**Пользовательский сценарий:**

Госслужащий (без навыков программирования) запускает ваше приложение на своем рабочем компьютере.

Он создает новый «проект» для своей госуслуги.

Загружает в приложение три типа файлов:

JSON-схему услуги (описание полей формы с ЕПГУ).

XSD-схему вида сведений (описание структуры данных, которую ожидает внутренняя система).

Один или несколько JSON-файлов тестовых заявлений (примеры заполненных форм).

Нажимает кнопку «Сгенерировать шаблон».

Приложение анализирует файлы и автоматически создает готовый код VM-шаблона (файл с расширением .vm).

Пользователь может скопировать полученный код или сохранить его как файл.

Функциональные требования к MVP

Графический интерфейс (GUI):

Приложение должно быть десктопным (мультиплатформенным.)

Интерфейс должен быть максимально простым и интуитивно понятным для пользователя, не знакомого с IT-терминологией.

Должны быть элементы для загрузки файлов и кнопка для запуска генерации.

**Управление проектами:**

Возможность создать новый проект (например, по названию услуги).

Возможность сохранить и загрузить проект (со всеми загруженными файлами и результатом).ж

Основная логика (ядро генератора):

Парсинг входных данных: Приложение должно уметь читать и анализировать структуру загруженных .json и .xsd файлов.

Сопоставление полей: Реализовать логику сопоставления полей из JSON-схемы услуги с полями из XSD-схемы. (Для MVP можно реализовать простое сопоставление по одинаковым или схожим названиям полей).

Генерация кода: На основе сопоставленных полей приложение должно сгенерировать синтаксически корректный файл на языке Apache Velocity, который будет формировать XML-документ нужной структуры.

Вывод результата:

Окно для предпросмотра сгенерированного .vm кода с подсветкой синтаксиса.

Кнопка «Сохранить в файл» или «Копировать в буфер обмена».