### Задание

1. *Написать краткое техническое задание к проекту. Описать только требования к основной функциональности системы. По этим требованиям вы далее будете строить систему.*

Телеграм бот, который предназначен для сортировки работ, которые присылают студенты по категориям: домашняя работа, контрольная работа, мат диктант, коллоквиум, тест, с возможностью сортировки по признакам: дата, студент, номер группы, преподаватель.

Интерфейс со стороны студент -> сервер должен включать в себя возможность

регистрации студента в системе

* /set фамилия имя курс группа

функциональные и нефункциональные требования

регистрация работы студента

* сообщение содержащее id работы, номер задания и фотографию (тригер - наличие фотографии)
* или сообщение содержащее только фотографии с решениями работы и id работы (тригер - наличие фотографий)

получение статистики по выбранной работе

* сообщение вида /stat work\_id

получение статистики по всему курсу, по всем работам

* сообщение вида /stat all

информацию о дедлайнах на определенные виды работы

* /deadlines

возможность получения условий задания для работы

* /task work\_id

Отношение сервер -> студент включают в себя

регистрация студента (внесение tg id, имени пользователя в tg, фамилии, имени, номера группы и курса в базу данных)

* сообщение об успешной или неуспешной регистрации

регистрация работы студента (первичная проверка на совпадение файлов фотографий или pdf-ок, проверка на плагиат файлов)

* сообщение об успешной/неуспешной регистрации

сообщение об ошибке при принятии работы

сообщение о состоянии сервера

сообщение со статистикой студента

Интерфейс преподаватель -> сервер

регистрация преподавателя (аналог студ)

возможность отвечать на вопросы студента

посмотреть информацию, статистику о студенте

посмотреть информацию по успеваемости группы

создание работы для студентов

* преподаватель присылает сообщение с фотографиями номером курса, группы, описанием работы, сроки сдачи
* преподаватель присылает сообщение с номером курса, группы, описанием работы, сроки сдачи

Основные требования к серверу и процессы, которые будут на нём происходить

регистрация работ:

уведомление студента -> сохранение в базе

конвертация работ в архив:

тригер преподавателя -> поиск в базе данных фотографий -> создание архива или pdf -> отправка преподавателю

тригер преподавателя об автопроверке -> загрузка работы студента из базы данных в сеть по распознаванию ответов, загрузка ответов на данную работу -> работа сети -> прислать сообщение о результате преподавателю или студенту

создание работы для студентов

тригер преподавателя о создании работы -> присвоение id\_work -> загрузка фотографий в базу данных о работах или установить флаг о том, что фотографий для данной работы нет -> прислать работу студенту, сообщить преподавателю о том, что работа зарегистрирована и доставлена студентам или вывести преподавателю отчет с ошибкой

Вывод статистики по группе

вывод сводной таблицы проверенных работ.

1. Указать требования к нагрузке:
   * 1. ограничения на количество запросов (чтение/сохранение) в единицу времени, средняя нагрузка, пиковая нагруза
     2. допустимые задержки при выполнении запросов, допустимое время недоступности системы
     3. объем, формат и специфика данных
   1. Средняя нагрузка ~100 запросов с клиентской части в час. Один запрос с клиента соответствует одному запросу на сохранение данных в БД и одной операции помещения значения в in-memory кэш.
   2. Пиковая нагруза ~100 запросов с клиента в минуту. Длительность нагрузки около 120 минут. А также, промежуток в ~150 запросов в минуту, длительностью в 10 минут.
   3. Latency при запросах с клиента не более 3 сек. Запись в БД и другие серверные процессы, иницированные клиентским запросом, происходят асинхронно и не заставляют клиент ждать.
   4. Downtime системы не более часа в день.
   5. Запрос с клиента сохраняет и обрабатывает несколько фотографий. Объем не более 1 мб. Формат JPEG, PNG.
2. *Построить архитектуру проекта. В качестве результата подготовить архитектурную диаграмму. В Вашем git репозитории нужно завести папку docs, где будут храниться материалы по проекту.*
3. *Выбрать технологии для проекта. Кратко обосновать свой выбор.*

Мы выбираем технологии .NET и Azure. Платформа .NET позволяет быстро разрабатывать нагруженные системы, а Azure предоставляет легкодоступные необходимые сервисы (файловое хранилище azure storage, service bus, functions, позволяет хостить docker контейнеры)

1. *Сделать оценку трудоемкости проекта. Сформулировать ограничения на выполняемую оценку.*

Image processor - эта часть требует большой работы со стороны дисциплины, не относящейся никак к данному предмету. Поэтому в данном проекте можно сделать заглушку, для симуляции процесса обработки

Развертывание бота, написание функциональности бота.

Развертывание баз данных, зависит от структуры данных, которая будет окончательно утверждена, возможно в процессе разработки структура будет меняться на более удобную

Подготовка окружения Azure

Развертывание и оркестрация в докере. Большое количество сервисов необходимо интегрировать друг с другом.

Разработка метрик и проведение нагрузочных тестов. Анализ полученных данных, поиск узких мест.