

# **Сервис контроля посещаемости**

**Саматов Дамир Ришатович**

**Научный руководитель: Бандеров В.В.**

## **Задачи**

- 1. Разработать автоматизированную информационную систему для контроля посещаемости студентами занятий с целью оптимизации процесса ведения и просмотра статистики**
- 2. Изучить и подобрать необходимые инструменты для разработки**
- 3. Смоделировать структуру системы и ее функционал**
- 4. Спроектировать и создать базу данных**
- 5. Внедрить технологию NgRx для контроля состояния приложения**

## Использование стека технологий



# Структура базы данных

university classroom
id : int(11)
name : varchar(255)
photo : varchar(255)
# id_building : int(11)
# seats : int(11)
# floor : int(11)

university building
id : int(11)
name : varchar(255)
photo : varchar(255)

university department
id : int(11)
# id_institute : int(11)
name : varchar(255)

university program
id : int(11)
# id_institute : int(11)
name : varchar(255)

university institute
id : int(11)
name : varchar(255)

university groups
id : int(11)
# id_program : int(11)
name : varchar(255)

university schedule_group
# id_schedule : int(11)
# id_group : int(11)

university schedule
id : int(11)
# id_lesson : int(11)
# id_teacher : int(11)
# id_classroom : int(11)
start : time
end : time
day : varchar(255)

university current_schedule
id : int(11)
# id_schedule : int(11)
date : date
photo : varchar(255)
# stats : int(11)

university student
id : int(11)
name : varchar(255)
surname : varchar(255)
last_name : varchar(255)
login : varchar(255)
password : varchar(255)
# group_id : int(11)
photo : varchar(255)
type : varchar(255)

university lesson
id : int(11)
name : varchar(255)

university teacher
id : int(11)
name : varchar(255)
surname : varchar(255)
last_name : varchar(255)
login : varchar(255)
password : varchar(255)
# id_department : int(11)
photo : varchar(255)
type : varchar(255)

university statistic
# id_student : int(11)
# id_current_schedule : int(11)
# here : tinyint(1)



# Модель спроектированной системы

## Компоненты

1. app
2. Main
3. app-signup
4. Workspace
5. Results
6. search
7. userinfo
8. News
9. Calendar
10. Pie-chart-component
11. Lesson-list.
12. Lesson-info
13. Searchbygroupresult
14. Statforgroup
15. Lessonliststudent
16. Statforstudent
17. Totalstat

## NgRx

1. userActions
2. scheduleActions
3. statActions
4. getUserReducer
5. getSchedule
6. statReducer
7. effects

## Сервисы

1. Auth
2. Request
3. authguard
4. studentguard
5. adminguard
6. teacherguard
7. userinfo

# Проблемы управления состоянием приложения

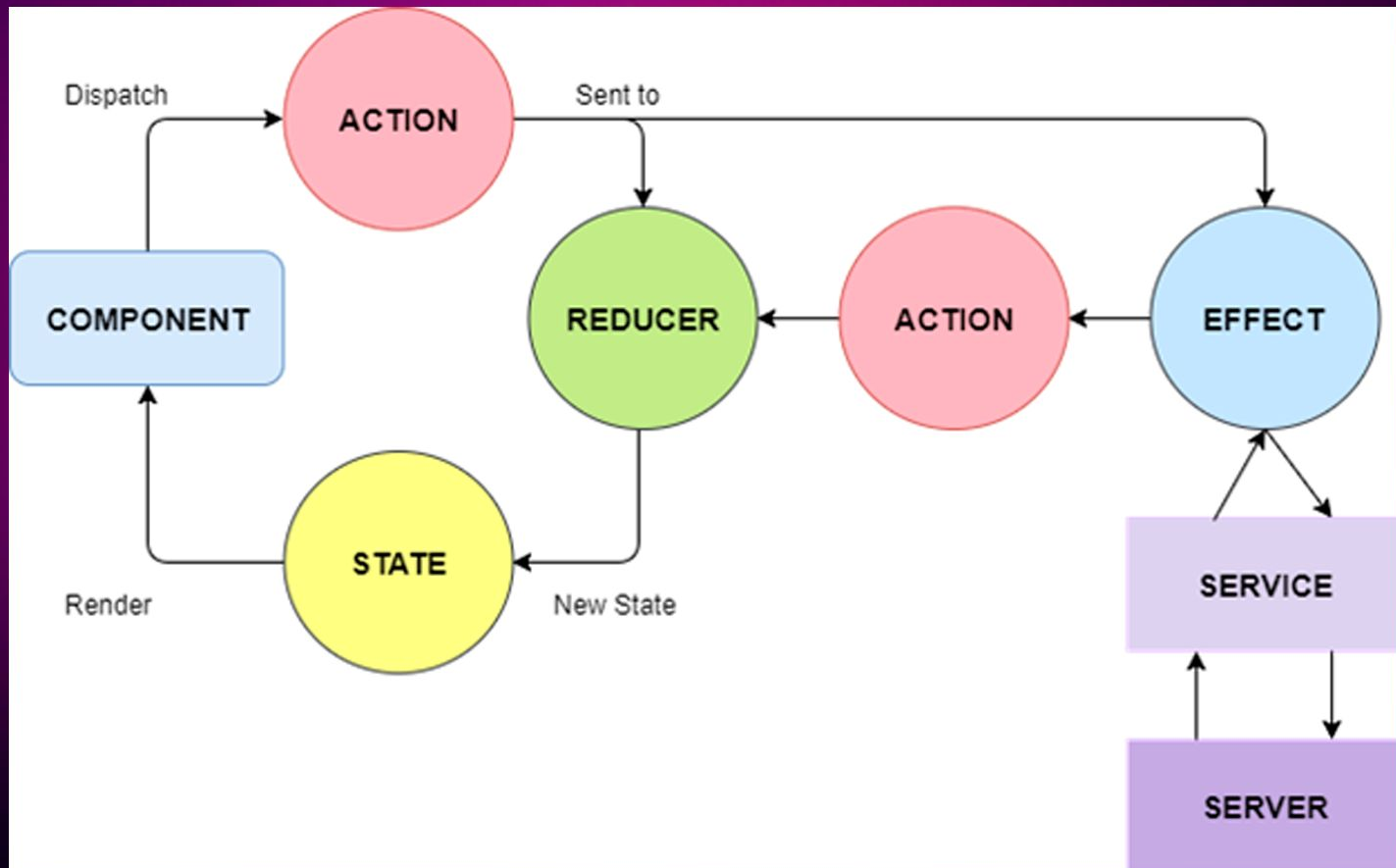
1. Состояние наших компонентов не должно быть хаотичным, мы должны видеть зависимости явно.
2. Источник изменения состояния должен быть известен. Если состояние было изменено, нам следует знать, кто являлся источником.
3. Изменения состояния не должны происходить во всех местах нашего приложения (компонентах, *service-ax*). Желательно отделить эту часть приложения для быстрого понимания архитектуры приложения.

# NGRX

**NGRX - пакет модулей, который имплементирует redux-паттерн и дает возможность воспользоваться некоторыми приемами, помогающими сделать наше приложение более гибким и масштабируемым.**

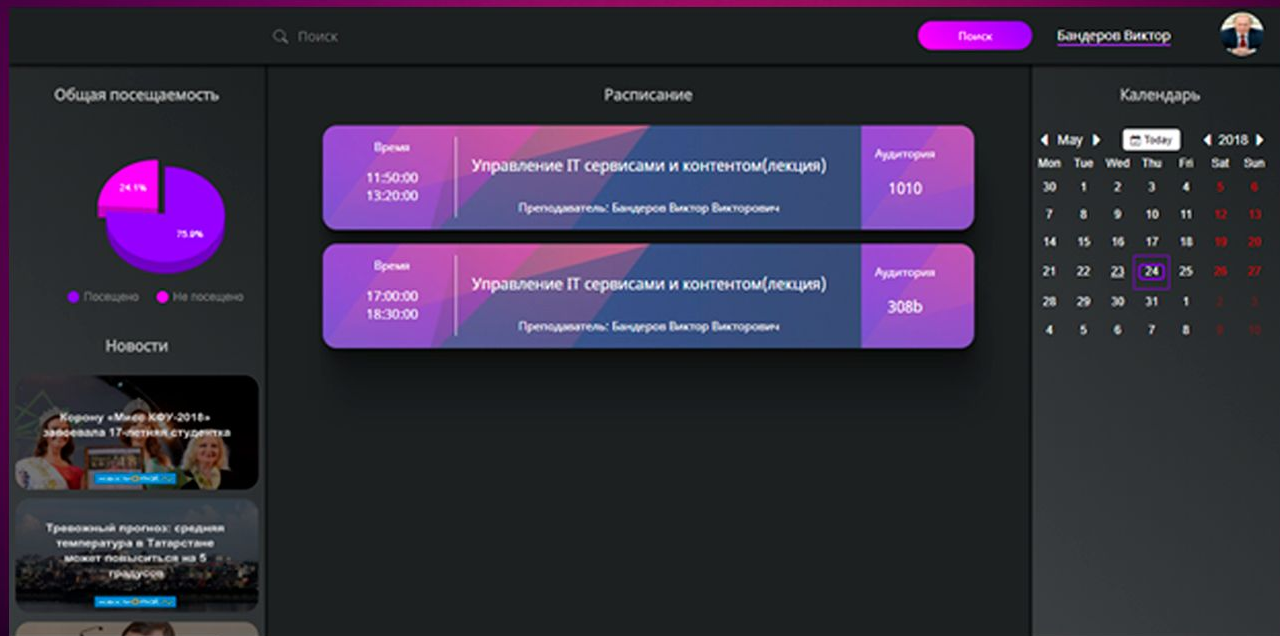
- 1. Единственно точный источник данных и его удобное использование в компонентах**
- 2. Изменение состояния только посредством чистых функций- reducers**
- 3. Инициализация изменений может осуществляться посредством отправки действий (actions)—описывающих тип изменений и содержащих полезную нагрузку (payload).**
- 4. 2 типа компонентов (контейнер\представление)**
- 5. Изоляцию side-effect-ов**

# NGRX

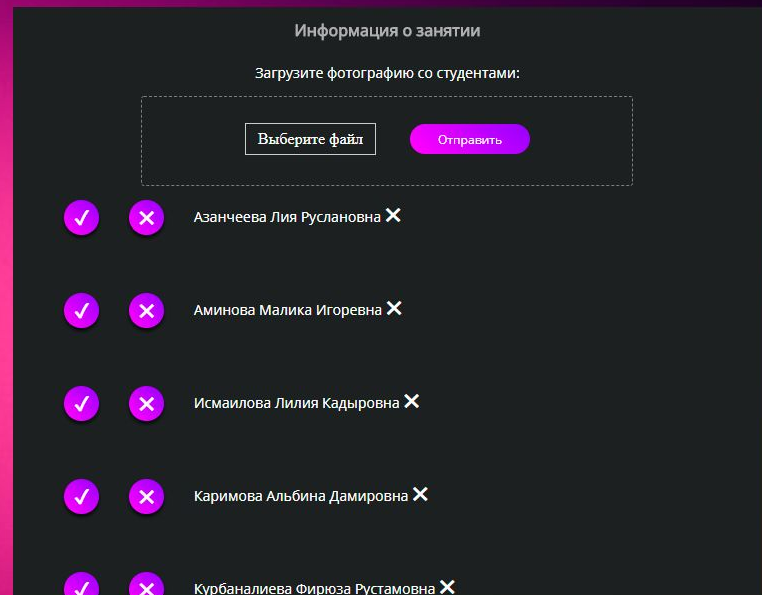
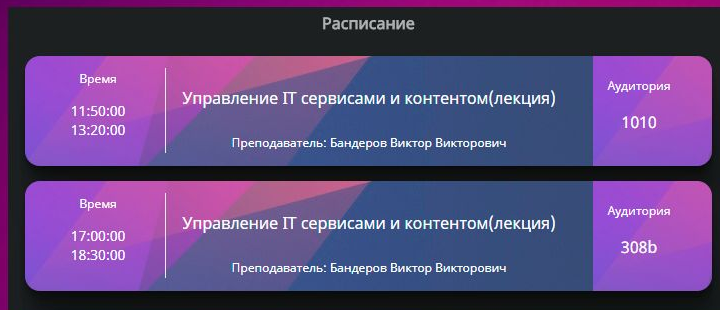
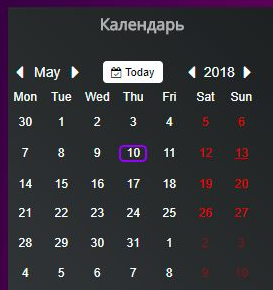




# Интерфейс личного кабинета преподавателя



# Принцип работы Web-интерфейса преподавателя

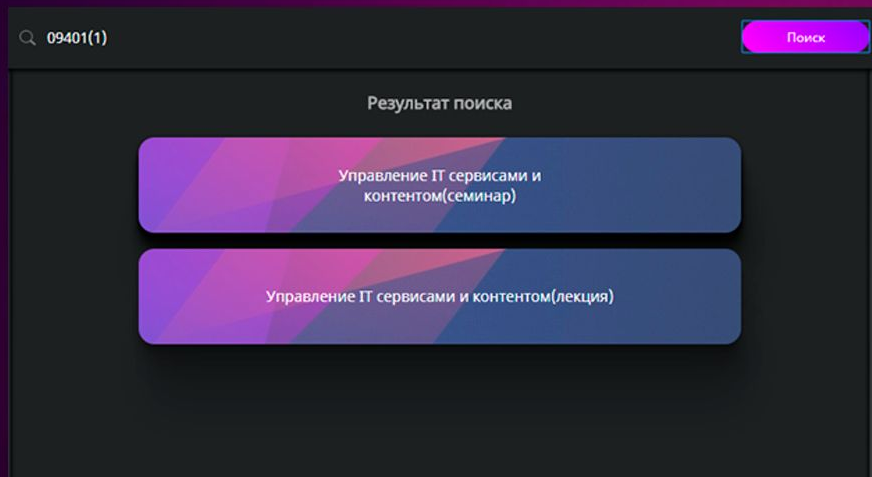


**1. Возможность выбора даты из календаря для просмотра расписания**

**2. Выбор текущего занятия из списка**

**3. Форма загрузки фотографии и автоматически обновляющийся список студентов, с возможностью корректировки вручную**

# Интерфейс личного кабинета преподавателя

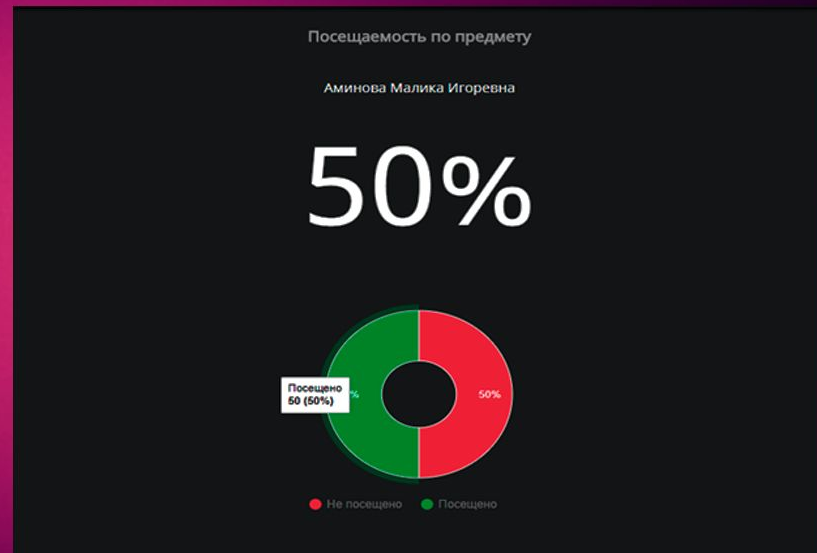
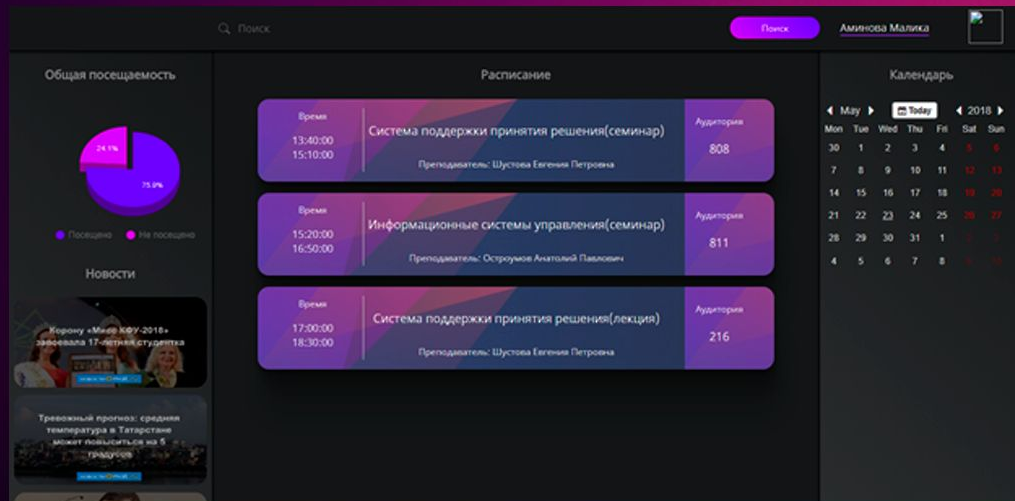


## 1. Результат поиска

## 2. Статистика

Статистика группы	
Азанчеева Лия Руслановна	- 100%
Аминова Малика Игоревна	- 50%
Исмаилова Лилия Кадыровна	- 0%
Каримова Альбина Дамировна	- 100%
Курбаналиева Фирюза Рустамовна	- 100%
Саматов Дамир Ришатович	- 100%

# Интерфейс личного кабинета студента





# Интерфейс личного кабинета администратора

Загрузите расписание

Количество учащихся в аудиториях

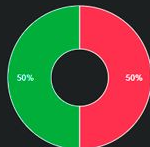
Время	Учеников	Аудитория
11:50:00 13:20:00	10	1010

1. Форма для загрузки расписания

2. Мониторинг количества учащихся в аудиториях

# Визуализация статистики средствами GoogleCharts

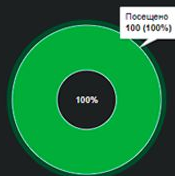
50%



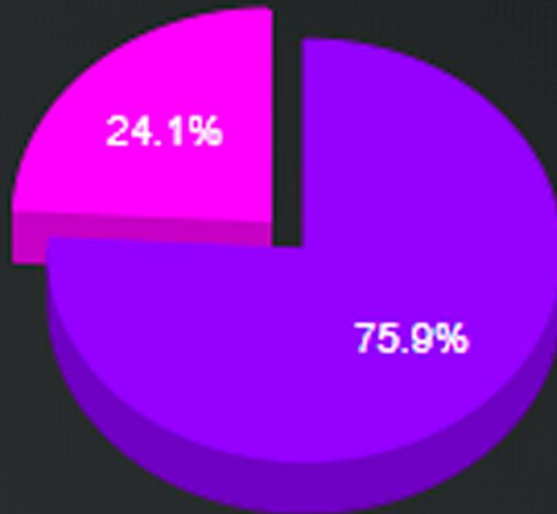
● Не посещено ● Посещено

Саматов Дамир Ришатович

100%



● Посещено



● Посещено ● Не посещено

# Итоги работы

**Мы воспользовались новейшими инструментами создания веб-сервисов и организации их архитектуры в коллаборации с передовыми методами машинного обучения. В итоге реализовано принципиально новое приложение, со следующими характеристиками:**

- **Удобный и простой интерфейс пользователя**
- **Современные технологии, обеспечивающие быстроту и стабильность сервиса**
- **Автоматизация процесса отметки посещения, путем простой загрузки фотографии в сервис**
- **Использование нейронных сетей для обработки полученных данных и формировании результатов**
- **Круглосуточный доступ ко всей нужной информации**
- **Вся необходимая статистика предоставляется быстро с возможностью ее визуализации в виде инфографики**