

## Описание диаграммы классов

В данном документе приведено описание диаграммы классов разрабатываемого приложения. Сама диаграмма приведена в приложении А настоящего документа.

Приведем описание каждого из классов на диаграмме и их роль.

Класс Transcript представляет собой модель стенограммы встречи и отвечает за хранение базовой информации: идентификатора, названия, даты, а также списков тегов и реплик. Он устанавливает связь один-ко-многим с сущностью Replica и связь многие-ко-многим с сущностью Tag. При этом задачи (Task) в структуру Transcript не включаются напрямую, что обеспечивает разделение ответственности и упрощает сопровождение модели.

Класс Replica описывает отдельный фрагмент речи (реплику), входящий в состав стенограммы. Он содержит сведения о спикере, тексте и временной метке. Replica ассоциирована с Transcript по схеме многие-к-одному и с Speaker — аналогично. Это отражает структуру, в которой каждая реплика принадлежит как одной стенограмме, так и одному участнику встречи.

Класс Task – модель задачи, сформулированной по результатам встречи. Каждая задача связана с конкретной стенограммой (Transcript) и назначается конкретному участнику (Speaker). Атрибуты включают идентификатор, описание, а также идентификаторы связанных сущностей. При этом список задач не включается в Transcript, а агрегируется отдельно, что повышает модульность архитектуры.

Класс Speaker представляет участника встречи. Он содержит основные атрибуты (идентификатор, имя, изображение) и устанавливает связи один-ко-многим с Replica и Task, что позволяет отслеживать, какие реплики и задачи связаны с конкретным участником.

Сущность Tag предназначена для классификации стенограмм с использованием ключевых слов. Tag участвует в связи многие-ко-многим с Transcript, позволяя одному тегу быть связанным с несколькими стенограммами и наоборот.

Класс Protocol отвечает за представление итогового вывода по встрече. Он содержит текст итогов и связан с Transcript по схеме один-к-одному. Такое разделение способствует четкому разграничению между исходной стенограммой и итоговым резюме встречи.

Класс MeetingMaterials объединяет в себе материалы, связанные с конкретной встречей: стенограмму (Transcript), итоговый протокол (Protocol) и задачи (Task). Именно здесь хранятся все связанные с встречей данные, включая список задач, что делает MeetingMaterials центральной точкой доступа к информации о встрече. Композиция с Transcript, Protocol и Task подчеркивает зависимость жизненного цикла этих объектов от MeetingMaterials.

Классы LLMService / LLMWrapper инкапсулируют взаимодействие с языковой моделью (LLM) и предоставляют интерфейс для извлечения задач, генерации итогов и других операций над текстами стенограмм. Они являются компонентами прикладной логики, взаимодействующими с Transcript и другими сущностями для обогащения данных на основе ИИ.

Классы TranscriptDao, TaskDao, SpeakerDao, TagDao, MeetingMaterialsDao реализуют паттерн доступа к данным (DAO) и предоставляют методы выполнения CRUD-операций. Они ассоциированы с соответствующими бизнес-сущностями и обеспечивают ручное взаимодействие с базой данных без использования ORM.

Отличия от предыдущей версии диаграммы классов:

- в новой архитектуре задачи (Task) и итоговый протокол (Protocol) не входят в состав Transcript, как это было ранее. Вместо этого они агрегируются в отдельном классе MeetingMaterials, что способствует более строгому разделению ответственности между сущностями;
- актуализированная диаграмма концентрируется на бизнес-сущностях: Transcript, Replica, Task, Speaker, Tag, Protocol, MeetingMaterials. Это отражает ориентацию на прикладную логику обработки встреч, в отличие от предыдущей диаграммы, включавшей низкоуровневые классы, связанные с аудио/видеопотоками (например, VoskRecognizer, MicrophoneStreamer), что указывало на техническую направленность;
- новая версия архитектуры предусматривает явное выделение DAO-классов для доступа к данным. Это обеспечивает четкое разграничение между слоями бизнес-логики и хранения данных, повышая модульность системы и облегчая тестирование.

В новой модели исключены компоненты, не относящиеся к бизнес-логике, а методы классов стали более специализированными, соответствующими конкретным прикладным задачам.

Диаграмма отражает прямую интеграцию ИИ: классы LLMService и LLMWrapper обеспечивают централизованное взаимодействие с языковой моделью. Это упрощает добавление интеллектуальных функций, таких как извлечение задач и формирование протоколов.

### **Причины внесения изменений**

Принцип единой ответственности: выделение MeetingMaterials позволяет каждому классу концентрироваться на строго определенной задаче: Transcript — на текстовом содержимом, Task и Protocol — на производных данных встречи.

Переход от технически-ориентированной к бизнес-ориентированной архитектуре: актуализированная модель исключает компоненты, связанные с обработкой аудиопотока, что отражает сдвиг фокуса в сторону обработки содержимого встреч и автоматизации делопроизводства.

Повышение гибкости и удобства сопровождения: использование DAO-классов и разделение логики на независимые модули обеспечивает масштабируемость и упрощает поддержку системы.

Глубокая интеграция ИИ-функциональности: наличие специализированных компонентов для взаимодействия с LLM способствует централизованному и расширяемому подходу к применению ИИ в системе.

Соответствие структуре хранения данных: новая диаграмма проектировалась с учетом физической структуры базы данных, что облегчает ручное написание SQL-запросов и обеспечивает согласованность между логикой и хранилищем данных.

# Приложение А

