

Есть **минимальный набор информации, который важен для документирования в любой системе**. Этот "минимальный жизнеспособный" документ по архитектуре должен включать:

- **Ключевые архитектурные решения и принципы:** Эти основные решения формируют общую структуру системы и влияют на все аспекты ее работы. Они должны быть четко документированы и объяснены.
- **Важные компоненты и их взаимодействия:** Центральные элементы системы и способы их взаимодействия друг с другом также необходимо описать. Это позволяет понять, как данные и работа перемещаются по системе.
- **Информация о выбранных технологиях:** Обоснование выбора определенных технологий, а также их основные характеристики и способы использования, должны быть частью документации. Это помогает понять, почему был выбран тот или иной инструмент и как он поддерживает требования бизнеса и пользователей.
- **Схемы данных и взаимосвязи:** Основные модели данных, их атрибуты и отношения между ними являются ключевой частью любой системы. Они должны быть документированы для понимания, как информация хранится и обрабатывается в системе.

Первый пункт вы можете описать текстом, опираясь на ранее обнаруженные и решённые (и нерешённые) компромиссы по архитектурно значимым требованиям и ограничениям к проекту.

Для описания важных компонентов системы, их взаимодействий, выбранных технологий и схем данных и взаимосвязей, можно использовать различные подходы и методологии. **Один из них — модель 4+1 представления архитектуры, предложенная Филиппом Крукенбергером.**

Модель 4+1 является эффективной методикой для представления архитектуры системы. Эта модель делит архитектуру на пять различных "представлений", каждое из которых фокусируется на определенном аспекте системы.

- **Логическое представление (или представление сценариев).** Это представление ориентировано на функциональность системы с точки зрения конечного пользователя. В этом представлении используются диаграммы классов и диаграммы последовательности, которые описывают функциональные аспекты системы. Например, диаграмма классов может отобразить структуру объектов в

системе и их взаимосвязи, а диаграмма последовательности может показать, как эти классы взаимодействуют во время выполнения определенного сценария.

- Процессное представление. Это представление фокусируется на аспекты системы, связанные с параллелизмом, синхронизацией и производительностью. Оно помогает системным инженерам понять, как система будет вести себя в процессе выполнения. Диаграммы деятельности и диаграммы состояний часто используются в этом представлении для иллюстрации этих аспектов.
- Компонентное представление. Оно ориентировано на архитектуру программного обеспечения, предназначенную для разработчиков и архитекторов. Это представление использует диаграммы компонентов для описания структуры системы на уровне модулей и подсистем.
- Физическое представление. Оно связано с инфраструктурой и оборудованием и включает диаграммы развертывания, которые отображают физическое расположение компонентов системы. Оно помогает понять топологию сети, конфигурации оборудования и другие аспекты, связанные с развертыванием системы.
- Представление использования (или "плюс один"). Это представление связано с основными сценариями использования системы. Оно позволяет выделить наиболее значимые сценарии и описывает, как система поведет себя во время их выполнения.

Как вы увидели, для отображения различных диаграмм используется нотация UML. Для нашего проекта это может выглядеть так.

- Логическое представление: Здесь мы используем диаграмму классов UML, чтобы описать основные классы нашей системы и их взаимосвязи. В вашем случае классы могут включать компоненты пользовательского интерфейса, классы бизнес-логики для каждого из трех модулей бэкенда, и классы, которые представляют данные, хранящиеся в реляционной базе данных. Важно также отобразить взаимодействие с внешним сервисом "Алёшка".
- Процессное представление: Здесь мы используем диаграммы последовательностей или диаграммы деятельности для описания того, как эти классы взаимодействуют друг с другом. Например, можно изобразить, как происходит процесс обработки пользовательского запроса от момента его получения на фронтенде до момента возврата ответа от одного из модулей бэкенда.

- Разработческое представление: Здесь мы используем диаграммы компонентов для описания модульной структуры нашей системы. В вашем случае, у вас есть фронтенд и три модуля бэкенда, которые можно изобразить как отдельные компоненты, а также база данных и внешний сервис "Алёшка".
- Физическое представление: Здесь мы используем диаграммы развертывания для описания того, как наши компоненты развернуты на физических или виртуальных машинах. В зависимости от вашей конкретной инфраструктуры, вы можете отобразить, где размещаются фронтенд, модули бэкенда, база данных и как они взаимодействуют с сервисом "Алёшка".
- Представление использования: Здесь мы используем диаграммы вариантов использования для описания основных функций системы с точки зрения пользователя. В вашем случае, это могут быть диаграммы, описывающие ключевые функции, которые пользователь может выполнять в системе, и какие модули бэкенда и сервисы используются для каждой из этих функций.

Важно помнить, что диаграммы UML — это только один из инструментов для документирования архитектуры. Вы также можете использовать текстовые описания, таблицы, списки и другие формы представления информации в зависимости от того, что наиболее подходит для вашей конкретной ситуации.

Со всеми UML диаграммами можете ознакомиться здесь: <https://plantuml.com/>