

Задание для группы разработки

0. Назначение документа

Сформулировать состав работ, критерии приёмки и артефакты по разработке мотивационного модуля геймификации для кадровой системы. Документ служит рабочим ТЗ для команды разработки.

1. Цели и ключевые показатели

Цель: связать разрозненные задачи кандидатов и сотрудников в прозрачные «миссии» с прогрессом и целями, повысить вовлечённость.

KPI пилота: - $\geq 70\%$ пользователей, начавших онбординг, завершают хотя бы 1 миссию. - $\geq 50\%$ пользователей, выполнивших 2+ миссии, поднимают ранг хотя бы на 1 уровень. - Время до «осязаемой награды» (опыт/мана/артефакт) ≤ 3 шага.

2. Объём работ (Scope)

2.1 Пользовательские функции (Mobile-first)

1) **Ранги:** линейная шкала; авто-повышение по 3 условиям (опыт, ключевые миссии, уровни компетенций). 2) **Миссии:** каталог по категориям (Квесты, Рекрутинг, Лекторий, Симулятор); карточка миссии; выполнение с возможностью загрузки файла/ссылки. 3) **Ветвление:** миссии объединяются в «ветки»/цепочки; пользователь видит прогресс по ветке. 4) **Бортовой журнал:** хронология действий, прогресса, наград; еженедельные/месячные/годовые топы. 5) **Компетенции:** экран навыков с уровнями; прокачка от миссий. 6) **Хранилище (магазин):** трата «маны/кредитов» на призы/бонусы; история покупок. 7) **Артефакты:** коллекция с карточками артефактов (изображение, название, описание, редкость). 8) **Онбординг и лор:** короткие сюжетные вставки, объясняющие механику и «космический» контекст.

2.2 Функции для HR/Организатора (Desktop)

1) **Конструктор миссий:** поля (название, описание, опыт, мана, доступность по рангу, эффекты на компетенции, артефакт по итогу). 2) **Редактор веток:** визуальная связь миссий; статусы, приоритеты. 3) **Управление рангами:** правила повышения (порог опыта, обязательные миссии, уровни компетенций). 4) **Модерация:** очередь подтверждений с прикрепленными файлами/ссылками. 5) **Артефакты:** CRUD + атрибуты артефактов. 6) **Статистика:** конверсии по миссиям/веткам/пользователям; выгрузки.

2.3 Интеграции и API

- SSO/аутентификация: JWT; интеграция с внешним порталом (на этапе пилота — мок).
- Источники задач (пул миссий из внешних систем) — мок-адаптер.
- Публичный API: чтение миссий, выполнение, прогресс, лидерборды.
- Webhooks: события выполнения/ранга/покупки.

Вне score пилота: расчёт денежной стоимости призов, платежи, сквозная поставка мерча.

3. Ограничения и допущения

- Пилот: один сценарий e2e «кандидат получает оффер»; одна ветка (3–5 миссий), 2–3 ранга, 5–7 компетенций, 2–3 артефакта, 3–5 товаров.
 - Данные и интеграции — замockаны; реальная интеграция после пилота.
 - Темизация «космос»: палитра/иконки/фон; переключаемые темы.
-

4. Архитектура и стек

Фронтенд: React + TypeScript, Router, Zustand/Redux Toolkit; CSS-модули или CSS-in-JS; Mobile-first для пользователя, Desktop для HR.

Бэкенд: Python (FastAPI либо Django), БД: SQLite (пилот) → PostgreSQL (прод); JWT; миграции.

Сервис прогресса (Rule Engine): отдельный модуль, реагирующий на события и проверяющий 3 условия ранга.

Событийная шина: минимально — таблица/очередь; в перспективе — брокер.

Файлы: объектное хранилище (локально — каталог + антивирус-сканер), подписанные ссылки.

Обсервабилити: логирование (структурированное), метрики (RPS, p95), трассировки.

5. Доменная модель (черновик)

User(id, role, rank_id, xp, mana, created_at)

Rank(id, name, order, xp_required, meta)

RankRule(rank_id, required_missions[], required_competency_levels{competency_id:level})

Competency(id, name, desc)

UserCompetency(user_id, competency_id, level, progress)

Mission(id, title, description, category, min_rank_id, rewards{x p, mana, artifact_id?}, effects{competency_id:delta})

Branch(id, name, desc)

BranchEdge(branch_id, mission_id, next_mission_id?)

MissionSubmission(id, mission_id, user_id, payload{file/url/text}, status)

Artifact(id, image_url, name, description, rarity)

StoreItem(id, name, price_mana, stock?, meta)

Order(id, user_id, item_id, status)

JournalEntry(id, user_id, type, payload, created_at)

— связи: User→Rank (M:1); User↔Competency (M:N); Branch↔Mission (M:N через Edge);

6. API (MVP)

Публичное API (пользователь)

- GET /api/me — профиль, прогресс, текущий ранг.
- GET /api/missions?rank=...&branch=... — список.
- GET /api/missions/{id} — карточка миссии.
- POST /api/missions/{id}/submit — выполнение (файл/ссылка/форм-данные).
- GET /api/journal — бортовой журнал, топы (?period=week|month|year).
- GET /api/skills — компетенции пользователя.
- GET /api/store/items — каталог; POST /api/store/purchase — покупка.

Админ API (HR)

- CRUD /api/admin/missions, /api/admin/branches, /api/admin/ranks, /api/admin/competencies, /api/admin/artifacts.
- GET /api/admin/moderation → очередь; POST /api/admin/submissions/{id}/approve|reject.
- GET /api/admin/analytics/* — агрегаты по конверсиям.

Контракты: спецификация OpenAPI 3.1 (в репозитории /openapi.yaml).

7. НФТ (Нефункциональные требования)

- **Производительность:** p95 отклик API ≤ 200 мс при 100 RPS (пилот); начальная нагрузка 1k MAU.
 - **Доступность:** 99,0% в неделю (пилот).
 - **Адаптивность:** мобильные экраны ≥ 320 px; HR-панель ≥ 1024 px.
 - **Доступность (a11y):** соответствие WCAG 2.2 AA ключевых пользовательских потоков.
 - **Безопасность:** базовый уровень соответствия (OWASP ASVS Level 2): аутентификация, управление сессиями, авторизация по ролям, валидация входных данных, безопасное хранение файлов, защита от XSS/CSRF/SQLi, журнал аудита админ-действий.
 - **Качество продукта:** ориентироваться на характеристики ISO/IEC 25010 (функциональная пригодность, надёжность, производительность, удобство, безопасность, сопровождаемость и др.).
-

8. UX/UI

- Космическая тема: палитра, фон, иконки; нейминг рангов.
- Экран пользователя: цель, прогресс-бар до следующего ранга, доступные ветки и миссии, быстрый старт.
- Экран HR: конструктор миссий (формы), редактор веток, модерация, аналитика.

Артефакты дизайн-системы: палитра, типографика, кнопки, карточки, формы, тосты, прогресс-бар, модальные окна.

9. Приёмка (Definition of Done)

- Реализованы фичи разделов 2.1–2.2 (MVP-объём).
 - Покрытие интеграционными тестами ключевых сценариев; unit-тесты Rule Engine.
 - Линтеры и форматтеры подключены; CI выполняет тесты и линт.
 - Сгенерирован `openapi.yaml`; README с инструкциями; сид-данные.
 - Демонстрация e2e «кандидат получает оффер» на стенде.
-

10. Критерии приёмки (Gherkin, выборочно)

Сценарий: Авто-повышение ранга по 3 условиям

```
Given у пользователя ранг R0 и 0 XP
And есть правило повышения до R1: XP≥500, миссии [M1, M2], компетенции {Аналитика≥1, Общение≥1}
When пользователь завершает M1 и M2 и набирает ≥500 XP
And уровни компетенций удовлетворяют порогам
Then система повышает ранг до R1
And в журнал добавляется запись о рангапе
```

Сценарий: Выполнение миссии с модерацией

```
Given миссия M3 требует загрузки файла и модерации
When пользователь отправляет отчёт
Then запись попадает в очередь модерации со статусом "На проверке"
When HR одобряет отчёт
Then пользователь получает награды (XP, мана, артефакт?) и эффекты на компетенции
And в журнале фиксируется операция
```

Сценарий: Покупка в магазине

```
Given у пользователя 150 маны и товар T ценой 100
When пользователь оформляет покупку T
```

Then списывается 100 маны
And создаётся заказ в статусе "Ожидает выдачи"

Сценарий: Ветвление миссий

Given ветка B: M1 -> M2 -> M3
When пользователь завершает M1
Then M2 становится доступной

11. Тестирование

- **Unit:** правила рангов, расчёт наград, транзакции маной.
- **Интеграционное:** миссии→журнал, модерация, покупки.
- **E2E:** ключевой пользовательский сценарий.
- **Безопасность:** статический анализ, базовые проверки ASVS L2.
- **Доступность:** линтеры a11y, ручные проверки по чек-листу AA.

12. CI/CD и качество

- GitHub Actions: линт, тесты, сборка докер-образов.
- Превью-развёртывания для PR (frontend + backend).
- Теги релизов, CHANGELOG, семантическое версионирование.

13. Сид-данные (минимум)

- 2-3 ранга (примерные названия), пороги XP и требования.
- 1 ветка на 3-5 миссий (микс категорий).
- 5-7 компетенций с базовыми уровнями.
- 2-3 артефакта разной редкости.
- 3-5 товаров магазина.
- 3 тестовых пользователя (Кандидат, HR, Организатор).

14. План работ (ориентир, 2 спринта по 2 недели)

Спринт 1: доменная модель, CRUD миссий/веток/артефактов, подача списка миссий, выполнение без модерации, журнал, магазин (чтение/покупка), сид-данные, базовый UI.

Спринт 2: Rule Engine (3 условия ранга), модерация, топы, аналитика HR (минимум), онбординг/лор, шлифовка UX, a11y, подготовка к демо.

15. Риски

- Сложность правил рангов → изоляция в Rule Engine + тесты.
 - Нагрузочные пики (рейтинги) → кэширование/агрегаты.
 - Мошенничество (накрутка) → rate-limit, аудиты, дедупликация событий.
-

16. Артефакты, которые нужно предоставить

- Код (frontend/backend), Docker-композиция, сид-данные.
 - OpenAPI 3.1 спецификация.
 - README: запуск, архитектура, ограничения, ссылки на стенд.
 - Демо-скрипт e2e и видео/скринкаст.
-

17. Глоссарий

Пользователь, HR, Организатор, Опыт (XP), Мана, Ранг, Артефакты, Компетенции, Миссии, Ветвление, Бортовой журнал.