

Экзаменационное задание Моделирование бизнес-процессов и формализация требований.

1. Контекст и Общая задача

Вы присоединяетесь к команде проекта «TechStats» — веб-сервиса для анализа востребованности технологий в IT-вакансиях на HeadHunter.ru.

Текущее состояние (MVP):

- Сервис работает и имеет базовую функциональность через WebSocket.
- Есть система кэширования с TTL для ускорения повторных запросов.
- Клиентский веб-интерфейс отображает прогресс выполнения анализа.

Проблема: Команда хочет перейти от MVP к production-решению. Процесс анализа сейчас монолитный, протекает в одной долгой WebSocket-сессии. Нет возможности:

- Ставить анализ в очередь при высокой нагрузке.
- Возобновлять прерванный анализ (при разрыве соединения, перезагрузке страницы).
- Управлять приоритетом анализов.
- Планировать периодический анализ (например, ежедневный сбор статистики по ключевым технологиям).
- Масштабировать обработку на несколько воркеров.

Ваша задача — проанализировать текущую архитектуру и **формализовать требования к модулю «Очередь задач анализа» (Task Queue)**, а также спроектировать соответствующие бизнес-процессы.

Исходные материалы проекта доступны в репозитории: <https://github.com/tsenturion/techstats>. Особое внимание уделите файлам:

- app.py — основная логика сервера и WebSocket-обработчик (`/ws/analyze`).
- README.md — описание возможностей и API.
- index.html — клиентская часть.

2. Задания

Задание 1. Анализ «как есть» и выявление ограничений (Макс. 4 балла)

На основе изучения кода, опишите текущую логику работы сервиса при выполнении запроса на анализ:

- Основные сущности и данные:** Перечислите ключевые сущности (Запрос на анализ, Вакансия, Результат, Запись кэша), их атрибуты и взаимосвязи.
- Текущий поток выполнения:** Опишите шаги, которые выполняет система от момента подключения клиента по WebSocket до отправки финального результата. Выделите этапы: аутентификация (если есть), валидация входных данных, поиск вакансий, их обработка, формирование ответа. Укажите, где находятся потенциальные точки отказа.
- Архитектурные ограничения:** Сформулируйте 3-5 ключевых ограничений текущей синхронной WebSocket-модели, которые должна решить внедрение асинхронной очереди задач (Task Queue). Например: «При разрыве WebSocket-соединения выполнение анализа прерывается, результат теряется».

Ожидаемый результат: Текстовый документ с разделами 1.1, 1.2, 1.3.

Задание 2. Моделирование бизнес-процесса «Анализ через очередь задач» (Макс. 5 баллов)

Спроектируйте целевой бизнес-процесс «Обработка запроса на анализ вакансий» с использованием асинхронной очереди. Процесс должен поддерживать статусы: PENDING, PROCESSING, SUCCESS, FAILED, CANCELLED.

Используйте нотацию BPMN 2.0.

Процесс должен включать:

- Создание задачи:** Клиент (UI) отправляет параметры анализа (название вакансии, технологию) через REST API. Система валидирует параметры, создает запись AnalysisTask в статусе PENDING и помещает задачу в очередь. Клиенту сразу возвращается task_id .
- Обработка воркером:** Отдельный фоновый воркер (Worker) забирает задачу из очереди, меняет статус на PROCESSING и выполняет анализ (текущая логика из app.py), периодически обновляя прогресс в БД.
- Отслеживание прогресса:** Клиент может подписаться на обновления по task_id через WebSocket или опрашивать статус через REST. Система отправляет обновления прогресса.
- Завершение:** При успешном завершении воркер сохраняет результат, меняет статус задачи на SUCCESS . При ошибке — на FAILED с записью ошибки.
- Получение результата:** Клиент запрашивает результат по task_id через REST API.
- Обработка прерывания:** Возможность отмены задачи пользователем (CANCELLED). Защита от повторного запуска одной и той же задачи (дедупликация по хешу параметров).

Требования к диаграмме:

- Использованы пулы/дорожки (минимум: «Клиент (UI)», «API Gateway», «Сервис задач (Task Service)», «Воркер анализа (Analysis Worker)», «Внешний API (HH.ru)»).
- Показаны сообщения (REST, WebSocket), таймеры (для опроса), ошибки.
- Отражена асинхронность (задачи и события).
- Процесс имеет четкое начало (получение запроса) и несколько конечных событий (успех, ошибка, отмена).

Ожидаемый результат: Диаграмма BPMN в виде изображения (PNG/JPEG) и редактируемого файла (например, .bpmn).

Задание 3. Формализация нефункциональных и функциональных требований (Макс. 3 балла)

На основе разработанного BPMN-процесса составьте структурированные требования к новой системе.

Часть А. Нефункциональные требования (Non-Functional Requirements, NFR):

- NFR1: Производительность и масштабируемость.** (Опишите ожидаемую нагрузку, требования к времени отклика API, возможность горизонтального масштабирования воркеров).
- NFR2: Надежность и отказоустойчивость.** (Требования к сохранности задач при падении воркера, гарантии доставки, время восстановления).
- NFR3: Удобство использования (Usability).** (Требования к отображению прогресса, работе с историей задач в UI).

Часть В. Функциональные требования (Functional Requirements, FR):

Сгруппируйте по логическим модулям. Используйте формат «Система должна...» или «Как <роль>, я хочу...».

- FR1: Модуль управления задачами (Task Service).**
 - Пример: FR1.1: Система должна предоставлять REST API endpoint POST /api/v1/tasks для создания новой задачи анализа с валидацией входных параметров.
 - Пример: FR1.2: Система должна обеспечивать дедупликацию задач на основе хеша входных параметров (вакансия, технология, регион) в течение последних N часов.
- FR2: Модуль воркеров (Worker).**
- FR3: Модуль уведомлений/WebSocket.**
- FR4: Модуль истории и результатов.**

Ожидаемый результат: Документ с 5-7 NFR и 10-12 конкретных, тестируемых FR.

3. Система оценивания

Общий балл складывается из суммы баллов за три задания. **Максимальный итоговый балл — 12.**

Балл	Критерии оценки (суммарно по всем заданиям)
12	Отлично. Анализ демонстрирует глубокое понимание кодовой базы, выявлены тонкие архитектурные проблемы. BPMN-диаграмма логически безупречна, элегантно решает задачи асинхронности, обработки ошибок, масштабирования. Требования полные, включают продвинутые аспекты (дедупликация, приоритеты, метрики). Оформление идеально.
11	Почти отлично. Анализ глубокий. BPMN-диаграмма корректна и полна, возможны 1-2 незначительных недочёта. Требования исчерпывающие. Работа на профессиональном уровне.
10	Очень хорошо. Анализ содержательный. BPMN-диаграмма верно отражает процесс, есть незначительные ошибки в элементах. Требования покрывают ключевые аспекты, часть может быть менее конкретна.
9	Хорошо, с заметными недочётами. Анализ выполнен, но поверхностен. BPMN-диаграмма в целом верна, но содержит несколько ошибок (логика, элементы). Набор требований неполный, некоторые расплывчаты.
8	Удовлетворительно. Анализ минимально достаточен. BPMN-диаграмма передаёт общую идею, но с серьёзными ошибками моделирования. Требования фрагментарны, неструктурированы.
7	Посредственно. Анализ крайне поверхностный. BPMN-схема схематична, с грубыми нарушениями нотации. Требования декларативны, не тестируемы. Работа формальна.
6	Минимальный зачёт. Выполнены только базовые части заданий. Анализ не демонстрирует понимания проекта. BPMN-схема присутствует, но нечитаема. Требования почти не formalizованы.
5	Недостаточно. Сделана лишь часть работы. Качество низкое. Серьёзное несоответствие условию.
4	Слабо. Работа сдана, но результат не демонстрирует понимания задач системного анализа.
3	Очень слабо. Представлены лишь разрозненные фрагменты.
2	Практически без результата. Сдана неудачная попытка.
1	Работа не выполнена. Представлены нерелевантные материалы.
0	Работа не сдана.

Примечание: Оценка выставляется за **качество, глубину и аккуратность** проработки каждого пункта.

4. Формат сдачи

Необходимо предоставить **архив (ZIP)**, содержащий:

- Readme.md — краткая пояснительная записка.
- Analysis_Asis.docx/pdf — результат **Задания 1**.
- Process_TovB.bpmn и Process_TovB.png — результат **Задания 2**.
- Requirements_NFR_FR.docx/pdf — результат **Задания 3**.