



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**РАСПРЕДЕЛЁННАЯ СИСТЕМА БРОНИРОВАНИЯ НОМЕРОВ
ГОСТИНИЦ СЕТИ APARTLUX**

Техническое задание

Листов 10

Инв.№ подл.	Подп. и дата	Взаим.инв.№	Инв.№ дубл.	Подп. и дата

Введение

По данным «Анализа рынка гостиничных услуг в России», подготовленного агентством BusinesStat, занимающимся исследованием рынка Российской Федерации и всех стран бывшего СНГ, подобным сервисом в 2022 году воспользовались 62.4 млн чел, что на 63% превысило значение 2020 года (38.3 млн человек). Такой рост объясняется сокращением выездного туризма и развитием внутреннего на фоне геополитической обстановки. Также численность гостиничных учреждений к концу 2022 года достигла 22.01 тыс., в то время как в 2020 она составляла 20.41 тыс.

Ввиду сильной конкуренции для привлечения клиентов каждая компания стремится улучшить свой сервис, особое внимание уделяется системам бронирования.

Данное техническое задание составлено для разработки распределённой системы для бронирования номеров гостиниц сети Apartlux. Техническое задание выполнено на основе ГОСТ 19.201–78 «ЕСПД. Техническое задание. Требования к содержанию и оформлению».

Глоссарий

- 1) Узел системы – региональный сервер, содержащий данные авторов и читателей указанного региона.
- 2) «Горячее» переконфигурирование системы – способность системы применять изменения без перезапуска и перекompиляции.
- 3) Медиана времени отклика – среднее время предоставления данных пользователю.
- 4) Валидация данных – проверка данных на соответствие заданным условиям и ограничениям.
- 5) REST (Representational State Transfer) – архитектурный стиль взаимодействия

ствия компонентов распределённого приложения в сети.

- 6) Фронтенд – серверное приложение, принимающее запросы от пользователя. На каждый из типов запросов определяется, как организовать его выполнение. Принимает запросы, анализирует их и в соответствии с заложенным алгоритмом выполняет запросы к бекендам.
- 7) Бекенд – серверное приложение, выполняющее определенную задачу, например, взаимодействие с СУБД. Бекенды принимают запросы от фронтенда.
- 8) ПО – программное обеспечение.

Основания для разработки

Разработка ведётся в рамках выполнения лабораторных работ по курсу «Методология программной инженерии» на кафедре «Программное обеспечение ЭВМ и информационные технологии» факультета «Информатика и системы управления» МГТУ им. Н.Э. Баумана.

Назначение разработки

Разрабатываемая система должна предоставлять пользователям возможность бронирования номеров сети Apartlux, в которую входят 14 гостиниц в Москве на таких станциях метро, как Бабушкинская, Рижская, Тверская, Авиамоторная и другие. Должен быть предусмотрен поиск подходящих номеров по таким параметрам, как этаж, дата и продолжительность бронирования, стоимость, число мест, наличие двуспальной кровати. В зависимости от количества сделанных ранее заказов система должна рассчитывать скидку на новые согласно условиям программы лояльности.

Существующие аналоги

У сети Apartlux уже есть действующий с 2011 года сайт для бронирования, который имеет ряд недостатков. При его создании разработчики придерживались подходам монолитной архитектуры, поэтому сейчас компания столкнулась с такими трудностями, как:

- 1) проблематичность масштабирования;
- 2) сложность внедрения появившихся технологий, которые используются повсеместно;
- 3) внесение даже незначительного изменения в функциональность существенно усложняет и замедляет разработку.

В то время как на российском рынке гостиничных услуг появляется всё больше компаний, например, Radisson, Azimut Hotels, Hilton, остановивших свой выбор на микросервисной архитектуре и внедряющих современные технологии: PostgreSQL, MongoDB, Kafka, ELK stack и т.д. У приведённых сетей отелей и гостиниц есть общий недостаток: непрозрачная программа лояльности, которая направлена лишь на ограниченный круг лиц.

По сравнению с существующим сайтом и указанными аналогами разрабатываемый проект должен иметь следующие преимущества:

- 1) в основе должна лежать микросервисная архитектура, решающая сложности с масштабированием, обслуживанием и внесением изменений в функциональность;
- 2) понятная бонусная программа, ориентированная на каждого из клиентов.

Описание системы

Разрабатываемый сервис должен представлять собой распределённую систему для бронирования номеров гостиниц сети Apartlux. Если клиент хочет оформить бронь, ему необходимо зарегистрироваться, указав информацию: фа-

милия, имя, отчество, дата рождения, номер телефона, электронная почта. В случае, если зарегистрированному ранее пользователю нужно отменить заказ, получить информацию о его бронированиях или статусе в программе лояльности, ему нужно авторизоваться. Для неавторизованных пользователей доступен только просмотр общей информации. На рисунке 1.1 отображена схема предметной области.

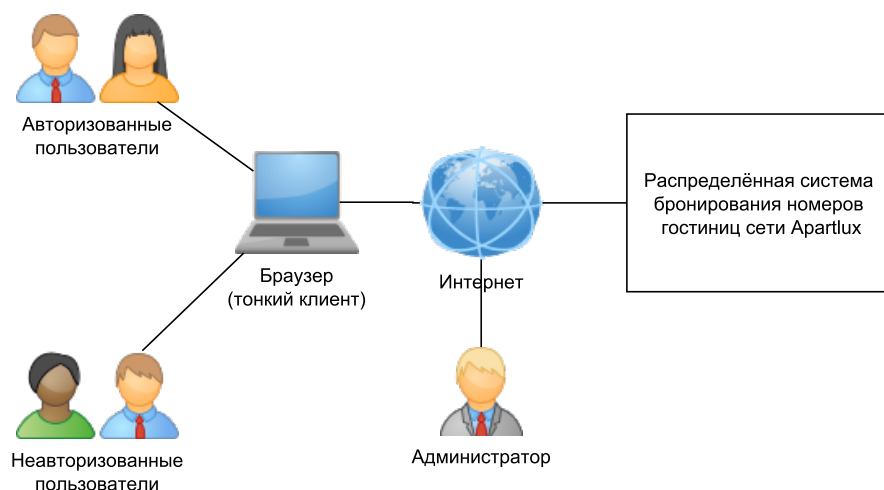


Рисунок 1.1 – Схема предметной области.

Общие требования к системе

Требования к системе следующие.

- 1) Разрабатываемое ПО должно поддерживать функционирование системы в режиме 24 часов, 7 дней в неделю, 365 дней в году (24/7/365) со среднегодовым временем доступности не менее 99.9%. Допустимое время, в течении которого система недоступна, за год должна составлять $24 \cdot 365 \cdot 0.001 = 8.76$ ч.
- 2) Время восстановления системы после сбоя не должно превышать 15 минут.
- 3) Каждый узел должен автоматически восстанавливаться после сбоя.
- 4) Система должна поддерживать возможность «горячего» переконфигурирования системы. Необходимо предусмотреть поддержку добавления нового узла во время работы системы без рестарта.
- 5) Обеспечить безопасность работоспособности за счёт отказоустойчивости узлов.

Требования к функциональным характеристикам

- 1) По результатам работы модуля сбора статистики медиана времени отклика системы на запросы пользователя на получение информации не должна превышать 3 секунд.
- 2) По результатам работы модуля сбора статистики медиана времени отклика системы на запросы, добавляющие или изменяющие информацию на портале не должна превышать 7 секунд.
- 3) Медиана времени отклика системы на действия пользователя должна быть менее 0.8 секунд при условии работы на рекомендованной аппаратной конфигурации, задержках между взаимодействующими сервисами менее 0.2 секунды и одновременном числе работающих пользователей менее

100 на каждый сервер, обслуживающий внешний интерфейс.

- 4) Система должна обеспечивать возможность запуска в современных браузерах: не менее 85% пользователей Интернета должны пользоваться ей без какой-либо деградации функционала.

Функциональные требования к системе с точки зрения пользователя

Система должна обеспечивать реализацию следующих функций.

- 1) Регистрация и авторизация пользователей с валидацией вводимых данных как через интерфейс приложения, так и через социальные сети.
- 2) Аутентификация пользователей.
- 3) Разделение всех пользователей на три роли:
 - Пользователь (неавторизованный пользователь);
 - Клиент (авторизованный пользователь);
 - Администратор.
- 4) Предоставление возможностей **Пользователю, Клиенту, Администратору** представленных в таблице 1.

Таблица 1 – Функции пользователей

Пользователь	Клиент	Администратор
1. просмотр списка гостиниц, входящих в сеть; 2. просмотр информации о возможности бронирования номера гостиницы по заданным реквизитам; 3. получение информации об условиях программы лояльности;		
	4. получение и изменение информации текущего аккаунта;	4. получение и редактирование информации о любом клиенте, зарегистрированном в системе;

Продолжение на следующей странице

Пользователь	Клиент	Администратор
	<p>5. просмотр всех бронирований, зарегистрированных на имя текущего клиента;</p> <p>6. получение детальной информации по конкретному бронированию текущего клиента;</p> <p>7. бронирование отеля на имя текущего клиента;</p> <p>8. отмена заказа, оформленного на имя текущего пользователя;</p> <p>9. получение информации о статусе текущего пользователя в программе лояльности.</p>	<p>5. просмотр и редактирование всех оформленных бронирований;</p> <p>6. получение детальной информации по конкретному бронированию;</p> <p>7. бронирование отеля на зарегистрированного в системе пользователя;</p> <p>8. отмена любого оформленного заказа;</p> <p>9. получение информации о статусе в программе лояльности любого зарегистрированного в системе клиента.</p>
		<p>10. изменение доступных дат для бронирования;</p> <p>11. редактирование информации об условиях программы лояльности.</p>

Входные данные

Входные параметры системы представлены в таблице 2.

Таблица 2 – Входные данные

Сущность	Входные данные
Клиент/Администратор	<ol style="list-style-type: none">1. <i>фамилия, имя и отчество</i> не более 256 символов каждое поле;2. <i>дата рождения</i> в формате д/м/гггг;3. <i>логин</i> не менее 10 символов и не более 128;4. <i>пароль</i> не менее 8 символов и не более 128, как минимум одна заглавная и одна строчная буква, только латинские буквы, без пробелов, как минимум одна цифра;5. <i>номер телефона</i>;6. <i>электронная почта</i>;
Гостиница	<ol style="list-style-type: none">1. <i>идентификатор</i>;2. <i>название</i> не более 256 символов;3. <i>страна</i>;4. <i>город</i>;5. <i>полный адрес</i> не более 1024 символа;6. <i>контактный телефон</i>;7. <i>электронная почта</i>;8. <i>описание</i> не более 2048 символов9. <i>количество звёзд</i>

Продолжение на следующей странице

Сущность	Входные данные
Номер	<ul style="list-style-type: none"> 1. <i>идентификатор</i>; 2. <i>идентификатор</i> соответствующей <i>гостиницы</i>; 3. <i>число мест</i>; 4. <i>этаж</i>; 5. <i>стоимость</i>; 6. <i>наличие двуспальной кровати</i>;
Бронирование	<ul style="list-style-type: none"> 1. <i>идентификатор</i>; 2. <i>логин клиента</i>, на которое оно оформлено; 3. <i>идентификатор гостиницы</i>; 4. <i>идентификатор</i> соответствующей <i>платёжной операции</i>; 5. <i>статус</i> (APPROVED/UNAPPROVED); 6. <i>дата въезда</i>; 7. <i>дата выезда</i>;
Оплата	<ul style="list-style-type: none"> 1. <i>идентификатор</i>; 2. <i>фамилия, имя, отчество</i> человека, совершившего операцию; 3. <i>статус</i> (PAID/CANCELED); 4. <i>сумма</i>; 5. <i>дата и время</i>.

Выходные параметры

Выходными параметрами системы являются web-страницы. В зависимости от запроса и текущей роли пользователя они содержат следующую информацию (таблица 3).

Таблица 3 – Выходные параметры

Пользователь	Клиент	Администратор
<p>1. список гостиниц, которые входят в сеть Apartlux, указывается:</p> <ul style="list-style-type: none"> • <i>название;</i> • <i>полный адрес;</i> • <i>описание;</i> • <i>контактная информация: телефон и электронная почта;</i> • <i>доступные для бронирования номера по заданным реквизитам;</i> 		
<p>2. информация об условиях текущей программы лояльности;</p>		
	<p>3. детальная информация о пользователе, вошедшем в систему;</p> <ul style="list-style-type: none"> • <i>фамилия, имя, отчество;</i> • <i>дата рождения;</i> • <i>логин;</i> • <i>номер телефона;</i> • <i>электронная почта;</i> • <i>размер скидки;</i> 	

Продолжение на следующей странице

Пользователь	Клиент	Администратор
	<p>4. список оформленных бронирований на пользователя, вошедшего в систему, предоставляется информация о:</p> <ul style="list-style-type: none"> • <i>дате въезда/выезда;</i> • <i>полный адрес</i> гостиницы и <i>контактные данные;</i> • информация о номере: <i>этаж, номер, количество мест;</i> • <i>статус оплаты;</i> • <i>статус бронирования;</i> • <i>сумма;</i> 	<p>4. список оформленных бронирований, предоставляются такие данные, как:</p> <ul style="list-style-type: none"> • <i>идентификатор;</i> • информация о клиенте, на которое оно оформлено: <i>логин, номер телефона, электронная почта;</i> • <i>дата въезда/выезда;</i> • информация о гостинице: <i>её идентификатор, полный адрес и контактные данные;</i> • информация о номере: <i>этаж, номер, количество мест;</i> • информация об оплате: <i>идентификатор, статус, ФИО оплатившего, дата и время операции;</i> • <i>статус бронирования;</i> • <i>сумма;</i>

Продолжение на следующей странице

Пользователь	Клиент	Администратор
		5. список зарегистрированных в системе клиентов с указанием <i>ФИО</i> , <i>логина</i> , <i>даты рождения</i> , <i>контактных данных</i> , <i>персональной скидки</i> , <i>число предыдущих бронирований</i> .

Топология Системы

На рисунке 1.2 изображён один из возможных вариантов топологии разрабатываемой распределенной Системы.

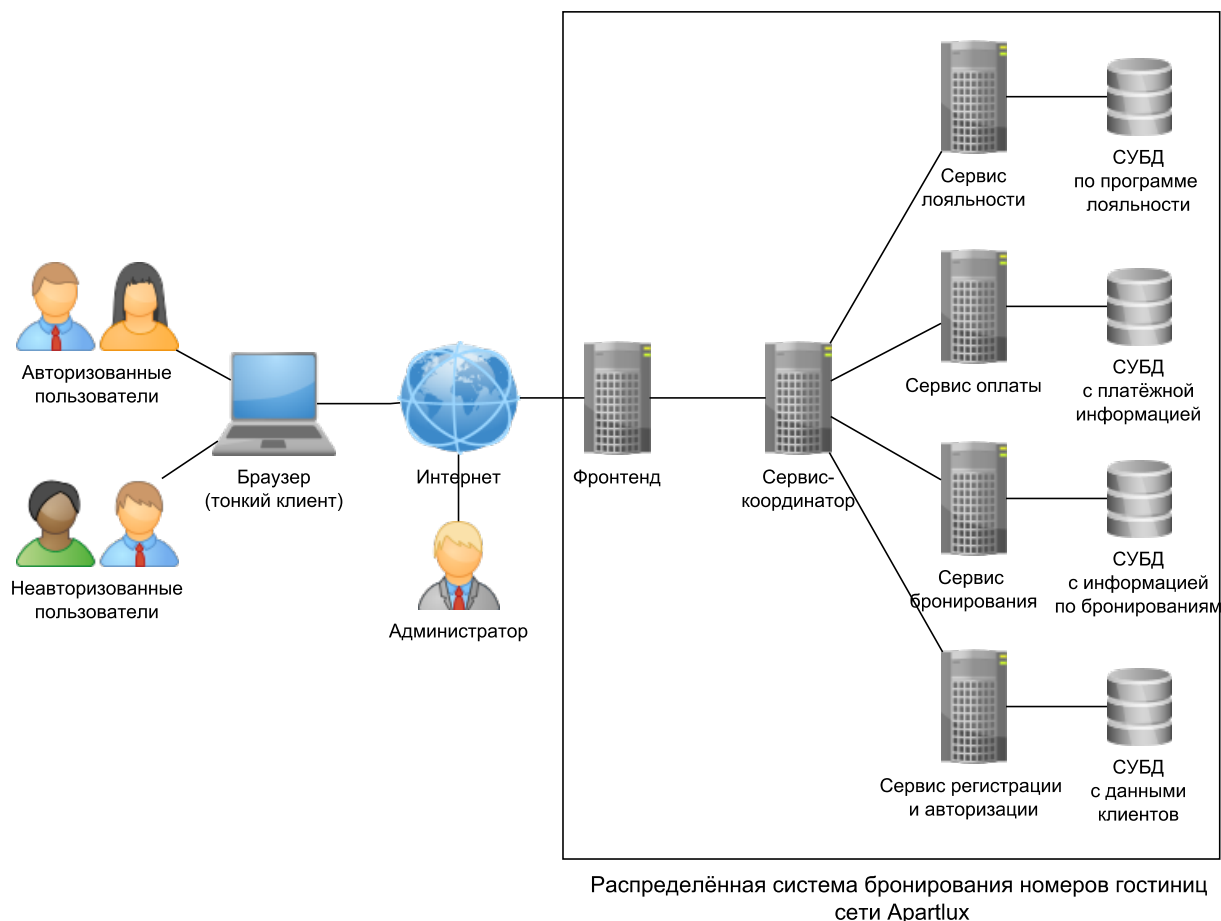


Рисунок 1.2 – Топология системы.

Система будет состоять из фронтенда и 5 подсистем:

- сервис-координатор;
- сервис регистрации и авторизации;
- сервис бронирования;
- сервис оплаты;
- сервис лояльности.

Фронтенд принимает запросы от пользователей по протоколу HTTP и анализирует их. На основе проведенного анализа выполняет запросы к микросервисам бекенда, агрегирует ответы и отправляет их пользователю.

Сервис-координатор – единая точка входа и межсервисной коммуникации.

Сервис-регистрации и авторизации отвечает за:

- возможность регистрации нового клиента;

- аутентификацию пользователя (клиента и администратора);
- авторизацию пользователя;
- выход из сессии.

Сервис бронирования реализует следующие функции:

- получение списка всех гостиниц, входящий в сеть Apartlux;
- получение информации о конкретной гостинице;
- получение, создание, отзыв бронирования.

Сервис оплаты реализует функции:

- проведение платежа от клиента к системе;
- получение статуса оплаты;
- отмену платежа.

Сервис лояльности отвечает за ведение статистики по количеству бронирований всех клиентов, на основе которой для каждого пользователя в индивидуальном порядке предоставляется скидка на будущие заказы.

Требования к программной реализации

- 1) Требуется использовать СОА (сервис-ориентированную архитектуру) для реализации системы.
- 2) Система состоит из микросервисов. Каждый микросервис отвечает за свою область логики работы приложения и должны быть запущены изолированно друг от друга.
- 3) При необходимости, каждый сервис имеет своё собственное хранилище, запросы между базами запрещены.
- 4) При разработке базы данных необходимо учитывать, что доступ к ней должен осуществляться по протоколу TCP.
- 5) Необходимо реализовать один web-интерфейс для фронтенда. Интерфейс должен быть доступен через тонкий клиент (браузер).
- 6) Для межсервисного взаимодействия использовать HTTP (придерживать-

ся RESTful).

- 7) Выделить Gateway Service как единую точку входа и межсервисной коммуникации. В системе не должно осуществляться горизонтальных запросов.
- 8) При недоступности систем портала должна осуществляться деградация функционала или выдача пользователю сообщения об ошибке.
- 9) Необходимо предусмотреть авторизацию пользователей, как через интерфейс приложения, так и через популярные социальные сети.
- 10) Валидацию входных данных необходимо проводить и на стороне пользователя, и на стороне фронтенда. Микросервисы бекенда не должны валидировать входные данные, поскольку пользователь не может к ним обращаться напрямую, они должны получать уже отфильтрованные входные данные.
- 11) Для запросов, выполняющих обновление данных на нескольких узлах распределенной системы, в случае недоступности одной из систем, необходимо выполнять полный откат транзакции.
- 12) Приложение должно поддерживать возможность горизонтального и вертикального масштабирования за счет увеличения количества функционирующих узлов и совершенствования технологий реализации компонентов и всей архитектуры системы.
- 13) Код хранить на Github, для сборки использовать Github Actions.
- 14) Gateway Service должен запускаться на порту 8080, остальные сервисы запускать на портах 8050, 8060, 8070.
- 15) Каждый сервис должен быть завернут в docker.

Функциональные требования к подсистемам

Подсистемы: фронтенд, бекенд-координатор, бекенд регистрации и авторизации, бекенд бронирования, бекенд оплаты, бекенд лояльности.

Фронтенд – серверное приложение, предоставляет пользовательский интерфейс и внешний API системы, при разработке которого нужно учитывать следующее:

- должен принимать запросы по протоколу HTTP и формировать ответы пользователям в формате HTML;
- в зависимости от типа запроса должен отправлять последовательные запросы в соответствующие микросервисы;
- запросы к микросервисам необходимо осуществлять по протоколу HTTP;
- данные необходимо передавать в формате JSON.

Сервис-координатор – серверное приложение, через которое проходит весь поток запросов и ответов, должен соответствовать следующим требованиям разработки:

- принимать и возвращать данные в формате JSON по протоколу HTTP;
- накапливать статистику запросов, в случае, если система не ответила N раз, то в N + 1 раз вместо запроса сразу отдавать fallback. Через некоторое время выполнить запрос к реальной системе, чтобы проверить её состояние;
- выполнять проверку существования клиента, также регистрацию и аутентификацию пользователей;
- получение списка всех гостиниц с возможными датами для бронирования и внесение изменений в перечень доступных дат (последнее только для администраторов);
- получение информации и обновление данных о зарегистрированном пользователе;
- оформление и отзыв созданного ранее бронирования;
- получение данных о бронированиях пользователя;
- получение статуса конкретного пользователя в программе лояльности и обновление её условий (последнее только для администраторов).

Сервис-регистрации и авторизации должен реализовывать следующие функ-

циональные возможности:

- принимать и возвращать данные в формате JSON по протоколу HTTP;
- возможность регистрации нового клиента и обновление данных уже существующего;
- проверка существования клиента;
- обеспечение авторизации пользователя через аккаунт как в системе, так и через предлагаемые социальные сети.

Хранимая в базе данных сущность, ассоциированная с сервисом, детально представлена в таблице 4.

Таблица 4 – Состав сущностей

Сущность	Поля	Обязательность
Аккаунт	<i>фамилия</i> , не более 256 символов	да
	<i>имя</i> , не более 256 символов	да
	<i>отчество</i> , не более 256 символов	нет
	<i>дата рождения</i>	да
	<i>логин</i> , является первичным ключом	да
	<i>захешированный пароль</i>	да
	<i>номер телефона</i>	да
	<i>электронная почта</i>	да

Сервис бронирования реализует следующие функции:

- получение и отправка данных в формате JSON по протоколу HTTP;
- получение таких данных, как:
 - список всех гостиниц, входящий в сеть Apartlux;
 - информация о конкретной гостинице по её идентификатору;
 - информация о свободных номерах по заданным реквизитам;
 - информация о конкретном номере по его идентификатору и идентификатору гостиницы, в которой он расположен;

- все бронирования, зарегистрированные на конкретного клиента;
- информация о конкретном бронировании по его идентификатору;
- вычисление стоимости бронирования за указанный период в выбранной гостинице за конкретный номер;
- создание, редактирование и отзыв бронирования.

Соответствующая база данных содержит три сущности, описание которых приведено в таблице 5.

Таблица 5 – Состав сущностей

Сущность	Поля	Обязательность
Гостиница	<i>идентификатор</i> , является первичным ключом	да
	<i>название</i> не превышает 256 символов	да
	<i>страна</i> не превышает 80 символов	да
	<i>город</i> не превышает 80 символов	да
	<i>адрес</i> не превышает 256 символов	да
	<i>контактный телефон</i>	да
	<i>электронная почта</i>	да
	<i>описание</i> не превышает 2048 символов	нет
	<i>количество звёзд</i> , по умолчанию 0	нет
Номер	<i>идентификатор</i> , является первичным ключом	да
	<i>идентификатор</i> гостиницы	да
	<i>число мест</i>	да
	<i>этаж</i>	да
	<i>стоимость</i>	да
	<i>признак наличия двуспальной кровати</i>	да

Продолжение на следующей странице

Сущность	Поля	Обязательность
Бронирование	<i>идентификатор</i> , является первичным ключом	да
	<i>логин клиента</i>	да
	<i>идентификатор платёжной операции</i>	да
	<i>идентификатор гостиницы</i>	да
	<i>статус</i> , APPROVED/UNAPPROVED, по умолчанию UNAPPROVED	да
	<i>дата въезда</i>	да
	<i>дата выезда</i>	да

Сервис оплаты реализует функции:

- получение и отправка данных в формате JSON по протоколу HTTP;
- предоставления информации об оплате по её идентификатору;
- проведения оплаты и её отмена;
- получения и обновления статуса оплаты.

Ассоциированная с этим сервисом база данных содержит сущность, детально представленная в таблице 6.

Таблица 6 – Состав сущностей

Сущность	Поля	Обязательность
Платёж	<i>идентификатор</i> , является первичным ключом	да
	<i>статус</i> , PAID/CANCELED	да
	<i>цена</i>	да
	<i>дата</i>	да

Сервис лояльности должен реализовывать представленные такие функциональные возможности, как:

- получение и отправка ответов на запросы в формате JSON по протоколу HTTP;
- получение величины скидки по конкретному пользователю;
- получение детальной информации о конкретном участнике программы лояльности;
- обновление числа бронирований и статуса по программе лояльности (предусмотреть, как повышение, так и понижение в случае отмены бронирования);
- внесение изменений в размер скидки по конкретному пользователю.

Соответствующая сущность базы данных имеет поля, представленные в таблице 7.

Таблица 7 – Состав сущностей

Сущность	Поля	Обязательность
Карта лояльности	<i>идентификатор</i> , является первичным ключом	да
	<i>логин клиента</i>	да
	<i>количество оформленных ранее заказов</i> , по умолчанию 0	да
	<i>статус</i> , BRONZE/SILVER/GOLD, по умолчанию BRONZE	да
	<i>скидка</i> , по умолчанию 10	да

Требования к составу и параметрам технических средств

Все серверные приложения должны потреблять суммарно не более 2 Гбайт оперативной памяти и работать на сервере с процессором Intel(R) Core(TM) i7-10510U CPU 1.80GHz.

Требования к надёжности

Система должна работать в соответствии с данным техническим заданием без рестарта. Необходимо использовать «зеркалируемые серверы» для всех подсистем, которые будут держать нагрузку в случае сбоя до тех пор, пока основной сервер не восстановится.

Требования к документации

Исполнитель должен подготовить и передать заказчику руководство:

- для Администратора Системы;
- для Пользователя Системы;
- для Клиента Системы;
- по развёртыванию Системы.

Сценарии функционирования системы

Регистрация клиента

- 1) Пользователь нажимает на кнопку «Зарегистрироваться» в интерфейсе.
- 2) Пользователь перенаправляется на страницу, которая содержит поля для заполнения его данных.
- 3) Пользователь вводит данные в форму и для завершения регистрации нажимает на кнопку «Готово», тем самым подтверждая верность своих данных, а также согласие на их обработку и хранение.
- 4) Если пользователь с введенным для регистрации логином уже существует, то клиент перенаправляется на страницу ошибки. При успешной регистрации клиент попадает на страницу своего профиля в системе.

Авторизация клиента

- 1) Пользователь нажимает на кнопку «Войти» в интерфейсе.

- 2) Пользователь перенаправляется на страницу авторизации, которая содержит поля для заполнения логина и пароля.
- 3) Пользователь завершает работу с формой авторизации нажатием кнопки «Готово».
- 4) При обнаружении ошибки в данных, пользователь перенаправляется на страницу ошибки; при совпадении данных с записью в базе данных аккаунтов пользователь получает доступ к системе.

Бронирование номера

- 1) Клиент нажимает кнопку «Бронирование».
- 2) Клиент перенаправляется на страницу, которая содержит список гостиниц.
- 3) Клиент нажимает на понравившуюся позицию и попадает на страницу доступных для бронирования номеров в выбранной гостинице с разными реквизитами.
- 4) При необходимости выставляет необходимые параметры фильтров (например, адрес, планировка, диапазон цен и дат), нажимает кнопку «Применить». После этого список обновляется, сверху находятся предложения, наиболее соответствующие желанию клиента.
- 5) Клиент нажимает кнопку «Оформить бронирование» напротив нужного номера, на экране появляется всплывающее окно, дублирующее его атрибуты.
- 6) Клиент нажимает кнопку «Готово», выражая своё согласие на оформление бронирования, и перенаправляется на страницу оплаты, где вводит реквизиты карты и код подтверждения. В случае успешной операции придёт смс- и email-оповещения.
- 7) Если клиент не хочет оформлять бронь, он нажимает на кнопку выхода – крестик, всплывающее окно пропадает.