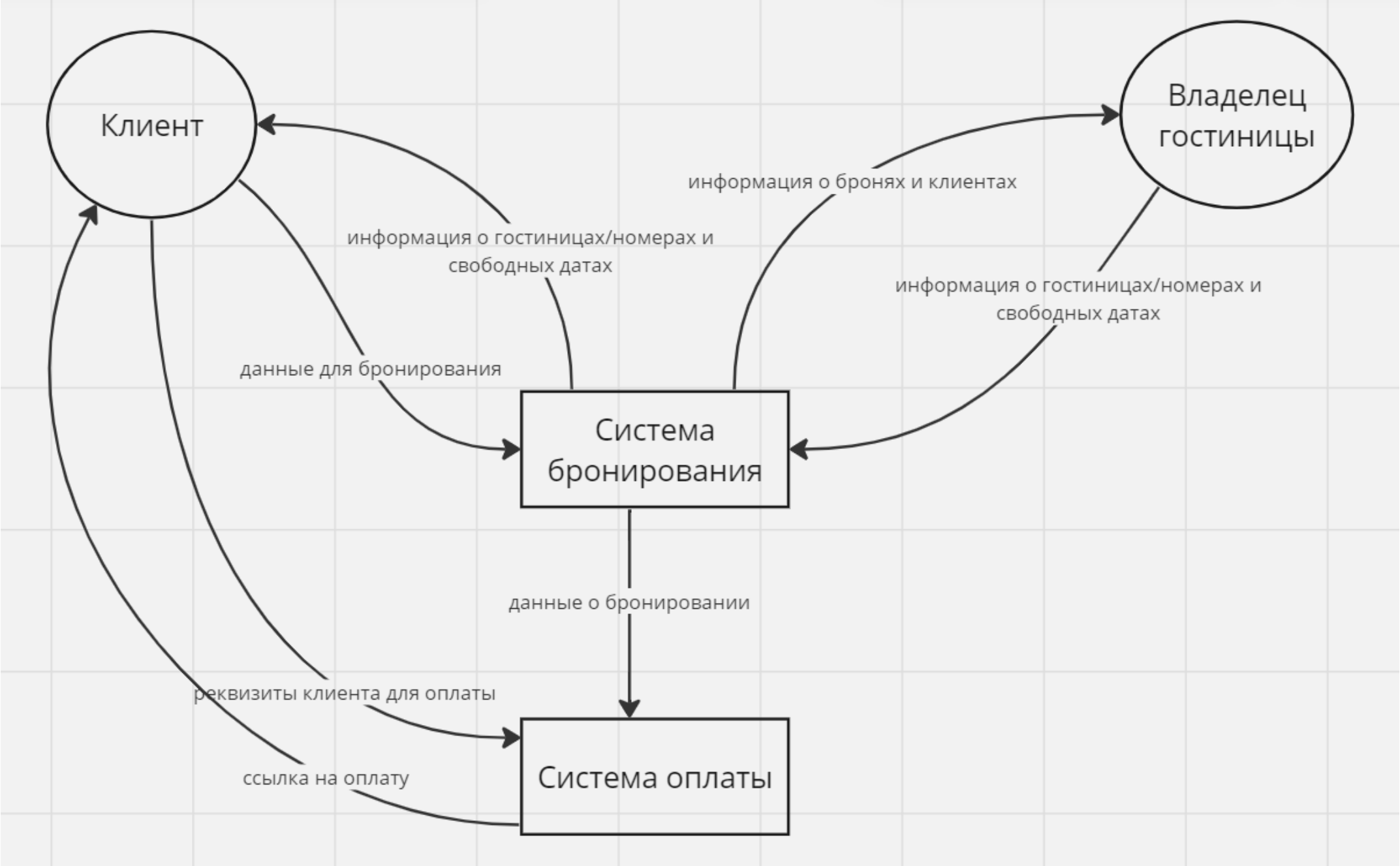


Моделирование. Сбор требований и контекстная диаграмма

Вот такая диаграмма получилась в итоге:



Как вы видите - мы не отмечаем стрелку от клиентов к нашей системе для функции "Клиенты: смотрят информацию о гостиницах\номерах и свободных датах". Потому что нам важен поток данных а не запрос данных (мы же проектируем данные а не API), а поток этих данных именно отправляет наша система бронирования а Клиенты уже просто ПОЛУЧАЮТ и СМОТРЯТ эту информацию.

Связь API и структуры базы данных - должны ли они быть похожи?

API играет важную роль в процессе сбора требований для проектирования данных. API действует как связующее звено между пользователем и системой, обеспечивая механизм, через который пользователи могут запрашивать и получать данные. Например, когда клиенты ищут информацию о гостиницах и доступных номерах, они взаимодействуют с системой через API, которое предоставляет необходимые данные в удобном для понимания формате.

API определяет, какие операции можно выполнять с данными, какие запросы можно отправлять и какие ответы ожидать. Таким образом, оно становится источником требований к данным, поскольку выявляет нужды пользователей в информации, способы ее представления и обработки (то есть на этапе сбора требований вы в том числе можете задавать вопросы - а какие данные нужны пользователям? какие API уже есть для пользователей?). Это помогает проектировщикам данных понять, какие данные важны для пользователей, как часто они будут запрашиваться и как должна быть организована их доступность.

Важно понимать, что, хотя API может влиять на требования к данным и их организацию, оно не должно напрямую определять структуру базы данных. API и база данных — это две разные сущности с различными целями и требованиями:

- **API:** Сфокусировано на предоставлении данных в необходимом для пользователя виде и управлении взаимодействиями с системой. API должно быть спроектировано таким образом, чтобы быть гибким, масштабируемым и удобным для пользователя, не раскрывая внутреннюю структуру данных.
- **База данных:** Должна быть оптимизирована для эффективного хранения, обработки и извлечения данных. Проектирование базы данных исходит из требований к надежности, целостности, безопасности и производительности, и не всегда совпадает с тем, как данные представлены в API.

Искусство проектирования — сделать так, чтобы структура базы данных поддерживала все необходимые операции API и была способна эффективно обслуживать запросы, но при этом оставалась независимой от интерфейса API. Это позволяет развивать и изменять API по мере изменения бизнес-требований, не затрагивая основу хранения данных.