

# Проектирование и разработка музыкального секвенсора на веб-платформе

Корчинский Сергей Олегович  
Кафедра компьютерных технологий и  
электронного обучения,  
4 курс

# Актуальность

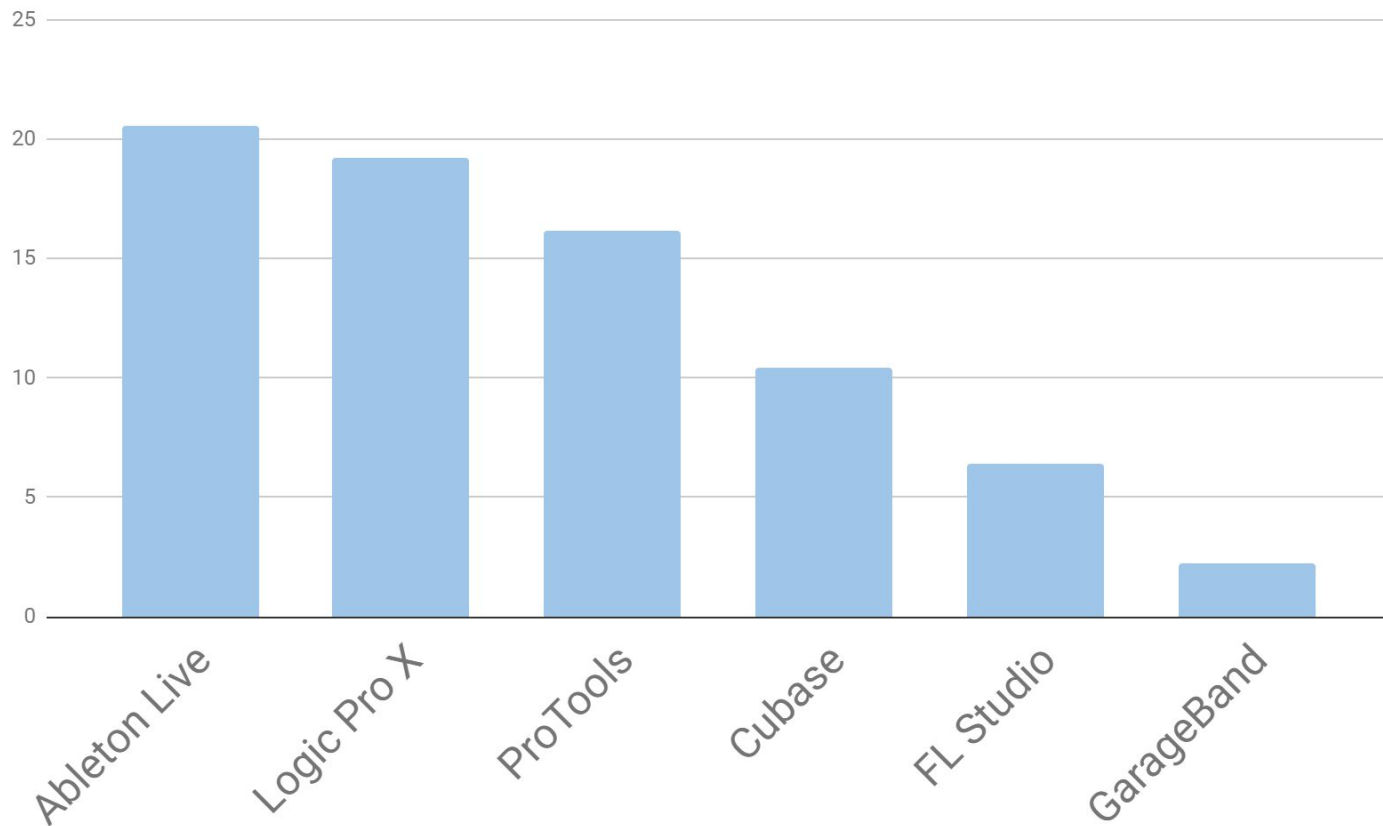
Несмотря на большое разнообразие DAW на рынке, ни один из них не является полностью кроссплатформенным решением. Стоимость подобных музыкальных секвенсоров часто является преградой для занятия любимым хобби. Необходимость создания DAW, доступной на разных платформах, определяет актуальность данной выпускной квалификационной работы.

# Цель и задачи

Цель: проектирование и разработка музыкального секвенсора на веб-платформе.

1. Анализ ключевых особенностей существующих DAW
2. Провести отбор наиболее подходящих инструментов разработки
3. Проектирование модели системы в соответствии с требованиями
4. Создание и тестирование программного продукта

# Анализ существующих DAW



# Основной функционал DAW

- Метроном
- Банк пресетов с различными инструментами и эффектами
- Пошаговый секвенсор для записи барабанной партии
- Поддержка клавиатуры как MIDI контроллера
- Поддержка внешних MIDI контроллеров
- Piano Roll для записи и редактирования партий виртуальных инструментов
- Рабочая область для расположения записанных партий

# Функциональные требования

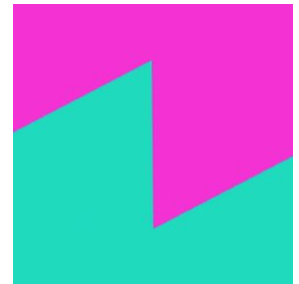
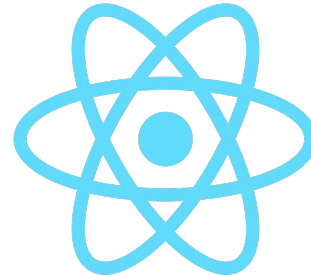
Создание полноценного музыкального секвенсора требует огромных ресурсов, поэтому было принято решение реализовать работоспособный прототип.

- Пошаговый секвенсор для записи барабанной партии
- Банк предустановленных инструментов
- Piano Roll для записи и редактировании партий виртуальных инструментов

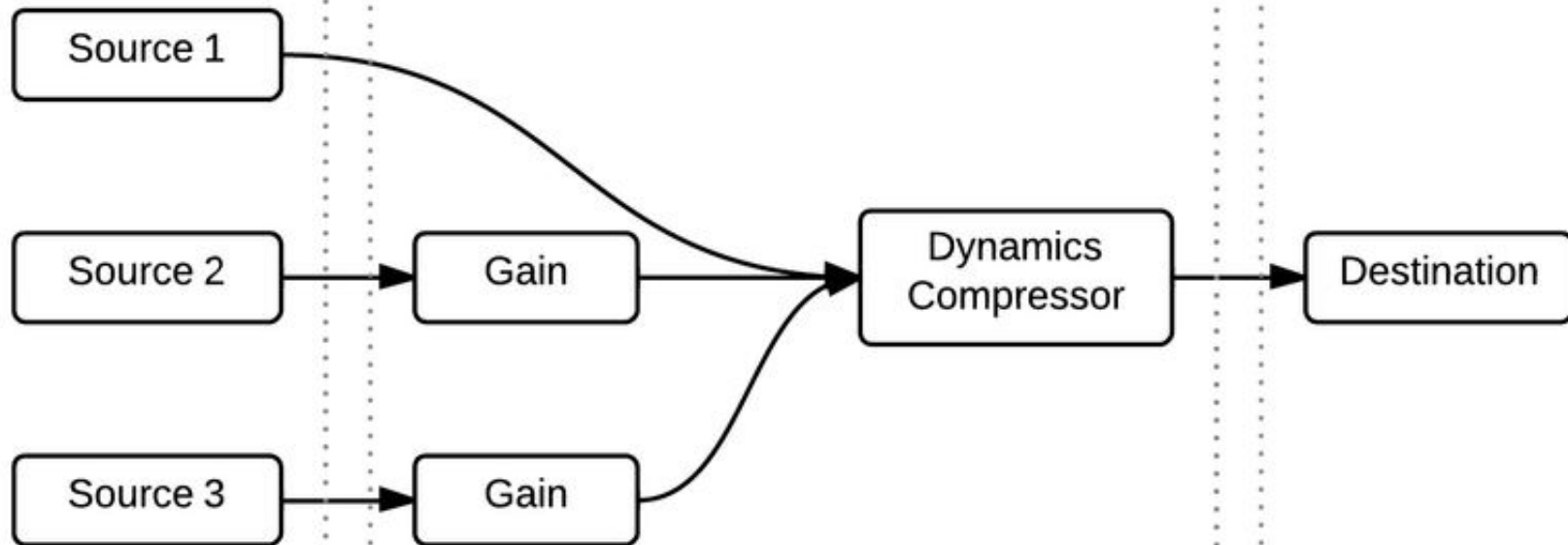
# Инструментальные средства

В качестве языка разработки был выбран JavaScript. Это единственный язык, который поддерживается всеми распространенными браузерами на данный момент.

Основной компонент системы - фреймворк Tone.js. В связке с React.js идеально подходит для реализации компонентного подхода.



## AudioContext



Input nodes

Processing nodes

Output nodes



# ReactJS и Tone.js

1. Записанная партия хранится в состоянии компонента
2. Объявляются функции-обработчики для событий интерфейса
3. После каждого обновления состояния, значения передаются в Tone.js
4. В соответствии с текущим состоянием вносятся изменения в интерфейс

Tempo (BPM): 128

## Управление темпом

Kick

HiHat

Snare

### Пошаговый секвенсор

B5

A5

G5

F5

E5

D5

C5

B4

A4

G4

F4

E4

D4

C4

### Piano Roll

# Заключение

В ходе выполнения работы были решены следующие задачи:

- Выделены основные возможности DAW и сформирован список функциональных требований;
- Проведен анализ и отбор оптимального инструментария для решения поставленной задачи.
- Спроектирована модель приложения. Описаны основные архитектурные особенности программного продукта.
- Разработан рабочий прототип продукта.