**Backend-разработчик C#**

1. Техническое задание на реализацию микросервиса получения и отображения курсов валют

**Задание:** В рамках тестового задания предлагается реализовать микросервис с использованием .Net Core.

Ожидаемые технологии: Entity Framework, EF Migrations, XUnit, Swagger, Serilog, PostrgeSQL, Docker/Docker Compose

Основной задачей данного микросервиса будет получение и отображение курсов валют с сайта НБ РБ (API национального банка: <https://www.nbrb.by/apihelp/exrates>).

Требуется создать 2 точки входа (endpoint) - формат данных, метод запросов, формат ответа и набор дополнительных данных запроса/ответа определяется разработчиком:

1. Endpoint с входящим параметром - дата.

Результат запроса - ответ, указывающий на корректность выполнения загрузки данных в разработанную систему о курсах за выбранную дату.

2. Endpoint с входящими параметрами - дата и код валюты. Результат запроса - информация о курсе валюты за указанный день.

Информация должна храниться во встроенной базе данных (далее БД). Создание БД и её сущностей должна происходить при первом старте сервиса.

**Что будет оцениваться:**

- Выполнение основной задачи;

- Качество кода (описание методов, прохождение тестов, использование ООП);

- Разработанный протокол взаимодействия (необходимо описание разработанного API);

- История взаимодействия git.

2. Техническое задание на проектирование системы учёта библиотечной книговыдачи\*

**Роли системных пользователей:**

Библиотекарь – выдаёт имеющиеся в наличии не повреждённые экземпляры зарегистрированным в системе посетителям, которые не имеют задолженности перед библиотекой (просроченные книги на руках), а также оформляет возврат книг, помечая повреждённые экземпляры.

Завхоз – вводит в систему новые книги и убирает из системы повреждённые экземпляры, которые не подлежат восстановлению.

Подсобный рабочий – просматривает имеющиеся в наличии экземпляры на предмет повреждений, пытается восстановить повреждённые и помечает те, которые не подлежат восстановлению.

**Решение должно содержать:**

- диаграмму классов и интерфейсов;

- пояснение к диаграмме в терминах паттернов проектирования.

**Дополнительные условия:**

- зарегистрированные в системе пользователи могут обладать сразу несколькими ролями;

- необходимо предусмотреть возможность добавления в систему новых пользовательских ролей и операций, а также редактирование списка операций, свойственных каждой роли, без необходимости перекомпиляции основного модуля;

- решение должно быть абстрагировано от реализации пользовательского интерфейса и репозитория.