### **Название задачи: Отображение списка депозитов с актуальными ставками**

### **Автор:**

### **Дата:**

### **Функциональные требования**

Опишите здесь верхнеуровневые Use Cases. Их нужно оформить в виде таблицы с пошаговым описанием:

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действующие лица или системы** | **Use Case** | **Описание** |
| 1 | Клиент, колл-центр, сотрудник колл-центра | Получение информации по доступным ставкам | Клиент звонит в колл-центр, оператор смотрит актуальную информацию в системе рассчета ставок и озвучивает предложение банка |
| 2 | Клиент, колл-центр, партнерский колл-центр, сотрудник партнерского колл-центра | Получение информации по доступным ставкам | Клиент звонит в колл-центр, но из-за загруженности, его звонок переводят в партнерский колл центр. Оператор видит актуальную информацию по ставкам и озвучивает ее пользователю. |
| 3 | Клиент, интернет банк | Получение информации по доступным ставкам | Клиент заходит в интернет-банк, авторизуется и переходит на страницу открытия депозита. На странице ему отображается список доступных предложений с актуальными ставками. |

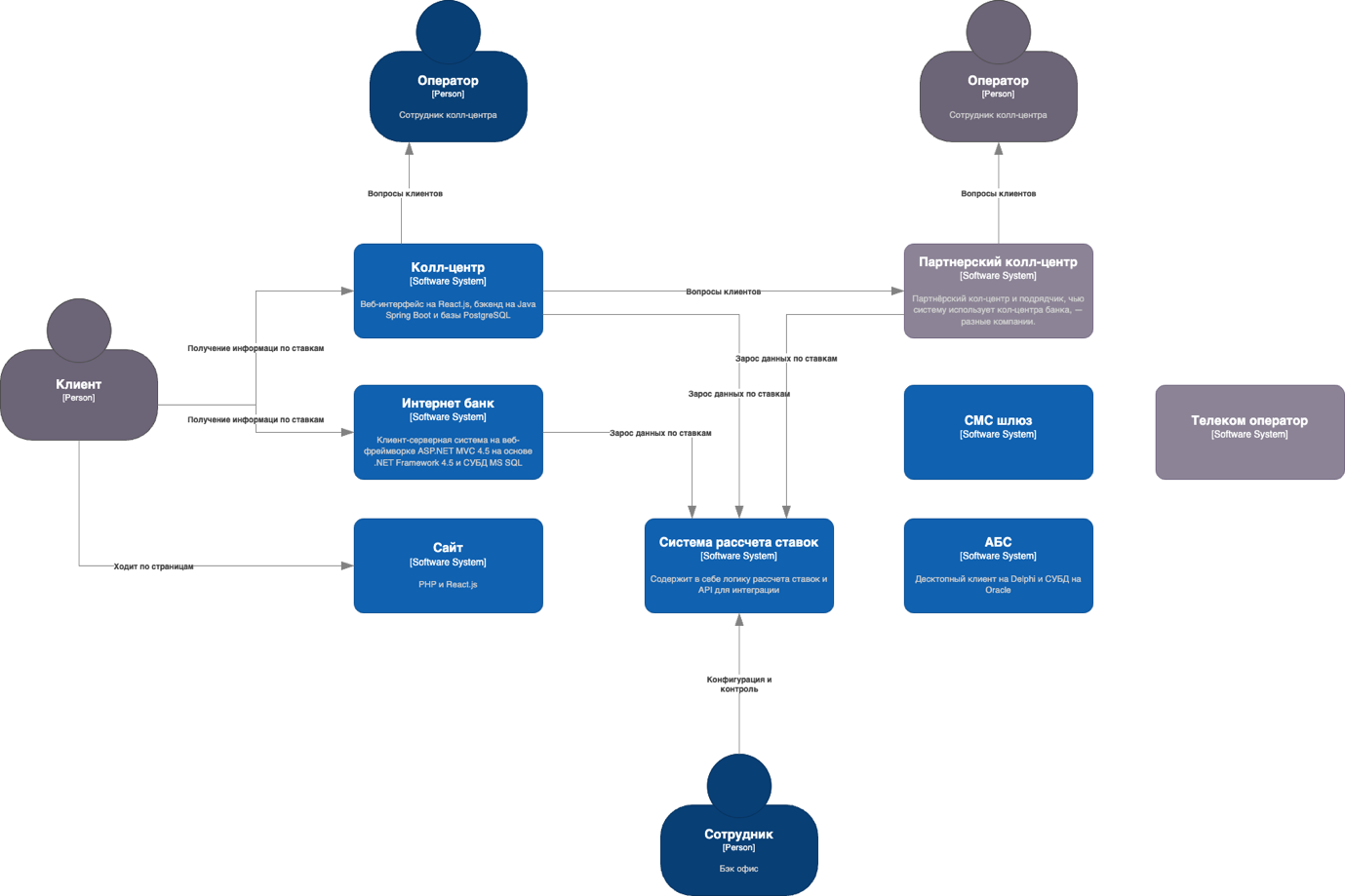
### **Нефункциональные требования**

Опишите здесь нефункциональные требования и архитектурно-значимые требования.

|  |  |
| --- | --- |
| **№** | **Требование** |
| 1 | Данные рассчета будут использоваться разнами системами |
| 2 | Данные должны отображаться пользователю максимально быстро |
| 3 | Нужно использовать принятую в банке системы дизайна |
| 4 | Нужно проектировать систему с учетом будующего расширения функционала и нагрузки |
| 5 | Нужно стараться использовать технологии, в которых в банке есть экспертиза |

### **Решение**

Судя по описанию структуры компании, для разработки новых модулей можно использовать команду разработки интернет-банка. Во первых, в текущий момент, это направление не является критически важным, во вторых .NET позволяет разрабатывать современные приложения в отличии от Delphi.



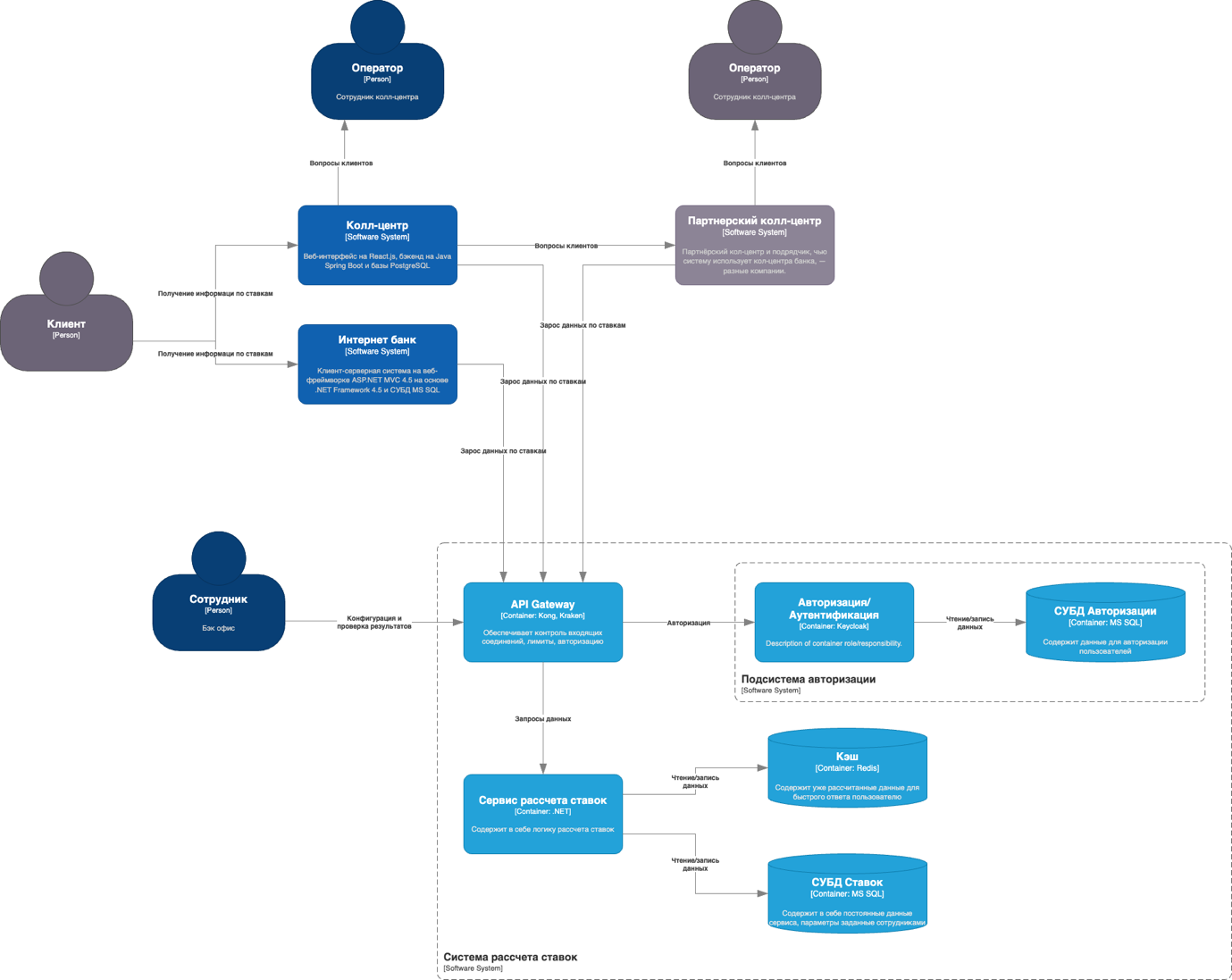
Изображение Диаграмма контекста

Судя по описанию системы, на старте, а возможно и позже, необходимо будет участие оператора в рассчете ставок. Использование .NET решает эту проблему тк позволяет создавать не только API, но и фронт с использованием движка шаблонов Razer.

Вспомогательные системы банка не указаны в описании, предполагаю, что есть один или несколько AD для управления пользователями.

Есть большое подозрение, что ставки по депозитам будут меняться не очень часто, не чеще одного раза в несколько минут, поэтому стоит использовать кэш для скорения возврата результата пользователю.

Использование Keycloak позволит разделить пользователей на группы (RBAC) и возвращать для них немного разную информацию. Имеется ввиду, что для оператора колл-цента, можно возвращать более подробную информацию, возможно с условиями по снижению ставки, которые можно озвучить пользователю. При этом, такая информация для интернет-банка, может не отображаться.



Изображение Диаграмма контейнеров

### **Альтернативы**

В качестве альтернативы, можно предложить разработку приложения, которое бы позволило мэнеджерам рассчитывать показатели в ручную без использоватьния почты. Например, общий документ в google-docs позволит нескольким менеджерам одновременно рассчитывать ставки. Google-docs позволяют сделать интеграцию, те можно отслеживать изменения в документе и рассылать их заинтересованным сторонам.

**Недостатки, ограничения, риски**

Малая команда разработки, без опыта создания сервисов может столкнуться с проблемами. В случае продолжения развития собственной разработки, нужно набирать еще людей.

При разработки данной системы, можно столкнуться с большим негативом и даже противодействием со стороны менеджеров бэк офиса, тк такая система предполагает сокращение их количества. Следует проводить аналитику как можно раньше, чтобы опираться на неискаженные данные от бэк офиса.

Стоит продумать масштабирование системы на второй ЦОД для повышение скорости ответов пользователям по гео-признаку. Одновременно, это поысит общую надежность системы.