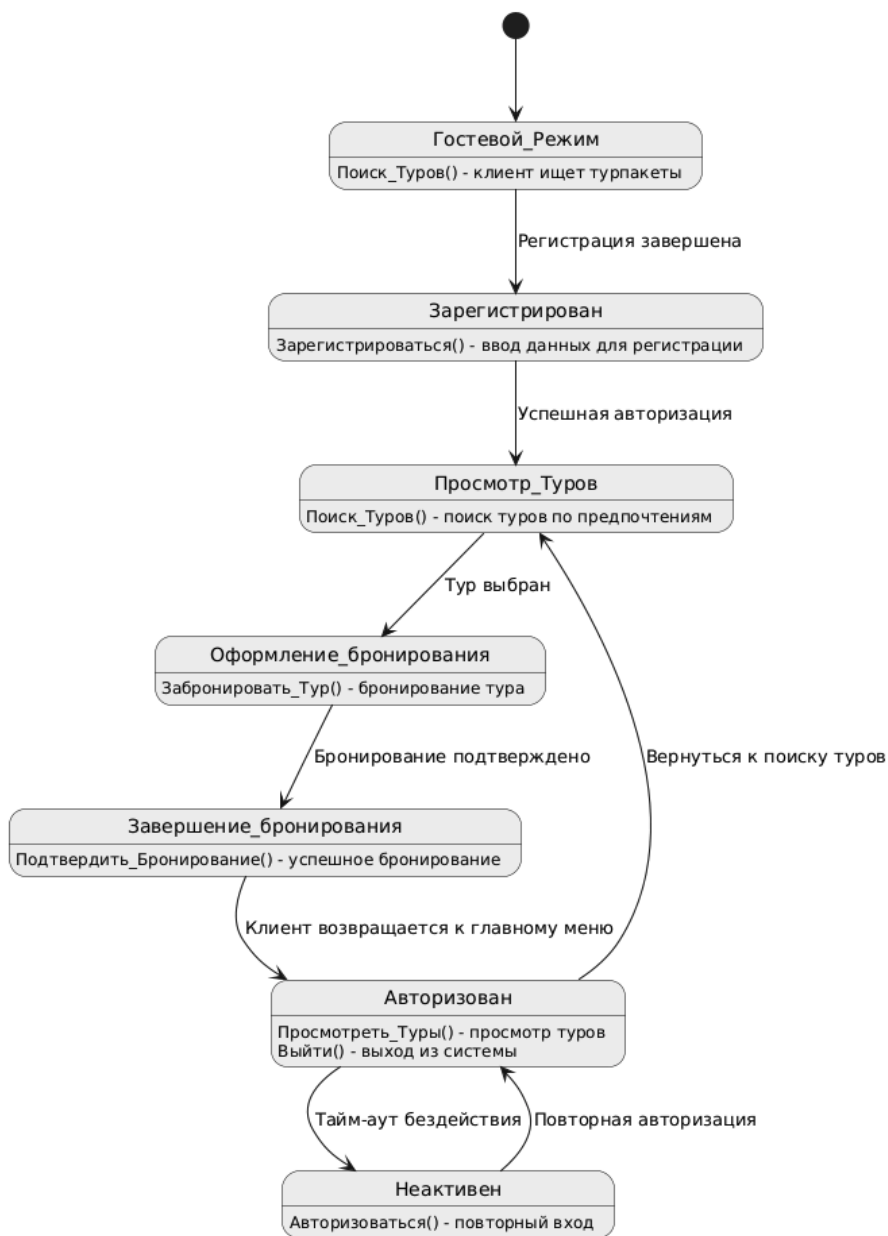
Support -> X Client: *Запрос завершен*: После того как запрос клиента в службу поддержки был обработан, объект службы поддержки завершает свой жизненный цикл и больше не используется в системе.

7.2.3.e. Завершение объекта

Client -> ЛК: *Просмотреть_бронирования и ЛК* --> Client: <<возврат>> Данные о бронированиях: Клиент отправляет запрос на просмотр своих бронирований, а после завершения операции личный кабинет возвращает данные. Это комбинация синхронного вызова и возврата данных.

8. Диаграмма состояния

8.1 Класс Клиент (Client)



a. Гостевой_Режим — клиент находится на сайте без авторизации и может просматривать турпакеты.

- Переход в Зарегистрирован: клиент завершает регистрацию для получения полного доступа к системе.

b. Зарегистрирован — клиент успешно завершил регистрацию.

- Переход в Просмотр_Туров: клиент авторизуется и может начать поиск туров.

c. Просмотр_Туров — клиент ищет доступные туры по предпочтениям.

- Переход в Оформление_бронирования: клиент выбрал тур и приступает к его бронированию.

d. Оформление_бронирования — клиент оформляет бронирование выбранного тура.

- Переход в Завершение_бронирования: бронирование подтверждено и успешно завершено.

e. Завершение_бронирования — клиент завершил бронирование тура.

- Переход в Авторизован: клиент возвращается к главному меню и может продолжать использование системы.

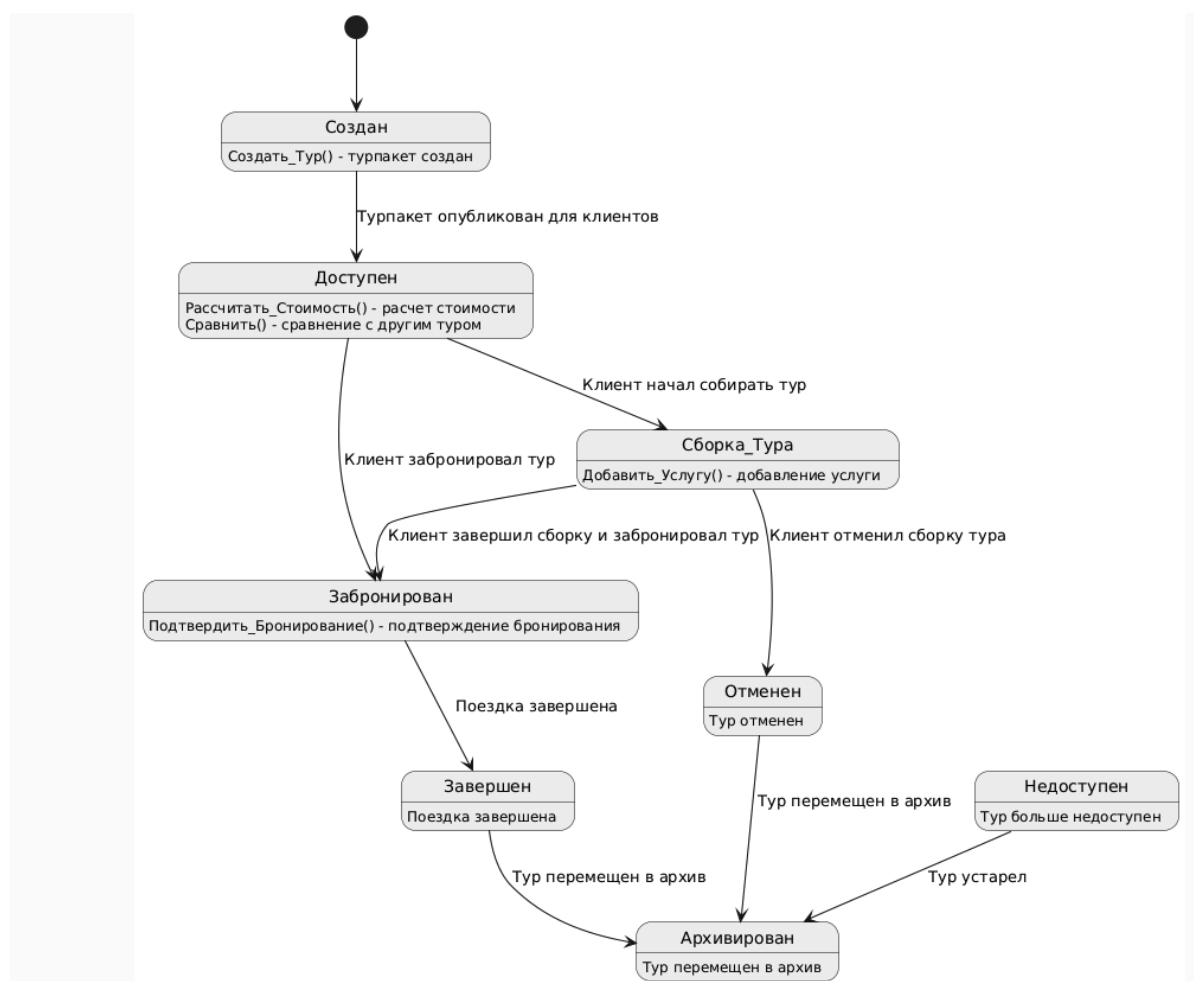
f. Авторизован — клиент авторизован и имеет доступ ко всем функциям системы, включая просмотр туров.

- Переход в Просмотр_Туров: клиент продолжает поиск туров.
- Переход в Неактивен: клиент неактивен в течение длительного времени.

g. Неактивен — клиент был неактивен в течение определенного времени.

- Переход в Авторизован: клиент повторно авторизуется и возвращается к функционалу системы.

8.2. Класс Турпакет (Tour_Package)



а. Создан — турпакет создан системой, но пока не опубликован для клиентов.

- Переход в Доступен: турпакет становится доступным для клиентов после публикации.

б. Доступен — турпакет доступен для бронирования клиентами.

- Переход в Забронирован: клиент забронировал готовый турпакет.
- Переход в Сборка_Тура: клиент начал собирать индивидуальный тур.
- Переход в Недоступен: все места в туре заняты или тур стал недоступен.

в. Сборка_Тура — клиент самостоятельно собирает турпакет.

- Переход в **Забронирован**: клиент завершил сборку и успешно забронировал тур.
- Переход в **Отменен**: клиент отменил процесс сборки тура.

d. Забронирован — турпакет успешно забронирован клиентом.

- Переход в **Завершен**: поездка по турпакету завершена.

e. Завершен — турпакет завершен, клиент воспользовался всеми услугами.

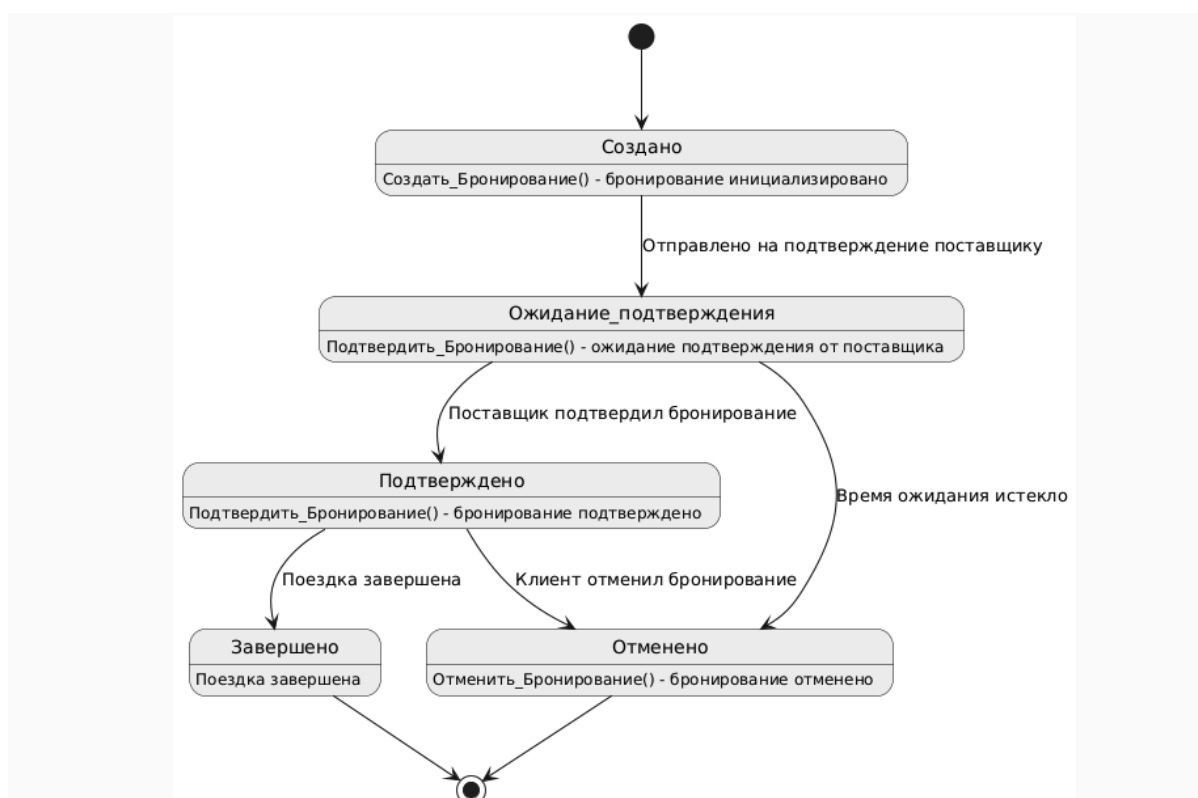
- Переход в **Архивирован**: тур перемещен в архив после завершения поездки.

f. Недоступен — турпакет больше недоступен для бронирования (например, все места заняты).

- Переход в **Архивирован**: тур перемещен в архив, так как больше неактуален.

g. Архивирован — финальное состояние для турпакета, когда он больше не актуален для клиентов (либо поездка завершена, либо тур снят с продаж).

8.3. Класс Booking



а. Создано — бронирование инициализировано и ожидает подтверждения от поставщика услуг.

- Переход в Ожидание_подтверждения: бронирование отправлено поставщику для проверки и подтверждения.

б. Ожидание_подтверждения — бронирование ожидает подтверждения от поставщика.

- Переход в Подтверждено: поставщик подтвердил бронирование, и оно стало активным.
- Переход в Отменено: время ожидания истекло, и бронирование автоматически отменяется.

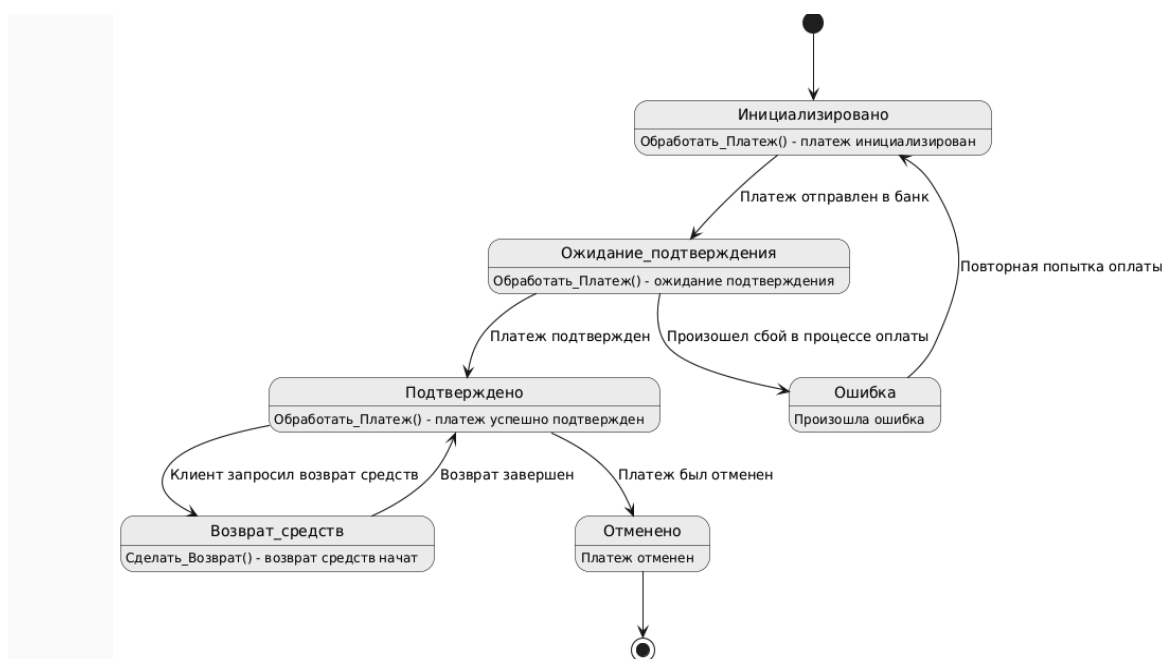
в. Подтверждено — бронирование подтверждено поставщиком.

- Переход в Завершено: поездка завершена, и бронирование закрыто.
- Переход в Отменено: клиент отменил бронирование до поездки.

d. Отменено — бронирование было отменено либо автоматически из-за истечения времени ожидания, либо по запросу клиента.

e. Завершено — бронирование завершено после того, как поездка состоялась

8.4. Класс платежная система



a. Инициализировано — платеж был создан и готов к обработке.

- Переход в **Ожидание_подтверждения**: платеж отправлен в банк или платежную систему для подтверждения.

b. Ожидание_подтверждения — платеж ожидает подтверждения от банка или процессинговой системы.

- Переход в **Подтверждено**: платеж успешно обработан и подтвержден.
- Переход в **Ошибка**: произошел сбой при обработке платежа, возможно из-за отказа банка или технической ошибки.

c. Подтверждено — платеж был успешно подтвержден.

- Переход в **Возврат_средств**: клиент запросил возврат средств, и процесс возврата начат.
- Переход в **Отменено**: платеж был отменен клиентом после подтверждения.

d. Возврат_средств — начат процесс возврата средств клиенту.

- Переход в **Подтверждено**: возврат средств успешно завершен.

e. Отменено — платеж был отменен до или после подтверждения.

f. Ошибка — произошла ошибка при оплате.

- Переход в **Инициализировано**: клиент может повторить попытку оплаты.

9. Матрица трассировки

Основное требование	Тест-кейс ID	Функциональное требование	Нефункциональное требование	Бизнес-требование
<i>Поиск и бронирование туров</i>	ТС_001 ТС_002	1. Возможность поиска туров по дате, стране и услугам.	1. Время отклика системы не более 2 секунд при поиске туров.	Увеличение числа бронирований через систему.
		2. Возможность применения фильтров к результатам поиска.	2. Устойчивость системы при большом количестве одновременных пользователей.	Повышение качества взаимодействия с клиентом.
		3. Отображение туров с полной информацией (цена, даты, услуги).	3. Обеспечение доступности системы 99.9% времени.	Привлечение новых клиентов за счет удобного поиска туров.
		4. Автоматическое обновление данных о доступных турах в реальном времени.		
<i>Бронирование тура</i>	ТС_003 ТС_004	1. Возможность создания бронирования выбранного тура.	1. Время создания бронирования не более 5 секунд.	Увеличение доходов от бронирований.
		2. Подтверждение и отправка информации на электронную почту клиента.	2. Надежность передачи данных о бронировании без сбоев.	Повышение лояльности клиентов за счет удобного процесса бронирования.
		3. Отображение статуса бронирования (подтверждено, отменено).	3. Гарантия выполнения бронирования даже при сбое соединения.	

		4. Валидация данных клиента перед созданием бронирования.	4. Система должна быть защищена от сбоя в процессе бронирования.	
Интеграция с поставщиками услуг	ТС_005	1. Интеграция с базами данных авиакомпаний и гостиниц.	1. Обеспечение безопасности данных при обмене через API.	Увеличение числа подключенных поставщиков услуг.
		2. Обновление данных о доступных услугах поставщиков в реальном времени.	2. Время ответа от внешних API не должно превышать 3 секунд.	Повышение оперативности предложений для клиентов.
		3. Проверка доступности мест и услуг на указанные даты.	3. Поддержка одновременных запросов от большого числа пользователей.	
		4. Возможность добавления новых поставщиков в систему.	4. Масштабируемость системы без снижения производительности.	
Обработка платежей	ТС_006 ТС_007	1. Поддержка различных методов оплаты (карты, электронные кошельки).	1. Все платежные данные должны быть зашифрованы (TLS).	Увеличение доходов от бронирований за счет удобства оплаты.
		2. Подтверждение успешной транзакции и уведомление клиента.	2. Время обработки платежа не более 5 секунд.	Повышение уровня доверия к системе через надежную обработку платежей.
		3. Обработка	3. Поддержка	

		возвратов при отмене бронирования.	масштабируемости системы для увеличения числа транзакций.	
		4. Валидация данных платежных методов перед обработкой транзакции.	4. Обеспечение безопасности транзакций через сторонние системы.	
Управление бронированиями	ТС_008	1. Возможность просмотра активных и прошлых бронирований.	1. Время отклика интерфейса не более 2 секунд.	Повышение удовлетворенности клиентов.
		2. Возможность изменения данных по бронированию (например, даты).	2. Гарантия надежности данных о бронированиях при сбоях системы.	
		3. Возможность отмены бронирования с возвратом средств.	3. Поддержка большого числа бронирований без снижения производительности системы.	
		4. Автоматическое обновление статуса бронирования после изменений.	4. Данные должны быть восстановлены в случае системного сбоя.	
Безопасность данных	ТС_009	1. Шифрование данных клиентов при хранении и передаче.	1. Использование современных протоколов шифрования (TLS, AES).	Соответствие нормативным требованиям (GDPR, ФЗ-152).
		2. Аутентификация пользователей с	2. Поддержка безопасности на уровне архитектуры	Повышение доверия пользователей к системе.

		использованием двухфакторной защиты.	системы.	
		3. Контроль доступа к данным в зависимости от роли пользователя.	3. Время обработки авторизационных данных не более 1 секунды.	
<i>Масштабируемость системы</i>	ТС_010	1. Поддержка увеличения числа пользователей без изменения архитектуры.	1. Масштабирование серверной архитектуры без снижения производительности .	Подготовка системы к увеличению клиентской базы.
		2. Устойчивость к высокой нагрузке в периоды пикового спроса.	2. Гарантия доступности системы 99.9% времени.	
		3. Обеспечение бесперебойной работы при одновременном доступе к системе.	3. Максимальное время восстановления после сбоев – 5 минут.	
<i>Управление пользователями</i>	ТС_011 ТС_012	1. Возможность создания, изменения и удаления учетных записей.	1. Время обработки каждого запроса не более 2 секунд.	Упрощение администрирования системы.
		2. Возможность назначения ролей пользователям (администратор, клиент).	2. Защита данных пользователей и учетных записей.	
		3. Поддержка восстановления пароля через электронную почту.	3. Масштабируемость системы для управления большим числом	

			учетных записей.	
<i>Автоматическое редактирование данных</i>	ТС_013	1. Возможность редактирования личных данных пользователя в личном кабинете.	1. Время отклика системы при редактировании данных не более 2 секунд.	Повышение удовлетворенности клиентов за счет простоты изменения данных.
		2. Сохранение и обновление данных после подтверждения пользователем.	2. Надежность обновления данных без потерь и ошибок.	Упрощение администрирования данных клиентов.
		3. Валидация данных на корректность (например, формат email, телефон).	3. Обработка ошибок при вводе некорректных данных.	Уменьшение числа ошибок при вводе личной информации.
<i>Использование мастера выбора турпакетов</i>	ТС_014	1. Возможность запуска мастера для выбора турпакетов.	1. Время отклика системы на каждом этапе выбора не более 3 секунд.	Увеличение удобства для пользователей при создании индивидуальных турпакетов.
		2. Возможность выбора перелета, отеля и экскурсий на каждом этапе.	2. Устойчивость системы при большом количестве одновременных пользователей.	Повышение лояльности клиентов за счет гибкости формирования туров.
		3. Сбор турпакета на основе введенных данных.	3. Обеспечение доступности услуг при формировании турпакета.	Увеличение количества успешно собранных и забронированных туров.
		4. Предложение альтернативных вариантов при недоступности услуг.	4. Стабильность работы мастера при изменениях данных (альтернативные услуги).	Увеличение числа бронирований при гибком предложении услуг.

Обращение в службу поддержки	ТС_015	1. Возможность отправки запроса в службу поддержки через интерфейс.	1. Время отклика системы при отправке запроса не более 2 секунд.	Повышение качества клиентской поддержки за счет своевременного ответа.
		2. Автоматическая обработка запросов и присвоение уникального ID.	2. Поддержка большого числа запросов без потери производительности.	Уменьшение времени решения проблем клиентов.
		3. Обновление статуса запроса на всех этапах обработки.	3. Обеспечение прозрачности обработки запроса и уведомления клиента.	Увеличение лояльности клиентов за счет быстрой обработки запросов.
		4. Перевод запроса в статус "ожидание" в случае необходимости.	4. Время ответа на запрос не более 24 часов.	Увеличение удовлетворенности клиентов за счет оперативного реагирования.

9.1. Описание тест-кейсов

1. Поиск доступных туров (ТС_001)

<i>Описание:</i> Проверка корректной работы функции поиска туров по заданным параметрам (например, даты, тип тура, страна).	
<i>Предусловие:</i> Пользователь авторизован в системе и находится на странице поиска туров.	
Шаги	Ожидаемый результат
1. Пользователь на главной странице переходит к форме поиска туров и последовательно выбирает страну, даты поездки и тип тура из предложенных системой списков значений.	Система выполняет поиск туров по выбранным пользователем параметрам и отображает список подходящих предложений, каждая запись включает краткое описание тура.
2. Пользователь применяет фильтры для уточнения поиска, такие как бюджет, категории туров и доступность	Система обновляет результаты поиска на основе примененных фильтров, исключая неподходящие варианты и

дополнительных услуг, и нажимает кнопку для обновления результатов.	показывая туры, удовлетворяющие всем выбранным критериям.
---	---

2. Отображение недоступных туров (ТС_002)

<i>Описание:</i> Проверка поведения системы, если по запросу пользователя нет доступных туров.	
<i>Предусловие:</i> Пользователь авторизован и ввел слишком узкие параметры поиска.	
Шаги	Ожидаемый результат
1. Пользователь на странице поиска вводит параметры, которые не соответствуют доступным турам (например, слишком узкий диапазон дат или недоступные направления).	Система уведомляет пользователя об отсутствии туров и предлагает изменить параметры поиска.

3. Создание бронирования тура (ТС_003)

<i>Описание:</i> Проверка процесса бронирования выбранного тура.	
<i>Предусловие:</i> Пользователь выбрал тур для бронирования и находится на странице бронирования.	
Шаги	Ожидаемый результат
1. Пользователь на странице бронирования нажимает на кнопку "Забронировать," после чего система отображает форму для заполнения данных бронирования.	Форма бронирования появляется с полями для ввода всех необходимых данных.
2. Пользователь вводит информацию, включая количество человек, даты поездки и прочие данные, затем система проверяет корректность введенных данных и подтверждает бронирование.	Система проверяет данные пользователя и уведомляет о корректности введенных данных, после чего создается подтвержденное бронирование.
3. Пользователь подтверждает свое бронирование, и система сохраняет заказ, отправляя пользователю уведомление на электронную почту.	Пользователь получает подтверждение бронирования, а система сохраняет всю информацию о заказе и отправляет электронное письмо с деталями.

4. Отмена бронирования (ТС_004)

<i>Описание:</i> Проверка возможности отмены ранее созданного бронирования.	
<i>Предусловие:</i> Пользователь имеет активное бронирование.	
Шаги	Ожидаемый результат
1. Пользователь заходит в раздел "Мои бронирования" и выбирает одно из активных бронирований для отмены.	Система отображает все активные бронирования пользователя с возможностью их отмены.
2. Пользователь нажимает на кнопку "Отменить" и подтверждает действие в появившемся диалоговом окне.	После подтверждения отмены система снимает бронирование и отправляет пользователю уведомление об успешной отмене на почту.

5. Проверка доступности номеров в отелях (ТС_005)

<i>Описание:</i> Проверка корректности интеграции системы с базой данных отелей.	
<i>Предусловие:</i> Пользователь авторизован и запрашивает информацию о доступных номерах.	
Шаги	Ожидаемый результат
1. Пользователь вводит даты проживания и отправляет запрос на получение списка доступных номеров отелей.	Пользователь видит доступные номера, соответствующие выбранным датам, и может выбрать любой из предложенных вариантов.
2. Система отправляет запрос к базе данных отелей и отображает пользователю список доступных номеров с возможностью их бронирования.	Система корректно интегрируется с базой данных и возвращает актуальную информацию о номерах для заданных дат.

6. Обработка оплаты бронирования (ТС_006)

<i>Описание:</i> Проверка корректной работы системы оплаты.	
<i>Предусловие:</i> Пользователь завершил бронирование и перешел к оплате.	
Шаги	Ожидаемый результат
1. Пользователь завершает процесс	Пользователь получает доступ к форме

бронирования и выбирает способ оплаты (например, банковская карта или электронный кошелек).	оплаты после завершения бронирования.
2. Система открывает форму для ввода данных оплаты, проверяет введенную информацию и обрабатывает транзакцию.	Система проверяет корректность платежных данных и успешно обрабатывает платеж.
3. Пользователь подтверждает оплату, и система отправляет уведомление об успешной транзакции.	Система подтверждает успешную оплату и отправляет электронное письмо с деталями транзакции пользователю.

7. Возврат средств при отмене бронирования (ТС_007)

<i>Описание:</i> Проверка процесса возврата средств при отмене бронирования.	
<i>Предусловие:</i> Пользователь имеет бронирование с подтвержденной оплатой.	
Шаги	Ожидаемый результат
1. Пользователь переходит в личный кабинет и выбирает опцию отмены активного бронирования, после чего система открывает интерфейс для подтверждения возврата средств.	Система успешно инициирует процесс возврата средств в соответствии с внутренней политикой компании, отображая уведомление о начале возврата.
2. Пользователь подтверждает отмену, и система отправляет запрос в платежную систему на обработку возврата средств.	Платежная система подтверждает успешное выполнение возврата средств, и пользователь получает соответствующее уведомление по электронной почте.

8. Просмотр и изменение бронирований (ТС_008)

<i>Описание:</i> Проверка корректной работы функции управления бронированиями (просмотр, изменение, отмена).	
<i>Предусловие:</i> Пользователь авторизован и имеет активные бронирования.	
Шаги	Ожидаемый результат
1. Пользователь на главной странице личного кабинета выбирает раздел "Мои бронирования" для просмотра списка	Система корректно отображает все активные и завершенные бронирования пользователя с возможностью их

активных и прошлых бронирований.	редактирования или отмены.
2. Пользователь редактирует данные активного бронирования (например, изменяет даты или количество гостей), после чего система проверяет введенные данные и подтверждает изменения.	Система обновляет информацию о бронировании после успешного внесения изменений, отправляя подтверждение на электронную почту пользователя.
3. Пользователь отменяет бронирование, нажимает на кнопку "Отменить" и подтверждает действие в появившемся диалоговом окне.	После подтверждения отмены система снимает бронирование и отправляет пользователю уведомление об успешной отмене на почту.

9. Шифрование данных клиента (ТС_009)

<i>Описание:</i> Проверка шифрования данных клиентов при передаче и хранении.	
<i>Предусловие:</i> Пользователь вводит данные при регистрации или авторизации.	
Шаги	Ожидаемый результат
1. Пользователь вводит личные данные, такие как имя, адрес электронной почты и пароль, при регистрации или авторизации на сайте.	Введенные данные надежно зашифрованы и переданы на сервер без утечек.
2. Система шифрует все введенные данные с использованием современных алгоритмов шифрования и сохраняет их в базе данных.	Данные пользователя успешно проверяются, и система завершает процесс авторизации или регистрации.

10. Тестирование системы при увеличении нагрузки (ТС_010)

<i>Описание:</i> Проверка масштабируемости системы при увеличении числа пользователей.	
<i>Предусловие:</i> Система должна выдерживать нагрузку от большого числа пользователей.	
Шаги	Ожидаемый результат
1. Провести симуляцию с несколькими сотнями пользователей, одновременно взаимодействующих с системой для выполнения операций поиска, бронирования и оплаты.	Система обрабатывает все запросы в условиях высокой нагрузки без замедления или сбоев, время отклика остается на допустимом уровне.

2. Проверить, как система реагирует на резкое увеличение нагрузки в часы пик и поддерживает стабильность работы всех модулей.	При массовых транзакциях система корректно сохраняет все бронирования и осуществляет платежи, поддерживая производительность на высоком уровне.
---	---

11. Изменение и удаление учетных записей (ТС_011)

<i>Описание:</i> Проверка работы управления пользователями (изменение, удаление учетных записей).	
<i>Предусловие:</i> Администратор авторизован в системе и имеет права управления учетными записями.	
Шаги	Ожидаемый результат
1. Администратор в разделе управления пользователями создает новую учетную запись, заполняя все необходимые поля (имя, электронная почта, роль) и подтверждает создание.	Система создает учетную запись с указанными данными и отправляет письмо с подтверждением на указанный email.
2. Администратор редактирует данные учетной записи пользователя, внося изменения в контактную информацию (например, телефон или электронная почта) и подтверждает внесенные изменения.	Система обновляет данные пользователя в базе данных и отображает уведомление о успешном изменении.
3. Администратор удаляет учетную запись пользователя, подтверждая действие в системе.	Система удаляет учетную запись пользователя и уведомляет администратора об успешном удалении.

12. Восстановление пароля (ТС_012)

<i>Описание:</i> Проверка корректной работы функции восстановления пароля.	
<i>Предусловие:</i> Пользователь забыл пароль и инициировал процесс восстановления через электронную почту.	
Шаги	Ожидаемый результат
1. Пользователь переходит на страницу авторизации и нажимает на ссылку "Забыли пароль?" для инициации процесса восстановления пароля.	Система корректно отображает форму восстановления пароля и принимает адрес электронной почты, проверяя его наличие в базе данных.

2. Система отображает форму для ввода адреса электронной почты, связанного с учетной записью пользователя.	Пользователь получает письмо с уникальной ссылкой для восстановления пароля, и система обновляет статус запроса на восстановление.
3. Пользователь вводит адрес электронной почты	Система отправляет сообщение с инструкциями по восстановлению пароля на указанный адрес.
4. Пользователь вводит новый пароль и подтверждает изменение.	Система обновляет пароль и уведомляет пользователя о его успешном изменении.

13. Редактирование личных данных в личном кабинете (ТС_013)

<i>Описание:</i> Проверка процесса редактирования личных данных пользователем в личном кабинете.	
<i>Предусловие:</i> Пользователь авторизован и находится на странице личного кабинета, где хочет изменить свои личные данные (например, имя, фамилия, email, телефон).	
Шаги	Ожидаемый результат
1. Пользователь входит в личный кабинет, переходит в раздел "Настройки" и редактирует свои личные данные (например, имя, номер телефона, почтовый адрес).	Система корректно отображает форму для редактирования данных пользователя, проверяя новые данные на корректность и дублирование.
2. Пользователь сохраняет изменения, и система проверяет введенные данные на корректность, после чего обновляет профиль.	Личные данные пользователя обновляются, и система уведомляет пользователя об успешном сохранении изменений через интерфейс и/или по электронной почте.

14. Создание турпакета с использованием мастера выбора (ТС_014)

<i>Описание:</i> Проверка корректной работы мастера выбора турпакетов, который позволяет пользователю создать индивидуальный турпакет, выбирая перелет, отель и экскурсии.	
<i>Предусловие:</i> Пользователь авторизован и запустил мастера выбора турпакета.	
Шаги	Ожидаемый результат
1. Пользователь на главной странице	Система корректно отображает шаги

запускает мастер создания турпакета, который открывает шаг выбора перелета.	выбора турпакета, начиная с перелета, и сохраняет каждый выбранный шаг.
2. Пользователь выбирает подходящий перелет из списка предложенных вариантов и переходит к выбору отеля.	Все выбранные параметры успешно сохраняются, и система переходит к следующему этапу выбора услуг.
3. Пользователь выбирает отель и подтверждает выбор, после чего система предлагает выбрать дополнительные экскурсии.	Система формирует полный турпакет с учетом всех выбранных услуг и предоставляет его для бронирования.
4. Пользователь выбирает одну или несколько экскурсий и завершает создание турпакета, нажав кнопку подтверждения.	Если выбранные услуги недоступны, система предлагает альтернативные варианты, позволяя пользователю завершить создание турпакета.

15. Обращение в службу поддержки (ТС_015)

<i>Описание:</i> Проверка процесса отправки запроса в службу поддержки и его обработки сотрудником.	
<i>Предусловие:</i> Пользователь авторизован в системе и хочет обратиться за помощью по технической проблеме или вопросам бронирования.	
Шаги	Ожидаемый результат
1. Пользователь через личный кабинет переходит в раздел "Поддержка" и заполняет форму для отправки запроса, указав суть проблемы и дополнительные детали.	Система корректно обрабатывает запрос, присваивая ему уникальный идентификатор и сохраняет его в базе данных для дальнейшей обработки.
2. Система присваивает запросу уникальный идентификационный номер и автоматически отправляет уведомление о приеме запроса пользователю.	Пользователь получает подтверждение о приеме запроса на свою электронную почту, включая информацию о статусе и идентификаторе запроса.

9.2. Выводы

Выводы касательно полноты и достаточности требований, на основе разработанной матрицы трассировки:

1. Функциональные требования:

- 1.1. *Полнота:* Функциональные требования охватывают все основные аспекты системы "НаКурорт", включая процессы поиска и бронирования туров, управления бронированиями, обработки платежей, интеграции с поставщиками, управления пользователями, обеспечения безопасности данных, и предоставления поддержки пользователям. Каждый из ключевых функциональных процессов имеет подробные требования и тест-кейсы, что указывает на полноту покрытия.
- 1.2. *Достаточность:* Функциональные требования обеспечивают достаточное описание процессов, которые необходимы для обеспечения нормальной работы системы. Для каждого процесса предусмотрены шаги взаимодействия, описание возможных ошибок и альтернативных действий. В описанных требованиях достаточно сценариев для проверки всех критических аспектов системы, включая поиск, выбор туров, бронирование, платежи и обращение в службу поддержки.
2. **Нефункциональные требования:**
 - 2.1. *Полнота:* Нефункциональные требования касаются производительности системы, безопасности данных, масштабируемости, устойчивости к нагрузкам, надежности передачи данных и времени отклика системы. Эти аспекты подробно описаны для всех ключевых процессов. Например, указано время отклика системы для поиска туров, бронирования и платежей, а также требования к шифрованию данных и поддержке высокого уровня безопасности.
 - 2.2. *Достаточность:* Нефункциональные требования включают достаточное количество параметров, которые обеспечивают устойчивую работу системы в условиях повышенных нагрузок и выполнения требований безопасности. Однако возможно добавить дополнительные требования по управлению отказами, например, время восстановления системы при сбое.
3. **Бизнес-требования:**
 - 3.1. *Полнота:* Бизнес-требования направлены на улучшение взаимодействия с пользователями, увеличение числа бронирований, повышение лояльности клиентов и расширение числа услуг, доступных через систему. Эти

требования соответствуют целям системы "НаКурорт" и поддерживаются функциональными и нефункциональными требованиями.

3.2. *Достаточность:* Бизнес-требования полностью описаны и сосредоточены на достижении ключевых целей проекта, таких как увеличение числа бронирований через удобство поиска и бронирования, поддержка работы с большим числом поставщиков и обеспечение безопасности данных пользователей. Они хорошо соответствуют бизнес-целям проекта и отражены в тест-кейсах, что гарантирует их проверку.

4. Тест-кейсы:

4.1. *Полнота:* Тест-кейсы охватывают все ключевые функциональные и нефункциональные требования системы. Они проверяют основные сценарии работы системы, включая поиск и бронирование туров, управление бронированиями, интеграцию с поставщиками, обработку платежей, управление учетными записями и поддержку пользователей. Тест-кейсы также включают проверку безопасности данных и масштабируемости системы, что является важным аспектом для большого числа пользователей.

4.2. *Достаточность:* Тест-кейсы достаточно детализированы и описывают все основные шаги взаимодействия пользователя с системой. Однако могут быть добавлены дополнительные тест-кейсы для проверки экстремальных ситуаций (например, пиковые нагрузки, сбои во внешних системах), что обеспечило бы еще более высокий уровень уверенности в надежности системы.

10. План реализации, тестирования и сопровождения. Бизнес-план

1. План реализации системы

I. Введение

1. Описание системы бронирования туристических пакетов.
2. Цели и задачи системы.
3. Целевая аудитория.
4. Функционал системы.
5. Состав команды ответственных.
6. Состав команды разработчиков.

II. Разработка системы

1. Разработка и создание базы данных для хранения данных о клиентах и поставщиках.
 - а. Сроки
 - б. Функционал
 - с. Ответственные
 - d. Затраты
2. Разработка и создание серверной инфраструктуры.
 - а. Сроки
 - б. Функционал
 - с. Ответственные
 - d. Затраты
3. Разработка и создание хранилища для данных о турпакетах.
 - а. Сроки
 - б. Функционал
 - с. Ответственные
 - d. Затраты
4. Создание интерфейса для пользователей (веб).
 - а. Сроки
 - б. Функционал
 - с. Ответственные
 - d. Затраты
5. Интеграция с платежной системой.
 - а. Сроки
 - б. Функционал
 - с. Ответственные
 - d. Затраты
6. Объединение всех компонентов системы.
 - а. Сроки

- b. Функционал
- c. Ответственные
- d. Затраты

III. Тестирование и отладка (план тестирования)

IV. Запуск системы

V. Переход к эксплуатации системы

2. План тестирования

1. Введение

1.1 Цель

Цель данного плана тестирования — обеспечить качественную работу системы "НаКурорт", соответствующую требованиям, функциональности и безопасности.

1.2 Область тестирования

Тестирование будет охватывать следующие основные аспекты системы:

- Регистрация и аутентификация пользователей.
- Выбор и бронирование турпакетов.
- Взаимодействие с поставщиками услуг (авиакомпания, отели и т.д.).
- Процесс оплаты и безопасности транзакций.
- Формирование и управление турпакетами.

2. Типы тестирования

2.1 Функциональное тестирование

- Проверка регистрации и авторизации пользователей.
- Тестирование поиска туров и их бронирования.
- Проверка взаимодействия с поставщиками для получения актуальных предложений.

- Тестирование процесса оплаты и интеграции с платежной системой.

2.2 Интеграционное тестирование

- Проверка взаимодействия между различными компонентами системы: модулями бронирования, оплаты и получения данных от поставщиков.
- Тестирование взаимодействия с внешними системами (авиакомпания, отели).

2.3 Производительность

- Тестирование системы при высокой нагрузке пользователей (например, во время сезона отпусков).
- Проверка времени отклика системы при массовом бронировании и оплатах.

2.4 Безопасность

- Проверка механизмов защиты данных пользователей.
- Тестирование безопасности финансовых транзакций.
- Тестирование авторизации и защиты от несанкционированного доступа.

3. Сценарии тестирования

3.1 Сценарий: Регистрация и авторизация пользователя

- Проверка корректности процесса регистрации нового пользователя.
- Проверка работы механизма сброса пароля.

3.2 Сценарий: Выбор и бронирование турпакета

- Тестирование поиска турпакетов с разными критериями (бюджет, даты, тип тура).
- Проверка процесса бронирования и подтверждения оплаты.

3.3 Сценарий: Интеграция с поставщиками услуг

- Проверка получения актуальных предложений от поставщиков (авиакомпаний, отелей).
- Тестирование актуальности цен и наличия услуг в реальном времени.

3.4 Сценарий: Оплата турпакета

- Проверка корректности оплаты через различные методы.
- Тестирование возврата средств и отмены бронирования.

3.5 Сценарий: Отчеты по продажам и бронированиям

- Проверка формирования отчетов по продажам туров для администрации системы.
- Тестирование вывода статистики по популярности туров и эффективности поставщиков.

4. Критерии приемки

- Все ключевые функции системы работают без сбоев и с минимальными задержками.
- Система способна обрабатывать до 500 пользователей одновременно без потери производительности.
- Процесс оплаты безопасен и защищает данные пользователей.
- Отчеты содержат актуальную информацию и соответствуют требованиям заказчика.

5. Риски

- **Низкая производительность:** Система может замедляться при большом количестве пользователей и запросов.
- **Проблемы с интеграцией:** Возможны ошибки при взаимодействии с системами поставщиков услуг.
- **Проблемы с масштабируемостью:** При росте числа пользователей возможны задержки в работе системы.

6. План выполнения тестирования

- **Подготовка тестового окружения:** Создание тестовых данных и конфигурация среды тестирования.
- **Функциональное тестирование:** Тестирование всех ключевых функций системы.
- **Интеграционное тестирование:** Проверка взаимодействия всех компонентов и модулей системы.
- **Тестирование производительности:** Тестирование под нагрузкой для проверки скорости работы системы.
- **Тестирование безопасности:** Проверка уязвимостей и защиты данных пользователей.
- **Пользовательское тестирование:** Оценка удобства системы реальными пользователями.

7. Отчетность

- **Еженедельные отчеты:** Прогресс тестирования, выявленные ошибки и статус их исправления.
- **Финальный отчет:** Результаты тестирования и рекомендации по улучшению системы.

8. Заключение

Следуя этому плану тестирования, система "НаКурорт" будет протестирована на соответствие функциональным требованиям, производительности и безопасности, что обеспечит ее успешную эксплуатацию.

3. План эксплуатации

1. Введение

- a. Определение целей и области применения системы:
Автоматизация процесса подбора, бронирования и оплаты туристических услуг.
- b. Описание структуры системы и ее ключевых компонентов:
Веб-интерфейс для пользователей, база данных клиентов и поставщиков, интеграция с платежными системами и внешними сервисами (отели, авиакомпанияи).

2. Административное управление

- a. Определение ролей и обязанностей сотрудников:
Операторы службы поддержки, администраторы системы, менеджеры по туризму.
- b. Установление процедур управления бронированиями, обновления предложений и синхронизации с поставщиками услуг.

3. Управление турпакетами и предложениями

- a. Создание и обновление базы данных туров, отелей, экскурсий и прочих туристических услуг.
- b. Разработка интерфейса для просмотра и бронирования туров клиентами.
- c. Управление актуальностью и обновлением предложений от поставщиков в режиме реального времени.

4. Управление пользователями

- a. Разработка интерфейса для регистрации и управления аккаунтами пользователей.
- b. Возможность для пользователей выбирать турпакеты, редактировать бронирования и управлять платежами.
- c. Обеспечение прозрачности для клиентов в отслеживании их заказов и получении уведомлений о статусах бронирований.

5. Оплата и финансовый учет

- a. Реализация механизмов безопасной оплаты турпакетов.
- b. Ведение истории платежей и взаимодействие с платежными системами для обработки транзакций.
- c. Поддержка различных методов оплаты (банковские карты, электронные платежи).

6. Безопасность и доступ

- a. Разработка механизмов аутентификации и авторизации для защиты данных пользователей и платежной информации.
- b. Регулярное обновление системы для обеспечения безопасности и защиты от киберугроз.
- c. Разделение уровней доступа для разных ролей: администраторы, менеджеры по туризму, пользователи.

7. Обучение и поддержка пользователей

- a. Подготовка обучающих материалов для операторов и администраторов системы, а также клиентов.
- b. Предоставление круглосуточной технической поддержки для пользователей системы через различные каналы (телефон, чат, email).

8. План обновления

- a. Разработка плана регулярного обновления системы, включая внедрение новых функций (дополнительные услуги, расширение выбора туров) и исправление ошибок.
- b. Установка регулярных проверок на наличие обновлений для поддержания актуальности системы.

9. Резервное копирование и восстановление

- a. Создание механизмов для автоматического резервного копирования данных (бронирования, платежи, пользовательские данные).

b. Разработка плана восстановления данных в случае сбоев или потерь информации.

10. Аудит и оценка

a. Проведение регулярных аудитов системы для обеспечения соответствия стандартам и требованиям безопасности.

b. Оценка пользовательского опыта и сбор обратной связи для постоянного улучшения системы.

11. Заключение

a. Подведение итогов и обзор ключевых аспектов плана эксплуатации системы.

b. Определение дальнейших шагов для улучшения системы и ее функциональности.

4. Бизнес-план

1. Объем инвестиций:

- **Разработка ПО и архитектуры:** \$50,000
- **Инфраструктура (серверы, оборудование, хостинг):** \$20,000
- **Маркетинг и продвижение:** \$30,000
- **Операционные расходы (зарплата команды, аренда и пр.):** \$60,000
- **Непредвиденные расходы:** \$10,000
- **Итого:** \$170,000

2. Суть бизнеса:

Система "НаКурорт" предоставляет пользователям возможность подбора и бронирования туристических пакетов, включающих авиаперелет, проживание, экскурсии. Система позволяет клиентам легко находить подходящие туры на основе их предпочтений и бюджета, а также безопасно оплачивать услуги онлайн. Проект предусматривает интеграцию с системами поставщиков туристических услуг в реальном времени для получения актуальных данных о ценах и наличии.

3. Конкурентные преимущества проекта:

- **Гибкость:** Система позволяет пользователям собирать туры под себя, включая различные опции по выбору гостиниц, перелетов и дополнительных услуг.
- **Простота использования:** Удобный интерфейс для пользователей с возможностью быстрого выбора и бронирования туров.
- **Интеграция с поставщиками:** Система в реальном времени получает данные от поставщиков (авиакомпаний, отели, туроператоры), что обеспечивает актуальные предложения.
- **Масштабируемость:** Возможность расширения функционала и добавления новых поставщиков по мере роста клиентской базы.

4. Период окупаемости и возврат инвестиций:

- **Предполагаемый ежемесячный доход:** \$40,000 (доход от комиссий за бронирования и дополнительных услуг).
- **Ежемесячные расходы (обслуживание, зарплаты):** \$15,000
- **Чистая прибыль в месяц:** \$25,000
- **Период окупаемости:** $\$170,000 / \$25,000 = 6.8$ месяцев.

5. Рентабельность:

- **Чистая прибыль в год:** $\$25,000 \times 12 = \$300,000$
- **ROI (коэффициент окупаемости инвестиций):**
(Чистая прибыль – Объем инвестиций) / Объем инвестиций =
 $(300,000 - 170,000) / 170,000 = 76.5$

11.Ревизия выявленных ошибок в ходе выполнения проекта

1.Техническое задание

- a. Конкретизировано необходимое количество персонала (п.4.1.2)
- b. Во всем ТЗ были уточнены числовые параметры
- c. Требования касающиеся безопасности и надежности переформулированы по ГОСТам (п. 4.1.4)

2. Добавлен п. 3.3 - сравнение с другими ЖЦ и методологиями

3. Переделана диаграмма прецедентов, все прецеденты (Must Have) с MoSCoW находятся в диаграмме.

4. Добавлено описания модулей службы поддержки и личного кабинета пользователя

5. Добавлено детализированное описание UseCase'ов

6. Добавлено описание класса интеграция с поставщиками

7. Добавление методов в диаграмму состояний

8. Более конкретное описание шагов тестирования и ожидаемых результатов

12. Итоговые результаты работы

В ходе проделанной работы были разработаны нормативные документы для реализации системы автоматизации туристических пакетов «НаКурорт». Составлены основные документы, такие как: техническое задание с полным перечнем требований, выбрана модель жизненного цикла и разработаны диаграммы прецедентов, состояний и классов для визуализации процессов. Определена архитектура системы, что позволит обеспечить масштабируемость и гибкость в дальнейшем. Также была составлена матрица трассировки требований, планы реализации, тестирования и сопровождения проекта, что позволит контролировать все этапы разработки. Все требования заказчика были четко и детально описаны, обеспечив прозрачность выполнения проекта и его соответствие поставленным целям.

13. Дальнейшие планы развития

В качестве основных планов для развития нашей системы можно выделить несколько направлений:

1. Расширение функциональности:

- **Уведомления:** Система будет отправлять клиентам уведомления о статусе бронирований, изменениях цен и доступности услуг, что улучшит пользовательский опыт.
- **Аналитика для администраторов:** Планируется внедрение модуля аналитики для отслеживания ключевых показателей эффективности, таких как количество проданных туров, средний чек и количество активных пользователей.
- **Оценки и отзывы:** Клиенты смогут оставлять отзывы и оценки о турах, что повысит доверие к системе и качество предоставляемых услуг.
- **Интеграция с платежными системами:** Будет предусмотрена возможность подключения различных платежных систем для расширения вариантов оплаты.

2. Улучшение взаимодействия с пользователями:

- **Персонализированные рекомендации:** Система может предлагать клиентам персонализированные турпакеты на основе их предыдущих бронирований и предпочтений, что улучшит качество взаимодействия.
- **Динамическое ценообразование:** Внедрение системы динамического ценообразования, которая будет корректировать цены на основе спроса, времени года и других факторов, позволяя предложить клиентам лучшие условия.

- **Кросс-продажи:** Клиентам будут предложены дополнительные услуги, такие как аренда автомобиля или страхование, что увеличит доход системы.

3. Развитие аналитики и отчетности:

- **Прогнозирование спроса:** Внедрение модуля аналитики для прогнозирования спроса на турпакеты в зависимости от сезона, экономических факторов и других данных, что поможет лучше планировать предложения и маркетинговые кампании.
- **Геоаналитика:** Добавление модуля для анализа популярности туров по регионам, что позволит компаниям адаптировать свои предложения под местные предпочтения и нужды.

4. Оптимизация интерфейса:

- **Адаптивный интерфейс для мобильных устройств:** Будет разработан адаптивный веб-интерфейс, который позволит клиентам удобно пользоваться системой на мобильных устройствах без необходимости создания отдельного мобильного приложения.

5. Международное расширение:

- **Мультиязычная поддержка:** Добавление дополнительных языков для выхода на международный рынок и привлечения клиентов из разных стран.
- **Локализация сервисов:** Адаптация предложений в зависимости от стран, включая поддержку локальных методов оплаты и учёт местных правовых требований.