

Use case:

Автоматический поиск и ранжирование кандидатов на HeadHunter	
Акторы	HR-менеджер, ИИ Агент рекрутинга, HeadHunter API
Предусловие	1. Пользователь авторизован в системе. 2. Система имеет доступ к HH API.
Ограничения	1. Работает только с публичными данными HH API. 2. LLM обрабатывает только текстовые данные (не файлы).
Триггер	HR-менеджер нажимает кнопку "Найти кандидатов" после заполнения формы.
Основной сценарий	<p>1. Пользователь вносит описание вакансии (Экран 1)</p> <ul style="list-style-type: none">- Система структурирует информацию (извлекает ключевые навыки, опыт, локацию).- Система отправляет запрос к HH API с параметрами вакансии. <p>2. Получает список резюме в JSON-формате</p> <ul style="list-style-type: none">- Сохраняет сырые данные в таблицу raw_resumes (PostgreSQL). <p>3. Извлекает текст резюме и передает его в LLM для анализа</p> <ul style="list-style-type: none">- LLM оценивает соответствие вакансии по критериям: навыки, опыт. <p>4. LLM возвращает оценку релевантности (0-100%)</p> <ul style="list-style-type: none">- Формат ответа: {"score": 85, "reason": "Совпадение по Python (5 лет), ML (3 года)"} <p>5. Система сохраняет топ-10 кандидатов в таблицу top_candidates (PostgreSQL)</p> <ul style="list-style-type: none">- Поля: id, name, skills (array), work_experience, score, hh_link. <p>6. Отображает список кандидатов с рейтингом (Экран 2)</p> <ul style="list-style-type: none">- Сортировка по убыванию score.
Альтернативный сценарий	<p>4а. LLM не может обработать резюме (например, язык ≠ русский/английский).</p> <p>→ Система помечает кандидата как needs_review и не включает в топ-10.</p>
Исключительный сценарий	<p>2а. HH API недоступен</p> <p>→ Система показывает ошибку: "Не удалось подключиться к HH. Попробуйте позже."</p>