

Use Case 5.2.2.1 Анализ резюме с помощью YandexGPT

Интеграционный документ

1. Общая информация

Параметр	Значение
Система	A-Работа
Функция	AI-анализ резюме
Внешняя система	YandexGPT API
Тип интеграции	Синхронная REST
Формат данных	JSON
Timeout	30 секунд
Retry	2 попытки (exponential backoff)

Хранение
результата

В БД

Логирование текстов Запрещено

2. Бизнес-описание

Авторизованный соискатель может выполнить AI-анализ своего резюме.

AI анализирует:

1. Раздел "О себе" (1 блок)
2. Каждый опыт работы отдельно:
 - Обязанности
 - Достижения
3. Формирует общий вывод

YandexGPT возвращает структурированный JSON-анализ:

- Анализ каждого поля
- Общий анализ резюме

Результат сохраняется в БД и отображается пользователю.

Ограничения анализа

Параметр	Значение
Минимальная длина поля	50 символов

Максимальное количество последних мест работы	За последние 5 лет, но не более 5 последних
Максимальный суммарный объем текста запроса	9 000 символов / 5000 токенов
Максимальный размер ответа AI	12 000 символов

Если опытов больше 5 → анализируются последние 5.

3. Участники процесса

Роль	Описание
Соискатель	Владелец резюме
Web Client	Web интерфейс
Mobile App	iOS / Android
Backend	Основная бизнес-логика
YandexGPT API	Внешняя LLM

4. API

4.1 Запуск анализа

```
POST /api/v1/resumes/{resumeId}/ai-analysis
```

Headers

```
Authorization: Bearer <JWT>
Idempotency-Key: <uuid>
Content-Type: application/json
```

Body

```
{
  "forceReanalysis": false
}
```

4.2 Ответ

```
{
  "analysisId": "UUID",
  "createdAt": "ISO8601",
  "nextAvailableAt": "ISO8601",
  "aboutAnalysis": {
    "strengths": "",
    "weaknesses": "",
    "recommendations": ""
  },
  "experienceAnalysis": [
    {
      "resume_experience_id": "integer",
      "responsibilityAnalysis": {
        "strengths": "",
        "weaknesses": "",
        "recommendations": ""
      },
      "achievementsAnalysis": {
        "strengths": "",
        "weaknesses": "",
        "recommendations": ""
      }
    }
  ]
}
```

```
        ],
    "generalAnalysis": {
        "strongSides": "",
        "weakSides": "",
        "overallRecommendations": ""
    }
}
```

5. Валидация перед запуском анализа

5.1 Проверка прав

- Пользователь авторизован
- Пользователь владелец резюме

5.2 Проверка входных данных

Поле	Условие
about	≥ 50 символов
responsibility	≥ 50
achievements	≥ 50
Максимум мест работы	5 за последние 5 лет
Максимальный суммарный текст	9 000 символов

- trim()
- HTML strip
- UTF-8
- max 3 000 символов для одного описательного поля

Если не проходит — 400 Bad Request.

6. Контракт интеграции с YandexGPT

6.1 Документация YandexGPT:

<https://yandex.cloud/ru/docs/ai-studio/quickstart/yandexgpt>

<https://yandex.cloud/ru/docs/ai-studio/operations/generation/create-prompt>

6.2 Endpoint

YandexGPT Lite 5

```
gpt://<идентификатор_каталога>/yandexgpt-lite
```

6.3 Авторизация

Authorization: API-ключ

<https://yandex.cloud/ru/docs/iam/concepts/authorization/api-key>

6.4 Request Contract

```
{
  "modelUri": "gpt://<идентификатор_каталога>/yandexgpt-lite",
  "completionOptions": {
    "temperature": 0.3,
    "maxTokens": 5000
  },
  "messages": [
    {
      "role": "system",
      "text": "SYSTEM PROMPT"
    },
    {
      "role": "user",
      "text": "USER PROMPT"
    }
  ]
}
```

6.5 Токен контроль

Перед отправкой:

- оценивается приблизительное количество токенов
- добавляется 20% buffer
- если превышает лимит → **уменьшается количество мест работы**

6.6 Требования к ответу модели

```
{  
  "aboutAnalysis": {  
    "strengths": "string",  
    "weaknesses": "string",  
    "recommendations": "string"  
  },  
  "experienceAnalysis": [  
    {  
      "experienceId": "uuid",  
      "responsibilityAnalysis": {  
        "strengths": "string",  
        "weaknesses": "string",  
        "recommendations": "string"  
      },  
      "achievementsAnalysis": {  
        "strengths": "string",  
        "weaknesses": "string",  
        "recommendations": "string"  
      }  
    }  
  ],  
  "generalAnalysis": {  
    "strongSides": "string",  
    "weakSides": "string",  
    "overallRecommendations": "string"  
  }  
}
```

Массив experienceAnalysis должен соответствовать количеству анализируемых опытов.

7. Промпты запроса в YandexGPT

7.1 Системный промпт

Ты -- профессиональный HR-аналитик с опытом более 15 лет в подборе IT-специалистов, оценке резюме и карьерном консультировании.

Твоя задача -- выполнить структурированный анализ текста резюме.

КРИТИЧЕСКИ ВАЖНО:

1. Верни ТОЛЬКО валидный JSON.
2. Не добавляй пояснений.
3. Не добавляй markdown.
4. Не добавляй комментариев.
5. Не добавляй текст вне JSON.
6. Не добавляй лишние поля.
7. Не обрамляй JSON в кавычки.
8. Ответ должен начинаться с { и заканчиваться }.
9. Все значения должны быть строками.
10. Если информации недостаточно -- укажи это явно текстом в соответствующем поле.

JSON должен строго соответствовать следующей структуре:

```
{  
    "aboutAnalysis": {  
        "strengths": "string",  
        "weaknesses": "string",  
        "recommendations": "string"  
    },  
    "experienceAnalysis": [  
        {  
            "experienceId": "uuid",  
            "responsibilityAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            },  
            "achievementsAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            },  
            "educationAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            },  
            "workExperienceAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            },  
            "softSkillsAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            },  
            "personalQualitiesAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            },  
            "languageSkillsAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            },  
            "otherAnalysis": {  
                "strengths": "string",  
                "weaknesses": "string",  
                "recommendations": "string"  
            }  
        }  
    ]  
}
```

```
        "weaknesses": "string",
        "recommendations": "string"
    }
}
],
"generalAnalysis": {
    "strongSides": "string",
    "weakSides": "string",
    "overallRecommendations": "string"
}
}
```

Массив `experienceAnalysis` должен соответствовать количеству анализируемых мест работы.

Требования к содержанию:

- Анализ должен быть профессиональным.
- Формулировки конкретные и практические.
- Избегай общих фраз.
- Давай рекомендации, которые можно применить.
- Используй деловой стиль.
- Не повторяй дословно текст пользователя.

Если ты добавишь текст вне `JSON`, это будет считаться ошибкой.

Если структура будет нарушена, ответ будет отклонен.

7.2 Пользовательский промпт

Проанализируй следующие разделы резюме:

О СЕБЕ:

`{{about_text}}`

ОПЫТ РАБОТЫ:

[

```
{
    "experience_id": "{{uuid_1}}",
    "company_name": "{{company_1}}",
    "position": "{{position_1}}",
    "responsibilities": "{{responsibilities_text_1}}",
    "achievements": "{{achievements_text_1}}"
},
```

{

```
        "experience_id": "{{uuid_2}}",
        "company_name": "{{company_2}}",
        "position": "{{position_2}}",
        "responsibilities": "{{responsibilities_text_2}}",
        "achievements": "{{achievements_text_2}}"
    }
]
```

Проанализируй каждый раздел отдельно, затем сделай общий вывод.

Критерии анализа:

1. Структурированность текста
2. Конкретика формулировок
3. Наличие измеримых результатов
4. Профессиональная зрелость
5. Уровень экспертности
6. Читаемость и ясность

Рекомендации должны быть:

- конкретными
- применимыми
- ориентированными на улучшение качества резюме

Ответ верни строго в формате JSON, определенном в system-сообщении.

8. Пример ответа YandexGPT

```
{
    "aboutAnalysis": {
        "strengths": "Четко обозначена профессиональная роль и опыт работы. Формулировки отражают специализацию и понимание своей зоны ответственности. Текст написан деловым языком.",
        "weaknesses": "Недостаточно конкретики по стеку технологий и типам проектов. Отсутствует позиционирование по уровню (Middle/Senior). Нет указания на бизнес-контекст выполняемых задач.",
        "recommendations": "Добавить ключевые технологии и инструменты. Уточнить специализацию (например, системный аналитик в финтех-проектах). Кратко обозначить масштаб проектов и тип заказчиков."
    },
    "experiences": [
```

```
{  
    "experience_id": "exp-id-1",  
    "strengths": [  
        "Указаны конкретные технологии",  
        "Описание обязанностей отражает участие в разработке",  
        "Есть упоминание оптимизации"  
    ],  
    "weaknesses": [  
        "Нет количественных показателей",  
        "Не указан масштаб проектов",  
        "Не раскрыто влияние на бизнес"  
    ],  
    "recommendations": [  
        "Добавить метрики производительности",  
        "Указать количество пользователей или нагрузку",  
        "Показать бизнес-результат реализованных решений"  
    ]  
},  
{  
    "experience_id": "exp-id-2",  
    "strengths": [],  
    "weaknesses": [],  
    "recommendations": []  
}  
],  
"generalAnalysis": {  
    "strongSides": "Резюме демонстрирует опыт, ориентацию на результат и понимание процессов разработки. Есть количественные показатели и структура.",  
    "weakSides": "Недостаточно технической детализации и позиционирования на рынке. Часть формулировок требует конкретизации.",  
    "overallRecommendations": "Добавить стек технологий, уточнить уровень позиции, усилить достижения через метрики и контекст. Сделать формулировки более конкретными и ориентированными на бизнес-ценность."  
}  
}
```

9. Валидация ответа AI

9.1 Проверки

- JSON парсится
 - Есть обязательные блоки
 - `experienceAnalysis.length == input.length`
 - Каждое поле строка
 - Размер ≤ 12 000 символов

9.2 Лишние поля

Игнорируются.

9.3 Некорректный JSON

→ Retry
→ При 3 ошибке → FAILED + refund

9. Доработка БД и маппинг данных

9.1 Бд → YandexGPT

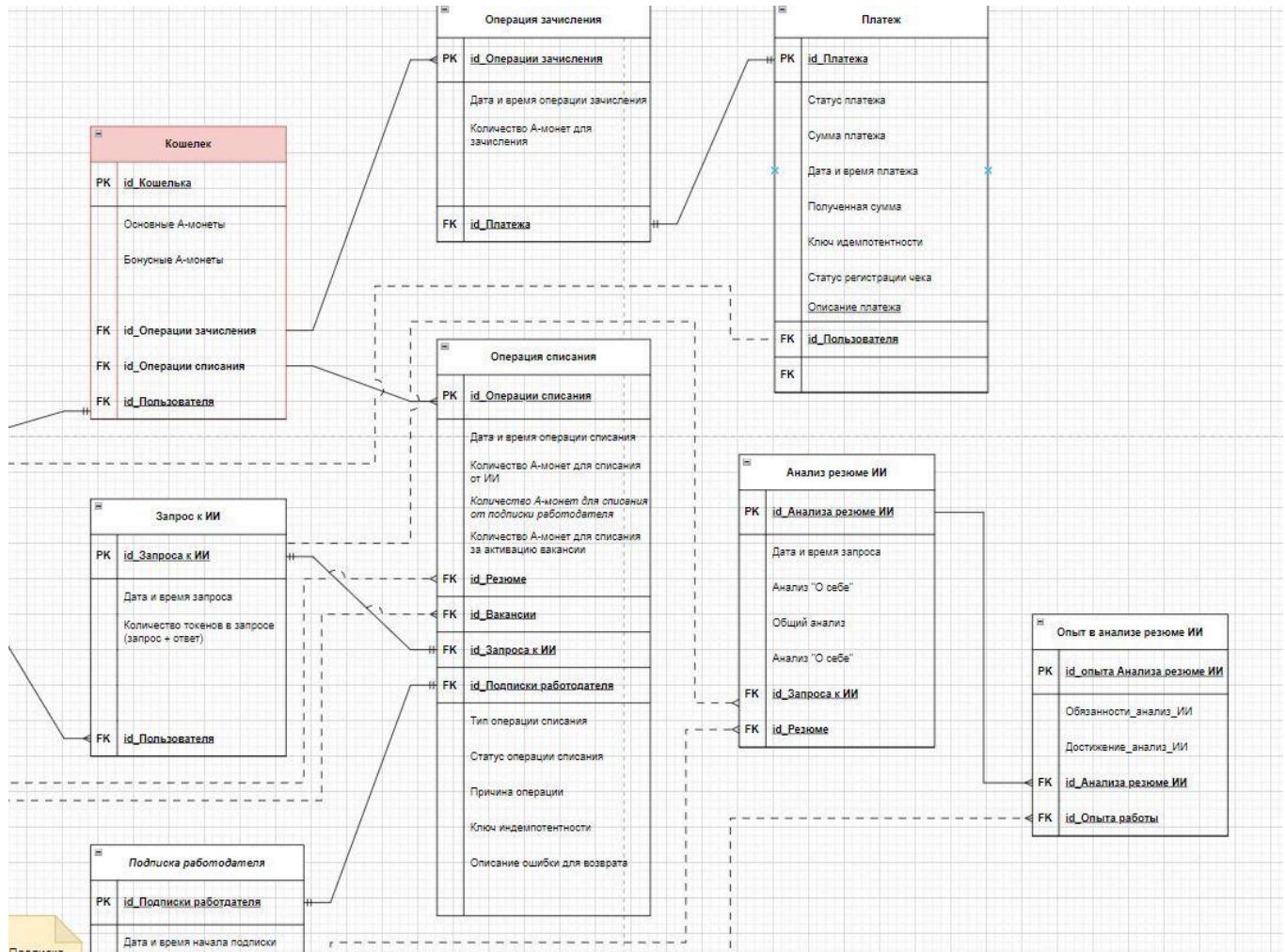
БД	Prompt
resume.about_self	О СЕБЕ
experience.responsibility	Обязанности
experience.achievements	Достижения

9.2 YandexGPT → Бд

Ответ А | Таблица | Поле

aboutAnalysis	resume_ai_analysis	about_analysis
generalAnalysis	resume_ai_analysis	general_analysis
experienceAnalysis[]	resume_experience_ai_analysis	responsibility_analysis / achievements_analysis

9.3 Схема ERD добавляемых таблиц



9.4 Добавляемые таблицы

9.4.1 Добавление в таблицу Операции списания:

Элемент данных	Описание	Состав или тип данных	Длина	Значения	Обязательность
type	Тип операции списания	varchar	20	DEBIT, CREDIT	да
status	Статус операции списания	varchar	20	PENDING, PROCESSING, SUCCESS, FAILED, REFUND_INIT, REFUND_SUCCESS, REFUND_FAILED	да
reason	Причина операции	varchar	100	AI_ANALYSIS, AI_ANALYSIS_REFUND	нет
indempotency_key	Ключ идемпотентности операции	varchar	100	Буквенно-числовое	нет
failure_reason	Описание ошибки операции	varchar	255	Буквенно-числовое	нет

9.4.2 Создание таблицы “Анализ резюме ИИ”

Элемент данных	Описание	Состав или тип данных	Длина	Значения	Обязательность
id	Уникальный номер анализа резюме ИИ	uuid			да
created_at	Дата и время анализа	timestampz			да

about_analyses	Анализ “О себе”	text	50-3000	Буквенно-числовое	да
general_analysis	Общий анализ резюме	text	50-3000	Буквенно-числовое	да

9.4.3 Создание таблицы “Опыт в анализе резюме ИИ”

Элемент данных	Описание	Состав или тип данных	Длина	Значения	Обязательность
id	Уникальный номер опыта анализа резюме ИИ	uuid			да
responsibility_analysis	Анализ Обязанностей	text	50-3000	Буквенно-числовое	да
achievements_analysis	Анализ достижений	text	50-3000	Буквенно-числовое	да

10. Retry-логика

3 попытки

exponential backoff (1s → 2s)

circuit breaker при 5 подряд ошибках

11. А-Монеты

11.1 SAGA - Бизнес-транзакция

Шаги:

1. Валидация входных данных
2. Проверка баланса
3. Создание транзакции DEBIT (PENDING)
4. Создание записи анализа (PROCESSING)
5. Вызов AI
6. Валидация полученных данных
7. Сохранение результата
8. Обновление транзакции SUCCESS

Компенсация

При любой ошибке:

- транзакция → FAILED
- создаётся CREDIT (AI_ANALYSIS_REFUND)

11.2 Идемпотентность

Используется заголовок:

[Idempotency-Key](#)

Backend:

- сохраняет ключ
- при повторе возвращает существующий analysisId
- предотвращает повторное списание

11.3 Списание А-монет

Создается транзакция:

Поле	Значение
------	----------

type	DEBIT
------	-------

reason	AI_ANALYSIS
--------	-------------

status	PENDING
--------	---------

После успешного ответа → SUCCESS.

11.4 Возврат А-Монет

Если 3 неудачных попытки:

Поле	Значение
------	----------

type	CREDIT
------	--------

reason	AI_ANALYSIS_REFUND
--------	--------------------

status	SUCCESS
--------	---------

11.5 State Machine — Транзакция А-Монет (AI-анализ)

```
@startuml
```

```

title State Machine -- A-Coins Transaction (AI Analysis)

[*] --> INIT : Создание транзакции

INIT --> VALIDATION : Проверка баланса

VALIDATION --> REJECTED : Недостаточно средств
VALIDATION --> PENDING : Баланс достаточен

state PENDING {
    PENDING : type = DEBIT\nstatus = PENDING

    PENDING --> PROCESSING : Отправка запроса в AI
    PROCESSING --> PROCESSING : Retry 1 (Timeout)
    PROCESSING --> PROCESSING : Retry 2 (Timeout)

    PROCESSING --> SUCCESS : Успешный ответ AI
    PROCESSING --> FAILED : 3-й Timeout
    PROCESSING --> FAILED : Некорректный JSON
    PROCESSING --> FAILED : Ошибка интеграции
}

SUCCESS : status = SUCCESS\nСредства списаны
FAILED : status = FAILED

FAILED --> REFUND_INIT : Запуск компенсации

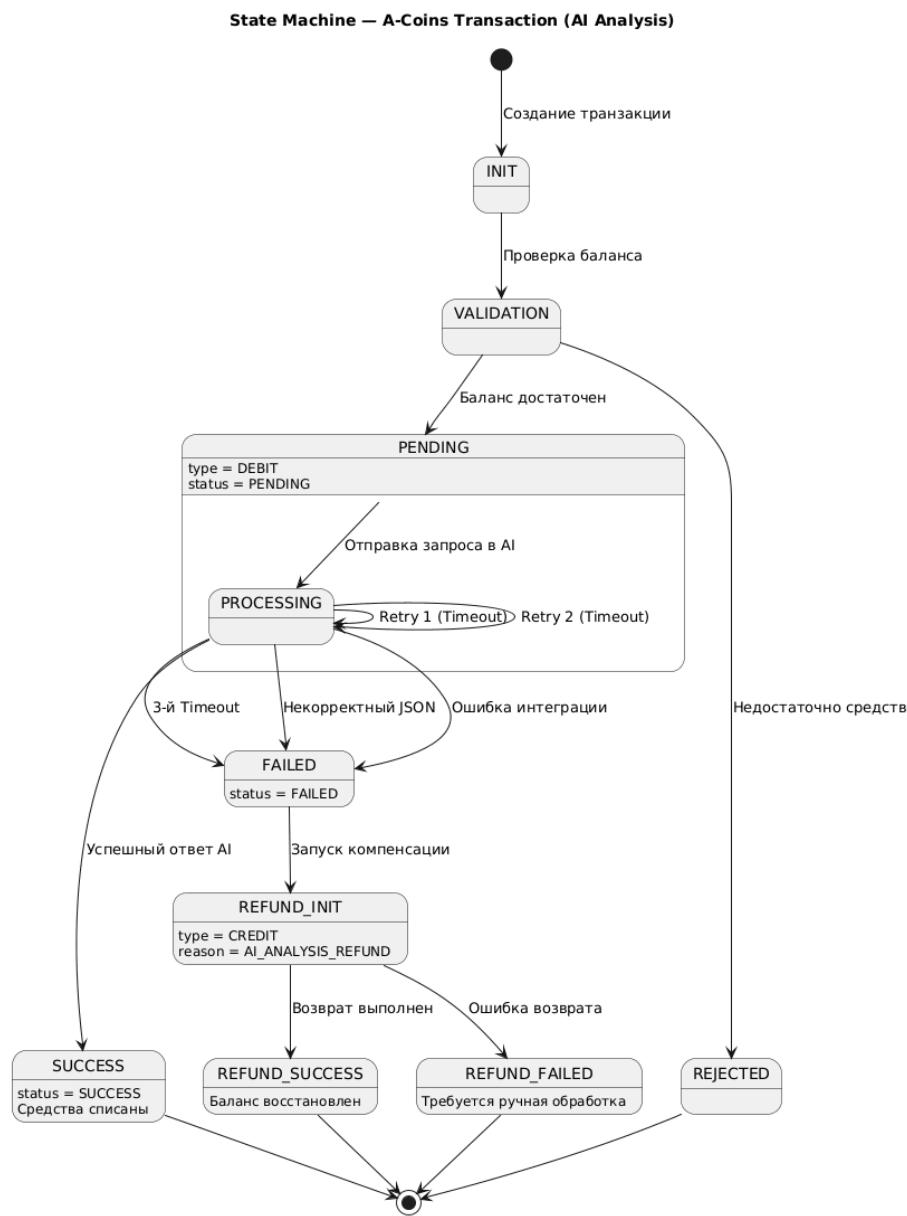
state REFUND_INIT {
    REFUND_INIT : type = CREDIT\nreason = AI_ANALYSIS_REFUND
    REFUND_INIT --> REFUND_SUCCESS : Возврат выполнен
    REFUND_INIT --> REFUND_FAILED : Ошибка возврата
}

REFUND_SUCCESS : Баланс восстановлен
REFUND_FAILED : Требуется ручная обработка

SUCCESS --> [*]
REJECTED --> [*]
REFUND_SUCCESS --> [*]
REFUND_FAILED --> [*]

@enduml

```



12. Статусы анализа

Статус	Описание
PROCESSING	Идёт анализ
SUCCESS	Завершено
FAILED	Ошибка

13. Нефункциональные требования

Требование	Значение
Максимальное время ответа	30 секунд
Логирование текста резюме	Запрещено
Логирование ответа AI	Запрещено
Соединение	HTTPS
Rate limit	Обработка 429
Размер запроса	$\leq 5\ 000$ токенов
Размер ответа	$\leq 12\ 000$ символов

14. Ошибки API

Код	Причина
400	Невалидные поля
401	Нет авторизации
403	Не владелец

402	Недостаточно средств
409	Idempotency conflict
429	Слишком частый запрос
500	Ошибка обработки
504	Timeout

15. Use Case

15.1 Общее описание

Use Case описывает процесс запуска AI-анализа резюме соискателя, отправки данных в YandexGPT, получения структурированного анализа, сохранения результата и отображения его пользователю.

15.2 Акторы

Актор	Тип	Описание
Соискатель	Основной	Авторизованный пользователь — владелец резюме
Web Client	Системный	Web-интерфейс
Mobile App	Системный	iOS / Android
Backend	Системный	Бизнес-логика
YandexGPT API	Внешняя система	LLM-анализ
Billing Service	Системный подсервис	Управление А-Монетами
БД	Системный компонент	Хранение данных

15.3 Предусловия

1. Пользователь авторизован.
2. Резюме существует.
3. Пользователь является владельцем резюме.
4. Поля:
 - "О себе"

У каждого опыта:

- обязанности \geq 50 символов
- достижения \geq 50 символов

Анализируются места работы за последние 5 лет, но не более 5 последних мест работы

5. У пользователя достаточно А-Монет.
6. С момента последнего анализа прошло \geq 30 секунд.

15.4 Постусловия

При успешном завершении:

- Анализ сохранен в БД.
- Транзакция А-Монет переведена в SUCCESS.
- Пользователь получил структурированный результат.

При неуспешном завершении:

- А-Монеты возвращены (если были списаны).
- Пользователь получил сообщение об ошибке.

15.5 Основной сценарий

1. Пользователь инициирует анализ

1.1 Соискатель нажимает кнопку «Анализировать с помощью ИИ».

1.2 Client отправляет:

```
POST /api/v1/resumes/{resumeId}/ai-analysis
```

2. Backend выполняет проверки

2.1 Проверка авторизации (JWT).

2.2 Проверка владения резюме.

2.3 Проверка обязательных полей:

- trim()
- not null
- длина ≥ 50
- длина поля ≤ 3 000
- удаление HTML
- проверка UTF-8

2.4 Проверка интервала 30 секунд.

2.5 Проверка достаточности А-Монет.

3. Списание А-Монет

3.1 Создание транзакции:

- type = DEBIT
- status = PENDING
- reason = AI_ANALYSIS

4. Формирование запроса к YandexGPT

4.1 Проведение токен контроля, что по предварительной оценке в запросе менее 5000 токенов

4.2 Формирование prompt (места работы за последние 5 лет, но не более 5)

4.3 Формирование JSON-запроса.

4.4 Отправка запроса в YandexGPT.

5. Получение ответа

5.1 Backend ожидает ответ ≤ 30 секунд.

5.2 Получает JSON.

5.3 Выполняет валидацию:

- JSON валиден
- 4 обязательных блока
- обязательные поля внутри
- нет лишних полей

- размер ≤ 12 000 символов
- валидность JSON
- наличие массива experienceAnalysis
- совпадение количества элементов

6. Сохранение результата

6.1 Сохранение анализа в БД.

- about_analysis
- general_analysis
- для каждого опыта отдельную запись

6.2 Обновление транзакции:

- status = SUCCESS

7. Ответ клиенту

7.1 Возврат 200 OK.

7.2 Передача анализа.

7.3 Передача nextAvailableAt.

15.6 Альтернативные сценарии

1А — Пользователь не авторизован

Возникает на шаге 2.1

Действия:

- Backend возвращает 401 Unauthorized
- Процесс завершается
- Транзакции не создаются

2А — Пользователь не владелец резюме

Возникает на шаге 2.2

Действия:

- Backend возвращает 403 Forbidden
- Процесс завершается

2B — Обязательные поля не заполнены / < 50 символов

Возникает на шаге 2.3

Действия:

- Backend возвращает 400 Bad Request
- В теле ответа — указание конкретного поля
- Кнопка на клиенте блокируется

2C — Нарушен интервал 30 секунд

Возникает на шаге 2.4

Действия:

- Backend возвращает 429 Too Many Requests
- В ответе передается nextAvailableAt
- Анализ не запускается

2D — Недостаточно А-Монет

Возникает на шаге 2.5

Действия:

- Backend возвращает 402 Payment Required
- Процесс завершается

3A — Ошибка при создании транзакции

Возникает на шаге 3.1

Действия:

- Backend возвращает 500
- Процесс завершается

4A — Ошибка формирования запроса

Возникает на шаге 4.1

Причины:

- Превышение лимита размера запроса

Действия:

- Backend убирает в файле запроса самый ранний опыт.
- заново проводится токен контроль (шаг 4.1)

5A — Timeout (1 попытка)

Возникает на шаге 5.1

Действие: Повторный запрос (Retry 1)

5B — Timeout (2 попытка)

Действие: Повторный запрос (Retry 2)

5C — Timeout (3 попытка)

Действия:

- Отмена операции
- Возврат А-Монет (CREDIT)
- Возврат 504 Gateway Timeout

5D — Некорректный JSON от YandexGPT

Возникает на шаге 5.3

Проверки:

- Не парсится JSON
- Нет обязательных блоков
- Лишние поля
- Пустые значения

Действия:

- Ошибка 500
- Возврат А-Монет

5E — Несовпадение количества мест работы

Если модель вернула:

- меньше блоков

- больше блоков

→ Ошибка
→ FAILED
→ Возврат монет

5F — Превышен лимит токенов

→ Анализируется только столько последних опытов, сколько помещается в лимит 12000 символов..

6A — Ошибка сохранения в БД

Возникает на шаге 6.1

Действия:

- Транзакция переводится в FAILED
- Возврат А-Монет
- 500 Internal Server Error

6B — Ошибка сохранения одного из места работы

→ Откат всей бизнес-транзакции
→ Возврат монет

7A — Ошибка при обновлении транзакции

Возникает на шаге 6.2

Действия:

- Логирование
- Создание compensating transaction
- 500 Internal Server Error

15.7 Гарантии целостности

- Списание и сохранение анализа — в рамках одной бизнес-транзакции
- Возврат А-Монет при любой технической ошибке
- Анализ сохраняется только при SUCCESS
- Exactly-once списание
- Token-контроль

15.8 Ограничения

Параметр	Значение
Минимальная длина текста	50 символов
Максимальная длина текста запроса	9 000 / 5000 токенов
Максимальная длина текста ответа	12 000
Таймаут	30 секунд
Retry	2
Повторный анализ	≥ 30 секунд

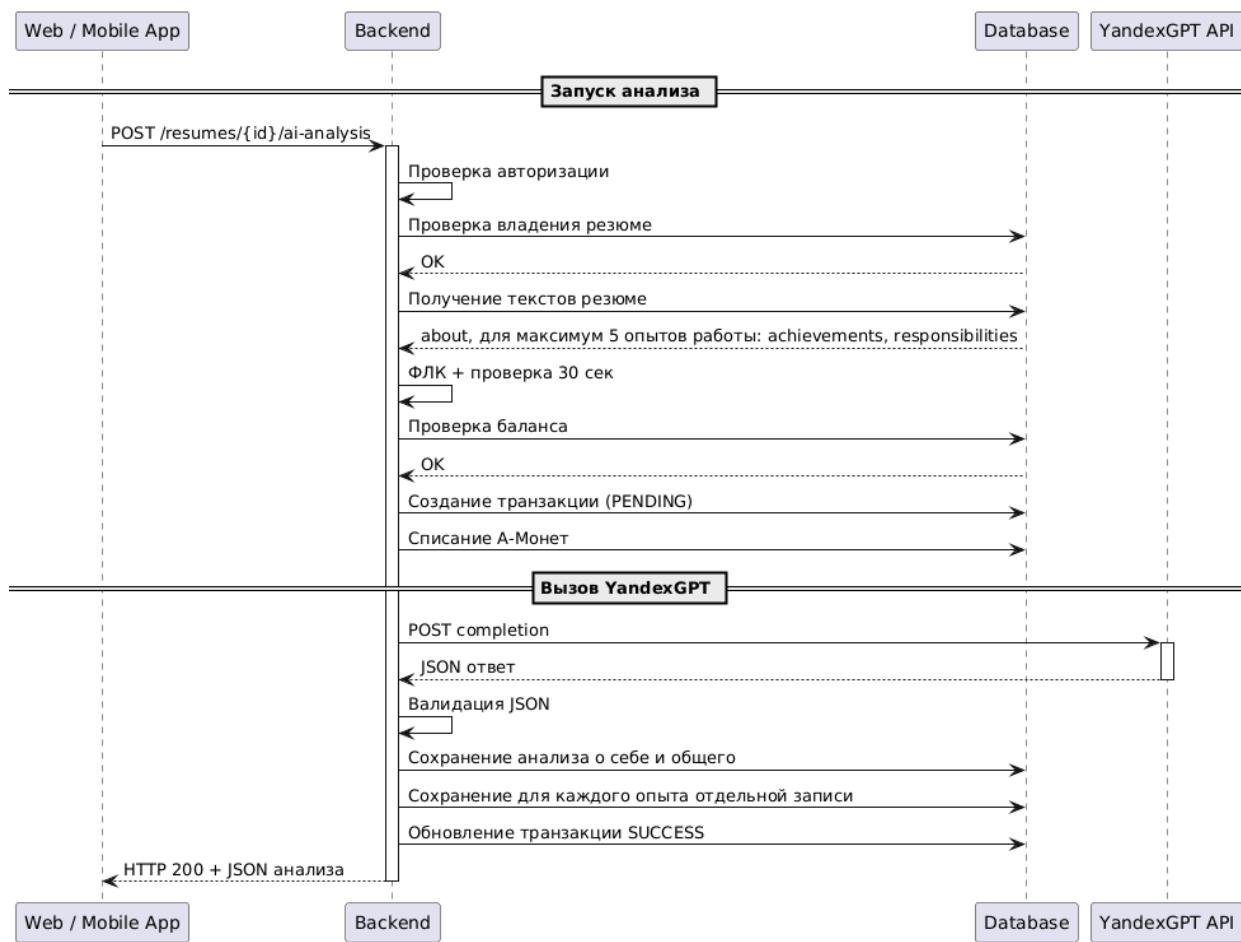
16. Sequence UML (PlantUML)

```
@startuml
participant "Web / Mobile App" as Client
participant "Backend" as Backend
participant "Database" as DB
participant "YandexGPT API" as GPT
== Запуск анализа ==
Client -> Backend: POST /resumes/{id}/ai-analysis
activate Backend
Backend -> Backend: Проверка авторизации
Backend -> DB: Проверка владения резюме
DB --> Backend: OK
Backend -> DB: Получение текстов резюме
DB --> Backend: about, для максимум 5 опытов работы: achievements,
responsibilities
Backend -> Backend: ФЛК + проверка 30 сек
Backend -> DB: Проверка баланса
DB --> Backend: OK
Backend -> DB: Создание транзакции (PENDING)
Backend -> DB: Списание А-Монет
== Вызов YandexGPT ==
Backend -> GPT: POST completion
activate GPT
```

```

GPT --> Backend: JSON ответ
deactivate GPT
Backend -> Backend: Валидация JSON
Backend -> DB: Сохранение анализа о себе и общего
Backend -> DB: Сохранение для каждого опыта отдельной записи
Backend -> DB: Обновление транзакции SUCCESS
Backend --> Client: HTTP 200 + JSON анализа
deactivate Backend
@enduml

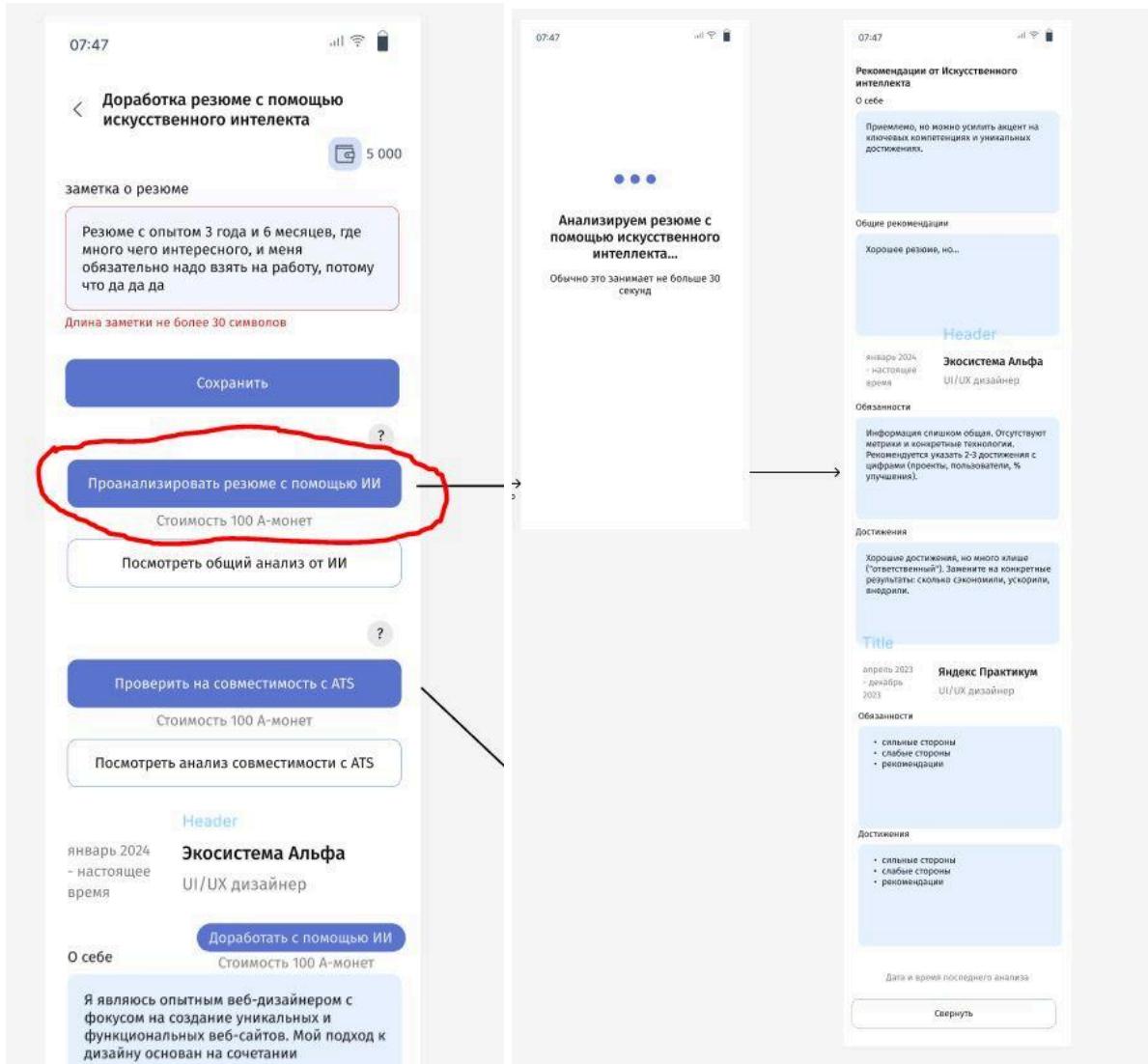
```



17. Экранные формы

Созданы предположительные экранные формы для Android

<https://www.figma.com/design/JJdiqKEZt6dn6tT3Mv3nJz/%D0%90-%D0%A0%D0%B0%D0%B0%D0%B1%D0%BE%D1%82%D0%B0-%D0%A1%D0%90?node-id=248-12372&m=dev>



18. Этапы разработки

- Дизайн
- Бэкэнд
- Фронтенд
- Тестирование