Домашнее предприятие «Музыкальная школа»

Направления: Вокал, Саксофон, Фортепиано

Микросервисная архитектура

Укрупненные контуры:

1. Контур Управление: получает сообщения-отчеты от сервисов Распсиания, Продвижения и Бухгалтерии,   
   в ответ шлет каждому сервису сообщения-руководства (набрать преподавателя, опубликовать акцию на обучение по курсу, выдать премию….)
2. Контур Расписание: управляет информацией о расписании по курсам  
   Информирует контур управления о нехватке педагогов/помещений,   
   Информирует контур Учеников и Преподавателей о свободных дата/время по курсам

Получает инфо об учениках и педагогах

1. Контур Учеников: личные данные учеников, курс, педагог, оплата.   
   Получает инфо от контура Продвижения о новых учениках

Отправляет инфо по ученикам и выбранным дата/время занятий на контур Расписание

В ответ получает подтверждение времени

1. Контур Преподавателей: личные данные педагогов, курс, ставка.   
   Получает инфо от контура Продвижения о новых преподавателях

Отправляет инфо по свободным дата/ на контур Расписание

В ответ получает подтверждение времени

1. Контур Продвижения:   
   Занимается рекламой курсов и поиском новых клиентов,   
   Передает информацию о новом ученике/преподавателе на соответствующие контуры
2. Контур Бухгалтерии:   
   Получает инфо о количестве часов преподавателя из контура Расписание  
   Получает инфо о ставке преподавателя из контура Преподавателей  
   Получает инфо об оплате из контура Учеников  
   Получает инфо с контура Управление (изменение ставки преподавателя, премировании…)  
   Отправляет фин отчет на контур Управления

Обоснование выбора архитектуры:

1. Гибкость приложения (предполагаем возможные изменения внешних условий и необходимость быстрой модификации приложения)
2. Масштабируемость (планируем добавлять направления, развернуть онлайн обучение в будущем)
3. Устойчивость к сбоям (приложение клиентоориентированное и каждый сбой может привести к потере клиента)
4. Обновление приложения по частям (обновления и доработки планируется, но они не должны привести к временной потере работоспособности приложения)
5. Возможность использовать разные технологии и языки на каждом контуре (предполагается и десктопная и веб части)
6. Логическое разделение приложения на модули (составляющие приложения логически и функционально отделимы, поэтому есть необходимость их разделить, чтобы они стали проще в реализации, поддержке и модификации)
7. Быстрее внести изменения (каждый модуль включает только свою логику)