**Тестовое задание**

**Часть 1.** Предложить архитектуру для целевой системы и процессы взаимодействия

**Часть 2.** Реализовать демонстрационный микросервис и UI к нему

**Краткое описание целевой системы (для разработки которой выбирается исполнитель)**

Цель разработки системы: информатизация и автоматизация широкого круга процессов ГК ГОРКА, основным направлением работы которой является инфраструктурное проектирование.

Разрабатываемая система включает в себя такие модули как управления задачами, проектами (участики, роли, связь с задачами), спринтами, совещаниями (повестки, протоколы, связь с задачами и др.), управление кадрами, электронный документооборот и ряд других модулей.

**Предполагаемый технологический стек**

* Kubernetes, Helm 3, Terraform
* Java, Spring Boot
* PostgreSQL
* React.js
* Kafka
* Gitlab CI
* SonarQube

**Принципы и подходы к реализации**

* Микросервисная архитектура
* API first, подробное описание в формате Open API Specification 3.0 (Swagger)
* Передача компетенций команде (владелец продукта, аналитики, небольшая команда разработки, тестировщики) заказчика (качественная документация, тесты, взаимодействие)
* Аутентификация OpenID Connect (OIDC)
* Scram

**Часть 1**

Предложить архитектуру для целевой системы и процессы взаимодействия с заказчиком

1. Предложить общую архитектуру решения (опираясь на описание выше)
2. Описать организацию команды (роли, взаимодействие, процессы)
3. Предложить схему взаимодействия с командой заказчика с учетом вводных

**Часть 2**

Реализовать демонстрационный микросервис и UI к нему

**Стек технологий**

Backend: Java, Spring Boot

База данных: H2 (in-memory)

Frontend: React.js

Docker (docker-compose для локального запуска)

**Описание реализуемой системы**

Система привелегий

Логика

| **Уровень** | **Сумма покупок, руб** | **Скидка, %** | **Дополнительные условия** |
| --- | --- | --- | --- |
| Бронзовый | <= 1000 | 3% | При покупке до 100 рублей скидка не предоставляется.  В будни скидка +1% |
| Серебряный | <= 2000 | 5% | При покупке до 100 рублей скидка не предоставляется.  В будни скидка +1% |
| Золотой | <= 5000 | 7% | При покупке до 100 рублей скидка не предоставляется.  В будни скидка +1% |
| Платиновый | > 5000 | 10% | В будни скидка +2% |

**Backend**

Методы API / параметры:

Покупка / # карты лояльности, сумма покупки

Возврат / # карты лояльности, сумма возврата

Уровень / # карты лояльности (возвращает текущий уровень)

Скидка / # карты лояльности, дата (возвращает текущий размер скидки, с учетом даты)

Требования:

* Проект должен быть реализован с использованием вышеуказанного стека
* Сервис должен реализовывать REST API
* API должен быть задокументирован с помощью Open API Specification 3.0 (Swagger)
* Предложить набор тест кейсов и соответствующие запросы для функционального тестирования
* Проект должен быть размещен в Git репозитории, инструкция по запуску должна быть описана в README.md

Основные критерии оценки:

* Соответствие требованиям
* Структура и формат кода
* Документация API
* Использование REST API
* Тесты
* Инструкция по запуску
* Контейнеризация

**Frontend**

Необходимо реализовать одностраничное web приложение, взаимодействующее с разработанным backend сервисом.

Описание экранной формы:

Поле ввода “Номер карты лояльности”

Кнопка “Текущий уровень и размер скидки” (дата в запрос автоматически проставляется текущая, в интерфейсе явно не задается)

Поле ввода “Сумма покупки”

Кнопка “Отправить”

Поле ввода “Сумма возврата”

Кнопка “Отправить”

Требования:

* Проект должен быть реализован с использованием вышеуказанного стека
* Проект должен быть размещен в Git репозитории, инструкция по запуску должна быть описана в README.md
* Допускается размещение Backend и UI проектов в одном репозитории

Основные критерии оценки:

* Соответствие требованиям
* Структура и формат кода
* Удобность, аккуратность и логичность интерфейса (красивый дизайн не требуется). От указанных в задании полей и кнопок можно отойти, при условии сохранения обозначенной функциональности
* Инструкция по запуску
* Тесты, если применимо
* Адаптивность верстки, если применимо
* Контейнеризация