

Государственное образовательное учреждение высшего профессионального образования



**«Московский государственный технический университет
имени Н.Э. Баумана»**

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____

КАФЕДРА _____

РАСЧЕТНО - ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту на тему:

Студент _____ (_____)
(ф.и.о.)

Руководитель курсового проекта _____ (_____)
(ф.и.о.)

Москва, 20 ____

Государственное образовательное учреждение высшего профессионального образования
«Московский государственный технический университет имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой _____

«_____» _____ 20____ г.

З А Д А Н И Е

на выполнение курсового проекта

по дисциплине _____

Студент _____
(фамилия, инициалы, индекс группы)

Руководитель _____
(фамилия, инициалы)

График выполнения проекта: 25% к _____ нед., 50% к _____ нед., 75% к _____ нед., 100% к _____ нед.

1. Тема курсового проекта

2. Техническое задание

3. Оформление курсового проекта

3.1. Расчетно-пояснительная записка на _____ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.) _____

Дата выдачи задания «_____» _____ 20____ г.

Руководитель курсового проекта _____ / _____ /

Задание получил _____ / _____ / «_____» _____ 20____ г.

Примечание:

1. Задание оформляется в двух экземплярах; один выдаётся студенту, второй хранится на кафедре.

Содержание

Введение	4
1 Аналитический раздел	5
1.1 Анализ предметной области	5
1.2 Описание системы	6
1.3 Требования	8
1.3.1 Требования к РСОИ	8
1.3.2 Требования к системе заказа путевок	9
1.3.3 Требования к системе платежей	10
1.3.4 Требования к системе бронирования авиабилетов	11
1.3.5 Требования к системе бронирования гостиничных номеров	12
1.4 Сценарии использования системы	13
1.4.1 Спецификации прецедентов использования	14
2 Конструкторский раздел	21
2.1 Архитектура системы	21
2.2 Протокол взаимодействия узлов	22
2.2.1 Последовательность обмена сообщениями	22
2.2.2 Передаваемые сообщения	24
2.3 Система бронирования	27
2.3.1 Прецеденты использования	27
2.3.2 Архитектура системы	27
2.3.3 Модель данных	30
2.4 Система заказа путевок	37
2.4.1 Прецеденты использования	37
2.4.2 Архитектура системы	37
2.4.3 Модель данных	40
3 Технологический раздел	44
3.1 Диаграммы классов	44
3.1.1 Сообщения	44
3.1.2 Транспорт	44
3.1.3 Менеджер сессий	45
3.1.4 Бизнес-логика	45

3.1.5 Контроллер БД	46
3.2 Язык программирования и среда разработки	46
3.3 Протоколы взаимодействия	47
3.3.1 Протокол синхронного взаимодействия	47
3.3.2 Протокол асинхронного взаимодействия	47
3.4 Сборка и запуск системы	47
3.5 Тестирование системы	49
Заключение	50
Список использованных источников	51
Приложение	52
Приложение А. Тестирование	52

Обозначения и сокращения

АИС — Автоматизированная информационная система.

РСОИ — Распределенная система обработки информации.

ОС — Операционная система

БД — База данных

Введение

Целью работы является создание РСОИ, позволяющую пользователям самостоятельно формировать туристическую путевку. Под формированием путевки понимается бронирование авиабилетов, бронирование номеров отелей и, при наличии входящей в РСОИ платежной системы, их последующая оплата. Разрабатываемая РСОИ состоит из четырех типов узлов: туристические агентства, системы бронирования авиабилетов, системы бронирования гостиничных номеров и система платежей. В данном проекте описана разработка всех узлов РСОИ, за исключением системы платежей. Таким образом РСОИ должна обеспечивать бронирование авиабилетов и гостиничных номеров пользователем.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать предметную область и выделить требования к системе
- спроектировать каждый узел разрабатываемой системы и разработать протокол взаимодействия между системами
- разработать узлы бронирования авиабилетов, гостиничных номеров и турагенств
- провести тестирование системы и проверить ее работоспособность

1 Аналитический раздел

В данном разделе анализируется предметная область и выделяются требования к разрабатываемой РСОИ.

1.1 Анализ предметной области

Процесс оформления туристических путевок включает в себя взаимодействие с несколькими независимыми системами, такими как системы бронирования билетов и системы бронирования номеров в гостиницах. Основная сложность этого процесса заключается в подборе вариантов, удовлетворяющих требованиям пользователя, поскольку поиск приходится производить среди множества различных транспортных компаний и гостиниц. Развитие сети Интернет позволило упростить и автоматизировать этот процесс благодаря созданию распределенных систем, включающих в себя взаимодействующие независимые друг от друга системы. В общем случае, такие системы состоят из некоторого количества узлов, объединенных каналами связи. Узел представляет собой полноценную систему, действующую независимо и согласно своему функционалу. Схему предметной области можно представить в виде схемы на Рис. 1.1. Основной сценарий взаимодействия пользователя с системой выглядит следующим образом:

- а) Пользователь обращается в турагенство, используя веб-сайт.
- б) Пользователь выбирает даты и места путешествия
- в) Система осуществляет поиск возможных вариантов гостиничных номеров и билетов, удовлетворяющим критериям запроса пользователя, и предоставляет пользователю выбрать конкретный вариант.
- г) Пользователь выбирает конкретный вариант, формируя туристическую путевку.
- д) Система бронирует гостиничные номера и билеты и ожидает оплаты пользователем в течение определенного интервала времени.
- е) Пользователь производит оплату тура через систему платежей.

ж) Система подтверждает оплату пользователем и выдает окончательную информацию о путешествии.

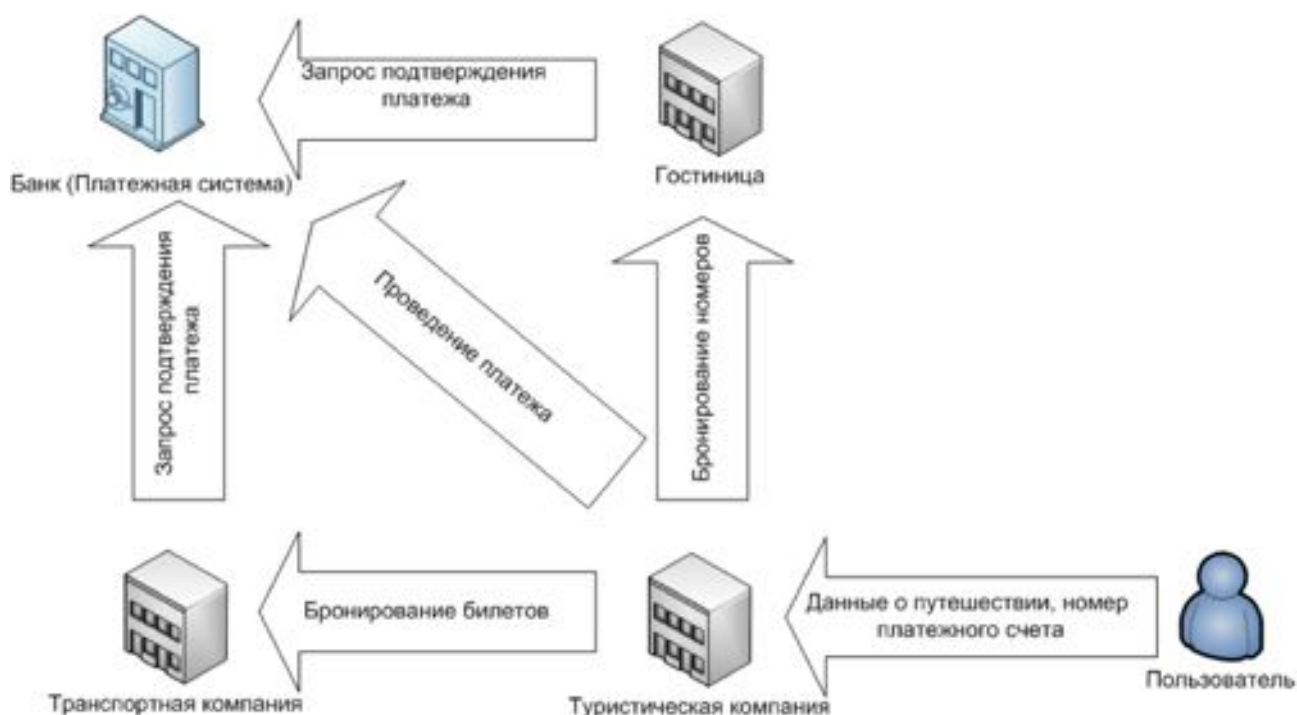


Рисунок 1.1 — Схема предметной области

В случае, если пользователь превысил заданный интервал времени, в течение которого должна была быть произведена оплата, то ему не гарантируется выкуп билетов или гостиничный номер, поскольку системы бронирования билетов или гостиница могут уже снять бронь.

1.2 Описание системы

В системе может находиться любое количество узлов. Узлы могут находиться под управлением различных ОС (Linux, Windows, MacOS и т.д.) и иметь различные технические платформы (например, различные процессоры: x86, Cell, Power, ARM), а также могут быть географически разнесены.

На Рис. 1.2 приведена топология РСОИ «Система заказа туристических путевок». У каждого из входящих в РСОИ узлов имеется собственный сервер БД. РСОИ состоит из следующих систем:

- а) Система платежей - осуществляет платежи между зарегистрированными в ней пользователями. БД системы платежей хранит ин-

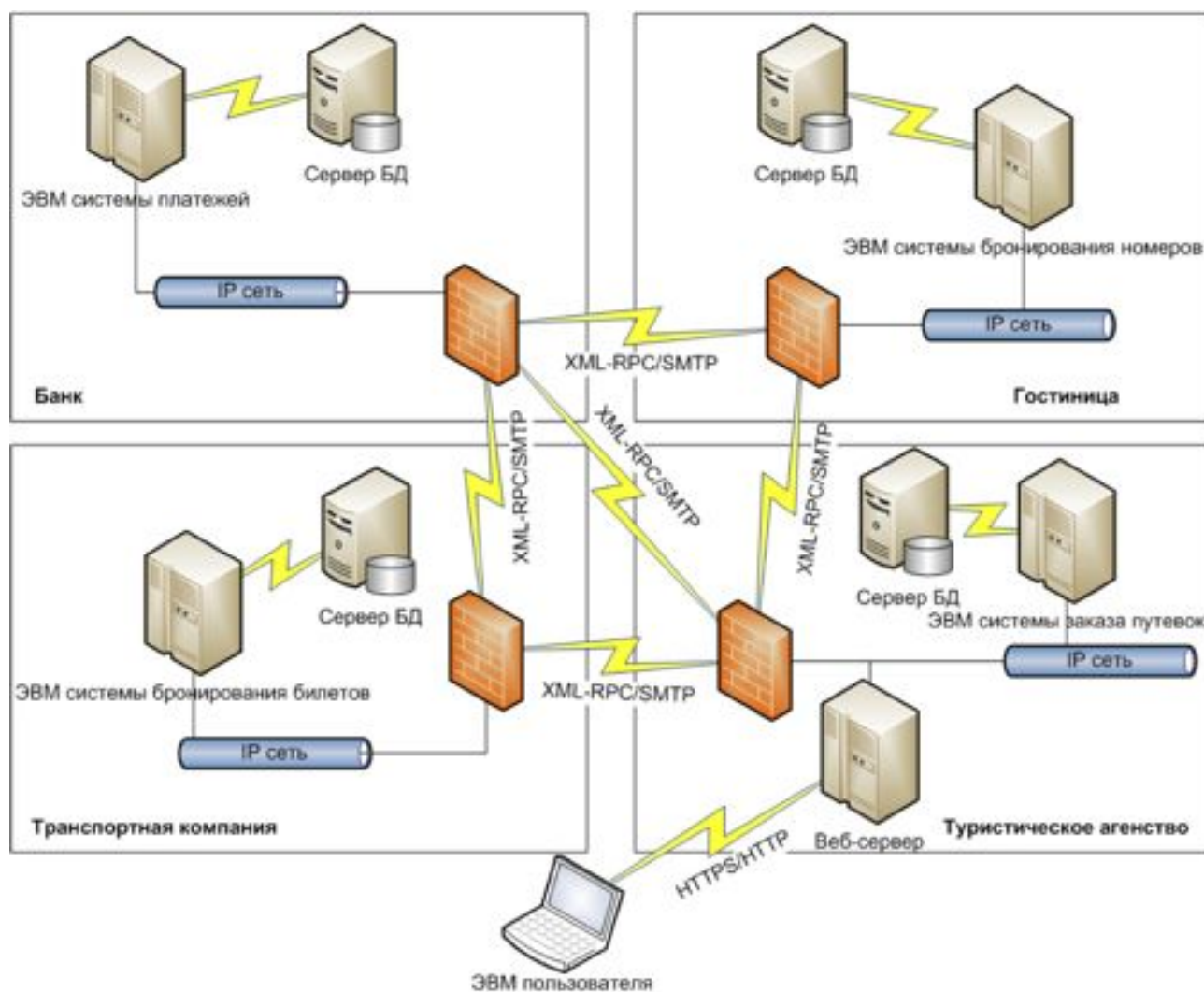


Рисунок 1.2 — Топология РСОИ "Система заказа туристических путе-
вок"

формацию о зарегистрированных в ней пользователях и проводимых платежах. Система платежей принадлежит какому-либо банку.

- б) Система бронирования билетов - позволяет бронировать и выкупать билеты через системы платежей. БД этой системы хранит информацию о наличии билетов. Система бронирования билетов принадлежит какой-либо конкретной транспортной компании (авиа или железнодорожной).
- в) Система бронирования гостиничных номеров, позволяющая бронировать номера в гостинице и оплачивать их через системы платежей. БД системы хранит информацию о наличии свободных и занятых номеров гостиницы. Система бронирования гостиничных номеров принадлежит какой-либо гостинице, либо нескольким гостиницам.
- г) Система заказа путевок, осуществляющая процесс формирования, заказа и оплаты туристической путевки пользователем. БД системы заказа путевок хранит информацию о пользователях системы и их заявках. Система взаимодействует с системами платежей, бронирования гостиничных номеров и билетов. Также система предоставляет веб-интерфейс для своих пользователей.

Узлы устанавливаются в представительствах (офисах) компаний (транспортные компании, гостиницы, банки и туристические агентства). Каждая компания, заинтересованная в своем участии в разрабатываемой системе, должна самостоятельно наполнять БД, расположенную у нее на сервере.

1.3 Требования

На основе проведенного анализа, сформулируем требования, предъявляемые как отдельным узлам, так и системе в целом.

1.3.1 Требования к РСОИ

- а) Система должна обеспечивать доступность на уровне 99,99

- б) Система должна допускать добавление новых узлов.
- в) Каждый узел системы должен конфигурироваться для обеспечения взаимодействия с другими узлами системы.
- г) Узлы РСОИ должны проектироваться исходя из предположения об отсутствии абсолютно надежной связи.
- д) Система должна предоставлять пользователю функции по формированию туристической путевки, ее оплате или отмене.
- е) Система должна предоставлять пользователю варианты путевок, если пользователю не удалось сформировать путевку самостоятельно.

1.3.2 Требования к системе заказа путевок

Требования к функциональным характеристикам

- а) Система должна обеспечивать одновременную работу до 100 пользователей.
- б) Время отклика системы не должно превышать более 5 секунд.
- в) Система должна запрещать одновременную работу пользователей под одними и теми же учетными данными.

Функциональные требования к системе с точки зрения пользователя

Пользователями подсистемы являются физические лица, которые хотят оформить туристическую путевку.

- а) Система должна предоставлять пользователям веб-интерфейс для работы.
- б) Система должна обеспечивать регистрацию пользователей
- в) Система должна обеспечивать аутентификацию пользователей
- г) Система должна обеспечивать пользователю функции по бронированию и выкупу билетов и номеров гостиниц.
- д) Система должна предоставлять пользователю возможность просмотра и изменения состояния заказа путевки.

Входные параметры системы

- а) Адреса узлов , входящих в РСОИ.
- б) Система должна содержать следующую информацию о пользователях:
 - имя пользователя;
 - пароль;
 - паспортные данные;
 - e-mail;
 - телефон;
 - номер пользователя в платежной системе.

Выходные параметры системы

Система выдает пользователю информацию о сформированной и оплаченной путевке.

1.3.3 Требования к системе платежей

Требования к функциональным характеристикам

- а) Система должна обеспечивать надежную передачу данных с использованием шифрования сообщений.
- б) Время отклика системы не должно превышать более 5 секунд.

Функциональные требования к системе с точки зрения пользователя

Пользователями системы являются другие узлы РСОИ, а также физические лица.

- а) Система должна обеспечивать аутентификацию пользователей
- б) Система должна обеспечивать пользователю функцию по переводу средств другим пользователям системы.

Входные параметры системы Система должна содержать следующую информацию о пользователях:

- имя пользователя;

- паспортные данные пользователя или реквизиты компании;
- e-mail пользователя;
- информация об остатке средств на счету пользователя;
- уникальный в пределах системы номер счета пользователя.

Выходные параметры системы

Система списывает средства со счета плательщика и перечисляет их на счет получателя и формирует отчет о проведенной операции.

1.3.4 Требования к системе бронирования авиабилетов

Требования к функциональным характеристикам

- а) Время отклика системы не должно превышать более 5 секунд.

Функциональные требования к системе с точки зрения пользователя

Пользователями подсистемы являются другие узлы, входящие в РСОИ.

- а) Система должна предоставлять пользователю информацию о свободных билетах.
- б) Система должна обеспечивать пользователю функции по бронированию билетов на определенный срок.
- в) В случае, если оплата брони не была произведена, система должна оставлять возможность оплаты.
- г) Система должна предоставлять возможность пользователю снять бронь.
- д) Система должна позволять пользователю получить информацию о статусе своих заявок.

Входные параметры системы

Система должна принимать следующие данные от пользователя:

- время вылета;
- даты вылета;
- пункт отправления;

- пункт назначения;
- номер счета пользователя в системе платежей для осуществления покупки билетов.

Выходные параметры системы

Система предоставляет пользователю информацию о статусе заявки и информацию о запрашиваемых рейсах.

1.3.5 Требования к системе бронирования гостиничных номеров

Требования к функциональным характеристикам

- а) Время отклика системы не должно превышать более 5 секунд.

Функциональные требования к системе с точки зрения пользователя

Пользователями подсистемы являются другие узлы, входящие в РСОИ.

- а) Система должна обеспечивать пользователю функции по бронированию гостиничных номеров на определенный срок и последующему выкупу.
- б) В случае, если оплата не была произведена, система должна оставлять возможность выкупа гостиничного номера.
- в) Система должна предоставлять возможность пользователю снять бронь.
- г) Система должна позволять пользователю получить информацию о статусе своих заявок.

Входные параметры системы

Система должна принимать следующие данные от пользователя:

- дата заезда;
- дата выезда;
- номер в гостинице;

- номер счета пользователя в системе платежей для осуществления покупки билетов.

Выходные параметры системы

Система должна предоставлять пользователю информацию о статусе заявки и информацию о свободных номерах гостиницы.

1.4 Сценарии использования системы

Пользователями системы являются физические лица, которые хотят сформировать и заказать туристическую путевку через систему. Таким образом, в системе выделяется одна роль пользователей - “Простой пользователь”. Диаграмма прецедентов использования системы изображена на Рис. 1.3.

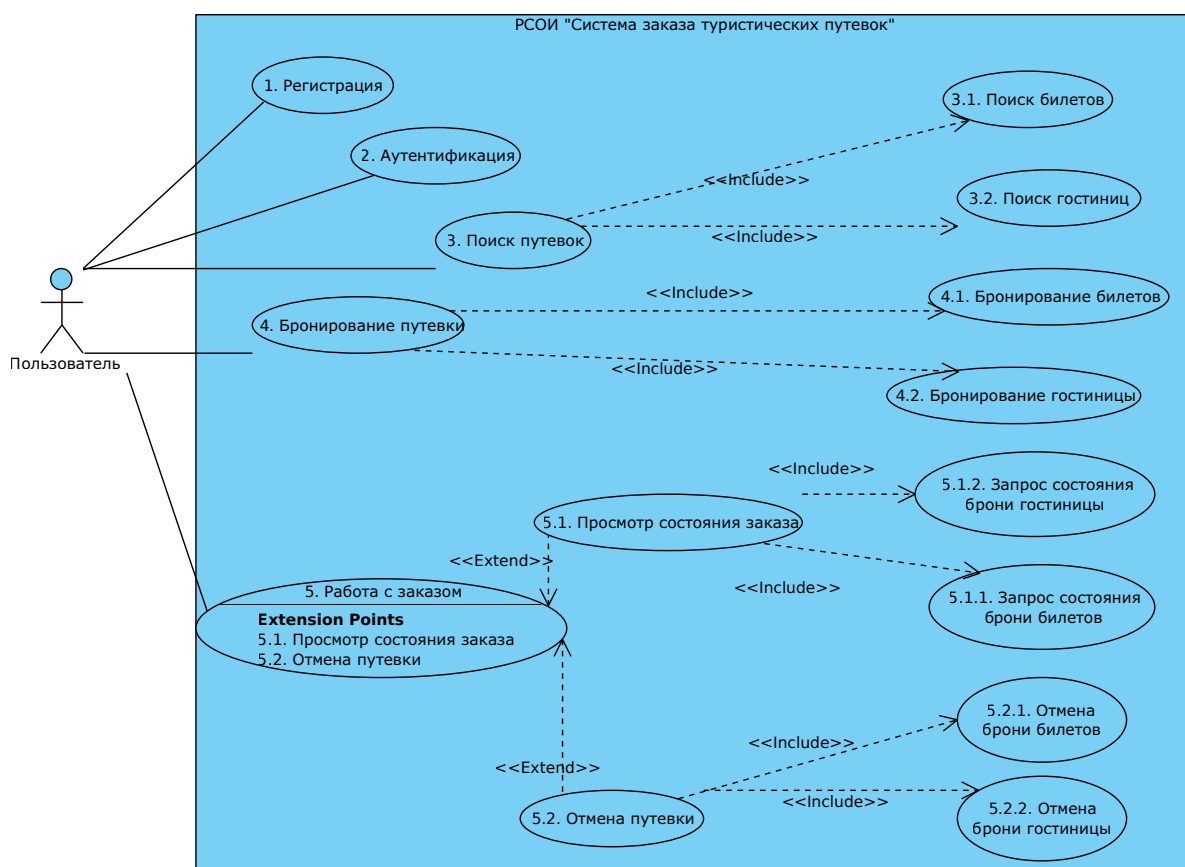


Рисунок 1.3 — Диаграмма прецедентов использования

1.4.1 Спецификации прецедентов использования

Спецификация прецедента "Регистрация"

Цель прецедента использования:прецедент использования позволяет действующему лицу зарегистрироваться в системе.

Сценарий:

- а) Действующее лицо заполняет форму на Web-сайте системы.
- б) Система сохраняет полученные через форму данные пользователя в базе данных.
- в) Система перенаправляет пользователя на страницу рабочего места зарегистрированного пользователя.
- г) Выполнение сценария заканчивается

Спецификация прецедента "Аутентификация"

Цель прецедента использования:прецедент использования позволяет действующему лицу войти в систему.

Сценарий:

Оптимистический сценарий

- а) Действующее лицо вводит имя пользователя и пароль доступа к системе.
- б) Система проверяет введенные данные.
- в) Если данные верны, система перенаправляет пользователя на страницу рабочего места зарегистрированного пользователя.
- г) Выполнение сценария заканчивается

Альтернативный сценарий

- а) Действующее лицо вводит имя пользователя и пароль доступа к системе.
- б) Система проверяет введенные данные.
- в) Если данные не верны, система выводит сообщение о неправильных аутентификационных данных.
- г) Выполнение сценария заканчивается.

Спецификация прецедента "Формирование заказа"

Цель прецедента использования: прецедент использования позволяет действующему лицу сформировать туристическую путевку.

Сценарий:

- а) Действующее лицо выбирает маршрут путешествия.
- б) Действующее лицо выбирает даты путешествия.
- в) Система получает информацию о гостиничных номерах от узла бронирования номеров.
- г) Действующее лицо выбирает номера в гостинице среди доступных.
- д) Действующее лицо подтверждает путевку.
- е) Система выполняет сценарий "Бронирование билетов".
- ж) Система выполняет сценарий "Бронирование номеров".
- з) Выполнение сценария заканчивается.

Спецификация прецедента "Бронирование билетов"

Цель прецедента использования: прецедент использования позволяет системе заказа путевок найти билеты на заданные числа среди систем бронирования билетов, с которыми она взаимодействует.

Сценарий:

Оптимистический сценарий:

- а) Система заказа путевок ищет систему бронирования билетов, у которой есть в наличии билеты на заданную дату.
- б) Система заказа путевок бронирует билеты.
- в) Система бронирования билетов сохраняет информацию о брони в своей базе данных.
- г) Система бронирования билетов устанавливает таймер на оплату брони.
- д) Система бронирования билетов проверяет проведение платежа через систему платежей.
- е) Если оплата была произведена, заявка считается обработанной и сохраняется в архиве обработанных заявок.

ж) Выполнение сценария заканчивается.

Альтернативный сценарий:

- а) Система заказа путевок ищет систему бронирования билетов, у которой есть в наличии билеты на заданную дату.
- б) Система заказа путевок бронирует билеты.
- в) Система бронирования билетов сохраняет информацию о брони в своей базе данных.
- г) Система бронирования билетов устанавливает таймер на оплату брони.
- д) Система бронирования билетов проверяет проведение платежа через систему платежей.
- е) Если таймер истек, а оплата не была проведена, бронь снимается, но заявка на билеты сохраняется в базе данных и возможность выкупа остается, но не гарантируется.
- ж) Выполнение сценария заканчивается.

Спецификация прецедента "Бронирование номеров"

Цель прецедента использования: прецедент использования позволяет системе заказа путевок найти гостиничные номера на заданные числа среди систем бронирования номеров, с которыми она взаимодействует.

Сценарий:

Оптимистический сценарий:

- а) Система заказа путевок ищет систему бронирования гостиничных номеров, у которой есть в наличии свободные гостиничные номера на заданную дату.
- б) Система заказа путевок бронирует номера.
- в) Система бронирования билетов сохраняет информацию о брони в своей базе данных.
- г) Система бронирования билетов устанавливает таймер на оплату брони.

- д) Система бронирования билетов проверяет проведение платежа через систему платежей.
- е) Если оплата была произведена, заявка считается обработанной и сохраняется в архиве обработанных заявок.
- ж) Выполнение сценария заканчивается.

Альтернативный сценарий:

- а) Система заказа путевок ищет систему бронирования гостиничных номеров, у которой есть в наличии номера на заданную дату.
- б) Система заказа путевок бронирует гостиничные номера.
- в) Система бронирования билетов сохраняет информацию о брони в своей базе данных.
- г) Система бронирования билетов устанавливает таймер на оплату брони.
- д) Система бронирования билетов проверяет проведение платежа через систему платежей.
- е) Если таймер истек, а оплата не была проведена, бронь снимается, но заявка на билеты сохраняется в базе данных и возможность выкупа остается, но не гарантируется.
- ж) Выполнение сценария заканчивается.

Спецификация прецедента "Оплата заказа"

Цель прецедента использования: прецедент использования позволяет действующему лицу оплатить сформированную ранее путевку через систему платежей.

Сценарий:

- а) Действующее лицо вводит свой номер в платежной системе.
- б) Действующее лицо проводит оплату через web-страницу рабочего места пользователя.
- в) Система заказа путевок выполняет сценарий "Оплата билетов".
- г) Система заказа путевок выполняет сценарий "Оплата номеров"
- д) Система выводит сообщение о результате проведенного платежа.

- е) Выполнение сценария заканчивается.

Спецификация прецедента "Оплата билетов"

Цель прецедента использования: прецедент использования позволяет системе заказа путевок оплатить ранее забронированные билеты в системе бронирования билетов.

Сценарий:

- а) Система заказа путевок удостоверяется, что период брони билетов не истек через систему бронирования билетов.
- б) Система заказа путевок проводит списание средств со счета пользователя на свой счет через систему платежей.
- в) Система заказа путевок оплачивает забронированные ранее билеты через систему платежей.
- г) Система заказа путевок сообщает пользователю о результате платежа.
- д) Выполнение сценария заканчивается.

Спецификация прецедента "Оплата номеров"

Цель прецедента использования: прецедент использования позволяет системе заказа путевок оплатить ранее забронированные гостиничные номера в системе бронирования номеров.

Сценарий:

- а) Система заказа путевок удостоверяется, что период брони номеров не истек через систему бронирования номеров.
- б) Система заказа путевок проводит списание средств со счета пользователя на свой счет через систему платежей.
- в) Система заказа путевок оплачивает забронированные ранее билеты через систему платежей.
- г) Система заказа путевок сообщает пользователю о результате платежа.
- д) Выполнение сценария заканчивается.

Спецификация прецедента "Отмена заказа"

Цель прецедента использования: прецедент использования позволяет действующему лицу отменить сформированную ранее путевку.

Сценарий:

- а) Действующее лицо вводит сообщает системе заказа путевок об отмене сформированной путевке через страницу рабочего места пользователя.
- б) Система заказа путевок выполняет сценарий “Отмена брони билетов”.
- в) Система заказа путевок выполняет сценарий “Отмена брони номеров”
- г) Система выводит сообщение о результате операции.
- д) Выполнение сценария заканчивается.

Спецификация прецедента "Отмена брони билетов"

Цель прецедента использования: прецедент использования позволяет системе заказа путевок снять бронь с ранее забронированных билетов в системе бронирования билетов.

Сценарий:

- а) Система заказа путевок сообщает системе бронирования билетов об отмене брони.
- б) Система бронирования билетов уничтожает запись в базе данных о брони билетов.
- в) Выполнение сценария заканчивается.

Спецификация прецедента "Отмена брони номеров"

Цель прецедента использования: прецедент использования позволяет системе заказа путевок снять бронь с ранее забронированных гостиничных номеров в системе бронирования номеров.

Сценарий:

- а) Система заказа путевок сообщает системе бронирования номеров об отмене брони.

- б) Система бронирования номеров уничтожает запись в базе данных о брони номеров.
- в) Выполнение сценария заканчивается.

2 Конструкторский раздел

В данном разделе на основе проведенного анализа описывается процесс проектирования РСОИ и ее узлов: узел бронирования гостиничных номеров, узел бронирования авиабилетов и узел заказа туристических путевок.

2.1 Архитектура системы

Проведенный анализ показал, что в состав РСОИ входят четыре типа узлов, проектирование трех из которых рассматривается в рамках данного проекта:

- а) Система бронирования гостиничных номеров
- б) Система бронирования авиабилетов
- в) Система заказа туристических путевок

Следует заметить, что система бронирования авиабилетов и система бронирования гостиничных номеров представляют собой идентичные системы, с единственным различием - тип бронируемых ресурсов. Система заказа путевок должна предоставлять Web-интерфейс пользователю для взаимодействия.

На Рис. 2.1 показаны компоненты РСОИ и интерфейсы, которыми эти компоненты соединены.

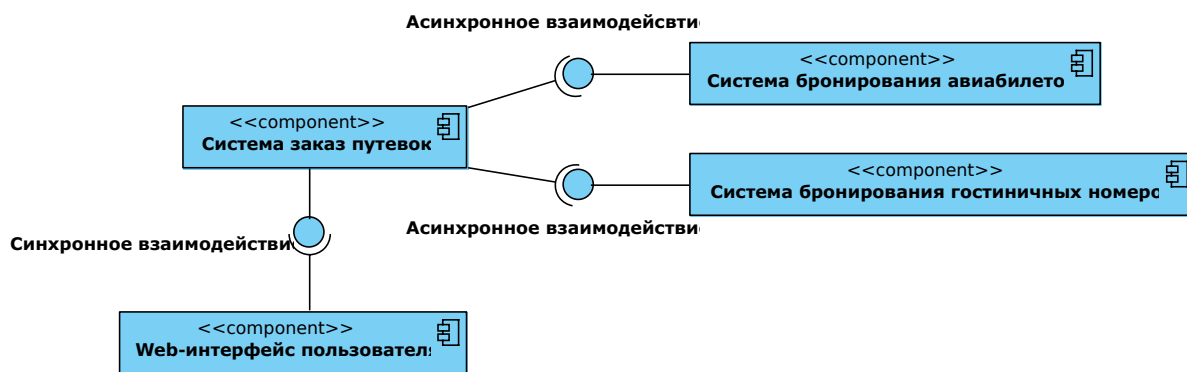


Рисунок 2.1 — Архитектура РСОИ

2.2 Протокол взаимодействия узлов

Для формализации взаимодействия узлов РСОИ, необходимо разработать протокол взаимодействия узлов, определяющий порядок и формат передаваемых сообщений.

2.2.1 Последовательность обмена сообщениями

В результате анализа прецедентов использования РСОИ, было описано взаимодействие систем и спроектирован протокол взаимодействия систем. Среди узлов разрабатываемой РСОИ можно выделить два типа:

- а) Узлы, получающие информацию для пользователя из других систем
- б) Узлы, предоставляющие доступ к своим ресурсам

К первому типу можно отнести системы заказа путевок, ко второму - системы бронирования.

На основе прецедентов использования, выделенных в предыдущем разделе, можно выделить те, в которых происходит взаимодействие между узлами:

- а) Поиск путевки
- б) Бронирование путевки
- в) Просмотр состояния заказа
- г) Отмена заказа

Опишем взаимодействие узлов в для каждого прецедента, используя диаграммы коммуникации.

Поиск путевки

Поиск путевки осуществляется системой на основе параметров поиска, таких как, например, пункт назначения, даты и количество человек. Система заказа путевок выполняет поиск по всем узлам бронирования, входящих в РСОИ. На Рис. 2.2 изображена диаграмма коммуни-

кации, отражающая взаимодействие узлов во время выполнения поиска путевки.

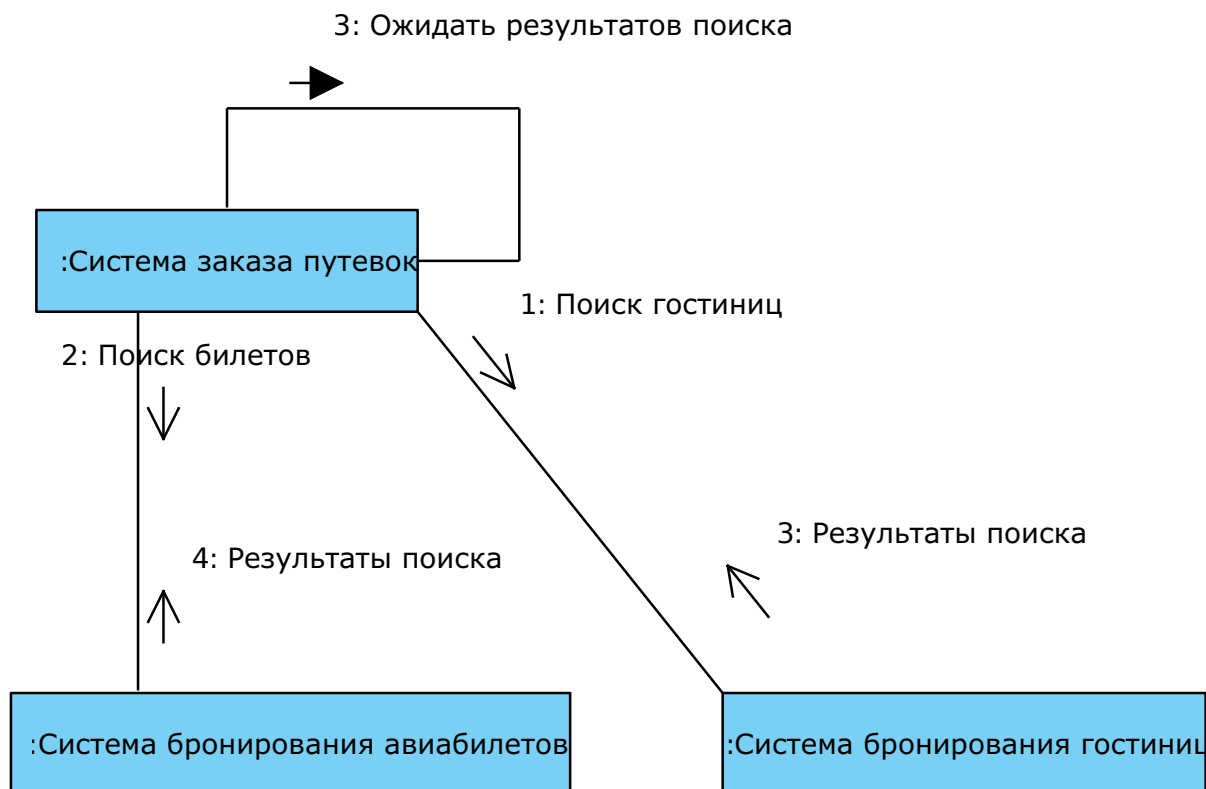


Рисунок 2.2 — Диаграмма коммуникации "Поиск путевки"

Бронирование путевки

На основе результатов поиска, пользователь способен выбрать варианты авиабилетов и гостиниц и сформировать заказ на путевку, забронировав эти ресурсы. На Рис. 2.3 изображена диаграмма коммуникации, отражающая взаимодействие узлов во время выполнения бронирования.

Просмотр состояния заказа

На Рис. 2.4 изображена диаграмма коммуникации, отражающая взаимодействие узлов во время выполнения прецедента "Просмотр состояния заказа".

Отмена путевки

На Рис. 2.5 изображена диаграмма коммуникации, отражающая взаимодействие узлов во время выполнения бронирования.

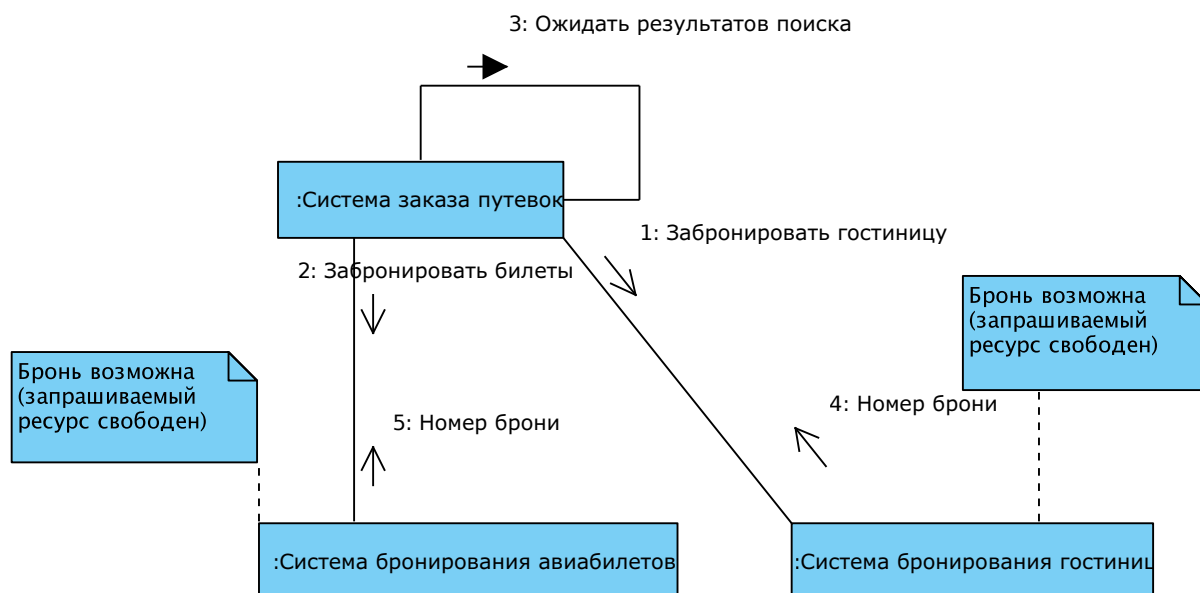


Рисунок 2.3 — Диаграмма коммуникации "Бронирование путевки"

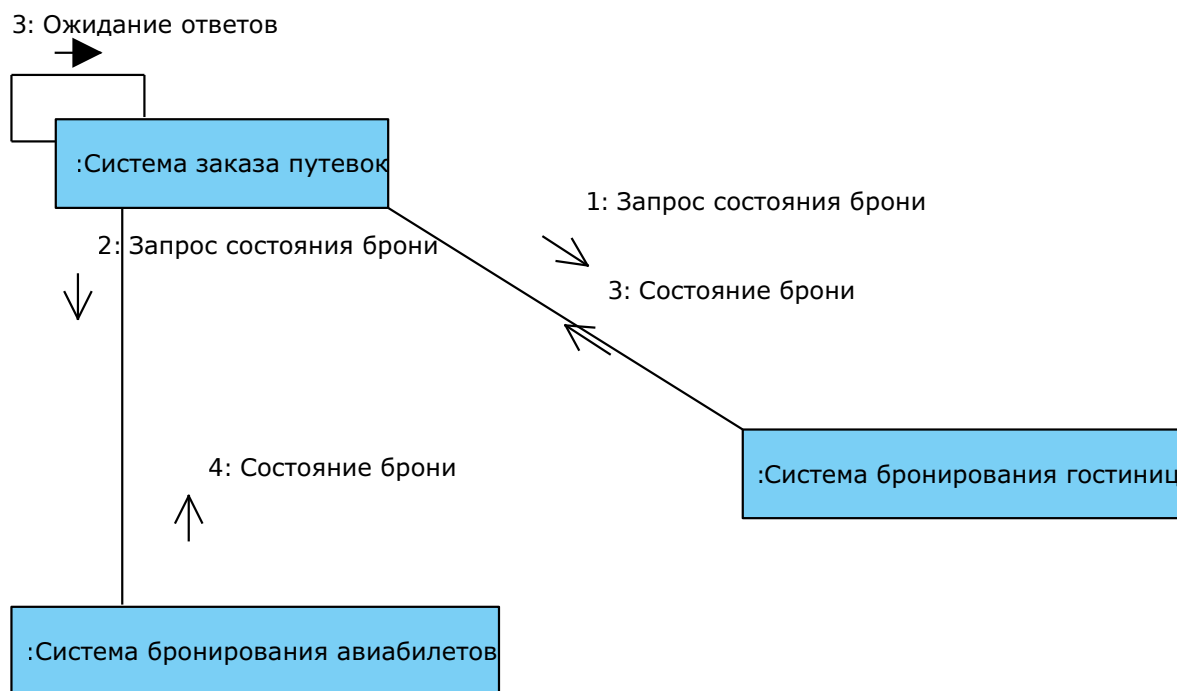


Рисунок 2.4 — Диаграмма коммуникации "Просмотр состояния заказа"

2.2.2 Передаваемые сообщения

В результате спроектированных моделей поведения узлов можно определить типы и форматы передаваемых сообщений (см. табл. 2.1). В РСОИ взаимодействуют всего два вида узлов (системы заказа путевок и системы бронирования), причем это взаимодействие носит клиент-серверный характер, поэтому все сообщения разделены на два типа:

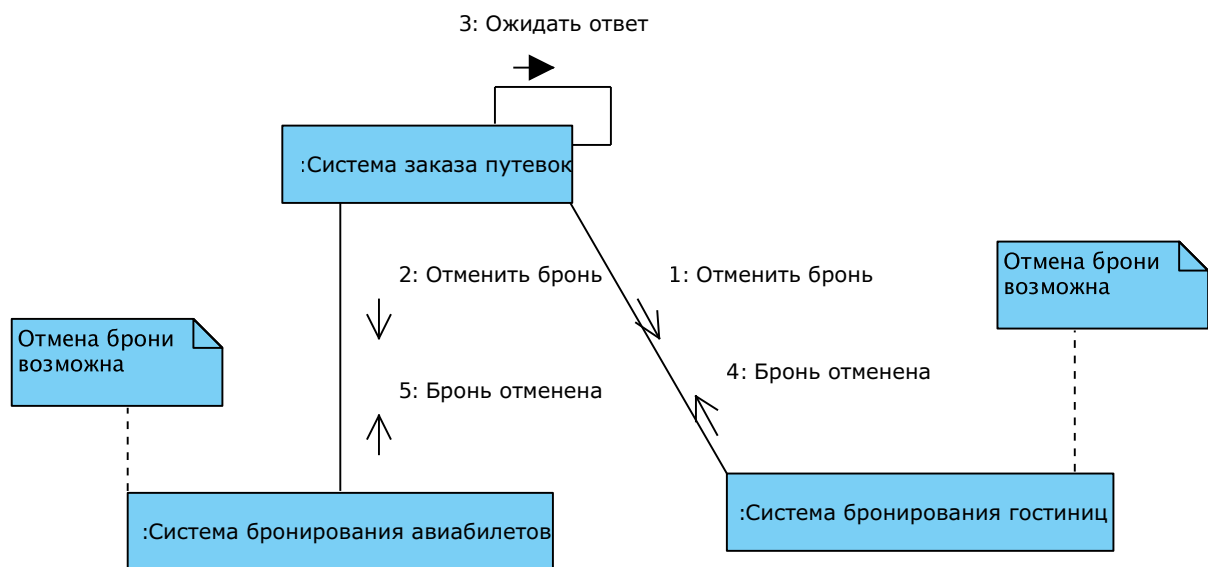


Рисунок 2.5 — Диаграмма коммуникации "Отмена путевки"

- а) Запрос - сообщения, посылаемые системой заказа путевок
- б) Ответ - сообщения, посылаемые системами бронирования в ответ на запрос

Сообщения также разумно разделить на два других класса:

- а) Авторизованные сообщения - сообщения, пересылаемые между авторизованным в системе бронирования пользователем и системой
- б) Анонимные сообщения - сообщения, не требующие авторизации пользователя в системе

Единственным различием между этими классами сообщений является то, что авторизованные сообщения содержат в себе поле идентификатора сессии пользователя, полученного после авторизации.

Таблица 2.1 — Типы сообщений, передаваемых между узлами

Тип	Название	Данные	Комментарий
Авторизованные сообщения			
Запрос	Бронирование ресурсов	Номера ресурсов	
Ответ	Результат брони	Номер брони	

Продолжение на след. стр.

Продолжение таблицы 2.1

Запрос	Проверка брони	Номер брони	
Ответ	Статус брони	Статус брони	Бронь может иметь несколько статусов: <ul style="list-style-type: none"> – ожидает оплаты – оплачена – отменена
Запрос	Отмена брони	Номер брони	
Запрос	Получить все заказы	Имя пользователя	
Ответ	Список заказов	Список всех заказов пользователя	
Анонимные сообщения			
Запрос	Поиск ресурсов	Параметры поиска	Параметры поиска содержат такие данные как, например, даты заезда, выезда и город пребывания
Ответ	Результаты поиска	Найденные ресурсы	Поскольку результаты поиска могут быть достаточно большими для отправки (в зависимости от протокола асинхронного взаимодействия, для XMPP - 65535 байт [1]), то разумно предусмотреть отправку нескольких сообщений, содержащих результат поиска
Ответ	Запрос успешно выполнен		

Продолжение на след. стр.

Продолжение таблицы 2.1

Ответ	Ошибка	Номер ошибки и комментарий	
Запрос	Авторизация	Имя пользователя и пароль	
Ответ	Открыта сессия	Идентификатор пользовательской сессии	

2.3 Система бронирования

Системы бронирования гостиничных номеров и авиабилетов имеют отличия между собой в типе бронируемых ресурсов, поэтому в данном разделе их проектирование будет объединено проектированием системы бронирования в общем случае.

2.3.1 Прецеденты использования

На основе разработанного протокола, можно выделить основные прецеденты использования системы бронирования, для дальнейшего анализа и проектирования системы. На Рис. 2.6 изображена диаграмма прецедентов использования системы бронирования.

2.3.2 Архитектура системы

В системе были выделены следующие подсистемы:

- а) Транспорт - выполняет задачи приема и отправки сообщений
- б) Менеджер сессий - управляет сессиями пользователей
- в) Бизнес-логика - логика работы системы бронирования
- г) Контроллер БД - предоставляет функции для доступа к БД

На диаграмме компонентов (см. Рис. 2.7) изображены выделенные подсистемы.

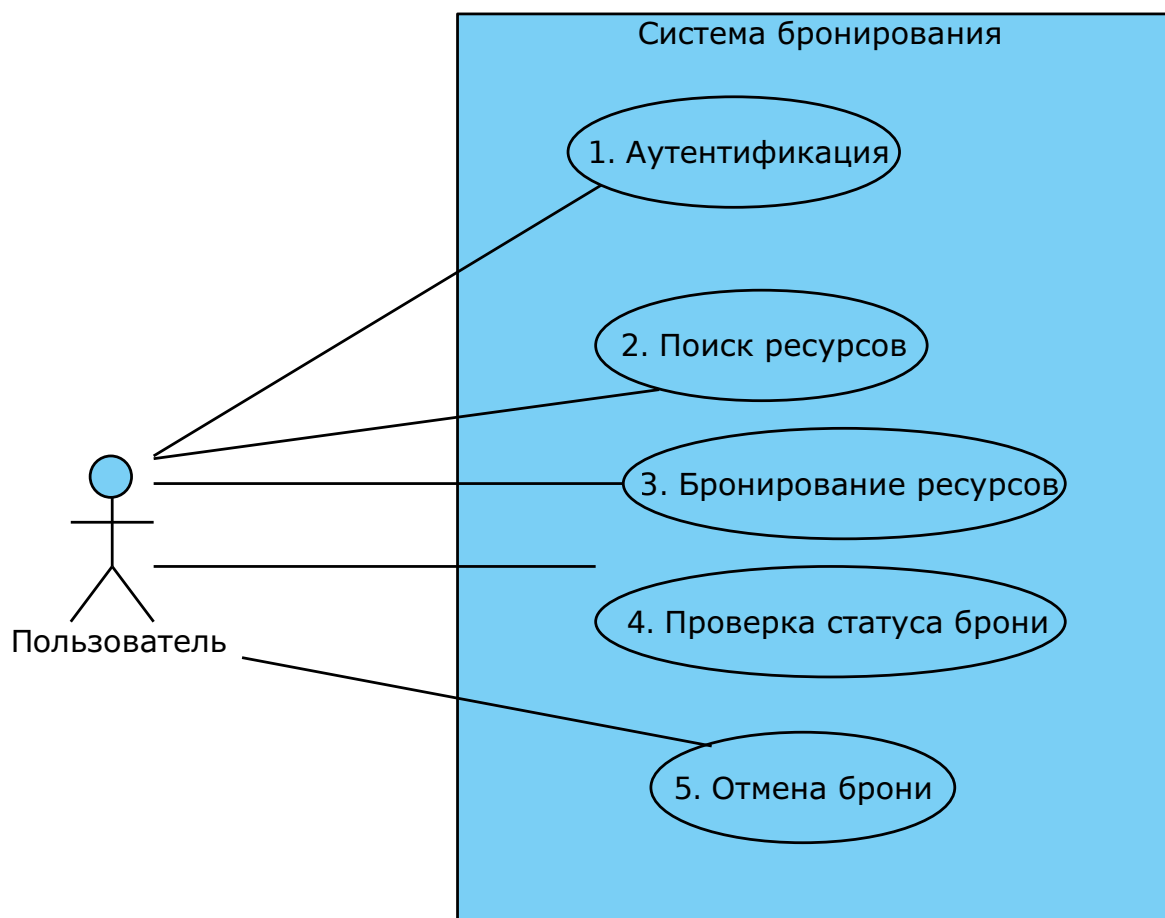


Рисунок 2.6 — Диаграмма прецедентов системы бронирования

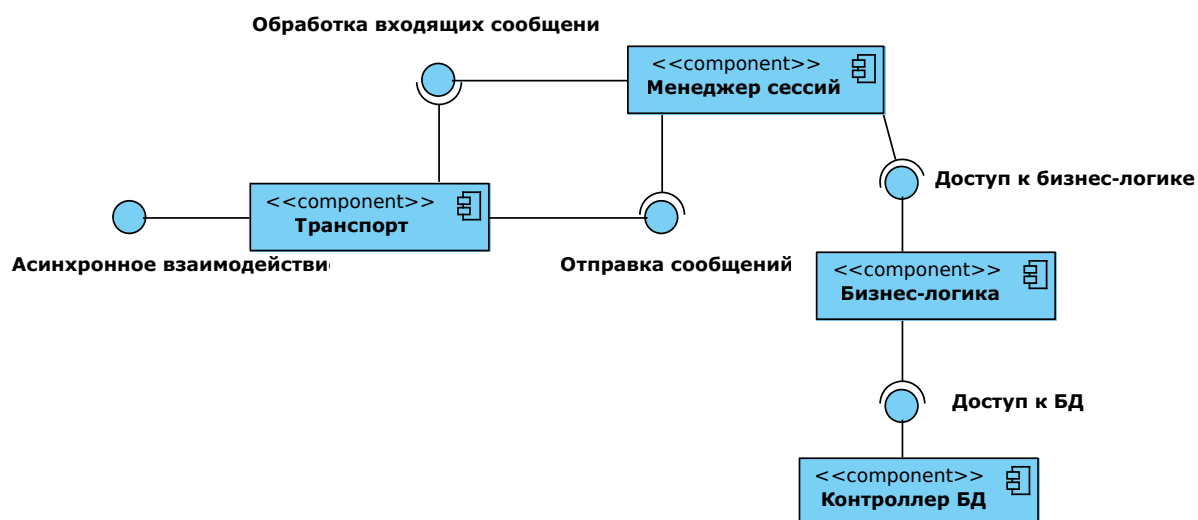


Рисунок 2.7 — Диаграмма компонентов системы бронирования

Транспорт

Можно разделить задачи транспортной подсистемы на две:

- прием и отправка синхронных сообщений
- прием и отправка асинхронных сообщений

Структура транспортной подсистемы изображены на Рис. 2.8

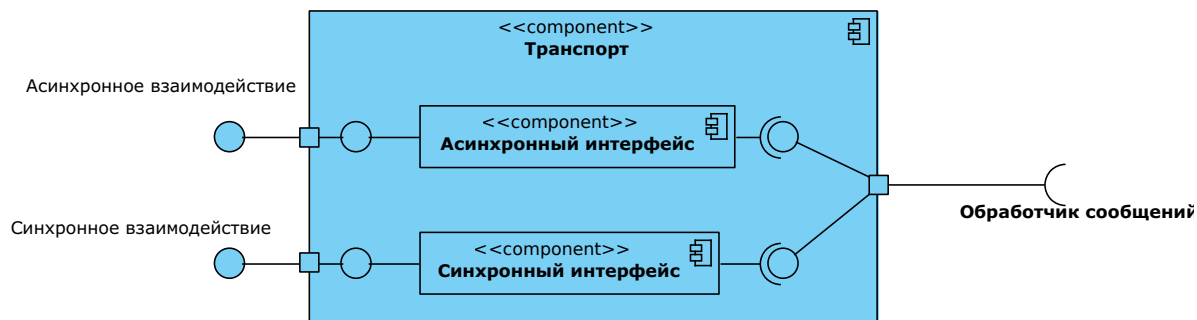


Рисунок 2.8 — Диаграмма компонентов транспортной подсистемы

Менеджер сессий

Менеджер сессий выполняет задачи диспетчеризации входящих сообщений и управление сессиями авторизованных клиентов. Структура менеджера сессий приведена на Рис. 2.9. Для каждого авторизованного пользователя создается новая сессия пользователя, основная задача которой - обработка сообщений от пользователя. Для анонимных сообщений используется анонимная сессия. Менеджер сессий также обязан следить за временем жизни сессий пользователей: если от пользователя не было сообщений в течение определенного интервала времени, сессия уничтожается.

На Рис. 2.10 описано поведение менеджера сессий при каждом получении входящего сообщения.

Бизнес-логика

Бизнес-логика системы бронирования берет на себя задачи бронирования ресурсов и управлением бронью. Бизнес-логика взаимодействует с базой данных через контроллер БД и сессиями пользователей.

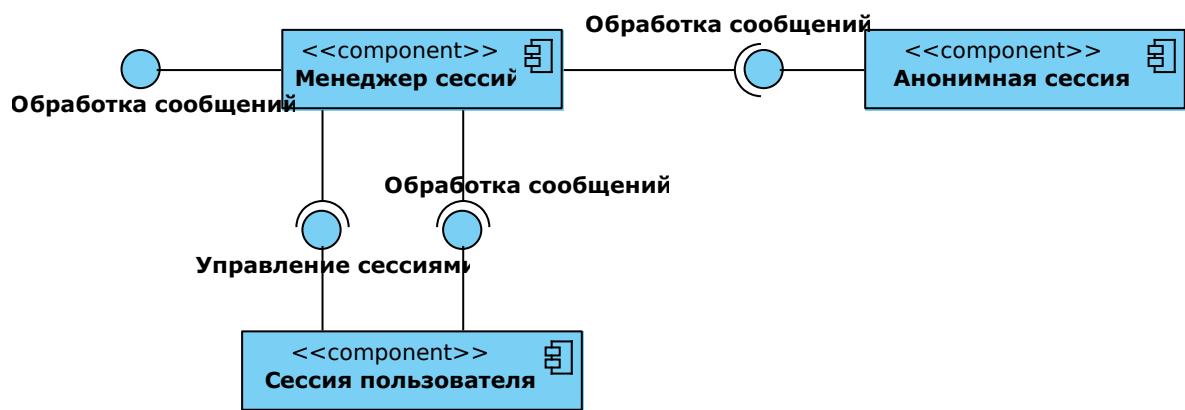


Рисунок 2.9 — Диаграмма компонентов менеджера сессий

Поведение системы

На Рис. 2.11, Рис. 2.12, Рис. 2.13 и Рис. 2.14 изображены последовательности действий для бронирования ресурса, проверки статуса заказа, отмены заказа и поиска ресурсов соответственно.

2.3.3 Модель данных

Для системы бронирования были выделены следующие сущности:

а) Пользователь

- e-mail
- пароль
- заказы

б) Заказ

- ресурсы
- дата
- общая стоимость
- статус

в) Ресурс

- данные ресурса
- стоимость
- состояние

На Рис. 2.15 изображена диаграмма сущность-связь для системы бронирования билетов, на Рис. 2.16 - для системы бронирования гостиничных номеров. [h!]

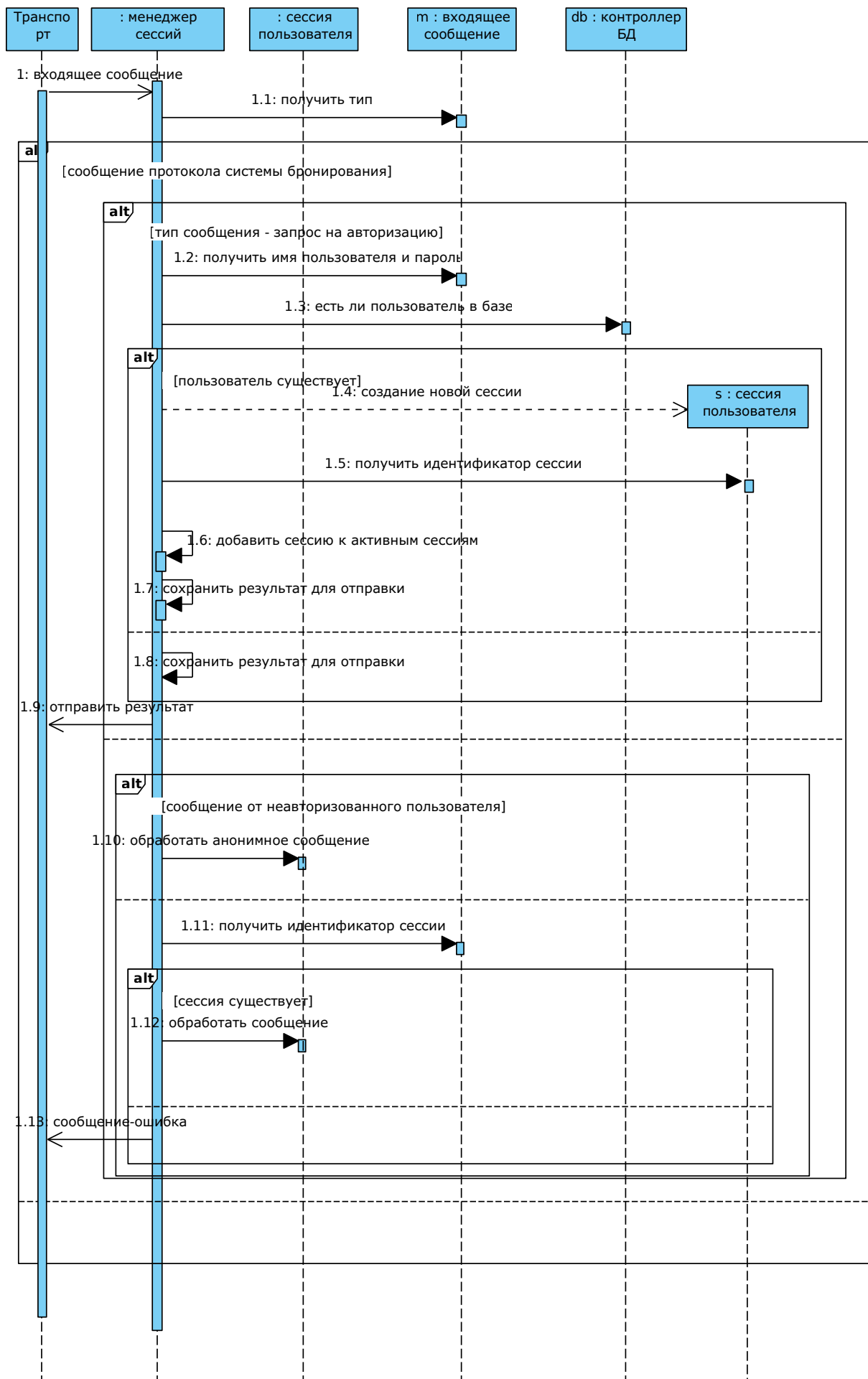


Рисунок 2.10 — Обработка сообщения (диаграмма последовательности)

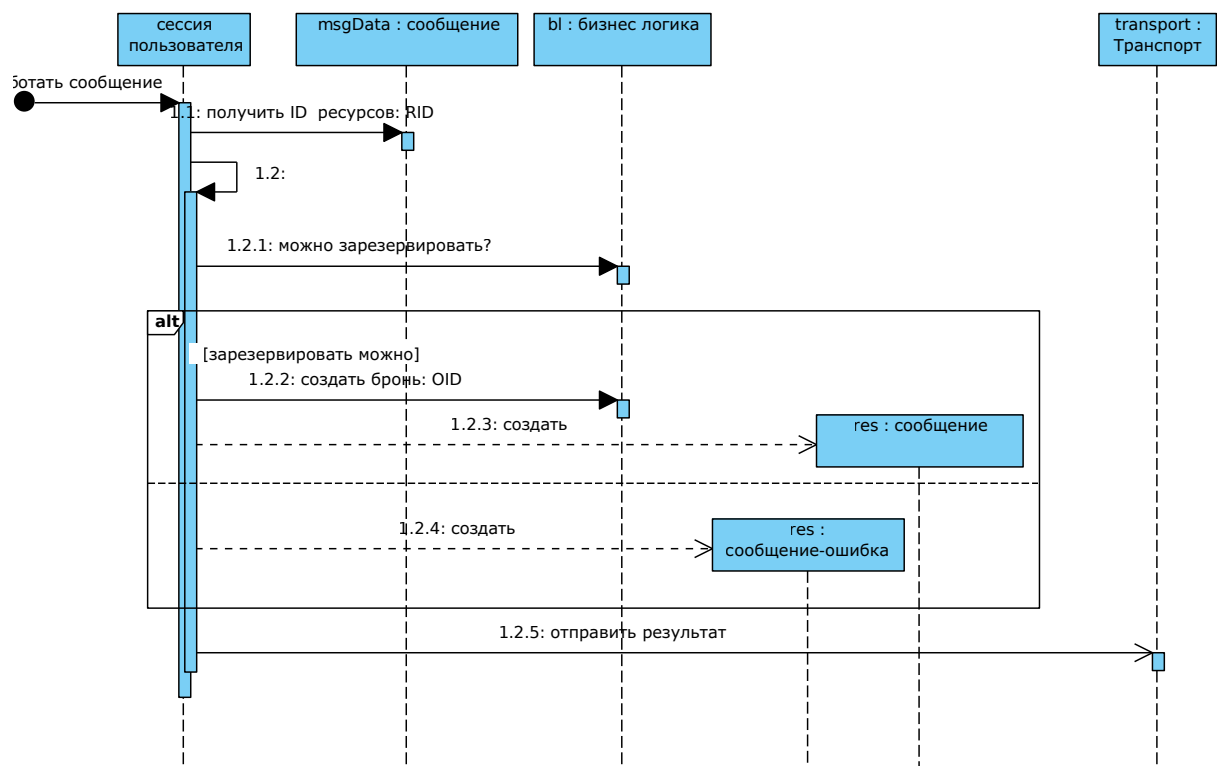


Рисунок 2.11 — Бронирование ресурсов

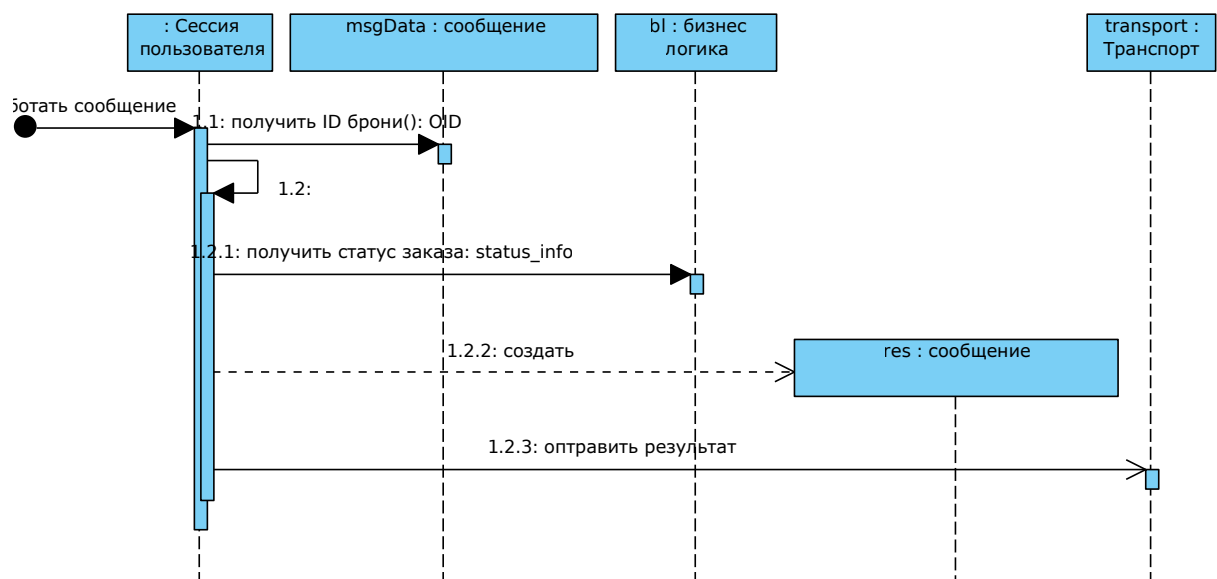


Рисунок 2.12 — Проверка статуса заказа

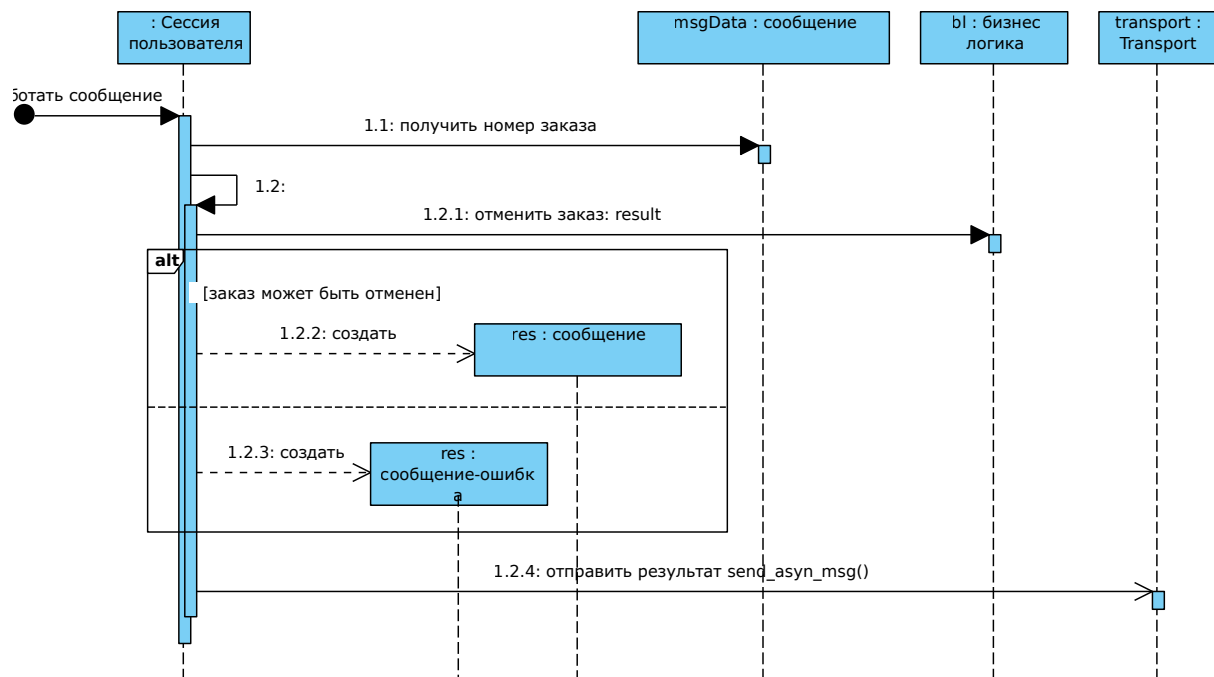


Рисунок 2.13 — Отмена заказа

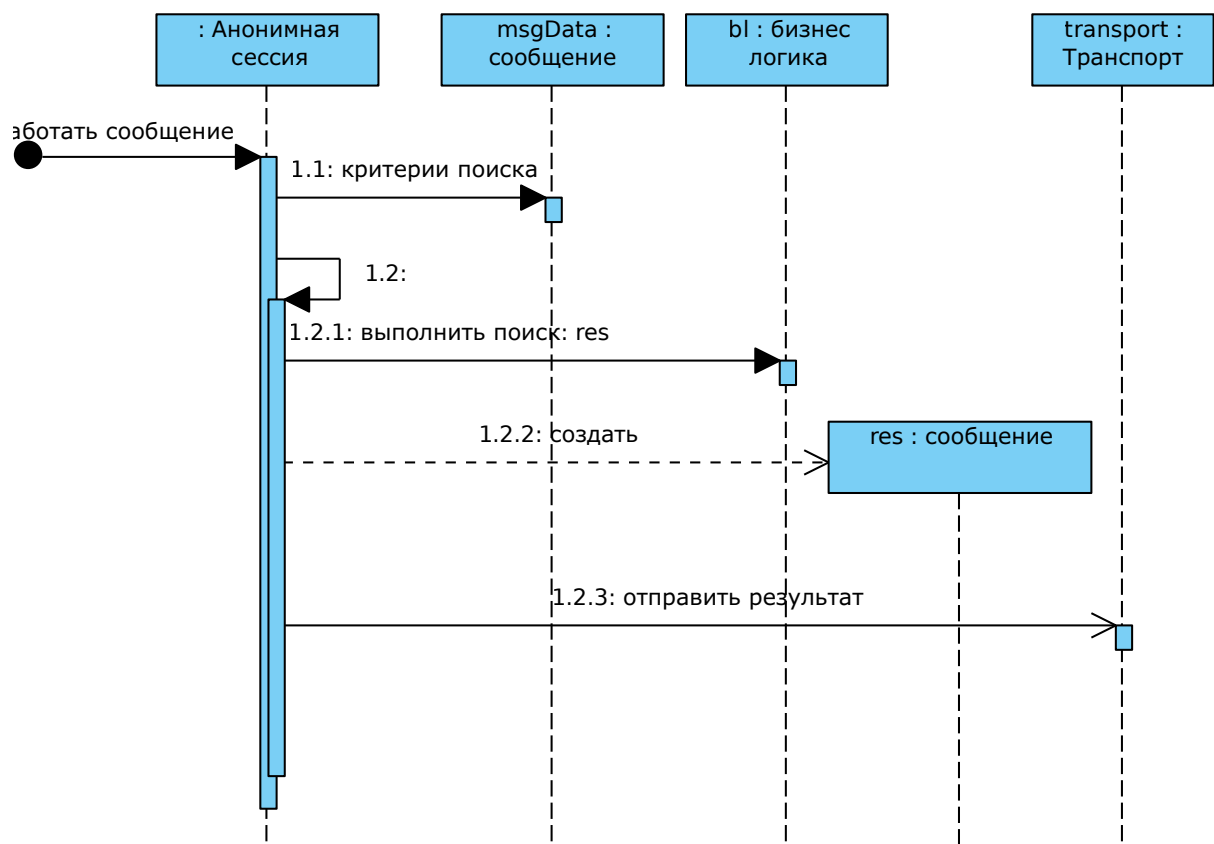


Рисунок 2.14 — Поиск ресурсов

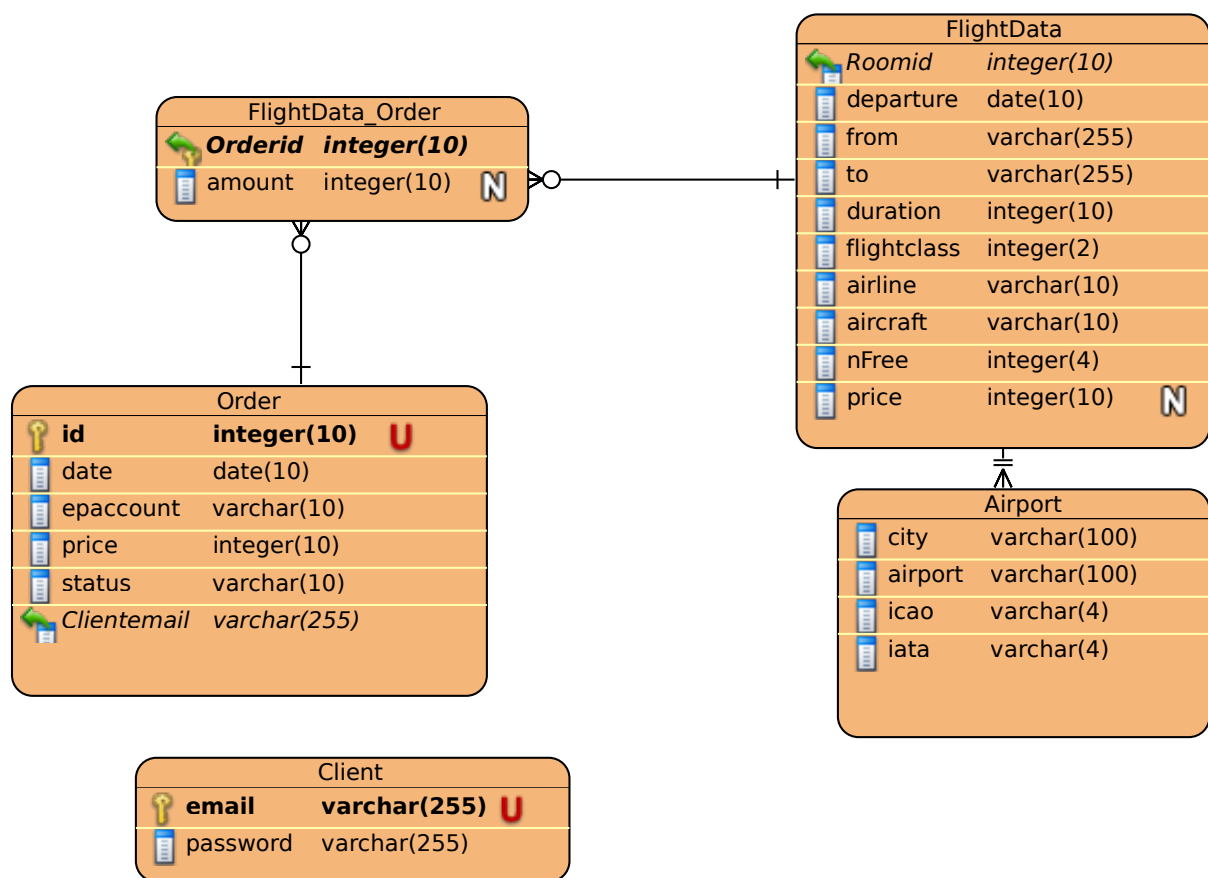


Рисунок 2.15 — ER-диаграмма для системы бронирования билетов

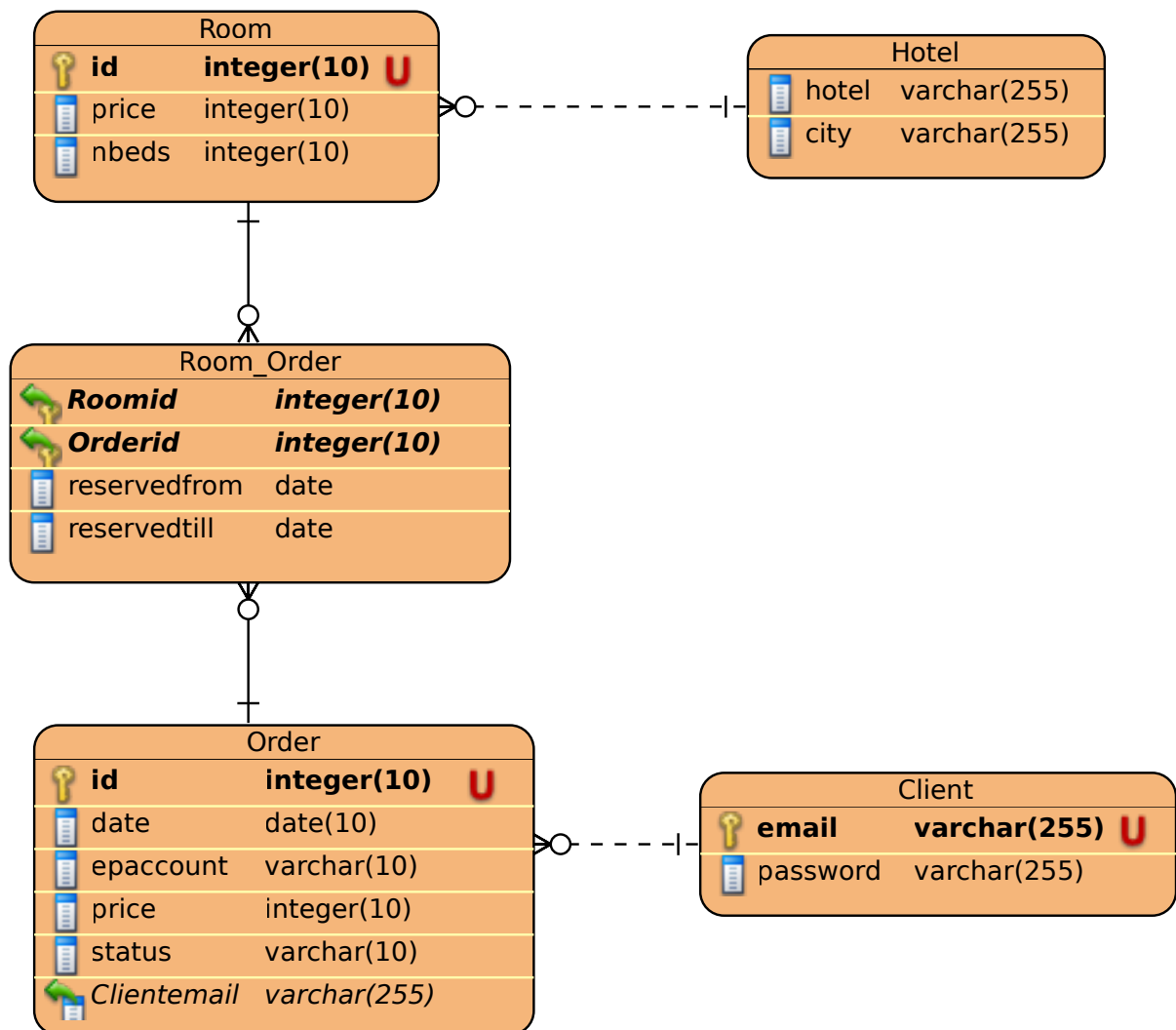


Рисунок 2.16 — ER-диаграмма для системы бронирования гостиничных номеров

2.4 Система заказа путевок

2.4.1 Прецеденты использования

В результате анализа было выявлено, что прецеденты использования системы заказа путевок совпадают с прецедентами использования РСОИ (см. Рис. 1.3).

2.4.2 Архитектура системы

В системе заказа путевок были выявлены следующие подсистемы:

- а) Транспорт - выполняет задачи приема и отправки сообщений
- б) Менеджер клиентов - управляет сессиями пользователей
- в) Бизнес-логика - логика работы системы
- г) Контроллер БД - предоставляет функции для доступа к БД
- д) Web-интерфейс - интерфейс пользователя системы

На диаграмме компонентов (см. Рис. 2.17) изображены выделенные подсистемы.

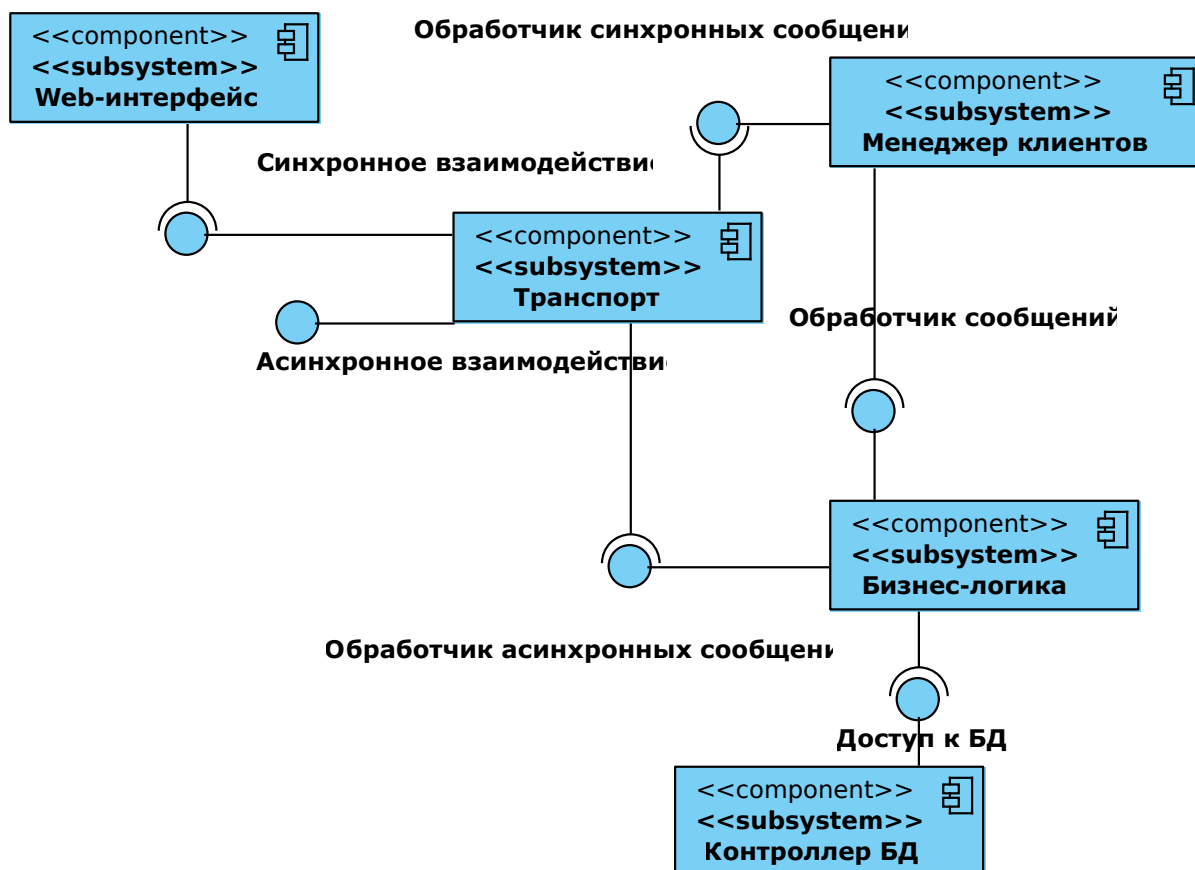


Рисунок 2.17 — Диаграмма компонентов системы заказа путевок

Транспорт

Подсистема транспорта занимается приемом и отправкой синхронных и асинхронных сообщений и полностью идентична рассмотренной транспортной подсистеме в предыдущем разделе.

Менеджер клиентов

Менеджер клиентов обрабатывает входящие синхронные сообщения от пользователей и отправляет их на обработку.

Бизнес-логика

В задачи бизнес-логики входит:

- обработка сообщений пользователя
- отправка запросов узлам РСОИ, согласно логике обработки
- обработка ответов от узлов РСОИ

Структура подсистемы бизнес-логики изображена на Рис. 2.18.

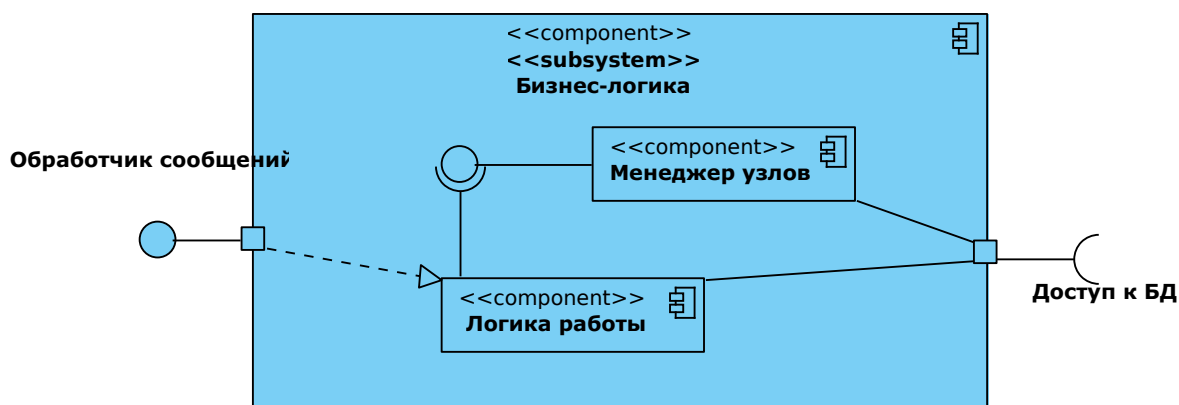


Рисунок 2.18 — Диаграмма компонентов подсистемы бизнес-логики

Web-интерфейс

С помощью данной подсистемы пользователь использует все функции РСОИ. Основная задача Web-интерфейса принимать команды от пользователя, формировать на их основе сообщения, отправлять их транспортной подсистеме, синхронно принимать ответы на них и выводить информацию пользователю в удобном и понятном для него виде.

В табл. 2.2 указаны все синхронные сообщения, которыми обмениваются Web-интерфейс и система заказа путевок.

Таблица 2.2 — Типы сообщений, передаваемых между подсистемой Web-интерфейса и системой транспортной подсистемой

Тип	Название	Данные	Комментарий
Авторизованные сообщения			
Запрос	Заказ путевки	Номера ресурсов	
Ответ	Сформирован заказ	Номер заказа	
Запрос	Проверка статуса заказа	Номер заказа	
Ответ	Статус заказа	Статус заказа	Заказ может иметь несколько статусов: <ul style="list-style-type: none"> – ожидает оплаты – оплачен – отменен – оплачены билеты – оплачена гостиница
Запрос	Отмена заказа	Номер заказа	
Запрос	Получить все заказы	Имя пользователя	
Ответ	Список заказов	Список всех заказов пользователя	
Анонимные сообщения			
Запрос	Поиск ресурсов	Параметры поиска	Параметры поиска содержат такие данные как, например, даты заезда, выезда и город пребывания

Продолжение на след. стр.

Продолжение таблицы 2.1

Ответ	Результаты поиска	Найденные ресурсы	Поскольку результаты поиска могут быть достаточно большими для отправки (в зависимости от протокола асинхронного взаимодействия, для XMPP - 65535 байт [1]), то разумно предусмотреть отправку нескольких сообщений, содержащих результат поиска
Ответ	Запрос успешно выполнен		
Ответ	Ошибка	Номер ошибки и комментарий	
Запрос	Авторизация	Имя пользователя и пароль	
Ответ	Открыта сессия	Идентификатор пользовательской сессии	

Поведение системы

Поведение системы определяется взаимодействием ее подсистем. Взаимодействие подсистем можно увидеть на диаграммах активностей (Рис. 2.20 и Рис. 2.19) для прецедентов бронирования и поиска путевок.

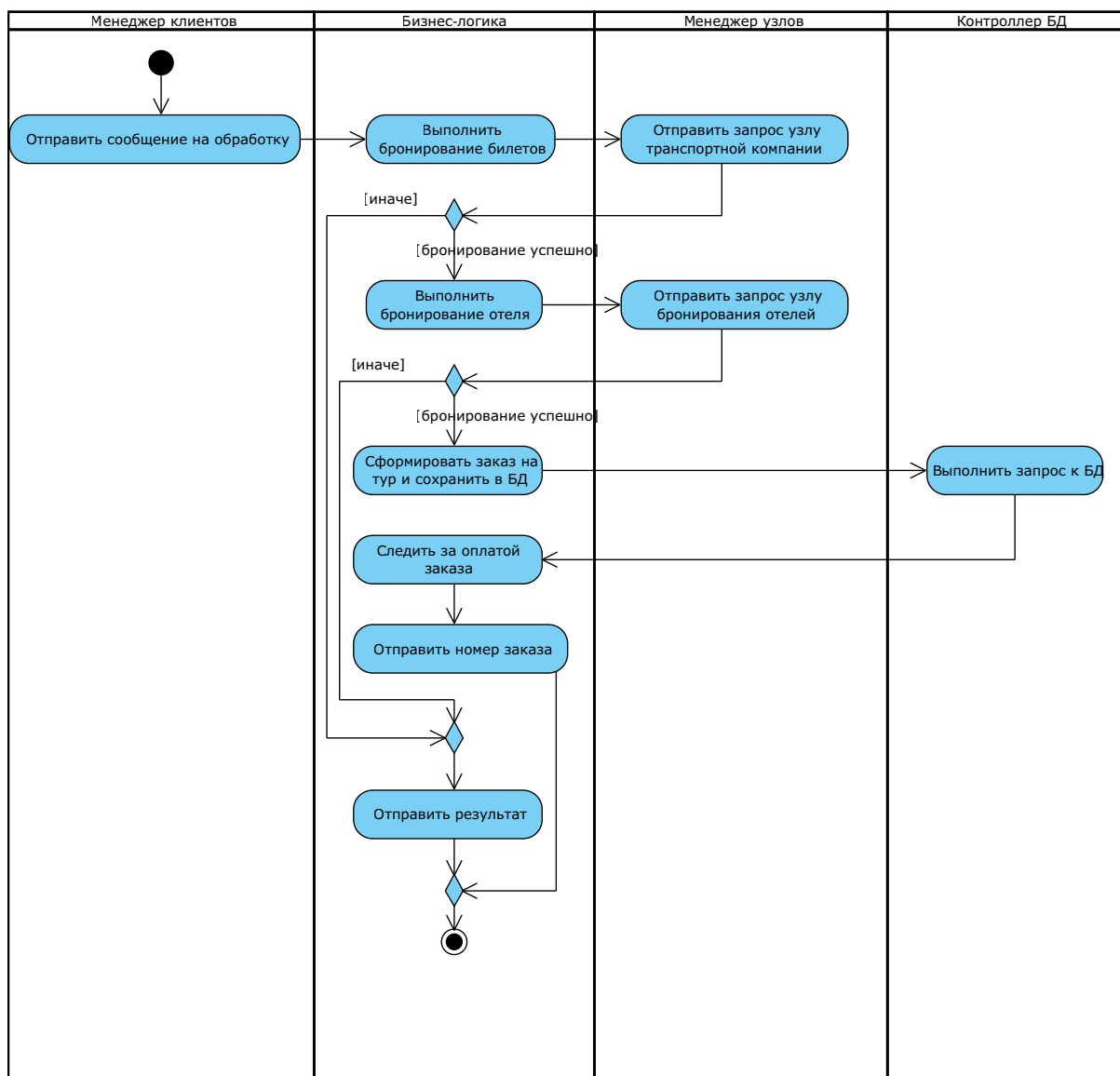


Рисунок 2.19 — Диаграмма активностей для прецедента "Бронирование путевки"

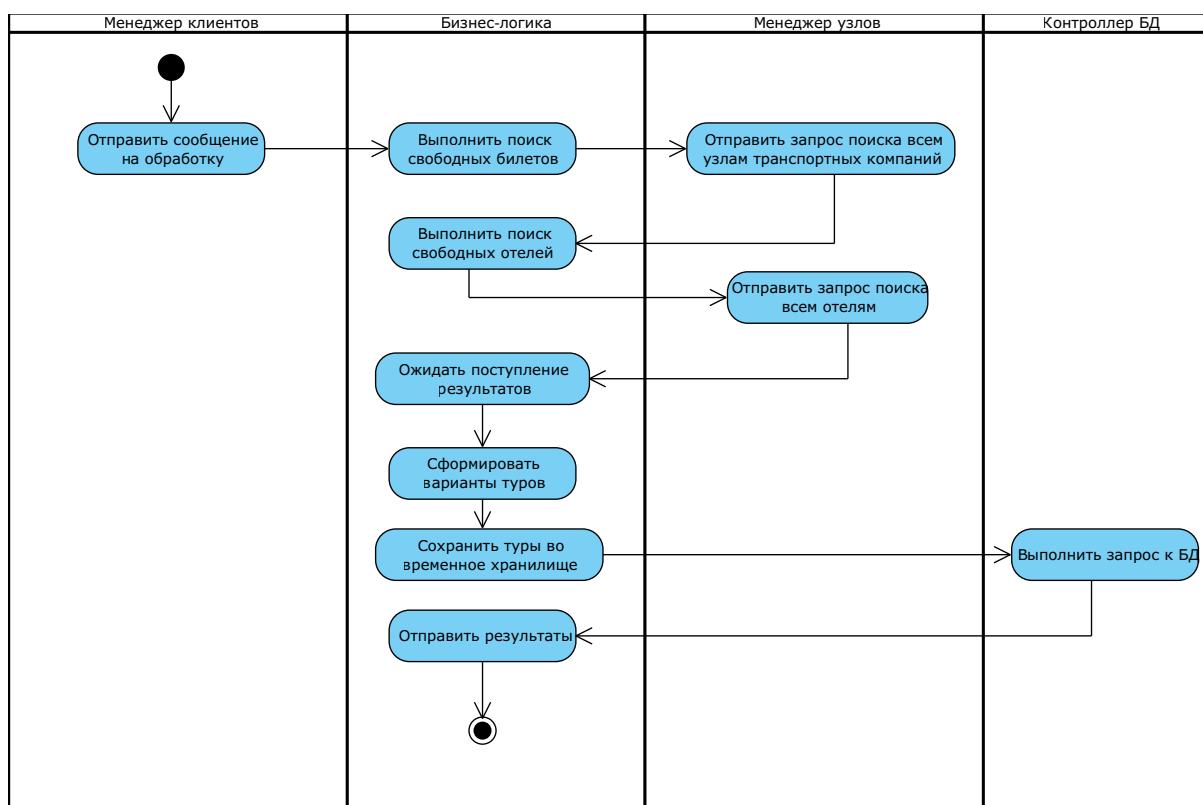


Рисунок 2.20 — Диаграмма активностей для прецедента "Поиск путевки"

2.4.3 Модель данных

Для системы заказа путевок были выделены следующие сущности:

а) Пользователь

- e-mail
- пароль
- заказы
- сессия

б) Заказ

- брони
- ресурсы
- дата
- общая стоимость
- статус

в) Бронь

- номер брони
- узел

г) Ресурс

- номер ресурса
- узел

д) Сессия

- идентификатор сессии
- время

На Рис. 2.21 изображена диаграмма сущность-связь для системы заказа путевок.

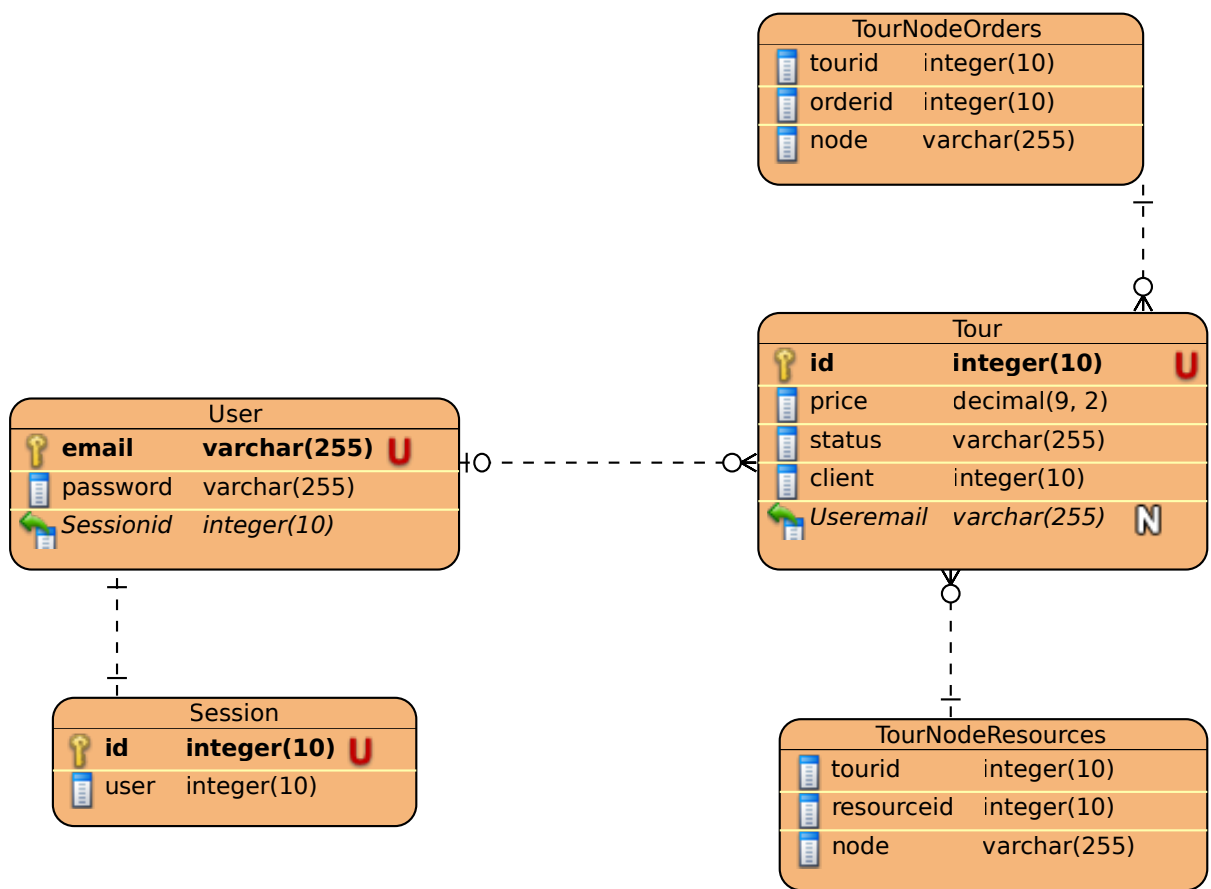


Рисунок 2.21 — ER-диаграмма для системы заказа путевок

3 Технологический раздел

3.1 Диаграммы классов

В данном разделе приведены диаграммы классов разработанных систем.

3.1.1 Сообщения

Класс DSMessage используется для обмена информацией между узлами и соответствует разработанному протоколу. Для передачи по сети, каждое сообщение упаковывается в формат YAML и отправляется. При получении сообщения, на основе полученного YAML формируется новый объект класса.

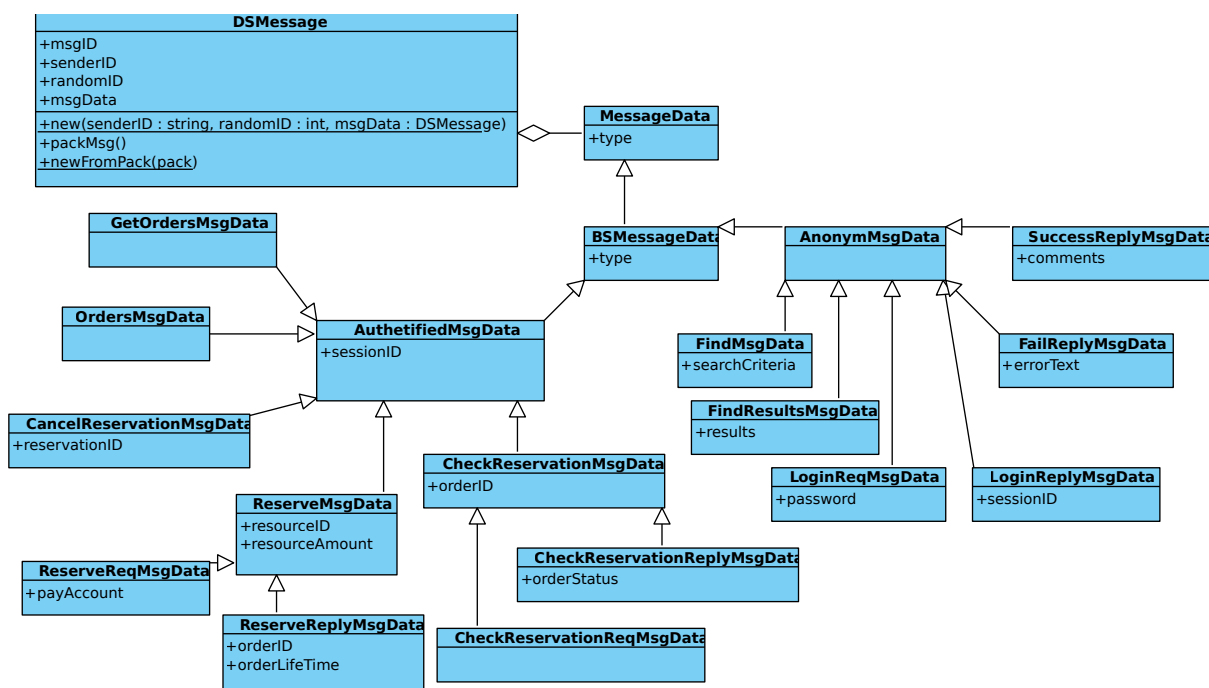


Рисунок 3.1 — Диаграмма классов для сообщений

3.1.2 Транспорт

На Рис. 2.8 представлена диаграмма классов транспортной подсистемы.

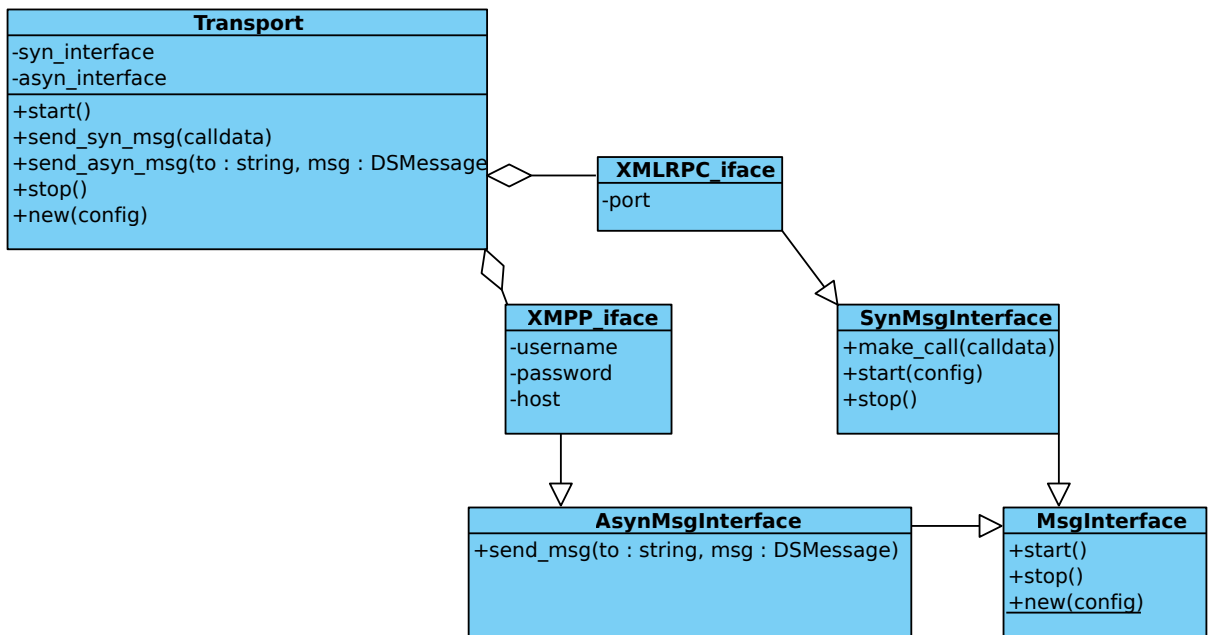


Рисунок 3.2 — Диаграмма классов для транспортной подсистемы

3.1.3 Менеджер сессий

На Рис. 3.3 представлена диаграмма классов подсистемы менеджера сессий.

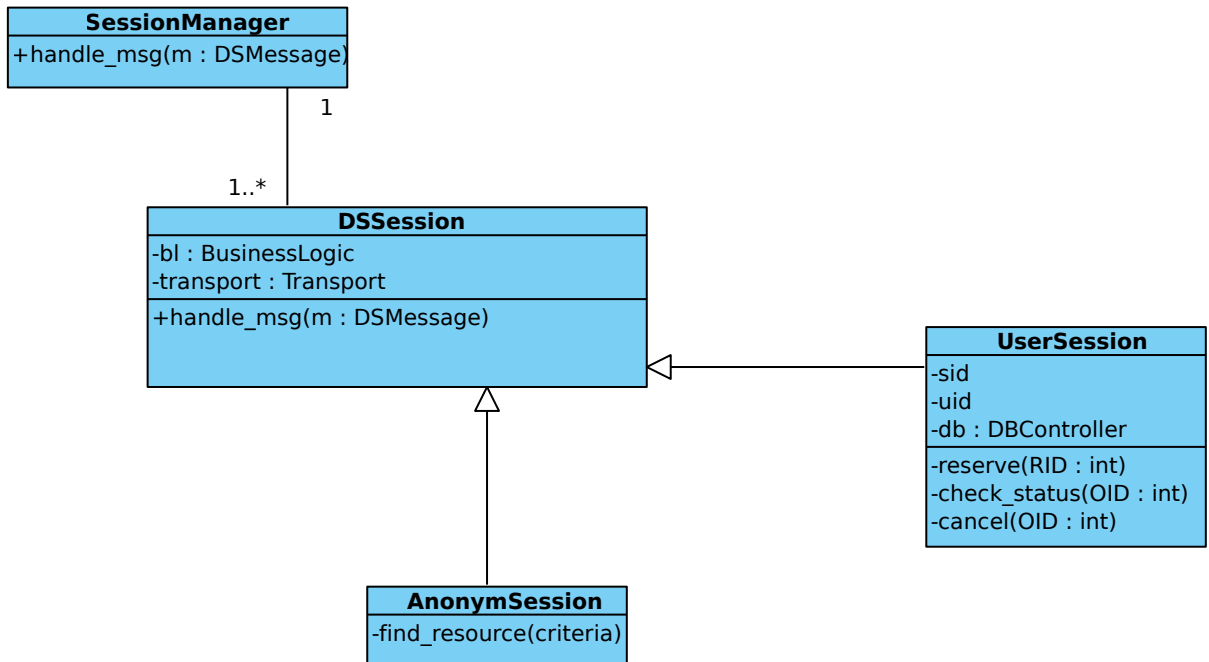


Рисунок 3.3 — Диаграмма классов для менеджера сессий

3.1.4 Бизнес-логика

На Рис. 3.4 представлена диаграмма классов бизнес-логики.

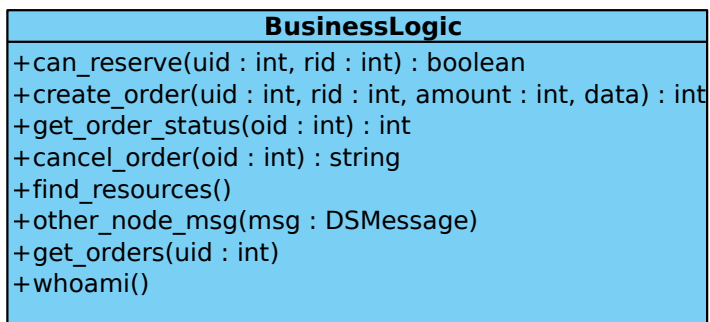


Рисунок 3.4 — Диаграмма классов для бизнес-логики

3.1.5 Контроллер БД

На Рис. 3.5 представлена диаграмма классов контроллера БД. В качестве механизма доступа к БД был использован ORM-паттерн ActiveRecord для Ruby.

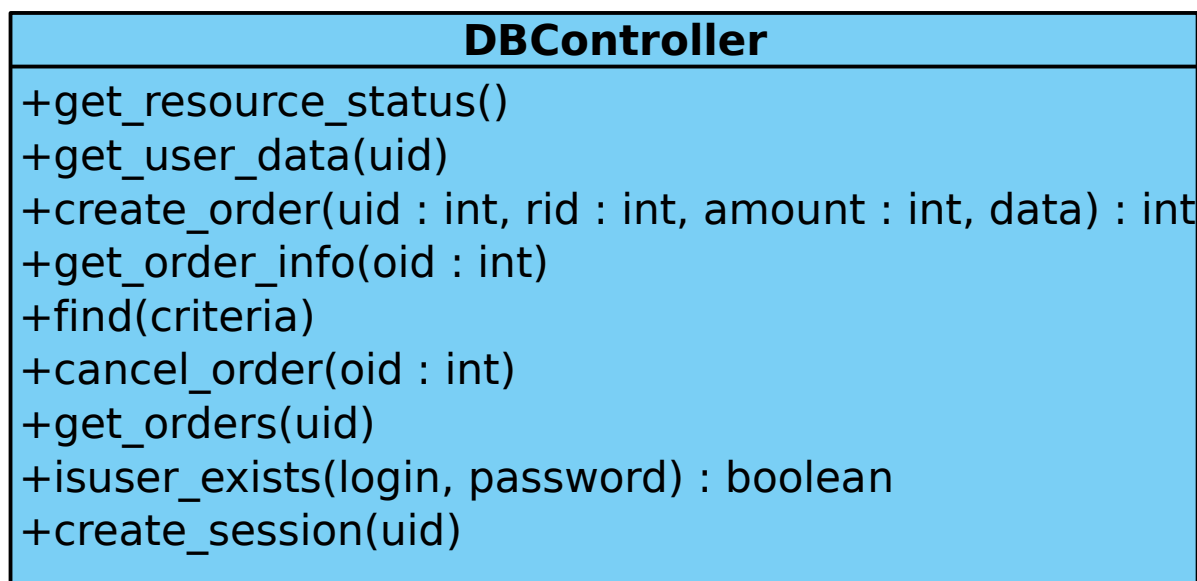


Рисунок 3.5 — Диаграмма классов для контроллера БД

3.2 Язык программирования и среда разработки

В качестве языка программирования был выбран Ruby 1.8.7. Этот язык является переносимым и имеет исчерпывающую библиотеку стандартных функций и классов, значительно ускоряющих разработку приложений.

Интерфейс пользователя для систем поиска и бронирования автомобилей был построен с использованием фрейворка Ruby On Rails. В качестве среды разработки была использована Netbeans по следующим причинам:

- данная среда разработки является бесплатным, свободно распространяющимся программным обеспечением;
- кроссплатформенность
- необходимый набор инструментов для создания и отладки Ruby-приложений;
- поддержка фреймворка Ruby On Rails.

3.3 Протоколы взаимодействия

3.3.1 Протокол синхронного взаимодействия

Для реализации синхронного взаимодействия между системами был использован протокол XML-RPC.

3.3.2 Протокол асинхронного взаимодействия

Для реализации асинхронного взаимодействия был использован протокол XMPP по следующим причинам:

- желание автора поближе познакомиться с протоколом;
- наличие стандартных классов для работы с этими протоколами в языке Ruby;
- простота отладки механизмов взаимодействия.

Таким образом, для реализации взаимодействия узлов необходим Jabber-сервер.

3.4 Сборка и запуск системы

В качестве системы сборки использовался Rake по следующим причинам:

- простота использования и написания файлов сборки;
- в качестве языка скрипта сборки используется язык Ruby.

Файл сборки имеет следующие цели:

- prepare - подготавливает конфигурационные файлы для каждого узла системы и генерирует тестовые базы данных для них;
- runnodes - запускает узлы РСОИ
- test - проводит юнит-тестирование системы

Конфигурационный файл узла представляет собой YAML-файл и имеет следующий формат:

```
—
sm_config:
  session_lifetime: времяжизнисессии
bl_config:
  whoami: имяузла
transport_config:
  asynconfig:
    friends:
      — имяузла 1
      — имяузла 2
      ...
    host: jabber сервер—
    password: пароль jabber аккаунта—
    user: имяпользователя jabber
  synconfig:
    port: порт XML-RPC
db_config:
  dbname: названиеБД
  username: имяпользователядлядоступаБД
  host: серверБД
  password: пароль
```

3.5 Тестирование системы

Было проведено модульное, интеграционное и нагрузочное тестирования системы бронирования. Результаты тестирования находятся в Приложении А.

Заключение

Результатом данной курсовой работы является разработанная распределенная система заказа туристических путевок, позволяющая находить авиабилеты в различных транспортных компаниях и свободные гостиничные номера с последующим их бронированием.

Разработанная система состоит из трех подсистем - система заказа путевки, система бронирования авиабилетов и система бронирования гостиничных номеров.

Каждая подсистема является независимой и может быть использована не только как узел распределенной системы, но и как программное обеспечение для организации деятельности компании и ведения учета.

В качестве развития проекта можно предложить добавление узла платежной системы для реализации функционала по покупке билетов и выкупу гостиничных номеров.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. xmpp.org/rfcs/
2. Sam Ruby, Dave Thomas, David Heinemeier Hansson. Agile Web Development with Rails. — USA.: 2009 — 760 с.
3. Оби Фернандес. Путь Rails. — М.: 2009 — 768 с.
4. <http://ruby-doc.org/>
5. <http://wiki.rubyonrails.org/>
6. <http://api.rubyonrails.org>

Приложение

Приложение А. Тестирование

Базовое тестирование

Тестирование подклассов класса AnonymMsgData

Таблица 3.1

Название теста	test_find_msg
Тестируемые методы	FindRequestMsg::type
Описание теста	Тестирование корректности определения типа сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение типа сообщения совпадает со значением, заданным в файле
Результат теста	Тест пройден

Таблица 3.2

Название теста	test_find_results_msg
Тестируемые методы	FindReplyMsg::type
Описание теста	Тестирование корректности определения типа сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение типа сообщения совпадает со значением, заданным в файле
Результат теста	Тест пройден

Таблица 3.3

Название теста	test_login_msg
----------------	----------------

Продолжение на след. стр.

Продолжение таблицы

Тестируемые методы	LoginRequestMsg::type
Описание теста	Тестирование корректности определения типа сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение типа сообщения совпадает со значением, заданным в файле
Результат теста	Тест пройден

Таблица 3.4

Название теста	test_login_reply_msg
Тестируемые методы	LoginReplyMsg::type
Описание теста	Тестирование корректности определения типа сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение типа сообщения совпадает со значением, заданным в файле
Результат теста	Тест пройден

Таблица 3.5

Название теста	test_success_reply_msg
Тестируемые методы	SuccessReplyMsg::type
Описание теста	Тестирование корректности определения типа сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение типа сообщения совпадает со значением, заданным в файле
Результат теста	Тест пройден

Таблица 3.6

Название теста	test_fail_reply_msg
Тестируемые методы	FailReplyMsg::type
Описание теста	Тестирование корректности определения типа сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение типа сообщения совпадает со значением, заданным в файле
Результат теста	Тест пройден

Тестирование подклассов класса AuthenticatedMsgData

Таблица 3.7

Название теста	test_auth_msg
Тестируемые методы	AuthenticatedMsgData::new, AuthenticatedMsgData::sessionID
Описание теста	Тестирование корректности создания нового сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение сессии должно совпадать со значением, переданным в конструкторе
Результат теста	Тест пройден

Таблица 3.8

Название теста	test_reserve_msg
----------------	------------------

Продолжение на след. стр.

Продолжение таблицы

Тестируемые методы	ReserveMsgData::new, ReserveMsgData::sessionID
Описание теста	Тестирование корректности создания нового сообщения
Степень важности	Фатальный
Ожидаемый результат	Полученное значение сессии должно совпадать со значением, переданным в конструкторе
Результат теста	Тест пройден

Интеграционное тестирование всей системы бронирования (BookingSystem)

Таблица 3.9

Название теста	test_1_login
Тестируемые методы	BookingSystem::start
Описание теста	Тестирование аутентификации клиента
Степень важности	Высокий
Ожидаемый результат	Получено сообщение об успешной аутентификации пользователя и идентификаторе сессии
Результат теста	Тест пройден

Таблица 3.10

Название теста	test_2_login_login
Тестируемые методы	BookingSystem::start
Описание теста	Тестирование некорректной последовательности сообщений

Продолжение на след. стр.

Продолжение таблицы

Степень важности	Высокий
Ожидаемый результат	Получено сообщение о том, что такая последовательность сообщений не предусмотрена протоколом
Результат теста	Тест пройден

Таблица 3.11

Название теста	test_3_session_expired
Тестируемые методы	BookingSystem::start
Описание теста	Тестирование превышения срока жизни сессии
Степень важности	Высокий
Ожидаемый результат	Получено сообщение о том, что сессия истекла и следует аутентифицироваться заново
Результат теста	Тест пройден

Таблица 3.12

Название теста	test_4_reserve
Тестируемые методы	BookingSystem::start
Описание теста	Тестирование резервирования ресурсов
Степень важности	Высокий
Ожидаемый результат	Получено сообщение с номером созданного заказа
Результат теста	Тест пройден

Таблица 3.13

Название теста	test_6_reservation_cancel
Тестируемые методы	BookingSystem::start
Описание теста	Тестирование отмены заказа
Степень важности	Высокий
Ожидаемый результат	Получено сообщение об успешной отмене заказа
Результат теста	Тест пройден

Интеграционное тестирование бизнес-логики (BusinessLogic)

Таблица 3.14

Название теста	test_2_create_order
Тестируемые методы	BusinessLogic::create_order
Описание теста	Тестирование создания заказа
Степень важности	Высокий
Ожидаемый результат	Создан новый заказ и сохранен в базе
Результат теста	Тест пройден

Таблица 3.15

Название теста	test_3_get_order_status
Тестируемые методы	BusinessLogic::get_order_status
Описание теста	Тестирование получения статуса заказа
Степень важности	Высокий
Ожидаемый результат	Получен статус заказа, если такой заказ существует в базе и принадлежит пользователю
Результат теста	Тест пройден

Таблица 3.16

Название теста	test_4_cancel_order
Тестируемые методы	BusinessLogic::cancel_order
Описание теста	Тестирование отмены заказа
Степень важности	Высокий
Ожидаемый результат	Отмена заказа возможна, если такой заказ существует в базе
Результат теста	Тест пройден

Таблица 3.17

Название теста	test_5_get_orders
Тестируемые методы	BusinessLogic::get_orders
Описание теста	Тестирование получения всех заказов клиента
Степень важности	Высокий
Ожидаемый результат	Получены все заказы клиента из базы
Результат теста	Тест пройден

Тестирование класса DBController

Таблица 3.18

Название теста	test_2reservation
Тестируемые методы	DBController::connect, DBController::disconnect
Описание теста	Тестирование создания нового заказа
Степень важности	Высокий

Продолжение на след. стр.

Продолжение таблицы

Ожидаемый результат	Ресурсы, указанные в запросе на резервацию получили статус "зарезервирован"
Результат теста	Тест пройден

Таблица 3.19

Название теста	test_3cancellation
Тестируемые методы	DBController::connect, DBController::disconnect
Описание теста	Тестирование отмены заказа
Степень важности	Высокий
Ожидаемый результат	Зарезервированные ресурсы получили статус "свободен"
Результат теста	Тест пройден

Таблица 3.20

Название теста	test_4payment
Тестируемые методы	DBController::connect, DBController::disconnect
Описание теста	Тестирование оплаты заказа
Степень важности	Высокий
Ожидаемый результат	Зарезервированные ресурсы получили статус "продан"
Результат теста	Тест пройден

Таблица 3.21

Название теста	test_5reserve_to
Тестируемые методы	DBController::connect, DBController::disconnect
Описание теста	Тестирование превышения срока брони заказа
Степень важности	Высокий
Ожидаемый результат	Зарезервированные ресурсы получили статус "свободен". Заказ получил статус "не гарантирован"
Результат теста	Тест пройден

Таблица 3.22

Название теста	test_7_user_exists
Тестируемые методы	DBController::connect, DBController::disconnect
Описание теста	Тестирование существования пользователя
Степень важности	Высокий
Ожидаемый результат	Пользователь существует, если имя пользователя и пароль совпадают со входными данными
Результат теста	Тест пройден

Тестирование класса DSMessage

Таблица 3.23

Название теста	test_serialization
Тестируемые методы	DSMessage::pack, DSMessage::new_from_pack

Продолжение на след. стр.

Продолжение таблицы

Описание теста	
Степень важности	Фатальный
Ожидаемый результат	Данные сообщения, полученного из упакованного формата совпадают с исходным
Результат теста	Тест пройден

Интеграционное тестирование менеджера сессий (SessionManager)

Таблица 3.24

Название теста	test_1start_stop
Тестируемые методы	SessionManager::start, SessionManager::stop
Описание теста	Тестирование запуска и остановки менеджера сессий
Степень важности	Средний
Ожидаемый результат	Корректный запуск и остановка
Результат теста	Тест пройден

Таблица 3.25

Название теста	test_2login
Тестируемые методы	SessionManager::handle_msg
Описание теста	Тестирование обработки сообщения аутентификации и создания новой сессии в случае успешной аутентификации
Степень важности	Высокий
Ожидаемый результат	Создана сессия для аутентифицированного клиента и получено сообщение с идентификатором сессии

Продолжение на след. стр.

Продолжение таблицы

Результат теста	Тест пройден
-----------------	--------------

Таблица 3.26

Название теста	test_3login_failure
Тестируемые методы	SessionManager::handle_msg
Описание теста	Тестирование обработки сообщения аутентификации в случае неправильного указания имени пользователя или пароля
Степень важности	Высокий
Ожидаемый результат	Получено сообщение об ошибке аутентификации
Результат теста	Тест пройден

Таблица 3.27

Название теста	test_4msg_handling
Тестируемые методы	SessionManager::handle_msg
Описание теста	Тестирование алгоритма обработки сообщений
Степень важности	Высокий
Ожидаемый результат	Принятое сообщение корректно перенаправляется в обработчик сообщений клиента
Результат теста	Тест пройден

Таблица 3.28

Название теста	test_5session_timeout
----------------	-----------------------

Продолжение на след. стр.

Продолжение таблицы

Тестируемые методы	SessionManager::watch_sessions
Описание теста	Тестирование алгоритма слежения за сроком жизни сессий
Степень важности	Высокий
Ожидаемый результат	Получено сообщение об истекшей сессии
Результат теста	Тест пройден

Таблица 3.29

Название теста	test_6login_and_login
Тестируемые методы	SessionManager::handle_msg
Описание теста	Тестирование некорректной последовательности сообщений
Степень важности	Высокий
Ожидаемый результат	Получено сообщение о том, что сообщение не предусмотрено протоколом
Результат теста	Тест пройден

Тестирование подклассов класса Session (обработчик сообщений клиентов)

Таблица 3.30

Название теста	test_0creation
Тестируемые методы	UserSession::new
Описание теста	Тестирование создания клиентской сессии
Степень важности	Высокий
Ожидаемый результат	Сессия корректно создана
Результат теста	Тест пройден

Таблица 3.31

Название теста	test_lreserve
Тестируемые методы	UserSession::handle_msg
Описание теста	Тестирование обработки сообщения о резервировании ресурсов
Степень важности	Высокий
Ожидаемый результат	Получен ответ об успешной резервации ресурсов
Результат теста	Тест пройден

Таблица 3.32

Название теста	test_badreservation
Тестируемые методы	UserSession::handle_msg
Описание теста	Тестирование обработки сообщения о резервировании ресурсов, если в резервировании отказано
Степень важности	Высокий
Ожидаемый результат	Получен ответ о неудачно попытке резервации ресурсов
Результат теста	Тест пройден

Таблица 3.33

Название теста	test_2cancel
Тестируемые методы	UserSession::handle_msg
Описание теста	Тестирование обработки сообщения об отмене заказа

Продолжение на след. стр.

Продолжение таблицы

Степень важности	Высокий
Ожидаемый результат	Получен ответ об успешном отмене заказа
Результат теста	Тест пройден

Таблица 3.34

Название теста	test_badcancel
Тестируемые методы	UserSession::handle_msg
Описание теста	Тестирование обработки сообщения об отмене заказа, если отменить заказ невозможно
Степень важности	Высокий
Ожидаемый результат	Получен ответ о неудачной попытке отмены заказа
Результат теста	Тест пройден

Таблица 3.35

Название теста	test_3get_orders
Тестируемые методы	UserSession::handle_msg
Описание теста	Тестирование обработки сообщения получения заказов клиента
Степень важности	Высокий
Ожидаемый результат	Получены заказы клиента
Результат теста	Тест пройден

Таблица 3.36

Название теста	test_4get_order_status
----------------	------------------------

Продолжение на след. стр.

Продолжение таблицы

Тестируемые методы	UserSession::handle_msg
Описание теста	Тестирование обработки сообщения получения состояния заказа
Степень важности	Средний
Ожидаемый результат	Получен статус заказа
Результат теста	Тест пройден

Таблица 3.37

Название теста	test_5find
Тестируемые методы	AnonymSession::handle_msg
Описание теста	Тестирование обработки сообщения поиска доступных ресурсов
Степень важности	Фатальный
Ожидаемый результат	Получен список доступных ресурсов
Результат теста	Тест пройден

Таблица 3.38

Название теста	test_6wrong_msgtypes
Тестируемые методы	UserSession::handle_msg, AnonymSession::handle_msg
Описание теста	Тестирование обработки некорректного сообщения
Степень важности	Средний
Ожидаемый результат	Система проигнорирует обработку некорректно составленное сообщение и сделает запись в лог-файл о некорректном сообщении
Результат теста	Тест пройден

Интеграционное тестирование транспортной подсистемы (Transport)

Таблица 3.39

Название теста	test_start_stop
Тестируемые методы	Transport::new, Transport::start, Transport::stop
Описание теста	Тестирование запуска транспортной подсистемы и ее остановки
Степень важности	Высокий
Ожидаемый результат	Транспортная система корректно запускает механизмы обмена сообщениями через XMPP и XML-RPC и останавливает их
Результат теста	Тест пройден

Таблица 3.40

Название теста	test_send_syn_msg
Тестируемые методы	Transport::send_syn_msg
Описание теста	Тестирование отправки синхронного сообщения (через механизм XML-RPC)
Степень важности	Фатальный
Ожидаемый результат	Отправленное сообщение корректно передано
Результат теста	Тест пройден

Таблица 3.41

Продолжение на след. стр.

Продолжение таблицы

Название теста	test_send_async_msg
Тестируемые методы	Transport::send_async_msg
Описание теста	Тестирование отправки асинхронного сообщения (через механизм XMPP)
Степень важности	Высокий
Ожидаемый результат	Отправленное сообщение корректно передано
Результат теста	Тест пройден

Таблица 3.42

Название теста	test_uncorrect_config
Тестируемые методы	Transport::new
Описание теста	Тестирование запуска подсистемы с некорректной конфигурацией
Степень важности	Высокий
Ожидаемый результат	Система выдает ошибку "Неправильная конфигурация"
Результат теста	Тест пройден

Тестирование механизма обменом сообщениями через XML-RPC (XMLRPC_iface)

Таблица 3.43

Название теста	test_1start_stop
Тестируемые методы	XMLRPC_iface::new, XMLRPC::start, XMLRPC::stop

Продолжение на след. стр.

Продолжение таблицы

Описание теста	Тестирование запуска механизма и его остановки
Степень важности	Высокий
Ожидаемый результат	Корректный запуск и остановка механизма
Результат теста	Тест пройден

Таблица 3.44

Название теста	test_2receive_msg
Тестируемые методы	XMLRPC_iface::new_msg
Описание теста	Тестирование приема сообщений от удаленных клиентов
Степень важности	Фатальный
Ожидаемый результат	Отправленные данные сообщения корректно передались
Результат теста	Тест пройден

Таблица 3.45

Название теста	test_3send_msg
Тестируемые методы	XMLRPC_iface::send_msg
Описание теста	Тестирование отправки сообщений
Степень важности	Фатальный
Ожидаемый результат	Отправленное сообщение корректно передавалось
Результат теста	Тест пройден

Тестирование механизма обмена сообщениями через XMPP (XMPP_iface)

Таблица 3.46

Название теста	test_aregister
Тестируемые методы	XMPP_iface::new, XMPP_iface::start, XMPP_iface::stop
Описание теста	Тестирование запуска xmpp интерфейса (с регистрацией нового пользователя на jabber-сервере) и его остановки
Степень важности	Высокий
Ожидаемый результат	Создан новый аккаунт на jabber-сервере
Результат теста	Тест пройден

Таблица 3.47

Название теста	test_msg_receive
Тестируемые методы	XMPP_iface::send
Описание теста	Тестирование механизма отправки сообщений через xmpp
Степень важности	Фатальный
Ожидаемый результат	Данные сообщения, принятые адресатом совпадают с исходным отправленным сообщением
Результат теста	Тест пройден

Тестирование пользовательских функций

Таблица 3.48

Продолжение на след. стр.

Продолжение таблицы

Прецедент	Описание класса эквивалентности	Состояние системы	Пример входных данных
Аутентификация пользователя	Известные системе имя пользователя и пароль	Аутентификация прошла успешно	Имя пользователя: tester, Пароль: secret
Аутентификация пользователя	Неизвестные системе имя пользователя и пароль	Сообщение об ошибке аутентификации	Имя пользователя: unknown, Пароль: secret
Бронирование ресурсов	Система доверяет пользователю	Бронирование прошло успешно, выслан номер заказа	Ресурс: авиабилеты на рейс Москва-Лондон, 31 мая 2010
Бронирование ресурсов	Система недоверяет пользователю	Сообщение о невозможности забронировать ресурс	Ресурс: авиабилеты на рейс Москва-Лондон, 31 мая 2010
Бронирование ресурсов	Система доверяет пользователю, но свободных ресурсов нет	Сообщение о невозможности забронировать ресурс	Ресурс: авиабилеты на рейс Москва-Лондон, 26 мая 2010
Проверка статуса заказа	Пользователь сформировал заказ	Сообщение о статусе заказа	Номер заказа: 210

Продолжение на след. стр.

Продолжение таблицы

Отмена заказа	Пользователь сформировал заказ, заказ можно отменить	Сообщение о результате операции	Номер заказа: 210
Отмена заказа	Пользователь сформировал заказ, заказ уже нельзя отменить	Сообщение о неудачной попытке отмены заказа	Номер заказа: 210

Тестирование функциональных возможностей

Таблица 3.49

Описание теста	Авторизация с именем пользователя и паролем известными системе
Входные данные	Имя пользователя: tester, пароль: secret
Ожидаемые результаты	Система присылает сообщение с номером сессии клиента
Реальные результаты	Получено сообщение с номером сессии
Результат теста	Тест пройден

Таблица 3.50

Описание теста	Авторизация с именем пользователя и паролем неизвестными системе
Входные данные	Имя пользователя: tester1, пароль: secret
Ожидаемые результаты	Система присылает сообщение о неудачной аутентификации
Реальные результаты	Получено сообщение "Неудачная аутентификация. Проверьте имя пользователя/пароль"

Продолжение на след. стр.

Продолжение таблицы

Результат теста	Тест пройден
-----------------	--------------

Таблица 3.51

Описание теста	Бронирование ресурса, система доверяет пользователю
Входные данные	Ресурс: авиабилеты на рейс Москва-Лондон, 31 мая 2010
Ожидаемые результаты	Система присылает сообщение с номером сформированного заказа
Реальные результаты	Получено сообщение с номером заказа - 210
Результат теста	Тест пройден

Таблица 3.52

Описание теста	Бронирование ресурса, система не доверяет пользователю
Входные данные	Ресурс: авиабилеты на рейс Москва-Лондон, 31 мая 2010
Ожидаемые результаты	Система присылает сообщение с отказом в бронировании
Реальные результаты	Получено сообщение с отказом брони
Результат теста	Тест пройден

Таблица 3.53

Описание теста	Бронирование ресурса, система доверяет пользователю, но ресурс уже забронирован
----------------	---

Продолжение на след. стр.

Продолжение таблицы

Входные данные	Ресурс: авиабилеты на рейс Москва-Лондон, 31 мая 2010
Ожидаемые результаты	Система присылает сообщение с отказом в бронировании
Реальные результаты	Получено сообщение с отказом брони
Результат теста	Тест пройден

Таблица 3.54

Описание теста	Проверка статуса заказа
Входные данные	Номер заказа: 210
Ожидаемые результаты	Система присылает сообщение со статусом заказа
Реальные результаты	Получено сообщение со статусом заказа: "ожидается оплата"
Результат теста	Тест пройден

Таблица 3.55

Описание теста	Отмена заказа, отмена возможна
Входные данные	Номер заказа: 210
Ожидаемые результаты	Система присылает сообщение об успешной отмене заказа
Реальные результаты	Получено сообщение: "заказ отменен"
Результат теста	Тест пройден

Таблица 3.56

Описание теста	Отмена заказа, отмена невозможна
Входные данные	Номер заказа: 210
Ожидаемые результаты	Система присылает сообщение о неудачной попытке отмене заказа
Реальные результаты	Получено сообщение: "заказ нельзя отменить"
Результат теста	Тест пройден

Тестирование надежности и доступности

В рамках тестирования надежности и доступности проводилось нагрузочное тестирование. Введем следующие обозначения:

λ - интенсивность запросов

T - общее время тестирования

t_1 - среднее время бесперебойной работы

t_2 - время восстановления после сбоя

$\theta = t_1 / (t_1 + t_2) * 100\%$ - доступность сервера

p - вероятность отказа сервера

Описание теста	Нагрузочное тестирование
Входные данные	$\lambda = 10$, $T = 2000c$, Запрос: поиск свободных ресурсов
Ожидаемые результаты	$\theta \geq 99\%$, $p \leq 0,1\%$
Реальные результаты	$\theta = 99,1$, $p = 0.034$
Результат теста	Тест пройден