



## ТЕХНИЧЕСКОЕ ЗАДАНИЕ ЗАДАЧА 9

Создание мотивационного  
модуля геймификации  
для кадровой системы «Алабуги»



## Суть

В ОЭЗ «Алабуга» работает более 26 000 сотрудников, и ежемесячно к нам приезжают тысячи кандидатов. Как перед трудоустройством, так и во время работы у специалистов есть определённые задачи, ведущие к достижению их целей.

Сейчас задачи приходят разрозненно, из-за чего между ними не хватает общей связи, и нет ощущения, что один шаг сейчас — это большой вклад в будущее. В связи с этим мы хотим разработать геймифицированную платформу, на которой пользователи смогут выполнять различные задачи на любом этапе, отслеживать свой прогресс и видеть, что ещё нужно сделать для достижения цели.

## Тематика

Корпоративная культура «Алабуги» всегда учила нас решать сверхзадачи и стремиться к звёздам. На крыше пирамиды одного из офисов «Алабуги» расположен прототип советского орбитального ракетоплана «Буран», напоминая о том, что в каждом из нас живёт частичка ДНК технологических изобретателей. А кроме того, в планах компании на ближайшие 25 лет — начать освоение космоса и колонизацию лун Юпитера.

В связи с этим тематику геймифицированной платформы хотелось бы видеть связанной с тематикой космоса, где пользователи проходят путь от космических пилотов-кандидатов до командиров космических подразделений.

## Терминология

- Пользователь — кандидат или сотрудник, пользователь платформы.
- HR — сотрудник, отвечающий за разработку заданий в платформе.
- Организатор — сотрудник, проводящий мероприятие (миссию).
- Опыт — очки прогресса, необходимые для повышения ранга.
- Мана — очки игровой валюты, за которые можно приобрести определённые бонусы.
- Ранг — игровое звание пользователя. Открывает новые задачи и цели.
- Артефакты — знаки отличия за миссии.
- Компетенции — определённые навыки со шкалами прогресса.

# 1. Основные механики

## 1. Ранги

Выполняя различные задачи на всех этапах, пользователи получают опыт и ману, которые повышают их ранг. Ранги расположены в линейной последовательности, и нельзя через них перескакивать. Для повышения ранга есть 3 условия:

- Первое — достаточное количество опыта, полученного при выполнении заданий.
- Второе — выполнение определённых заданий, необходимых для желаемого грейда.
- Третье — получение необходимого уровня прокачки конкретных компетенций.

### **Пример:**

кандидат хочет получить оффер. Чтобы получить оффер, кандидату необходимо набрать 500 очков опыта, выполнить задания (загрузка документов, заполнение резюме, выбор направлений) и прокачать компетенции «Общение» и «Аналитика» до 1 балла. Выполнив все условия, кандидат сможет получить оффер.

<b>Примеры</b>	<b>для</b>	<b>нейминга</b>	<b>рангов:</b>
искатель, разведчик, навигатор, пилот-кандидат, принятый в экипаж, пилот-испытатель, лидер эскадрильи, командир космического поселения и т. д.			

Со стороны HR необходимо сделать возможность настраивать условия для получения рангов:

- Опыт: [NNN]
- Ключевые задания: [mission1, mission2]
- Уровень компетенций: [competention=N]

## 2. Миссии

Миссии — это список заданий, доступных пользователю. Список миссий должен меняться в зависимости от ранга. Открыв миссию, пользователь должен ознакомиться со всеми условиями и иметь возможность перейти к действию.

Примеры миссий: сбор документов, заполнение резюме, прохождение бизнес-симуляций, проезд на очный этап, прохождение собеседования, прохождение онбординга, выполнение плана на месяц, участие в ежегодном ассессменте и т. д.

Минимальный список параметров для миссии:

- Название миссии
- Описание миссии
- Награда в опыте
- Награда в мане
- Доступность по рангу
- Какие компетенции на сколько прокачиваются
- Дополнительно: будет здорово за некоторые миссии выдавать особые награды — артефакты

### 3. Ветвление

Миссии должны быть связанными, а не сами по себе. В списке миссий пользователь должен видеть, какие ветви пути у него есть, например:

#### **Ветка 1 — Блогерская:**

- Миссия 1 — пост с фото
- Миссия 2 — сторис с хэштегом
- Миссия 3 — съёмка видеоблога про компанию

#### **Миссии могут делиться по категориям:**

- Квесты — базовые онлайн и офлайн задачи
- Рекрутинг — задания, направленные на привлечение новых кандидатов
- Лекторий — задания, направленные на обучение коллег и кандидатов
- Симулятор — задания, направленные на проверку знаний, например тесты, соревнования

При этом нельзя делать список миссий статичным, так как время, люди, задачи, цели — всё меняется. Соответственно, со стороны HR мы должны иметь возможность создания и редактирования миссий, чтобы поддерживать интересные и актуальные задачи в списке миссий у пользователей.

### 4. Бортовой журнал

История действий, прогресса пользователя и рейтинга. Пользователь может видеть свой прогресс, сколько он выполнил и к чему это привело. Также доступен просмотр ТОПов за месяц, неделю или год.



## 5. Навыки

Список всех имеющихся компетенций с текущим уровнем прокачки:

- Вера в дело
- Стремление к большему
- Общение
- Аналитика
- Командование
- Юриспруденция
- Трёхмерное мышление
- Базовая экономика
- Основы аэронавигации

*Прокачивать компетенции можно, выполняя миссии.*

## 6. Хранилище

Магазин, в котором можно приобрести за ману разнообразный мерч, товары, билеты и прочие бонусы.

## 7. Онбординг

Для большего погружения в тематику необходимо не просто выдавать задачи и поощрять баллами, а периодически предоставлять интересные отрывки лора. Онбординг должен рассказывать о работе отдельных блоков на платформе, подкрепляя это интересными научными и историческими фактами о космосе.

## 8. Статистика для HR-специалистов

HR-специалистам важно иметь доступ к информации для анализа конверсии выполнения миссий, веток и прогресса пользователей. Если результат миссии можно увидеть удалённо, будет здорово, чтобы пользователи прикрепляли его при закрытии миссии. В таком случае также необходим функционал модерации выполнения заданий.

## 9. Артефакты

Артефакты — уникальные награды, которые можно получить за прохождение миссий. Необходим функционал создания артефактов со стороны HR.

## У артефакта есть атрибуты:

- Изображение
- Название
- Краткое описание
- Дополнительно: редкость артефакта

## 2. Наши ресурсы

ОЭЗ «Алабуга» [[alabuga.ru](http://alabuga.ru)]

— основной сайт компании

HR-платформа [[hr.alabuga.ru](http://hr.alabuga.ru)]

— основная платформа для авторизации в экосистеме «Алабуги». На этой платформе расположены бизнес-симуляции, в которые играют кандидаты и сотрудники

Карьера.100 лидеров [[career.alabuga.space](http://career.alabuga.space)]

Карьера.Политех — [в разработке]

Карьера.Старт — [в разработке]

— платформы для трудоустройства кандидатов. В этих сервисах кандидаты заполняют резюме, документы, проходят симуляции, записываются на очные этапы и проходят собеседования

Алга.Алабуга [[alga.alabuga.ru](http://alga.alabuga.ru)]

— профориентационные экскурсии, которые запомнятся каждому участнику!

## 3. Программно-аппаратные требования

### 3.1. Аппаратные требования и подход к разработке

*Mobile First* (для пользователей-сотрудников/кандидатов): интерфейс должен быть адаптирован под мобильные устройства (ширина viewport от 320px). Предполагается, что большинство задач пользователи будут выполнять на ходу: проверять задания, загружать фотоотчёты, тратить ману в магазине.

*Desktop/Tablet* (для HR-специалистов): административный интерфейс для создания миссий, отслеживания статистики требует большого экрана. Минимальная ширина — 1024px.

### **3.2. Программные требования (стек технологий)**

Данный стек рекомендованный, но не обязательный.

#### **Frontend:**

Фреймворк: React (предпочтительно с использованием функциональных компонентов и хуков)

Язык: TypeScript (строгая типизация критически важна для избежания ошибок в рангах, миссиях и наградах)

Стили: CSS-in-JS (Styled-components, Emotion) или modern CSS с модулями. Важно обеспечить тему, легко меняемую под космический стиль

Состояние: Redux Toolkit / MobX / Zustand для управления сложным состоянием приложения (ранги, миссии, инвентарь)

Роутинг: React Router

#### **Backend:**

Фреймворк: Python + FastAPI (современный, высокопроизводительный) или Django (более богатый из коробки, но тяжелее)

База данных: SQLite на время хакатона — для простоты разработки и демонстрации. В продакшене — PostgreSQL

Аутентификация: JWT-токены. Необходима интеграция с «hr.alabuga.ru» (на хакатоне можно замочать или использовать простой вход по логину/паролю)

#### **Прочее:**

Контроль версий: Git

## **4. Требования к презентации/демонстрации**

Презентация должна быть в формате последовательного пользовательского сценария (User Flow).

Цель: показать не набор разрозненных кнопок, а историю одного пользователя.

### **Пример сценария для демонстрации:**

Кандидат (ранг: «Искатель») заходит на платформу, проходит онбординг.

Видит свою цель — «Получить оффер» (требуется ранг «Пилот-кандидат»).

Переходит в раздел «Миссии», видит доступные ветки: «Рекрутинг» (загрузка резюме, документов) и «Лекторий» (просмотр видео о компании).

Выбирает миссию «Загрузить резюме» → загружает файл → получает награду (опыт, мана, прокачка компетенции «Аналитика»).

В «Бортовом журнале» видит запись о выполненной миссии и рост progress bar до следующего ранга.

После выполнения всех ключевых миссий система автоматически повышает его ранг до «Пилот-кандидат» и выдаёт уведомление об успехе.

Пользователь заходит в «Хранилище» и тратит заработанную ману на мерч (например, «Футболка Алабуга» за 100 маны).

Для HR: показать один экран создания/редактирования миссии с полями (название, описание, опыт, мана, ранг, компетенции).

## **5. Требования к сопроводительной документации**

Краткий документ (README.md в репозитории), описывающий:

Команда и роли. Кто за что отвечал.

Архитектура. Краткое описание структуры фронтенда и бэкенда (какие основные модули, как взаимодействуют).

Реализованные механики. Список того, что получилось сделать (например: «Реализована система рангов с проверкой 3 условий», «Реализован CRUD для миссий со стороны HR»).

Что не реализовано и почему. Честность приветствуется.

Инструкция по запуску. Как установить зависимости и запустить приложение локально.

Ссылки. Ссылка на рабочий демо-сайт, ссылка на репозиторий с кодом.

## 6. Ресурсы

Основные референсы: Предоставлены ниже.

В оформлении использовать логотип (из приложенного файла) и брендовые цвета:



CMYK → 100/94/0/0  
RGB → 40/58/151  
Pantone 99-8U



CMYK → 100/68/0/0  
RGB → 0/93/172  
Pantone 104-8U



CMYK → 100/0/0/0  
RGB → 0/174/239  
Pantone 115-8U



CMYK → 51/0/0/0  
RGB → 106/207/246  
Pantone 115-5U

### Ключевые интеграции:

“hr.alabuga.ru” -> Аутентификация.

“career.alabuga.space” -> Источник задач для кандидатов.

Бизнес-симуляции с “hr.alabuga.ru” -> Задачи типа "Симуляция".

На хакатоне: Достаточно замочать данные интеграции (например, сделать несколько пользователей в БД и эмулировать успешный возврат с бизнес-симуляции).

## 7. Требования к сдаче решений

### 7.1. Промежуточная сдача (Концепт)

*Визуальный концепт:*

- Макеты ключевых экранов (как минимум: ЛК пользователя с прогрессом, список миссий, карточка миссии, ЛК HR для создания миссии) в Figma/Adobe XD.
- Проработанный космический UI-kit: цветовая палитра, кнопки, типографика, иконки.

*Предпочтительная реализация ключевых механик:*

- Работающий фронтенд на React с роутингом между пустыми страницами.
- Работающий бэкенд на Python с 2-3 API-эндпоинтами (например, "GET /api/missions", "POST /api/missions").
- Реализована хотя бы одна сложная механика на выбор: система проверки условий для повышения ранга ИЛИ система ветвления миссий.

*User Stories в формате:*

- «Как Кандидат, я хочу видеть свой прогресс в виде progress bar, чтобы понимать, сколько еще нужно сделать для оффера».
- «Как HR, я хочу иметь возможность указать награду в мане за миссию, чтобы мотивировать пользователей выполнять ее».

### 7.2. Финальная сдача

Полная реализация одного end-to-end процесса. Например, процесса «Кандидат выполняет миссии для получения оффера»:

- Пользователь регистрируется/логинится (мок).
- Видит свой текущий ранг и цели.
- Выполняет 2-3 связанные миссии из одной ветки (квест + симулятор).
- Система начисляет опыт, ману, прокачивает компетенции.
- При выполнении всех условий система автоматически повышает ранг пользователя.

- Пользователь видит это изменение в UI (уведомление, изменение в бортовом журнале).
- Демонстрация – это сквозной сценарий, а не показ отдельно взятого экрана миссий и отдельно взятого экрана ранга.

## 8. Критерии оценки (на что делать акцент)

### 1. Подход коллектива к решению задачи (20%):

- Понимание бизнес-цели: Не просто «сделать игру», а «повысить вовлеченность и связанность этапов».
- Работа в команде: Использование Git (много мелких коммитов, а не один огромный), распределение задач (Trello/Notion/Kanban-доска). \*Желательно
- Качество визуального концепта. Умение перенести космическую тему в интерфейс.

### 2. Техническая проработка решения (30%):

- Чистота и структура кода (отступы, именование переменных, компонентный подход).
- Архитектурные решения: Как спроектированы модели данных (ранг, миссия, пользователь), API.
- Сложность реализованной механики: Реализация системы рангов с тремя условиями будет оценена выше, чем простой линейный прогресс.

### 3. Соответствие решения поставленной задаче (25%):

- Решение решает проблему «разрозненных задач» (ветвление миссий, общая цель).
- Учтена тематика космоса (лоре, нейминг, визуал).
- Есть разделение на интерфейсы User и HR.

### 4. Эффективность решения (15%):

- Работоспособность: Приложение не падает при базовых сценариях.
- Удобство использования (Usability): Понятно ли пользователю, что делать? Ведет ли интерфейс его к цели?

### 5. Выступление на питч-сессии (10%):

- Ясность: Четкое объяснение проблемы и решения.
- Убедительность: Акцент на то, как продукт решает боль заказчика.



- Демонстрация: Упор на живой сквозной сценарий, а не на слайды.
- Тайминг: Уложиться в отведенное время.

## Референсы:





