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

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

2023

Введение

По данным «Анализа рынка гостиничных услуг в России», подготовленного агенством 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 перекомпиляции.
3) Медиана времени отклика - среднее время предоставления данных пользователю.
4) Валидация - проверка данных на соответствие заданным условиям и ограничениям.
5) REST (Representational State Transfer) – архитектурный стиль взаимодей-

ствия компонентов распределённого приложения в сети.

6) 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, указ информацию: фамилия, 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 (неавторизированный пользователь);
    - Клиент (авторизированный 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; 5. просмотр всех бронирований, зарегистрированных на имя текущего клиента; 6. получение детальной информации по конкретному бронированию текущего клиента; 7. бронирование отеля на имя текущего клиента; 8. отмена заказа, оформленного на имя текущего пользователя;	4. получение и редактирование информации о любом клиенте, зарегистрированном в системе; 5. просмотр и редактирование всех оформленных бронирований; 6. получение детальной информации по конкретному бронированию; 7. бронирование отеля на зарегистрированного в системе пользователя; 8. отмена любого оформленного заказа;

Продолжение на следующей странице

Пользователь	Клиент	Администратор
	9. получение информации о статусе текущего пользователя в программе лояльности.	9. получение информации о статусе в программе лояльности любого зарегистрированного в системе клиента.
		10. изменение доступных дат для бронирования; 11. редактирование информации об условиях программы лояльности.

Входные данные

Входные параметры системы зависят от функции, полный перечень представлен в таблице 2.

Пользователь Бронирование Гостинца

Таблица 2 – Входные данные

Сущность	Входные данные
Клиент/Администратор	1. фамилия, имя и отчество не более 256 символов; 2. дата рождения; 3. логин; 4. пароль; 5. номер телефона; 6. электронная почта;

Продолжение на следующей странице

Сущность	**Входные данные**
Гостиница	1. идентификатор; 2. название; 3. страна; 4. город; 5. полный адрес; 5. контактный телефон; 6. электронная почта; 7. описание 8. количество звёзд
Номер	1. идентификатор; 2. число мест; 3. этаж; 4. стоимость; 5. наличие двуспальной кровати;
Бронирование	1. идентификатор; 2. логин клиента, на которое оно оформлено; 3. идентификатор гостиницы; 4. идентификатор соответствующей платёжной операции; 5. статус; 5. дата въезда; 6. дата выезда;

Продолжение на следующей странице

Сущность	**Входные данные**
Оплата	1. идентификатор; 2. число мест; 3. этаж; 4. стоимость; 5. наличие двуспальной кровати;

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

Неавторизированный пользователь	**Авторизированный пользователь**	**Администратор**
1. список гостиниц, которые входят в сеть Apartlux 2. список доступных для бронирования дат по конкретной гостинице; 3. информация об условиях текущей программы лояльности;		
	4. детальная информация о пользователе, вошедшем в систему; 5. список доступных бронирований; 6. детальная информация о доступном конкретном бронировании; 7. форма для регистрации бронирования;	
		8. список зарегистрированных в системе клиентов;

Продолжение на следующей странице

Неавторизированный пользователь	Авторизированный пользователь	Администратор
		9. список оформленных заказов;

Топология Системы

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

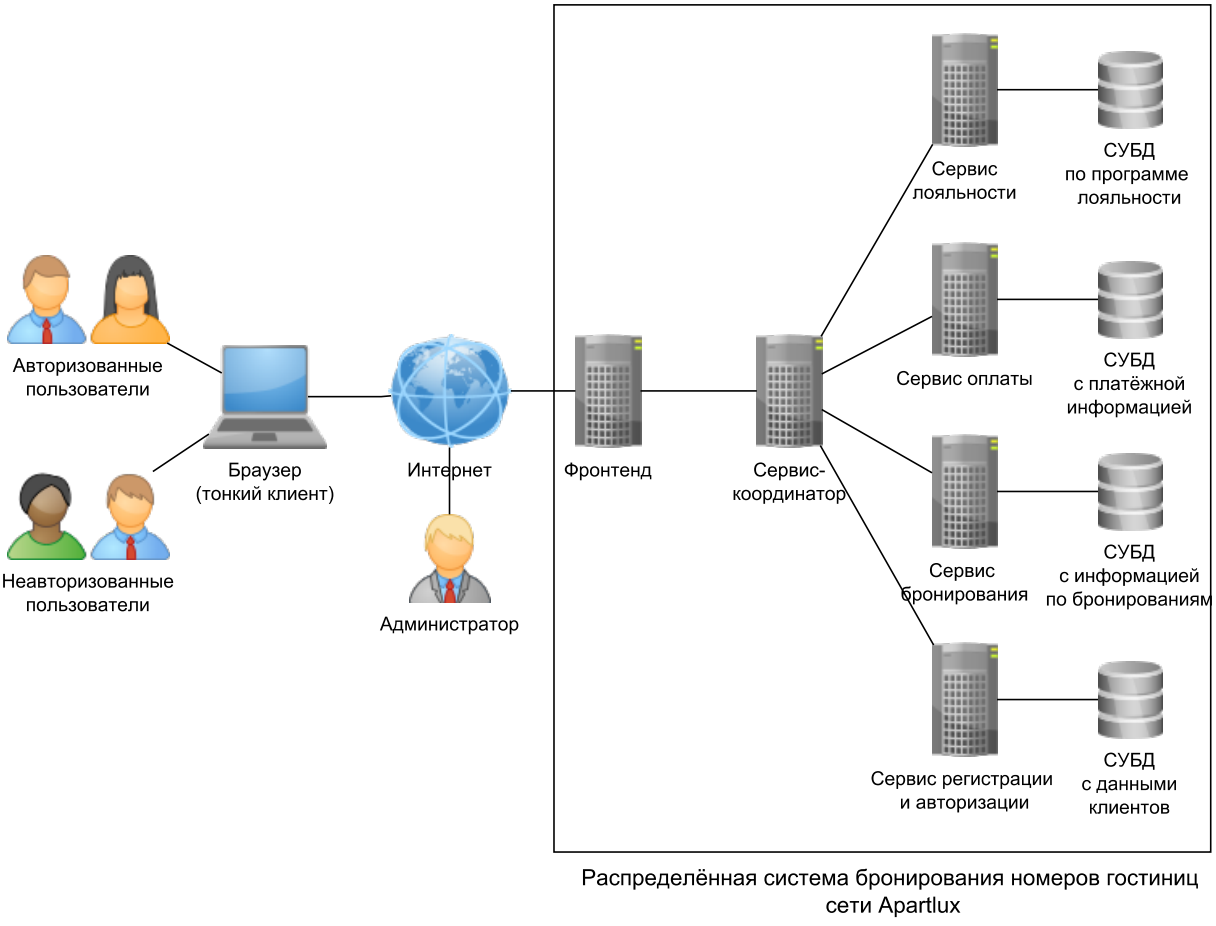

Рисунок 1.2 – Топология системы.

Система будет состоять из фронтенда и 5 подсистем:

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

Фротенд принимает запросы от пользователей по протоколу HTTP и анализирует их. На основе проведенного анализа выполняет запросы к микросервисам бекенда, агрегирует ответы и отсылает 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) Необходимо реализовать один web-интерфейс для фронтенда. Интерфейс должен быть доступен через тонкий клиент (браузер).
5) Для межсервисного взаимодействия использовать HTTP (придерживаться RESTful).
6) Выделить Gateway Service как единую точку входа и межсервисной коммуникации. В системе не должно осуществляться горизонтальных запросов.
7) При недоступности систем портала должна осуществляться деградация функционала или выдача пользователю сообщения об ошибке.
8) Необходимо предусмотреть авторизацию пользователей, как через интерфейс приложения, так и через популярные социальные сети.
9) 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.
10) Для запросов, выполняющих обновление данных на нескольких узлах распределенной системы, в случае недоступности одной из систем, необходимо выполнять полный откат транзакции.
11) Приложение должно поддерживать возможность горизонтального и вер-

тикального масштабирования за счет увеличения количества функционирующих узлов и совершенствования технологий реализации компонентов и всей архитектуры системы.

12) Код хранить на Github, для сборки использовать Github Actions.
13) Gateway Service должен запускаться на порту 8080, остальные сервисы запускать на портах 8050, 8060, 8070.
14) 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 функциональные возможности:

- принимать и возвращать данные в формате JSON по протоколу HTTP;
- возможность регистрации нового клиента и обновление данных уже существующего;
- 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	*фамилия*, не более 256 символов	да
	имя, не более 256 символов	да
	отчество, не более 256 символов	нет
	дата рождения	да
	логин, является первичным ключом	да

Продолжение на следующей странице

Сущность	Поля	Обязательность
	захешированный пароль	да

Сервис бронирования реализует следующие функции:

- получение и отправка данных в формате JSON по протоколу HTTP;
- получение списка всех гостиниц, входящий в сеть Apartlux;
- получение информации о конкретной гостинице по её идентификатору;
- вычисление стоимости бронирования за указанный период в выбранной гостинице;
- получение всех бронирований, зарегистрированных на конкретного клиента;
- получение конкретного бронирования по его идентификатору;
- создание, редактирование и отзыв бронирования.

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

Таблица 5 – Состав сущностей

Сущность	Поля	Обязательность
Гостиница	*идентификатор*, является первичным ключом	да
	идентификатор, отвечающий стандартам Open Software Foundation	да
	название, не превышает 256 символов	да
	страна, не превышает 80 символов	да
	город, не превышает 80 символов	да
	адрес, не превышает 256 символов	да
	количество звёзд	нет

Продолжение на следующей странице

Сущность	Поля	Обязательность
Бронь	*идентификатор*, является первичным ключом	да
	идентификатор, отвечающий стандартам Open Software Foundation	да
	логин клиента	да
	идентификатор оплаты	да
	идентификатор гостиницы	да
	статус оплаты, PAID/CANCELED	да
	дата начала брони	да
	дата окончания брони	да

Сервис оплаты реализует функции:

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

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

Таблица 6 – Состав сущностей

Сущность	Поля	Обязательность
Платёж	*идентификатор*, является первичным ключом	да
	идентификатор, отвечающий стандартам Open Software Foundation	да
	статус, PAID/CANCELED	да

Продолжение на следующей странице

Сущность	Поля	Обязательность
	цена	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 по его логину;
- обновление числа заказов 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	*идентификатор*, является первичным ключом	да
	логин клиента	да
	количество оформленных ранее заказов, по умолчанию 0	да
	статус, BRONZE/SILVER/GOLD, по умолчанию BRONZE	да
	скидка, 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;
- авторизированного пользователя Системы;
- неавторизированного пользователя Системы;
- по развёртыванию Системы.