1. Гоша  
   Как мы знаем и каждый день видим, расписание в нашем колледже представляет из себя PDF документ с огромной таблицей который помещает в себя огромное количество групп и из-за этого можно легко запутаться, кроме этого на телефоне этот самый документ еще нужно скачивать, а это плюс к неудобности.  
     
   Хотелось бы спросить у студентов, как бы хотели видеть расписание нашего колледжа?
2. Алмаз  
   Примерно такими же мыслями мы думали когда начали разрабатывать это приложения, удобное расписание которое приходит в общую беседу группы, при этом прописывая все, от названия предмета до самых мельчайших деталей которые могут там быть(время, кабинет, домашнее задание). Всем нужна удобность, как говориться «Лень двигатель прогресса»
3. Гоша  
   Вот такими образом мы хотим начать рассказать вам о том как разрабатывается и продолжает разрабатываться прототип нашей информационной системы.
4. Алмаз  
   В качестве примера данного расписания было взято расписание УКРТБ в котором онлайн расписание уже давно перенесено в учебный портал
5. Алмаз  
   Разработка началась с базы данных, она составляет основу всей информационной системы и отталкиваясь от нее продумывается архитектура и вся логика веб-приложения. На слайде мы видим графическую схему базы данных. В ней огромное количество таблиц и связей, но за счет этой сложности, в этой базе данных все продумано.
6. Алмаз  
   После того как база данных продумана можно начинать строить архитектуру и логику приложения. Расскажу в общих чертах о структуре и архитектуре кода максимально просто:  
   1 изначально приложение задумано как одностраничное, это значит то что страница в браузере при каких либо взаимодействиях не будет перезагружаться полностью, она будет обновлять только те данные которые были вызваны при взаимодействии  
   2 На серверной части используется методология разделения кода на 3 основных сущности это – модели, представления, контроллеры  
   Модели отвечают за взаимодействия с базой данных и выбором оттуда нужной информации  
   Представления обрабатывают полученную информацию и приводят ее в готовые Html страницы, которые будут выводиться для пользователя   
   Контролеры же являются связным звеном между представлениями и моделями, через них проходят все запросы к серверу проводятся проверки данных и происходит взаимодействие с моделями, затем сырые данные из моделей передаются в представления,
7. Алмаз  
   Так же хотелось бы упомянуть про архитектурный стиль REST который используется в данной ИС.  
   Этот стиль представляет из себя полноценное использование http протокола и методику взаимодействия между клиентом и сервером:  
   - Клиент обращается к серверу с помощью HTTP запросов  
   - Затем сервер отправляет клиенту только данные в виде json строк  
   Данный стиль позволяет построить наглядный программный интерфейс приложения и дает возможность использовать данный интерфейс для различных технологий (front-end часть проекта, вк бот, мобильное приложение )  
   По причине того что используется данный стиль (REST) «представления» в серверной части не были использованы
8. Гоша  
   Давайте отвлечемся от серверной части проекта и перейдем к клиентской,   
   Я разрабатывал клиентскую часть веб-приложения   
   Мне хотелось сделать легкий стиль в котором все понятно и не слепит глаз, поэтому я взял темные тона для дизайна данного проекта.
9. Гоша  
   Мной были использованы технологии Vue, bootstrap, Vuex, они в свою очередь представляют из себя фреймворки для более удобной разработки клиентской части
10. Vue и Vuex это очень мощные инструменты для создания реактивных веб приложений, они постоянно следят за изменениями данных, что позволяет незамедлительно изменять контент на html документе  
    Vue диктует компонентный подход, это значит он рекомендует разделять верстку и логику всей клиентской части на мелкие переиспользуемые компоненты, а также занимается работой с дом деревом(HTML документ) страницы   
    Vuex представляет из себя централизованное хранилище всех данных приложения, и является единственным истинным источником информации для всей клиентской части, он отправляет запросы на сервер, получает ответ, и заключает данные из ответа сервера внутри себя.
11. Гоша  
    Bootstrap в свою очередь из себя набор готовых и сбалансированных стилей для более быстрой верстки  
    Главным плюсом этого инструмента является то что в нем использются уже готовые классы для стилей.
12. Гоша   
    Вернемся к теме REST API   
    Клиентская и серверная часть нашего приложения взаимодействует между собой , отправляя друг другу http запросы с json строками тоесть, например клиентская часть отправляет запрос на сервер о создании расписания и получает в зависимости от успешности различные ответы, например Успешно создано расписание, что то в данных не так не успешно
13. ГОША и АЛМАЗ  
    Мы рассказали самые важные и самые нужные вещи в простом формате о нашей разработанной информационной системе