Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

ОТЧЁТ

по производственной практике

по модулю ПМ.02. ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

по специальности 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Зам. директора по учебной работе  \_\_\_\_\_\_\_\_\_\_\_\_/ З.З. Курмашева  подпись ФИО | Руководитель практики  от предприятия  \_\_\_\_\_\_\_\_\_\_\_\_/ Туманов Е.А.  подпись ФИО  М.П. |
| Руководитель практики  от учебного заведения  \_\_\_\_\_\_\_\_\_\_\_\_/ Каримова Р.Ф.  подпись ФИО | Студент группы 20П-1  \_\_\_\_\_\_\_\_\_\_\_\_/ Красильников Р.А.  подпись ФИО  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 год |

УФА - 2024 год

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| Введение |  |
| 1. Характеристика предприятия |  |
| 1. Техническое задание |  |
| 1. Описание предметной области |  |
| 1. Проектирование программного продукта |  |
| 1. Описание входной информации |  |
| 1. Описание выходной информации |  |
| 1. Описание структуры базы данных |  |
| 1. Контрольный пример |  |
| 1. Разработка программного продукта |  |
| 1. Тестирование программного продукта |  |
| 1. Техническая документация |  |
| Заключение |  |
| Список использованных источников |  |
| Приложение А. ХХХХХХХХХХХХХХХХХ |  |

ВВЕДЕНИЕ

Процесс интеграции программных модулей – это этап в жизненном цикле разработки программного обеспечения, который происходит после завершения разработки отдельных компонентов. Во время интеграции осуществляется объединение и тестирование различных модулей для обеспечения их взаимодействия и корректной работы в единой системе. Основная цель этого этапа – обеспечить совместную работу модулей, исключить конфликты и ошибки, а также обеспечить полноценное функционирование программного продукта.

Если в процессе интеграции обнаруживаются несоответствия или проблемы между модулями, специалист по интеграции может провести отладку кода и внести необходимые коррективы для обеспечения правильной работы системы в целом. Важным аспектом является обеспечение надежности программы, то есть гарантия ее способности соответствовать требованиям и обеспечивать стабильную работу после интеграции модулей.

Цели производственной практики: развитие профессионального взгляда на выбор методов проектирования, разработки, реализации и поддержки сложных программных продуктов, типа информационных систем, а также введение в круг проблем обеспечения правильности работы и надежности функционирования программных систем.

Задачи практики:

* разработка концептуальной модели задачи;
* использование современных инструментальных средств разработки программных средств;
* создание программного продукта;
* тестирование программного продукта;
* создание документации на сопровождение программного продукта.

# 1 Характеристика предприятия

Акционерное общество создано в декабре 2007 года. Наиболее известный проект компании — «Социальная карта Башкортостана». В апреле 2017 года впервые в России была выпущена многофункциональная Социальная карта Башкортостана на базе национальной платежной системы Мир. Сегодня АО «Башкирский регистр социальных карт» — это команда высококвалифицированных специалистов, способных решать задачи высочайшего уровня сложности в сфере информационных технологий. Наработанный опыт позволяет сделать сложнейшие технологии доступными для бизнеса и жителей Республики Башкортостан.

Компания предоставляет широкий спектр услуг: информационная безопасность, IT-услуги, продажа, регистрация, перерегистрация и обслуживание контрольно-кассовой техники, сопровождение автоматизированной информационной системы "Образование", а также многофункциональные карточные продукты: карта школьника, карта сотрудника, транспортная карта.

Организационная структура организации представлена на рисунке 1.1.

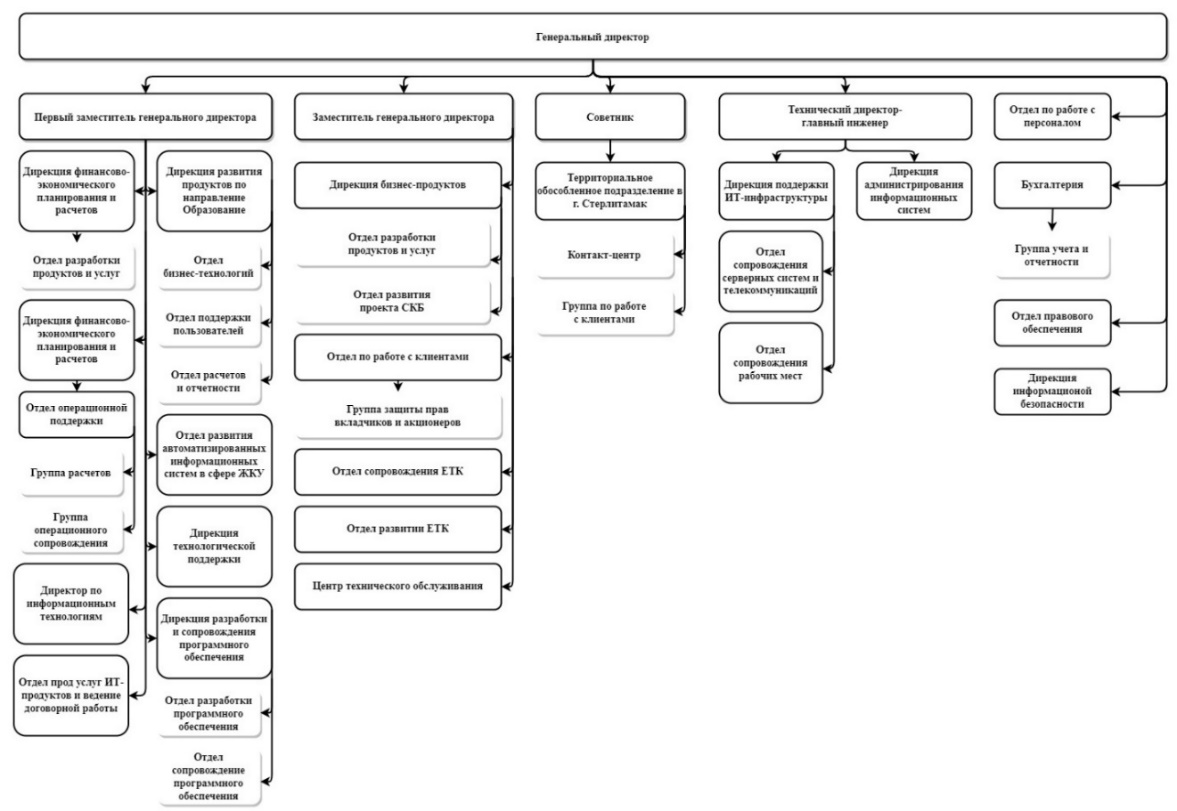


Рисунок 1.1 – Организационная структура организации

Схема расположения рабочих мест в отделе представлена на рисунке 1.2.



Рисунок 1.2. – Схема расположения рабочих мест в отделе разработки программного обеспечения группы образование

Программная архитектура:

* операционная система: Windows Server 2012 и Linux;
* сервер баз данных: Microsoft SQL Server 2016 и PostgreSQL;
* платформа управления контентом: OpenProject;
* электронная почта: Outlook 2010.

Техническая архитектура:

* серверы: Серверы под управлением Windows Server 2012 и Linux;
* базы данных: Базы данных Microsoft SQL Server 2016 и PostgreSQL;
* пользовательские устройства: Компьютеры с Windows 10 для доступа к системе;
* каналы связи: Используются Ethernet для связи между серверами и клиентскими устройствами.

2 Техническое задание

2.1 Общие сведения

Наименование проекта: Разработка онлайн секвенсора для создания музыки. Веб-сайт «ReSound» должен быть предназначен для создания собственных музыкальных произведений и их публикации. Пользователями программы должны выступать люди, которые хотят начать свою музыкальную карьеру. Веб-сайтом будет легко пользоваться новичкам, которые до этого не пользовались профессиональными программами.

2.2 Функциональные требования

2.2.1 Возможность создания музыкальных произведений с указанием следующих параметров:

* название;
* описание;
* приватность.

2.2.2 Возможность редактирования основных сведений музыкальных произведений:

* изменение названия проекта;
* изменение описания проекта;
* изменение приватности.

2.2.3 Возможность регистрации пользователя, где необходимо заполнить следующие поля:

* логин;
* пароль;
* электронная почта.

2.2.4 Возможность авторизации пользователя, где необходимо заполнить следующие поля:

* логин;
* пароль.

2.2.5 Возможность работать с содержимым музыкального произведения:

* шаблон;
* общая скорость воспроизведения;
* общая громкость;
* содержимое главной дорожки.

2.2.6 Возможность создавать новые шаблоны:

* название шаблона.

2.2.7 Возможность редактирования шаблонов:

* название шаблона;
* ноты;
* эффекты;
* громкость;
* звук.

2.2.8 В шаблоне должна быть возможность изменять ноты, а именно следующие параметры:

* название ноты;
* расположение на сетке;
* громкость;
* длина.

2.2.9 Возможность редактирования главной дорожки:

* добавление новых шаблонов;
* удаление шаблонов;
* перемещение шаблонов.

2.2.10 Возможность оставлять комментарии к опубликованным музыкальным произведениям.

2.2.11 Возможность оставлять оценку к опубликованным музыкальным произведениям.

2.2.12 Возможность просмотра опубликованных музыкальных произведений пользователей:

* поиск музыкальных произведений по названию или исполнителю;
* сортировка музыкальных произведений по дате, оценкам, количеству прослушиваний, количеству комментариев и количеству оценок;
* получение списка всех комментариев у музыкальных произведений;
* получение количества оценок и общей оценки музыкальных произведений.

2.2.13 Возможность просмотра профиля пользователей:

* просмотр музыкальных произведений пользователя;
* поиск музыкальных произведений по названию;
* просмотр данных пользователя.

2.2.14 Возможность просмотра собственного профиля:

* просмотр своих музыкальных произведений;
* поиск музыкальных произведений по названию;
* отслеживание статистики своих произведений;
* возможность просмотра своих данных.

2.2.15 Возможность отслеживания статистики своих произведений:

* вывод статистики количества прослушивания своих музыкальных произведений в выбранный период;
* вывод статистики популярных произведений.

2.3 Нефункциональные требования

* + 1. Кроссплатформенность:
* поддержка работы на OC семейства Windows, Linux и Mac;
* поддержка работы во всех браузерах.

2.3.2 Безопасность:

* логин и пароль для доступа к приложению.
  + 1. Удобство использования:
* простой и интуитивный интерфейс;
* информативные уведомления и подсказки.
  + 1. Производительность
* приложение должно иметь быстрый доступ к данным;
* минимальное время отклика на запросы пользователя.

2.4 Требования к реализации

* + 1. Языки программирования:
* для backend ASP .NET CORE C#;
* для frontend React TypeScript.
  + 1. Среды разработки:
* Microsoft Visual Studio 2022 Community;
* Visual Sudio Code.

2.4.3 СУБД: PostgreSQL

2.4.4 Среда для визуального проектирования баз данных: pgAdmin 4 v7

2.5 Требования к документации

2.5.1 Техническое задание на разработку программного модуля.

2.5.2 Руководство системному программисту.

2.5.3 Руководство пользователя.

3 Описание предметной области

Пусть требуется разработать онлайн секвенсор для создания музыкальных произведений и их публикации.

Каждое музыкальное произведений должно иметь следующие атрибуты:

* идентификатор произведения;
* идентификатор пользователя;
* название;
* описание;
* дата создания;
* дата последнего изменения;
* оценки;
* комментарии;
* содержимое проекта;
* приватность;
* обложка.

У каждого пользователя должна быть возможность создания своих музыкальны произведений. Для этого пользователю должно быть необходимо зарегистрироваться и войти в свой аккаунт.

Для регистрации пользователю будет необходимо заполнить следующее:

* логин;
* пароль;
* повтор пароля;
* электронная почта.

После регистрации веб-сайт должен перенаправлять пользователя на страницу его профиля. На этой странице у пользователя должны быть следующие возможности:

* изменение данных своего профиля;
* создания новых музыкальных произведений;
* просмотр списка своих музыкальных произведений;
* поиск своих музыкальных произведений по их названию;
* просмотр статистики своих музыкальных произведений;
* открытие своих музыкальных произведений для дальнейшей работы над ними;
* просмотр списка понравившихся музыкальных произведений;
* выход из аккаунта.

На странице профиля пользователь должен иметь возможность изменять свой профиль, а именно:

* добавление аватара;
* изменение логина;
* изменение пароля;
* изменение электронной почты.

После того как пользователь выйдет из аккаунта его должно перенаправить на страницу авторизации, где ему будет необходимо заполнить:

* логин;
* пароль.

Чтобы пользователь смог создавать свои музыкальные произведения, ему необходимо авторизоваться и перейти на страницу своего профиля, где должна быть кнопка для создания нового музыкального произведения.

После нажатия на кнопку создания нового музыкального произведения должно открываться модальное окно с полями, которые нужно заполнить для создания произведения:

* название (обязательно);
* описание (по желанию);
* приватность (по умолчанию true).

После заполнения полей и нажатия кнопки создать, пользователя должно перенаправить на главную страницу созданного музыкального произведения.

В содержимое музыкального произведения должно входить следующее:

* шаблоны;
* общая скорость воспроизведения;
* общая громкость;
* содержимое дорожек.

На главной странице музыкального произведения пользователь должен иметь возможность создавать свои шаблоны, в которых хранится следующая информация:

* ноты;
* эффекты;
* громкость;
* звук.

В шаблонах музыкального произведения пользователь должен иметь возможность работать с нотами, которые должны иметь следующие атрибуты:

* название ноты;
* расположение на сетке;
* громкость ноты;
* длина ноты.

На главной странице музыкального произведения должно быть множество дорожек, на которые можно размещать следующее:

* шаблоны;
* отдельные звуки.

Должна быть возможность экспортировать музыкальное произведение в такие форматы как:

* mp3;
* wav.

Экспортироваться должно только содержимое дорожек, на которые пользователь должен расположить свои шаблоны.

Музыкальное произведение должно иметь возможность сохраняться в файл собственного типа «.resound», с помощью которого можно будет открывать музыкальное произведение со всеми его параметрами и шаблонами. Также музыкальное произведение должно сохраняться и на самом веб-сайте, чтобы пользователь мог открывать свои работы через свой профиль.

Перед тем как заканчивать работу над своим музыкальным произведением пользователь должен сохранить свою работу. Если пользователь внесет какие-нибудь изменения в свою работу, то ему должно прийти сообщение что есть не сохраненные изменения, где пользователь может отказаться от внесенных раннее изменений или сохранить последние изменения.

Чтобы другие пользователи смогли просмотреть проект, необходимо изменить его приватность.

Все пользователи должны иметь возможность просматривать опубликованные проекты, но оставлять комментарии и оценку могут только авторизированные пользователи.

Комментарии должны иметь следующие атрибуты:

* id пользователя;
* id произведения;
* текст комментария;
* дата отправки.

Оценки должны иметь следующие атрибуты:

* id пользователя;
* id произведения;
* оценка.

В профиле пользователя должен находится список его музыкальных произведений, которые можно сортировать по дате изменения, дате создания и по количеству оценок.

В списке со всеми проектами пользователей должна отображаться следующая информация для каждого произведения:

* логин пользователя;
* аватар пользователя;
* название проекта;
* обложка;
* оценка;
* количество комментариев.

Обложка проекта генерируется сама при публикации проекта если пользователь сам не загружал обложку.

Должна быть возможность просмотра списка топ музыкальных произведений за неделю, месяц и все время.

В своем профиле у пользователя должна быть возможность просматривать статистику своих музыкальных произведений:

* вывод статистики количества прослушивания своих музыкальных произведений в выбранный период;
* вывод статистики популярных произведений.

4 Проектирование программного продукта

4.1 Диаграмма прецедентов

Диаграмма прецедентов или диаграмма вариантов использования в UML — диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе. Диаграмма прецедентов представлена на рисунке 4.1.1.

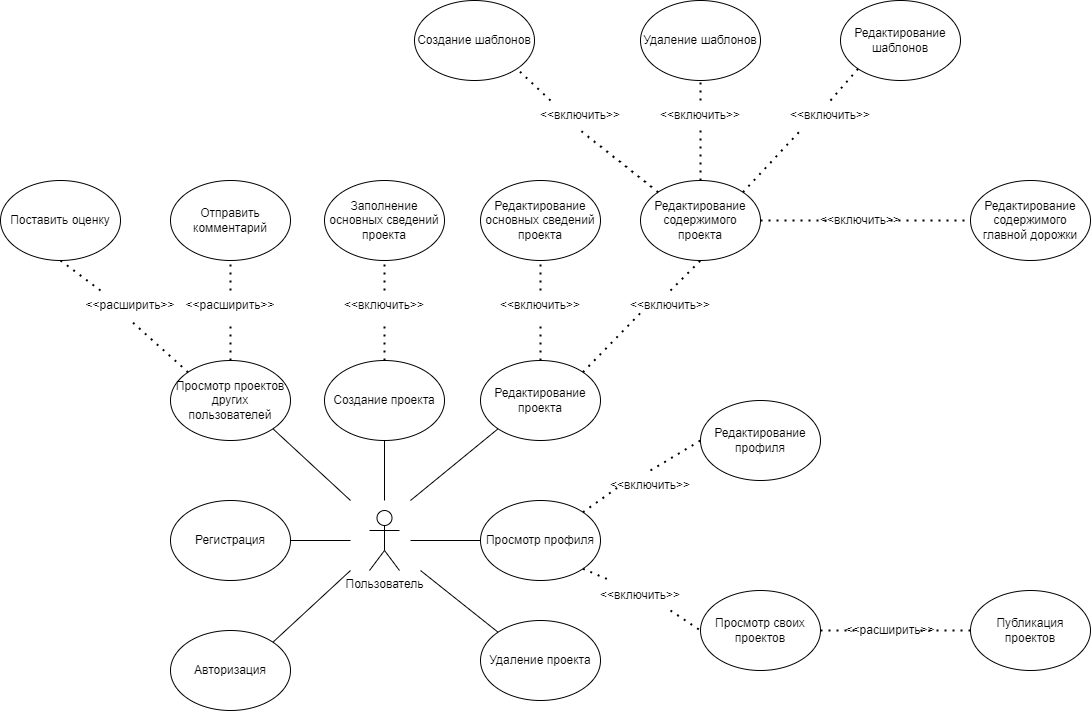


Рисунок 4.1.1 – Диаграмма прецедентов

5 Описание входной информации

Входной информацией является:

* аудио-семплы: Звуковые файлы, которые пользователь может загружать в секвенсор для использования в создании музыкальных композиций;
* шаблоны: Готовые музыкальные шаблоны или лупы, которые пользователь может использовать в своих композициях;
* настройки проекта: Общая скорость воспроизведения, темп, количество тактов и другие параметры проекта;
* список пользователей;
* опубликованные произведения
* комментарии пользователей под опубликованными произведениями
* оценки пользователей под опубликованными произведениями

6 Описание выходной информации

Выходной информацией является:

* аудио-файлы: Созданные пользователем музыкальные композиции, которые могут быть экспортированы в формате WAV, MP3 или других аудио-форматах;
* проектные файлы: Файлы проекта, содержащие информацию о композиции, расположении треков, настройках эффектов и других параметрах для возможности дальнейшего редактирования;
* ссылка на онлайн-проигрыватель: Пользователь может поделиться своей композицией, отправив другим уникальную ссылку на онлайн-проигрыватель, где можно прослушать результат;
* список топ опубликованных произведений.

7 Описание структуры базы данных

Описание структуры базы данных представлено в таблицах 7.1 – 7.10.

Таблица 7.1- Comment (Комментарии)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор комментария | Id | uuid(36) | первичный ключ |
| Идентификатор пользователя | UserId | uuid(36) | внешний ключ (к User) |
| Идентификатор проекта | ProjectId | uuid(36) | внешний ключ (к Project) |
| Текст комментария | Text | text | обязательное поле |
| Дата отправки | CreatedTime | timestamp with time zone(8 байт) | обязательное поле |

Таблица 7.2 - MainTrack(Главная дорожка)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор главной дорожки | Id | uuid(36) | первичный ключ |
| Контент | Content | text | обязательное поле |

Таблица 7.3 - Mark(Оценка)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор оценки | Id | uuid(36) | первичный ключ |
| Идентификатор пользователя | UserId | uuid(36) | внешний ключ (к User) |
| Идентификатор проекта | ProjectId | uuid(36) | внешний ключ (к Project) |
| Оценка | Value | integer(4 байта) | обязательное поле |

Таблица 7.4 - Project(Проект)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор проекта | Id | uuid(36) | первичный ключ |
| Идентификатор пользователя | UserId | uuid(36) | внешний ключ (к User) |
| Идентификатор главной дорожки | MainTrackId | uuid(36) | внешний ключ (к MainTrack) |
| Название проекта | Name | text | обязательное поле |
| Описание проекта | Description | text | обязательное поле |
| Дата создания | CreatedTime | timestamp with time zone(8 байт) | обязательное поле |
| Дата последнего обновления | UpdatedTime | timestamp with time zone(8 байт) | обязательное поле |
| Приватность | Private | boolean(1 байт) | обязательное поле |
| Обложка | PicturePath | text | обязательное поле |
| Общая скорость воспроизведения проекта | PlaybackSpeed | integer(4 байта) | обязательное поле |
| Общая громкость проекта | Volume | integer(4 байта) | обязательное поле |

Таблица 7.5 - ProjectTemplate(Шаблоны проектов)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор | Id | uuid(36) | первичный ключ |
| Идентификатор шаблона | TemplateId | uuid(36) | внешний ключ (к Template) |
| Идентификатор проекта | ProjectId | uuid(36) | внешний ключ (к Project) |

Таблица 7.6 - Template(Шаблон)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор | Id | uuid(36) | первичный ключ |
| Идентификатор звука | SoundId | uuid(36) | внешний ключ (к Sound) |
| Название шаблона | Name | text | обязательное поле |
| Ноты | Notes | text | обязательное поле |
| Громкость | Volume | integer(4 байта) | обязательное поле |

Таблица 7.7 - Sound(Звук)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор | Id | uuid(36) | первичный ключ |
| Название звука | Name | text | обязательное поле |
| Путь к звуку | Path | text | обязательное поле |

Таблица 7.8 - Users(Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор | Id | uuid(36) | первичный ключ |
| Логин | Login | text | обязательное поле |
| Пароль | Password | text | обязательное поле |
| Электронная почта | Email | text | обязательное поле |
| Аватар | Picture | text | обязательное поле |

Схема данных – это структура базы данных, описанная на формальном языке, поддерживаемом СУБД (системой управления базы данных). В реляционных базах данных схема определяет таблицы, поля в каждой таблице и ограничения целостности, такие как первичный и внешний ключи.

Схема данных представлена на рисунке 7.1.

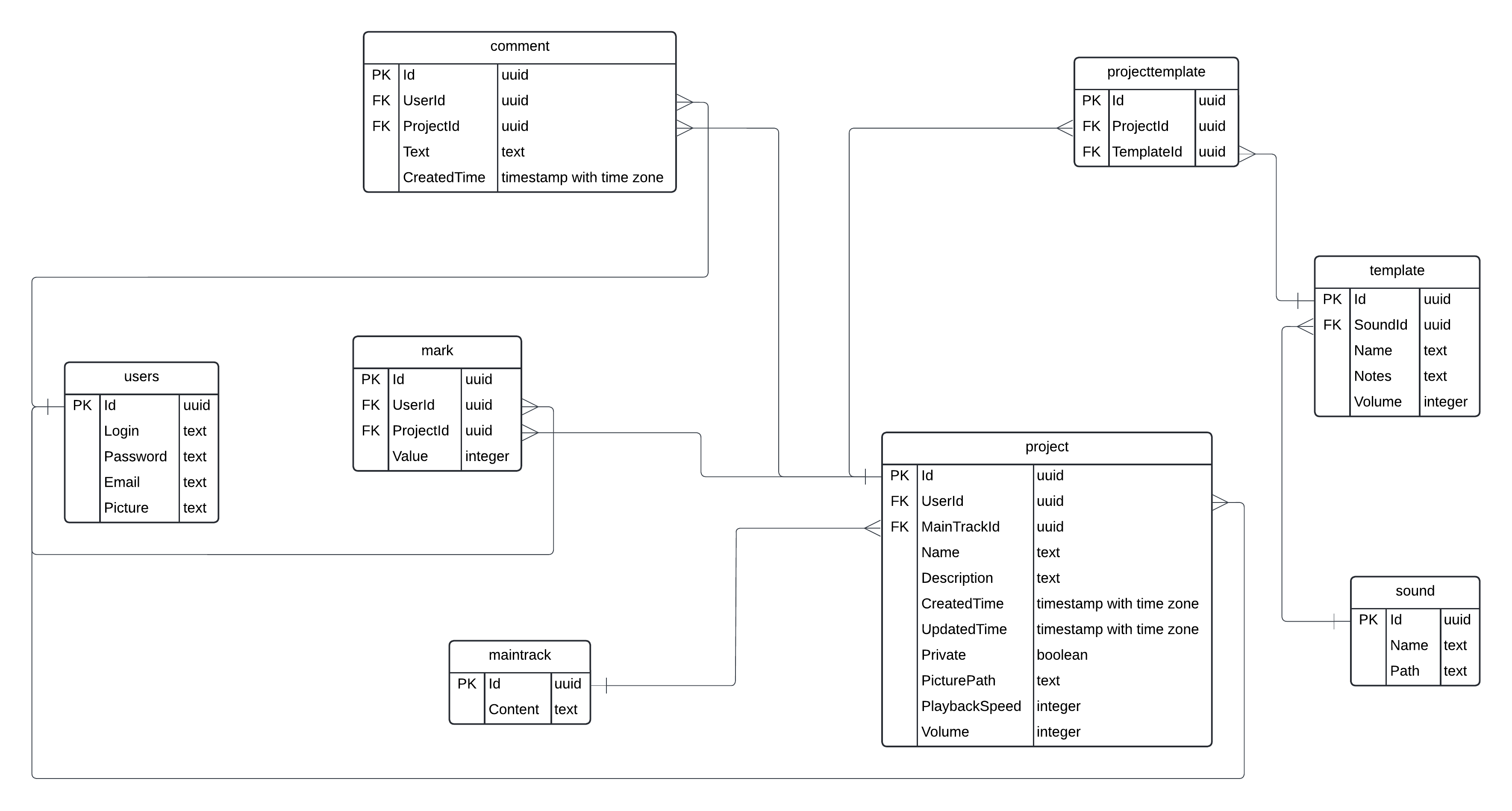


Рисунок 7.1 – Схема базы данных

8 Контрольный пример

Контрольный пример является ручным подсчётом задачи. По составленной программе обрабатываются исходные данные контрольного примера. Полученные результаты сравниваются с известными результатами контрольного примера. При несовпадении результатов производится поиск, исправление ошибок, и снова производится выполнение программы.

Таблица 8.1 Контрольный пример

|  |  |  |
| --- | --- | --- |
| Действия | Входные данные | Выходные данные |
| 1 | 2 | 3 |
| Страница «Шаблон» | | |
| Поле изменение скорости проигрывания | Некорректные:  BPM: 0 | BPM: 10 |
|  | Некорректные:  BPM: 777 | BPM: 522 |
| Очистка сетки с нотами | Ноты: A4, C5, D5  Столбцы:1,2,3 | Ноты: пусто |
| Выбор звука | Название звука: Kalimba | URL: https://tonejs.github.io/audio/berklee/  Filename: Kalimba\_1.mp3 |
| Запись шаблона | Ноты: A4, C5, D5  Столбцы:1,2,3 | Файл: recording.ogg |
| Сохранение шаблона | Ноты: A4, C5, D5  Столбцы:1,2,3  Название файла: template | Файл: template.resound |
| Загрузка шаблона | Файл: template.resound | Ноты: A4, C5, D5  Столбцы:1,2,3 |

9 Разработка программного продукта

Клиентская часть веб-сайта «ReSound» была написана при помощи фреймворка React на языке TypeScript. Программа имеет модульную структуру. Модульная схема представлена на рисунке 9.1.1.

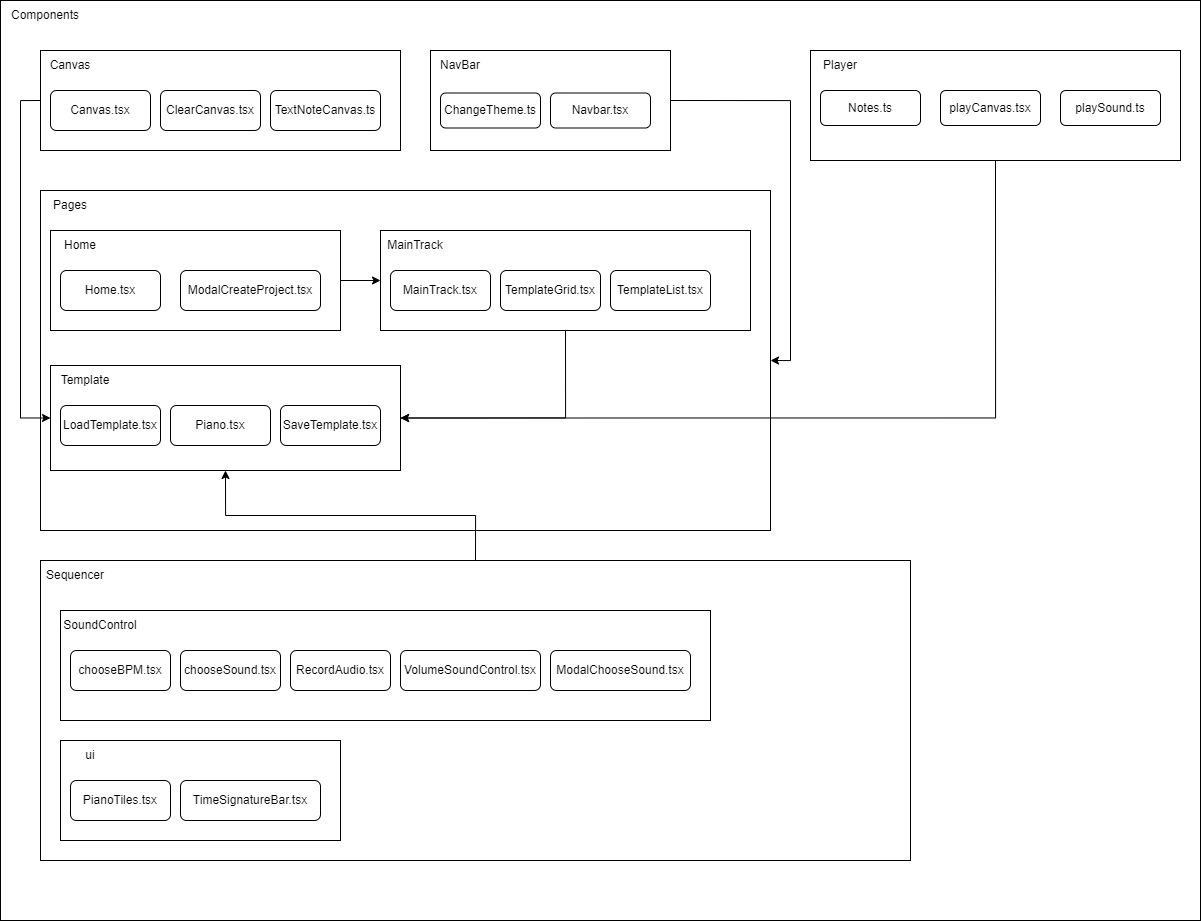


Рисунок 9.1.1 – Модульная схема

Описание модулей и процедур указано в таблице 9.1.1.

Таблица 9.1.1 – Описание модулей и процедур

|  |  |
| --- | --- |
| Процедуры | Назначение |
| 1 | 2 |
| Модуль Canvas.tsx | |
| function ClearCanvas() | Очистка нот |
| function drawCells() | Отрисовка ячеек с нотами |
| function handleClick(row: number, col: number) | Событие при нажатии на ячейку сетки |
| function handleMouseMove(e: React.MouseEvent<HTMLCanvasElement>) | Событие при перемещении зажатой кнопки мыши |
| function handleMouseUp() | Событие при отпускании кнопки мыши |
| function loadTemplate() | Загрузка шаблона |
| Модуль ModalCreateProject.tsx | |
| showModal() | Отображение модального окна создания проекта |
| HandleOk() | Создание нового проекта |
| handleCancel() | Отмена |
| Модуль MainTrack.tsx | |
| showDrawer() | Открытие меню слева для сохранения файла |
| onClose() | Закрытие меню слева |
| closePage() | Закрытие страницы и переход на профиль |
| handleCreateTemplate() | Создание шаблона |
| Модуль TemplateList.tsx | |
| openTemplate() | Открытие шаблона |
| Модуль loadTemplate.tsx | |
| const loadNotes = (file: File): Promise<number> | Загрузка шаблона |
| Модуль saveTemplate.tsx | |
| const saveNotes = () | Сохранение шаблона |
| Модуль Piano.tsx | |
| const addCols = () | Добавление столбцов |
| Модуль changeTheme.ts | |
| function changeTheme() | Смена темы |
| Модуль chooseBPM.tsx | |
| const handleBpmChange = (value: number | null) | Событие при изменении значения текстового поля, устанавливается выбранная скорость воспроизведения |
| Модуль chooseSound.tsx | |
| function playExample(url: string, filename: string) | Демонстрация выбранного звука |
| const ChooseSound = (soundname: string) | Выбор звука |
| Модуль RecordAudio.tsx | |
| function GetNotesPlay() | Получение нот для их записи |
| function playNote() | Проигрыш нот для записи |
| Модуль VolumeSoundControl.tsx | |
| const handleVolumeChange = (value: number) | Событие при изменении значения слайдера громкости |
| Модуль PianoTiles.tsx | |
| function keyColor(note: string): string | Изменение цвета нот в зависимости от того диез это или бемоль |
| Модуль Notes.ts | |
| function Getnotes(): string[] | Получение списка нот |
| Модуль playCanvas.tsx | |
| function Getnotes(): string[] | Получение списка нот |
| function GetNotesPlay() | Получение нот которые можно воспроизвести |
| function checkNotesPlay() | Проверка на существование нот в следующих секторах |
| function playNote() | Воспроизведение нот |
| function play() | Запустить проигрыш нот |
| function changePosition(newPosition: number) | Смена позиции воспроизведения |
| async function stop() | Остановка воспроизведения и сброс позиции |
| function pause() | Остановка воспроизведения |
| Модуль playSound.ts | |
| function newsampler(newurl: string, newfile: string) | Обновление звука |
| function newsamplervolume(volume: number) | Обновление громкости |

10 Тестирование программного продукта

Таблица 10.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта | ReSound |
| Номер версии | 0.6 |
| Имя тестера | Красильников Роман Арсеньевич |
| Даты тестирования | 11.04.2024 |

Таблица 10.2 – Тестирование создания проекта

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования | Создания нового проекта |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * нажатие на кнопку «+»; * открытие модального окна создание проекта; * заполнение обязательных полей; * нажатие на кнопку «Создать» * переход на страницу нового проекта. |
| Данные тестирования | Название: новый проект  Описание:  Приватность: false |
| Ожидаемый результат | Переход на страницу нового проекта |
| Фактический результат | Переход на страницу нового проекта |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница профиля пользователя |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результаты тестирования создания проекта представлены на рисунках 10.2.1 и 10.2.2.

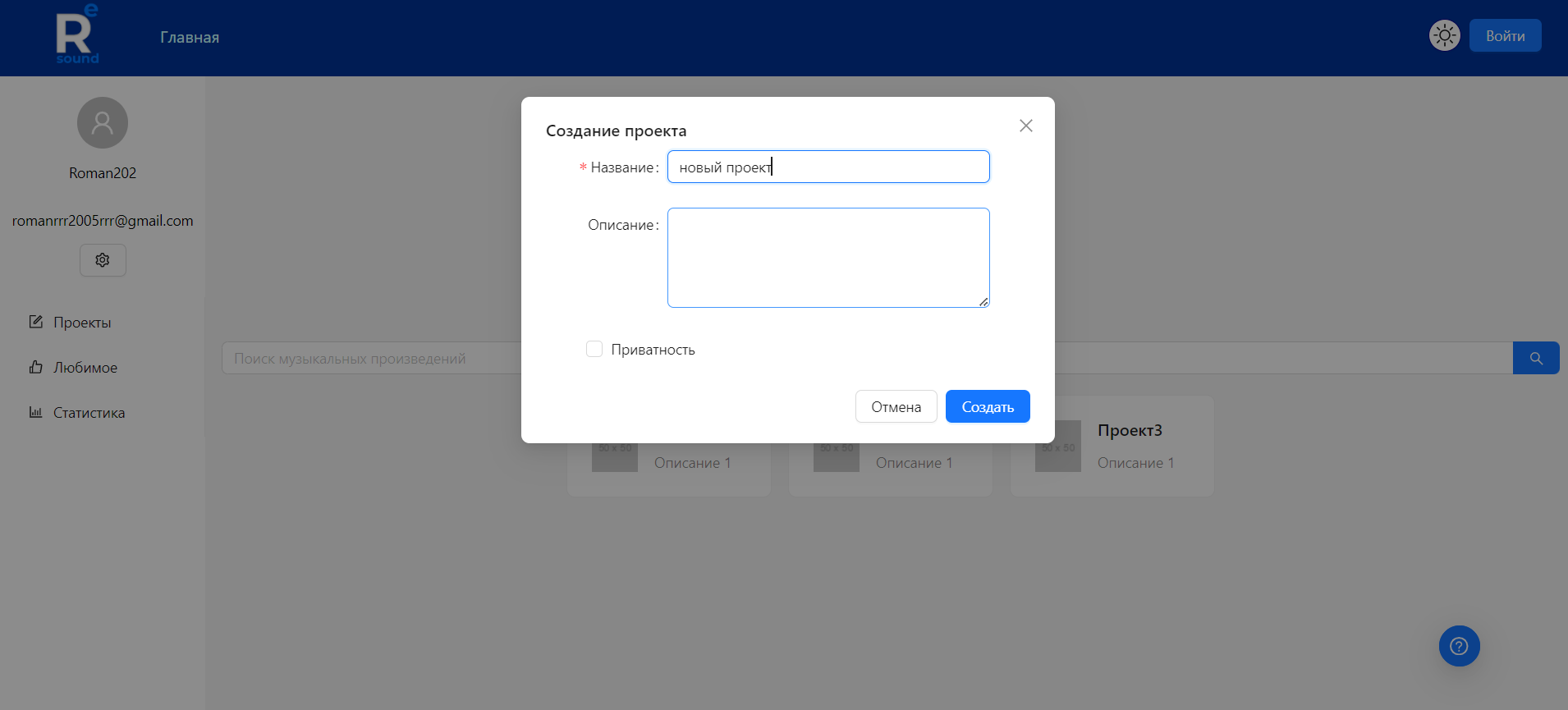


Рисунок 10.2.1 – Модальное окно создания проекта

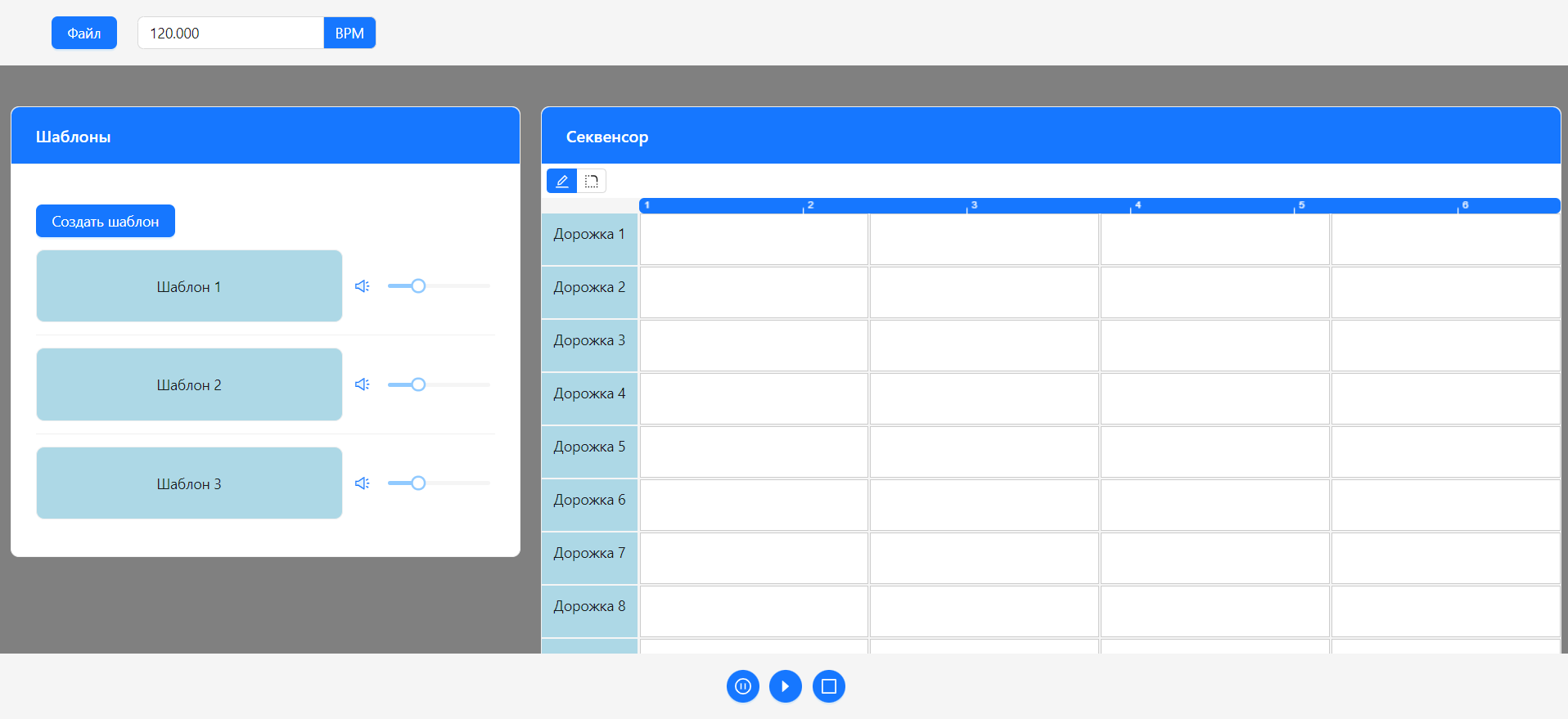


Рисунок 10.2.2 – Успешное создание проекта и переход на главную страницу проекта

Таблица 10.3 – Тестирование перехода на страницу шаблона

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_2 |
| Приоритет тестирования | Высокий |
| Название тестирования | Переход на страницу шаблона |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | Нажатие на один из шаблонов |
| Данные тестирования | Шаблон: Шаблон 1 |
| Ожидаемый результат | Переход на страницу шаблона |
| Фактический результат | Переход на страницу шаблона |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница проекта |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результат тестирования перехода на страницу шаблона представлен на рисунке 10.3.1.

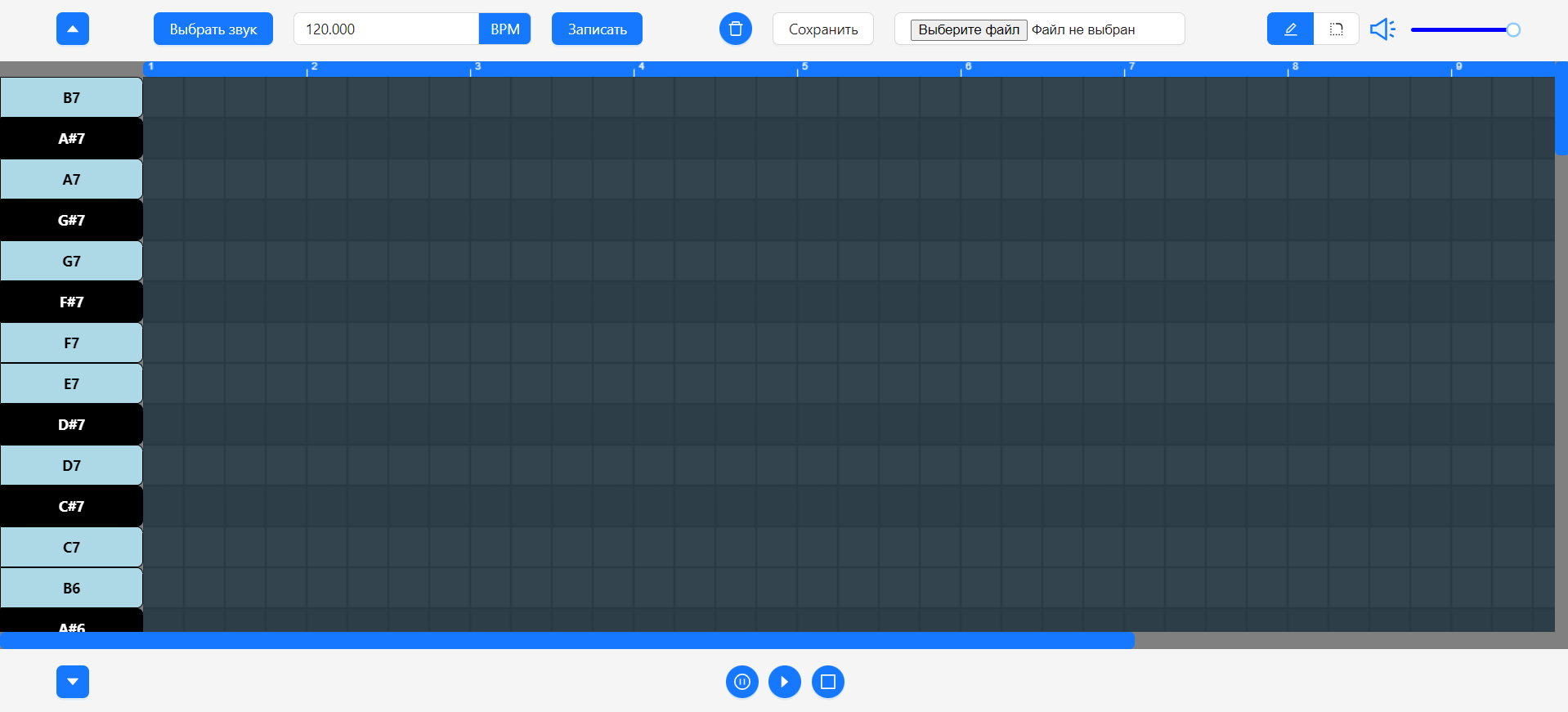


Рисунок 10.3.1 – Тестирование перехода на страницу шаблона

Таблица 10.4 – Тестирование размещения нот на сетке

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_3 |
| Приоритет тестирования | Высокий |
| Название тестирования | Размещение нот на сетке |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | Нажатие на ячейку сетки |
| Данные тестирования | Нота: A4  Столбец: 3 |
| Ожидаемый результат | Ячейка закрашивается и проигрывается звук ноты |
| Фактический результат | Ячейка закрашивается и проигрывается звук ноты |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результат тестирования размещения ноты на сетке представлен на рисунке 10.4.1.

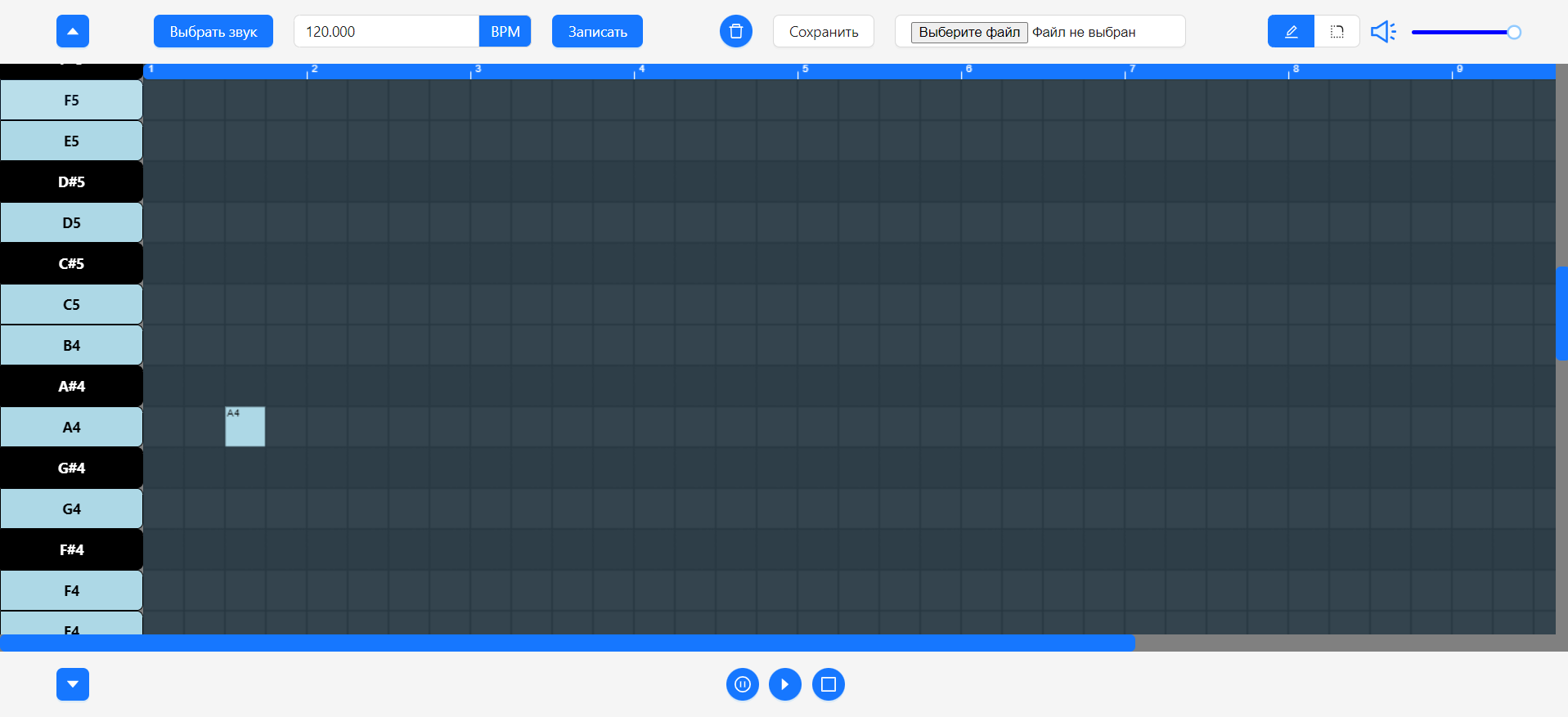


Рисунок 10.4.1 – Тестирование размещения ноты на сетки

Таблица 10.5 – Тестирование смены звука

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_4 |
| Приоритет тестирования | Высокий |
| Название тестирования | Смена звука |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * нажатие на кнопку «Выбрать звук»; * открытие модального окна выбора звука; * нажатие на кнопку «Kalimba»; * воспроизведение выбранного звука; * нажатие на кнопку «Ок»; * нажатие на ячейку сетки. |
| Данные тестирования | Нота: A4  Столбец: 3 |
| Ожидаемый результат | Ячейка закрашивается и проигрывается нота с выбранным звуком |
| Фактический результат | Ячейка закрашивается и проигрывается нота с выбранным звуком |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результат тестирования смены звука представлен на рисунке 10.5.1.

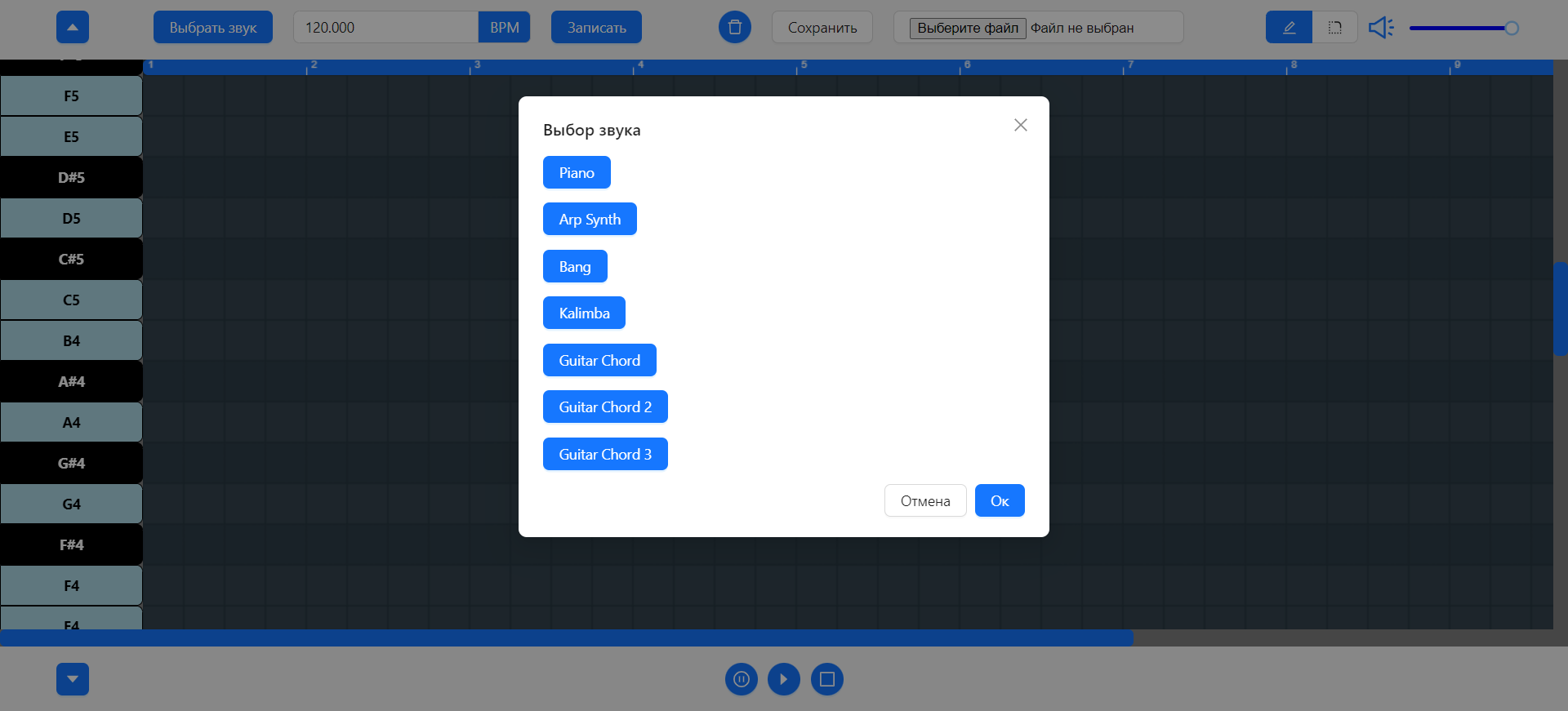


Рисунок 10.5.1 – Модальное окно смены звука

Таблица 10.6 – Тестирование очистки всех нот на сетки

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_5 |
| Приоритет тестирования | Высокий |
| Название тестирования | Очистка всех нот на сетке |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * нажатие на пару ячеек сетки; * нажатие на кнопку «Очистить». |
| Данные тестирования | Ноты: A4, C5  Столбец: 3, 4 |
| Ожидаемый результат | Очистка всех нот на сетке |
| Фактический результат | Очистка всех нот на сетке |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результаты тестирования очистки всех нот на сетке представлены на рисунках 10.6.1 и 10.6.2.

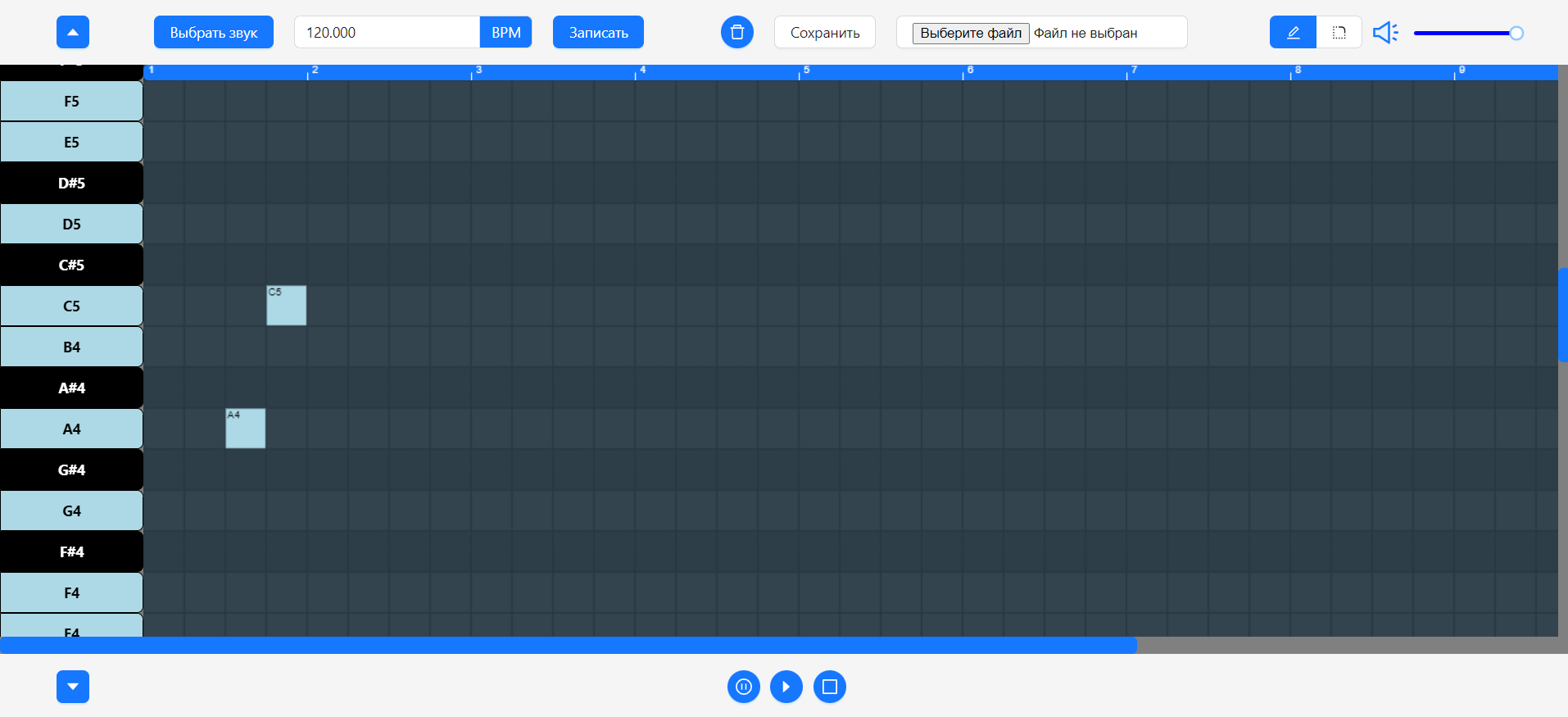


Рисунок 10.6.1 – Сетка до очистки

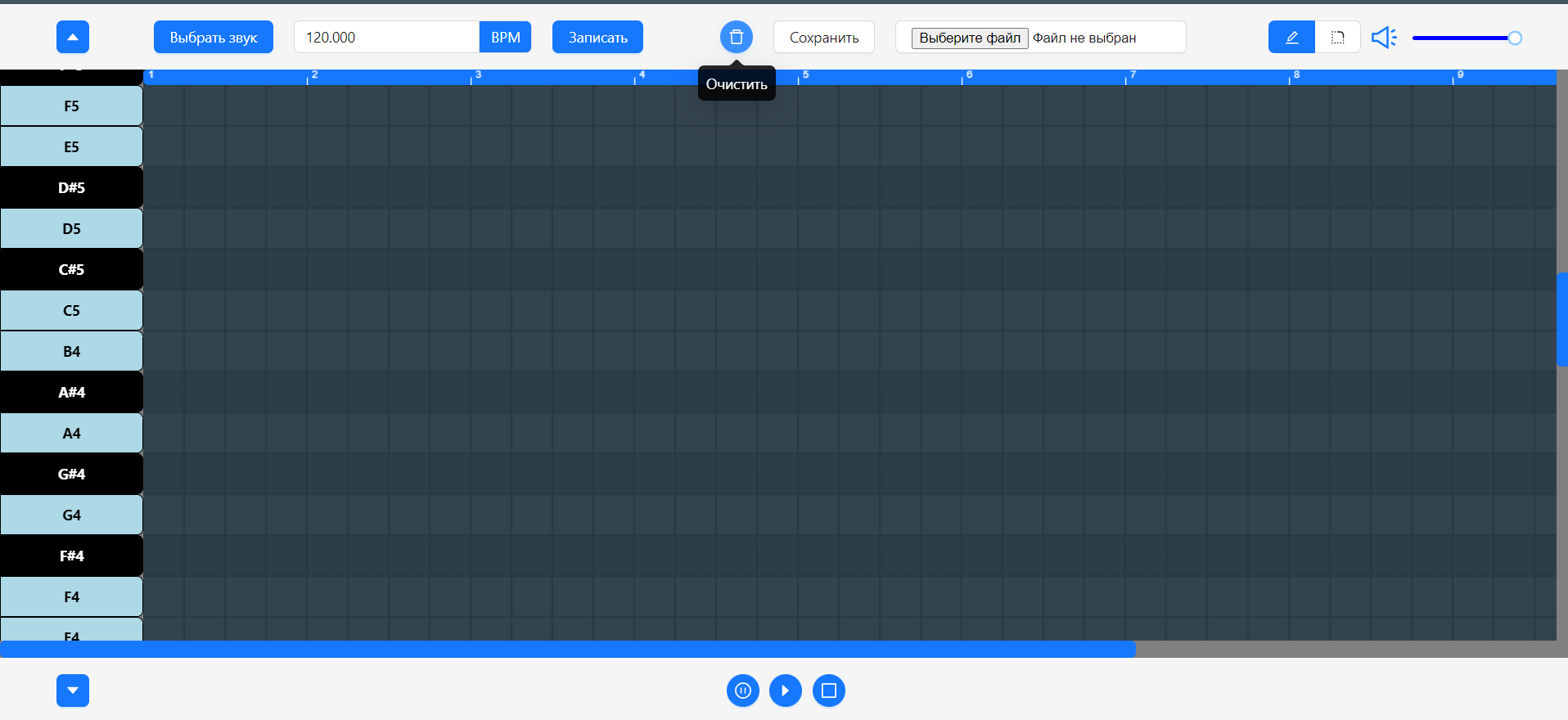


Рисунок 10.6.2 – Сетка после очистки

Таблица 10.7 – Тестирование добавление новых столбцов на сетке нот

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_6 |
| Приоритет тестирования | Высокий |
| Название тестирования | Добавление новых столбцов на сетке нот |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | Нажатие на кнопку «+» |
| Данные тестирования | Количество столбцов на сетке: 48 |
| Ожидаемый результат | Добавление еще 48 столбцов |
| Фактический результат | Добавление еще 48 столбцов |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результаты тестирования добавления новых столбцов на сетке нот представлены на рисунках 10.7.1 и 10.7.2.

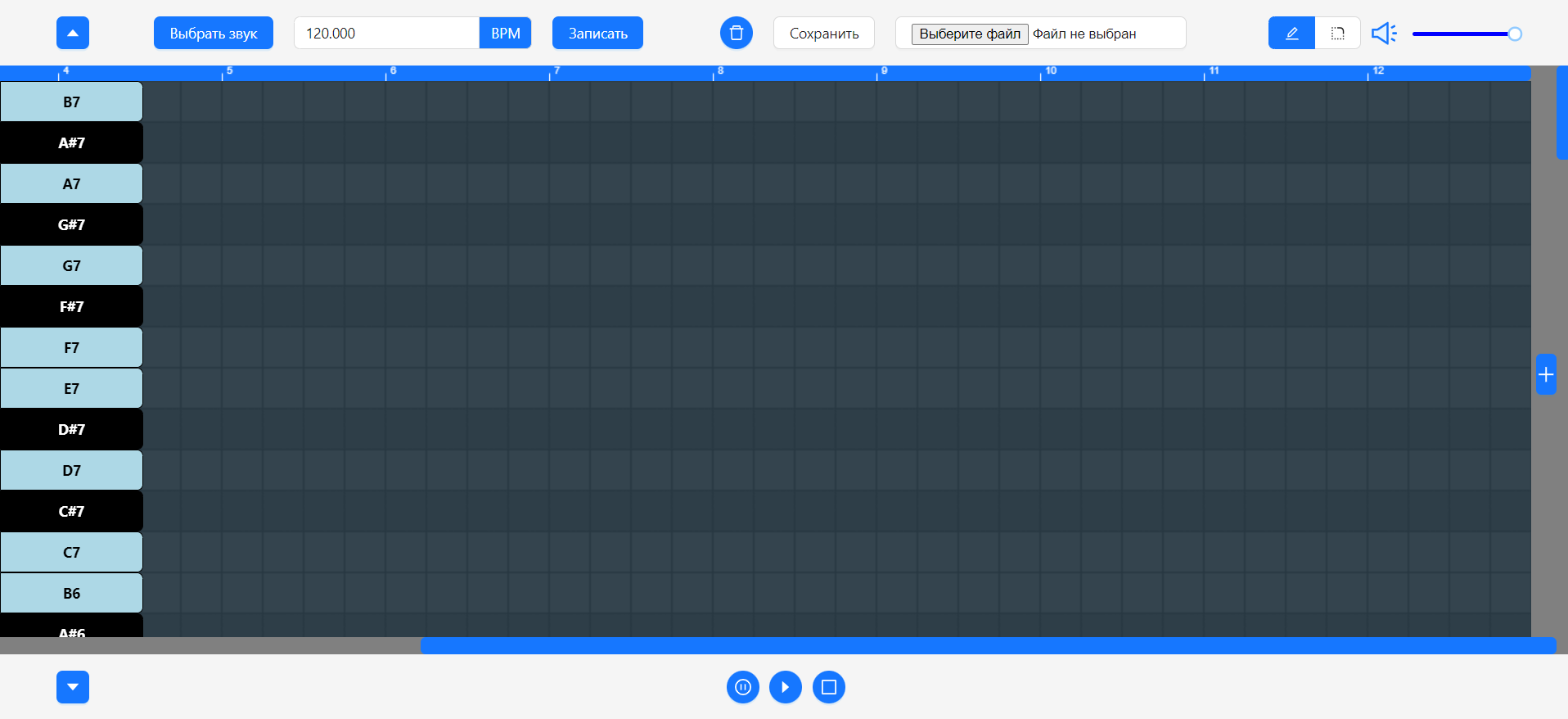


Рисунок 10.7.1 – Сетка до добавления новых столбцов

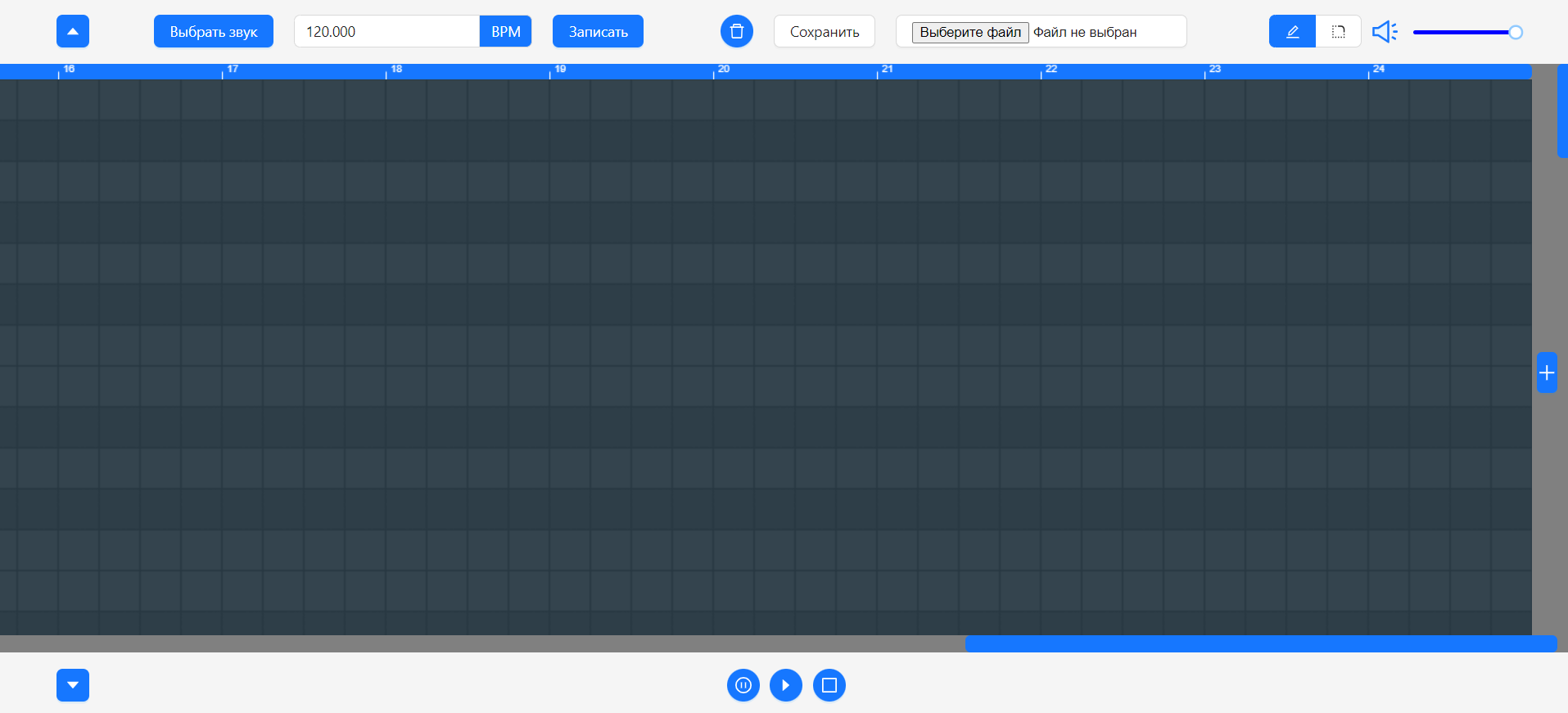


Рисунок 10.7.2 – Сетка после добавления новых столбцов

Таблица 10.8 – Тестирование воспроизведения мелодии

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_7 |
| Приоритет тестирования | Высокий |
| Название тестирования | Воспроизведение мелодии |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * размещение нот на сетки; * нажатие на кнопку «Пуск». |
| Данные тестирования | Ноты: D4, F4, A4  Столбцы: 1, 2, 3 |
| Ожидаемый результат | Воспроизведение мелодии |
| Фактический результат | Воспроизведение мелодии |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результаты тестирования воспроизведения представлены на рисунках 10.8.1 и 10.8.2.

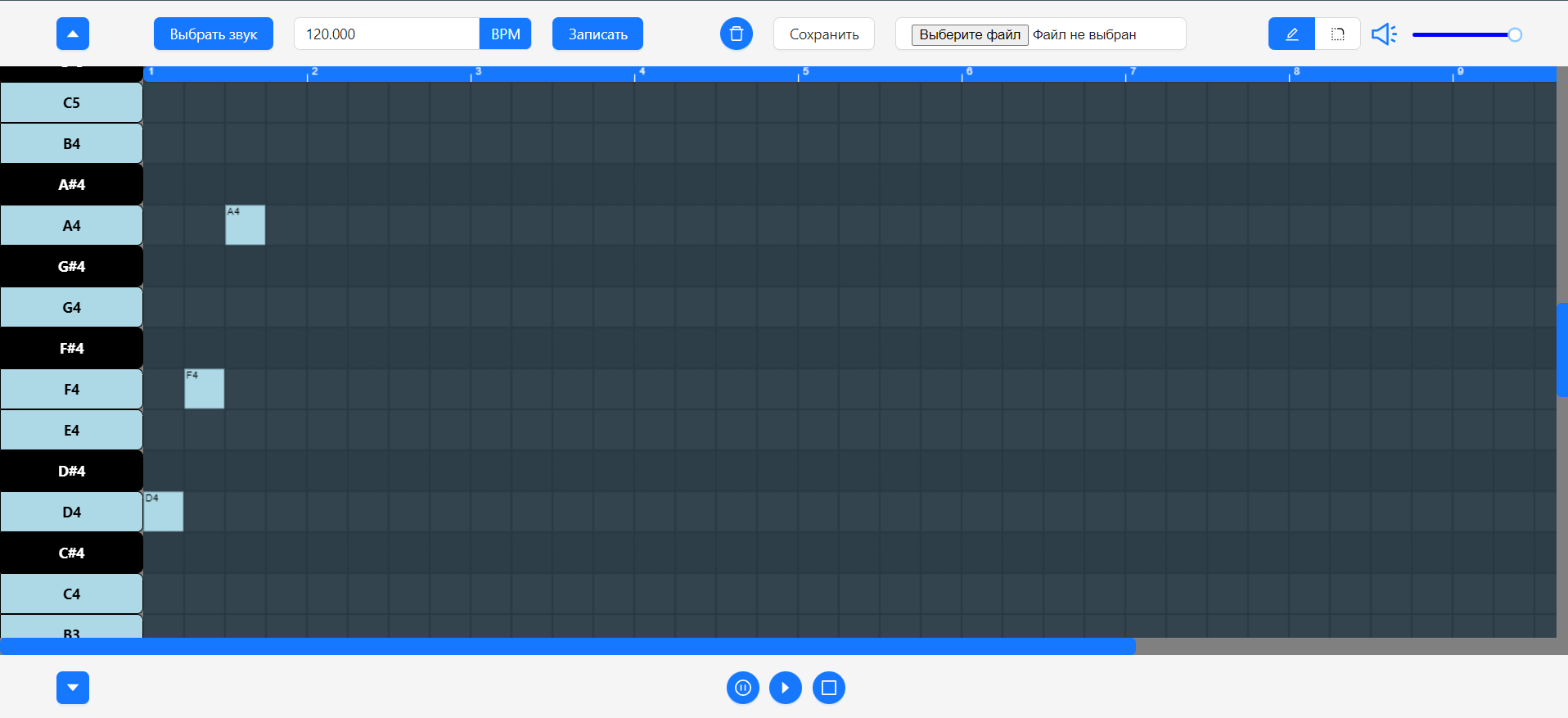


Рисунок 10.8.1 – Сетка нот перед запуском мелодии

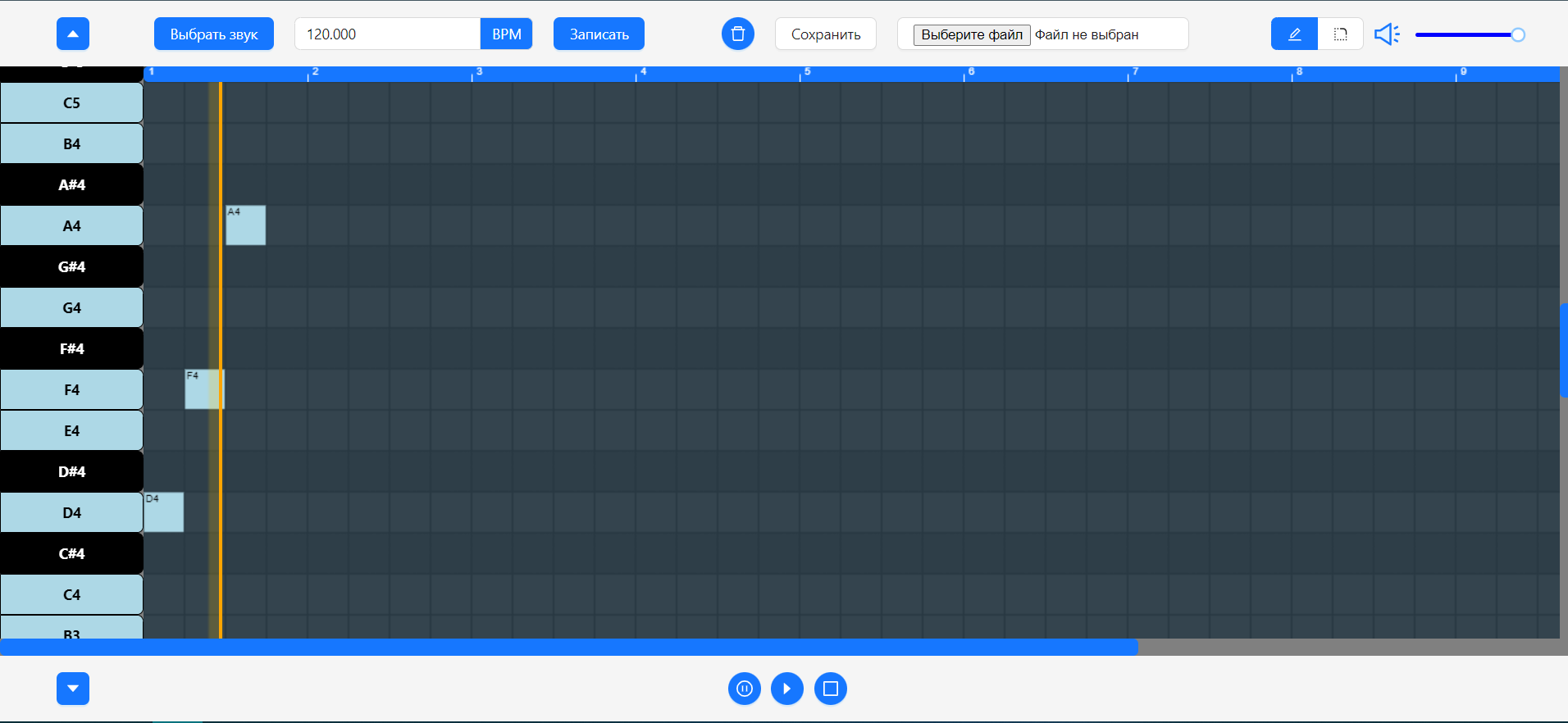


Рисунок 10.8.2 – Сетка нот после запуска мелодии

Таблица 10.9 – Тестирование изменения скорости воспроизведения мелодии

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_8 |
| Приоритет тестирования | Высокий |
| Название тестирования | Изменения скорости воспроизведения мелодии |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * размещение нот на сетки; * изменить значение скорости в поле «BPM»; * нажатие на кнопку «Пуск». |
| Данные тестирования | Ноты: D4, F4, A4  Столбцы: 1, 2, 3  BPM: 200 |
| Ожидаемый результат | Воспроизведение ускоряется |
| Фактический результат | Воспроизведение ускоряется |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Таблица 10.10 – Тестирование паузы

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_9 |
| Приоритет тестирования | Высокий |
| Название тестирования | Пауза |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * размещение нот на сетки; * нажатие на кнопку «Пуск»; * когда временная полоса достигнет 2 столбца нажать на кнопку «Пауза». |
| Данные тестирования | Ноты: D4, F4, A4  Столбцы: 1, 2, 3 |
| Ожидаемый результат | Воспроизведение останавливается |
| Фактический результат | Воспроизведение останавливается |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Таблица 10.11 – Тестирование кнопки «Стоп»

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_10 |
| Приоритет тестирования | Высокий |
| Название тестирования | Стоп |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * размещение нот на сетки; * нажатие на кнопку «Пуск»; * когда временная полоса достигнет 2 столбца нажать на кнопку «Стоп»; * нажатие на кнопку «Пуск». |
| Данные тестирования | Ноты: D4, F4, A4  Столбцы: 1, 2, 3 |
| Ожидаемый результат | Воспроизведение прекращается и при повторном запуске начинается сначала |
| Фактический результат | Воспроизведение прекращается и при повторном запуске начинается сначала |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Таблица 10.12 – Тестирование изменения громкости

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_11 |
| Приоритет тестирования | Высокий |
| Название тестирования | Изменение громкости |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * размещение нот на сетки; * изменение значения громкости на слайдере; * нажатие на кнопку «Пуск»; |
| Данные тестирования | Ноты: D4, F4, A4  Столбцы: 1, 2, 3  Громкость: 68 |
| Ожидаемый результат | Громкость воспроизводимой музыки становится меньше |
| Фактический результат | Громкость воспроизводимой музыки становится меньше |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Таблица 10.13 – Тестирование записи мелодии из шаблона

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_12 |
| Приоритет тестирования | Высокий |
| Название тестирования | Запись мелодии из шаблона |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * размещение нот на сетки; * нажатие на кнопку «Записать»; |
| Данные тестирования | Ноты: D4, F4, A4  Столбцы: 1, 2, 3 |
| Ожидаемый результат | Проигрывается мелодия, а после загружается файл с записанной мелодией |
| Фактический результат | Проигрывается мелодия, а после загружается файл с записанной мелодией |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результат тестирования записи мелодии представлен на рисунке 10.13.1.

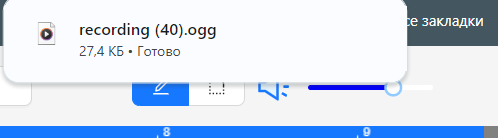


Рисунок 10.13.1 – Сохранение записи мелодии

Таблица 10.14 – Тестирование сохранения шаблона

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_13 |
| Приоритет тестирования | Высокий |
| Название тестирования | Сохранение шаблона |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * размещение нот на сетки; * нажатие на кнопку «Сохранить»; * ввести название файла. |
| Данные тестирования | Ноты: D4, F4, A4  Столбцы: 1, 2, 3 |
| Ожидаемый результат | Загрузка файла шаблона |
| Фактический результат | Загрузка файла шаблона |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результаты тестирования сохранения шаблона представлены на рисунках 10.14.1 и 10.14.2.

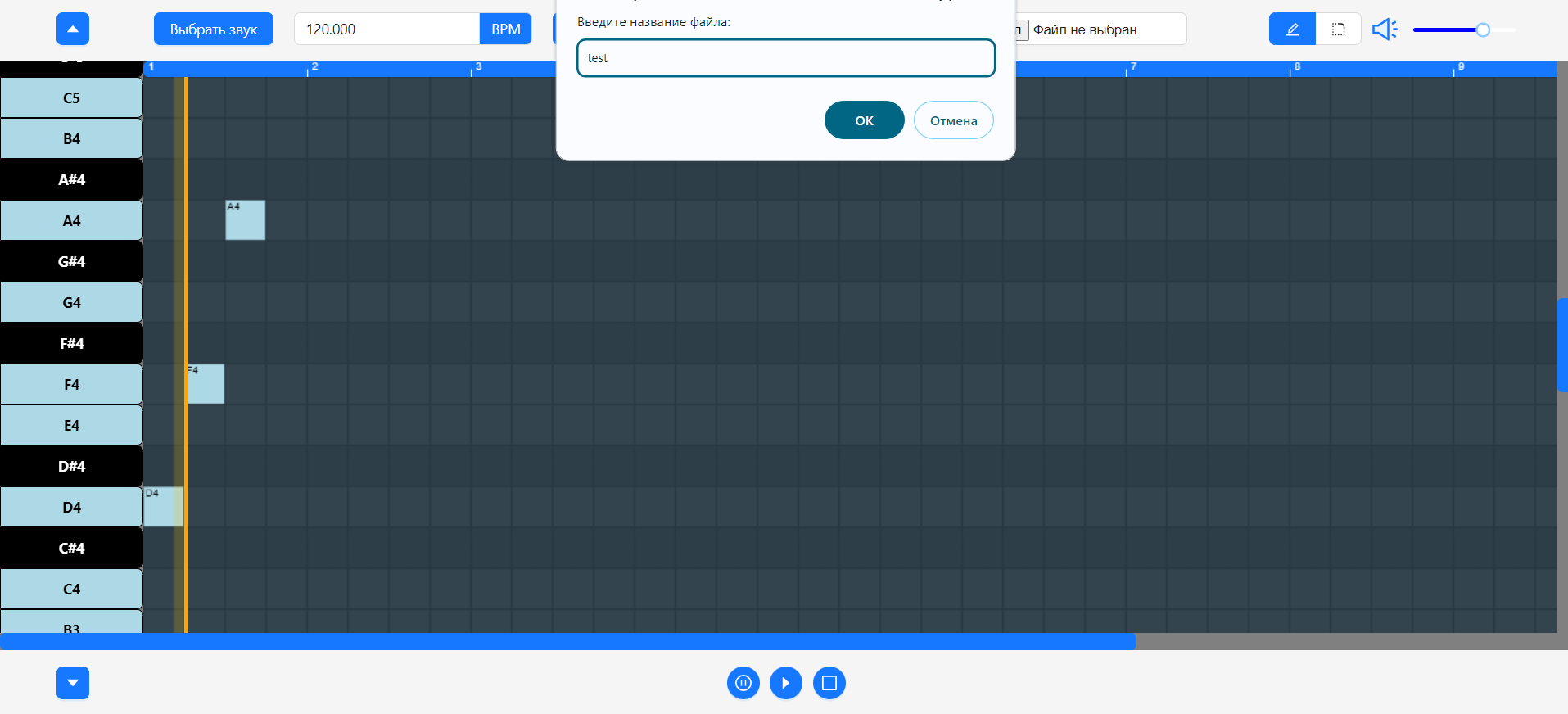


Рисунок 10.14.1 – Сообщение о том, что нужно ввести название файла

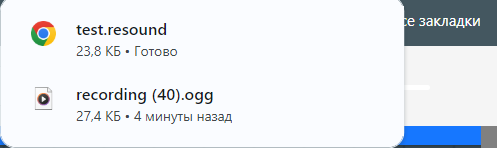


Рисунок 10.14.2 – Успешное сохранение файла шаблона

Таблица 10.15 – Тестирование загрузки шаблона

|  |  |
| --- | --- |
| Наименование | Описание |
| Test Case # | TC\_UI\_14 |
| Приоритет тестирования | Высокий |
| Название тестирования | Сохранение шаблона |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе корректных данных |
| Шаги тестирования | * нажатие на кнопку «Загрузить»; * выбор файла типа .resound. |
| Данные тестирования | Файл шаблона: test.resound |
| Ожидаемый результат | Отображение на сетки нот, которые были записаны в файл |
| Фактический результат | Отображение на сетки нот, которые были записаны в файл |
| Предпосылки | Запуск программы |
| Постусловия | Открыта страница шаблона |
| Статус *(Pass/Fail)* | Pass |
| Комментарии | Работает |

Результаты тестирования загрузки шаблона представлены на рисунках 10.15.1 и 10.15.2

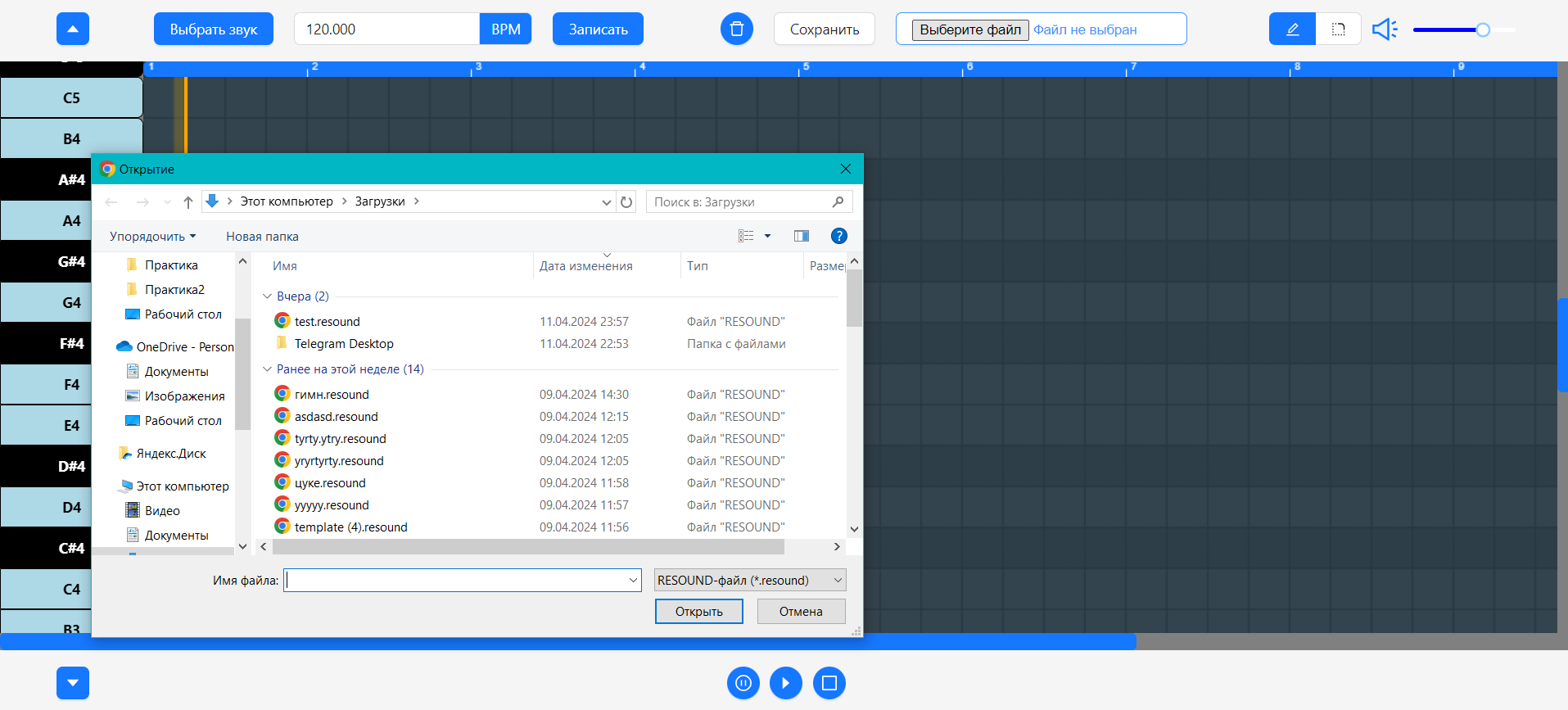


Рисунок 10.15.1 – Выбор файла шаблона

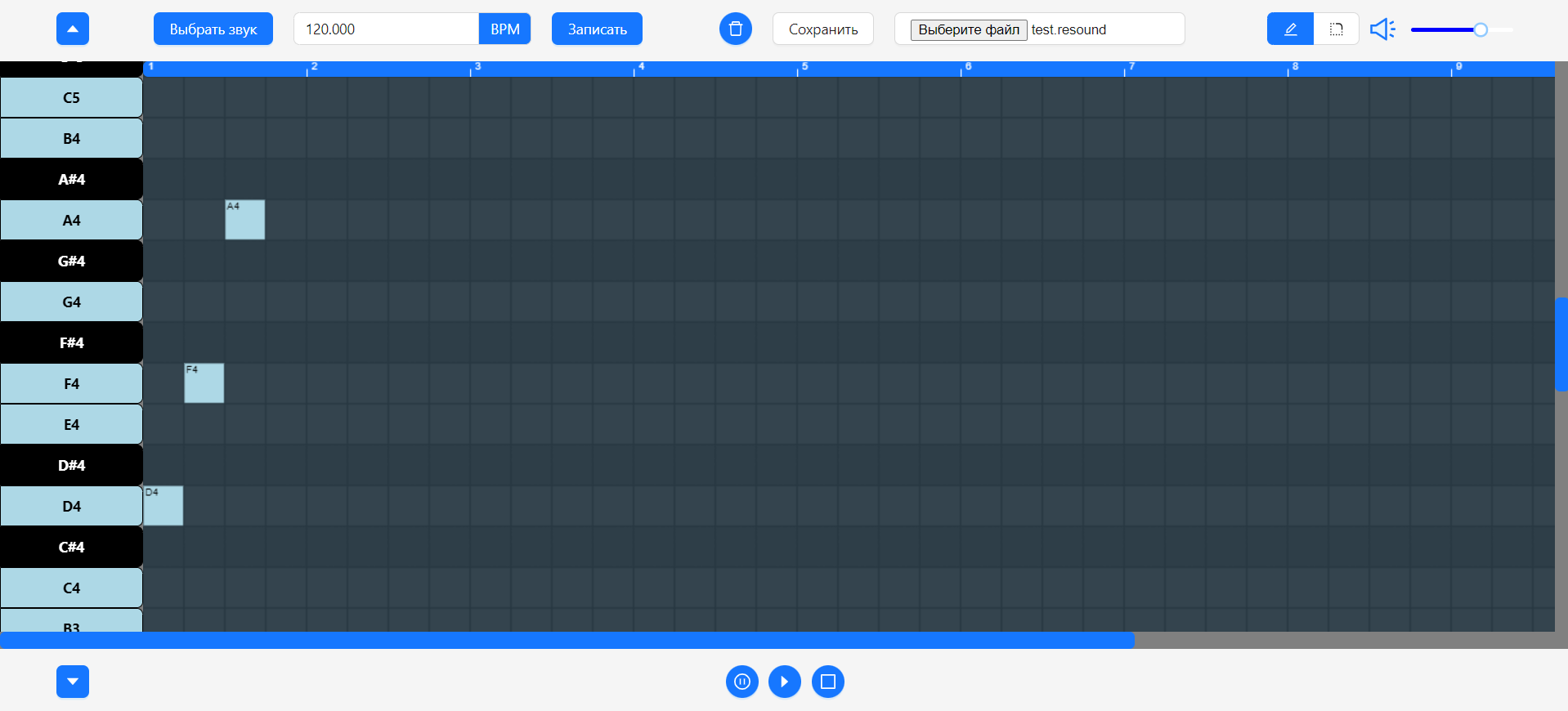


Рисунок 10.15.2 – Успешная загрузка шаблона

11 Техническая документация

11.1 Руководство пользователя

11.1.1 Введение

Цель руководства заключается в расписании подробной инструкции и информации для пользователя, чтобы он смог самостоятельно пользоваться программой и правильно ее эксплуатировал.

11.1.2 Назначение и условие применения

Веб-сайт «ReSound» предназначена для пользователей, имеющих как минимум первоначальные навыки работы с графической операционной системой, на которой будет запускаться данная программа.

11.1.3 Подготовка к работе

Для того чтобы пользователь мог начать пользоваться сайтом, ему необходимо перейти по ссылке в браузере, после чего откроется страница профиля. Эта страница показана на рисунке 11.1.3.1.

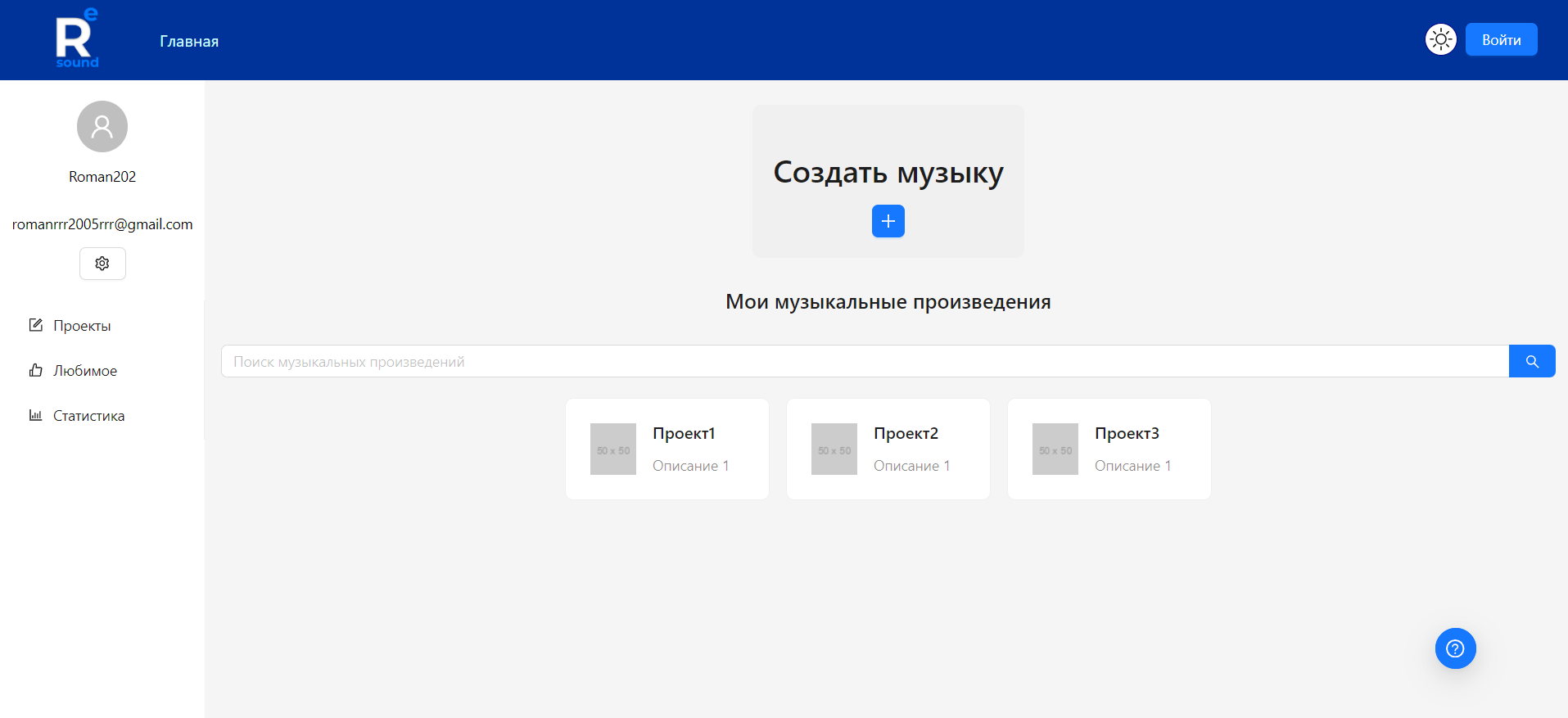


Рисунок 11.1.3.1 – Страница профиля пользователя

11.1.4 Описание операций

Чтобы создать новый проект необходимо нажать на кнопку «+», после чего откроется модальное окно создание нового проекта. Модальное окно представлено на рисунке 11.1.4.1.

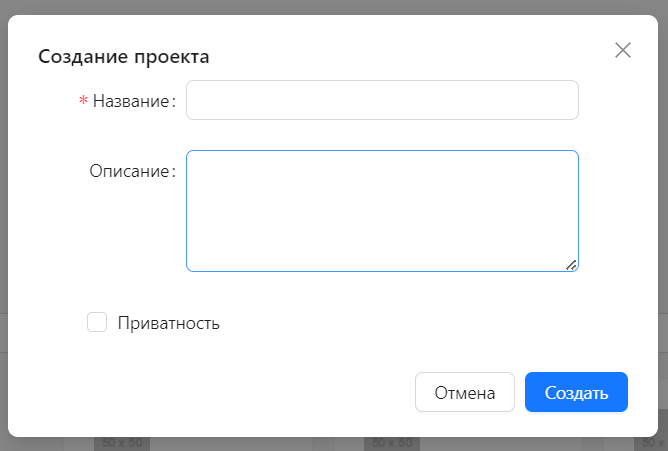


Рисунок 11.1.4.1 – Модальное окно создание проекта

После того как откроется модальное окно, необходимо заполнить поля, после чего нажать на кнопку «Создать». После этого откроется главная страница нового проекта, она представлена на рисунке 11.1.4.2.

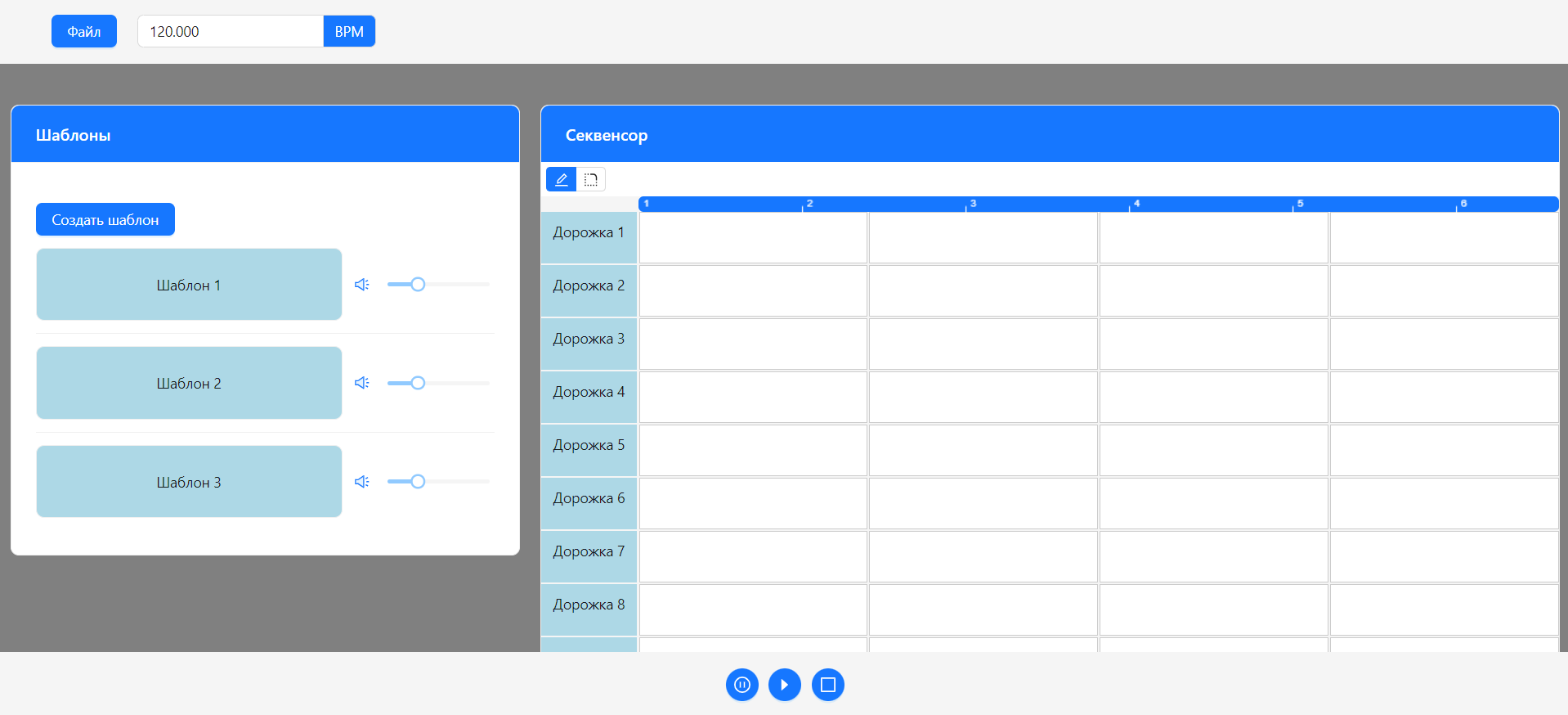


Рисунок 11.1.4.2 – Главная страница проекта

На главной странице проекта можно создавать новые шаблоны при помощи кнопки «Создать шаблон», а также переходить на страницу шаблона при помощи клика по нему. После нажатие на шаблон пользователь перейдет на страницу шаблона, которая представлена на рисунке 11.1.4.3.

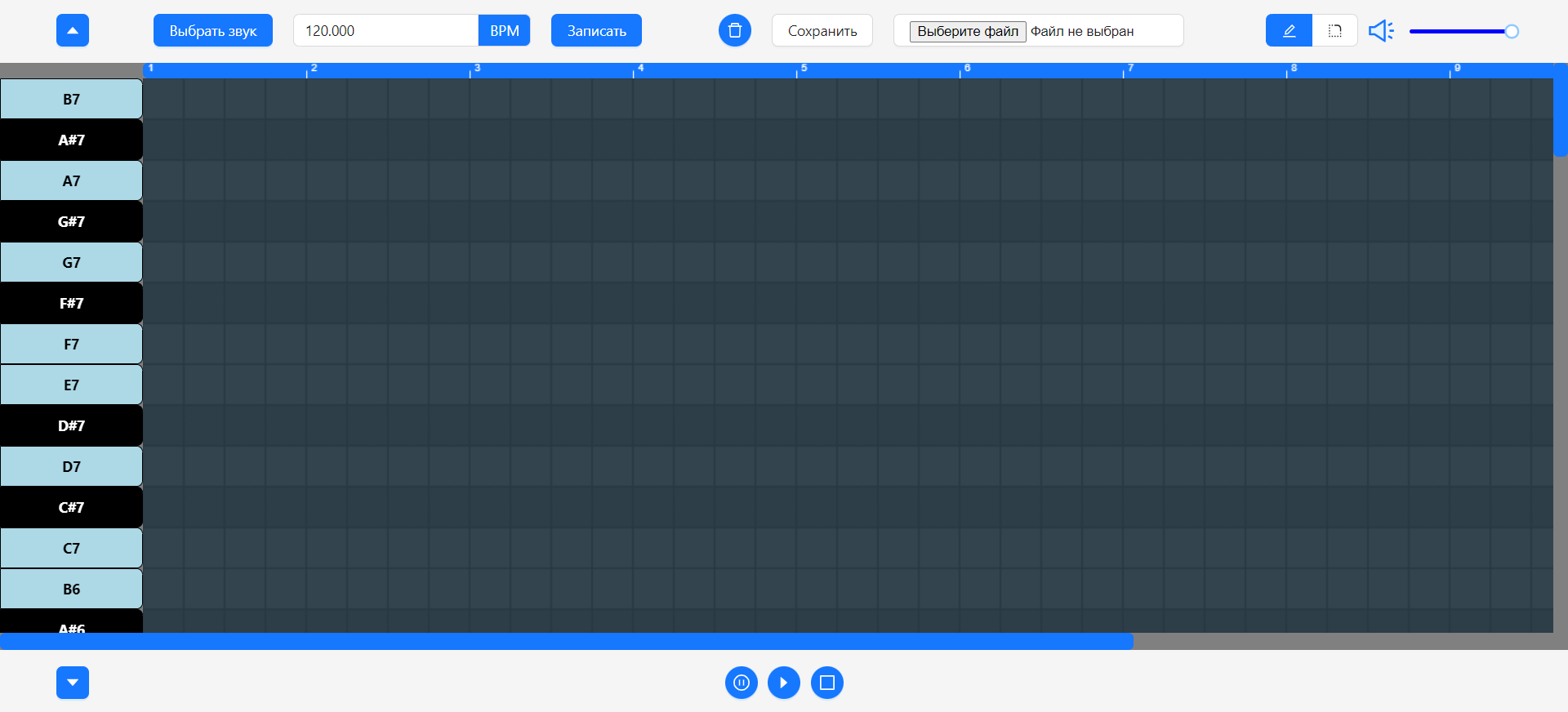


Рисунок 11.1.4.3 – Страница шаблона

На странице шаблона можно размещать ноты на сетки, а после чего их воспроизводить с помощью кнопки «Пуск». Также воспроизведение можно остановить и поставить на паузу с помощью кнопок «Стоп» и «Пауза». Чтобы быстро перемещаться по мелодии можно использовать временную шкалу сверху. Для этого необходимо нажать на нее в том месте, откуда вы хотите чтобы началось воспроизведение музыки. Пример страницы с размещенными нотами представлен на рисунке 11.1.4.5.

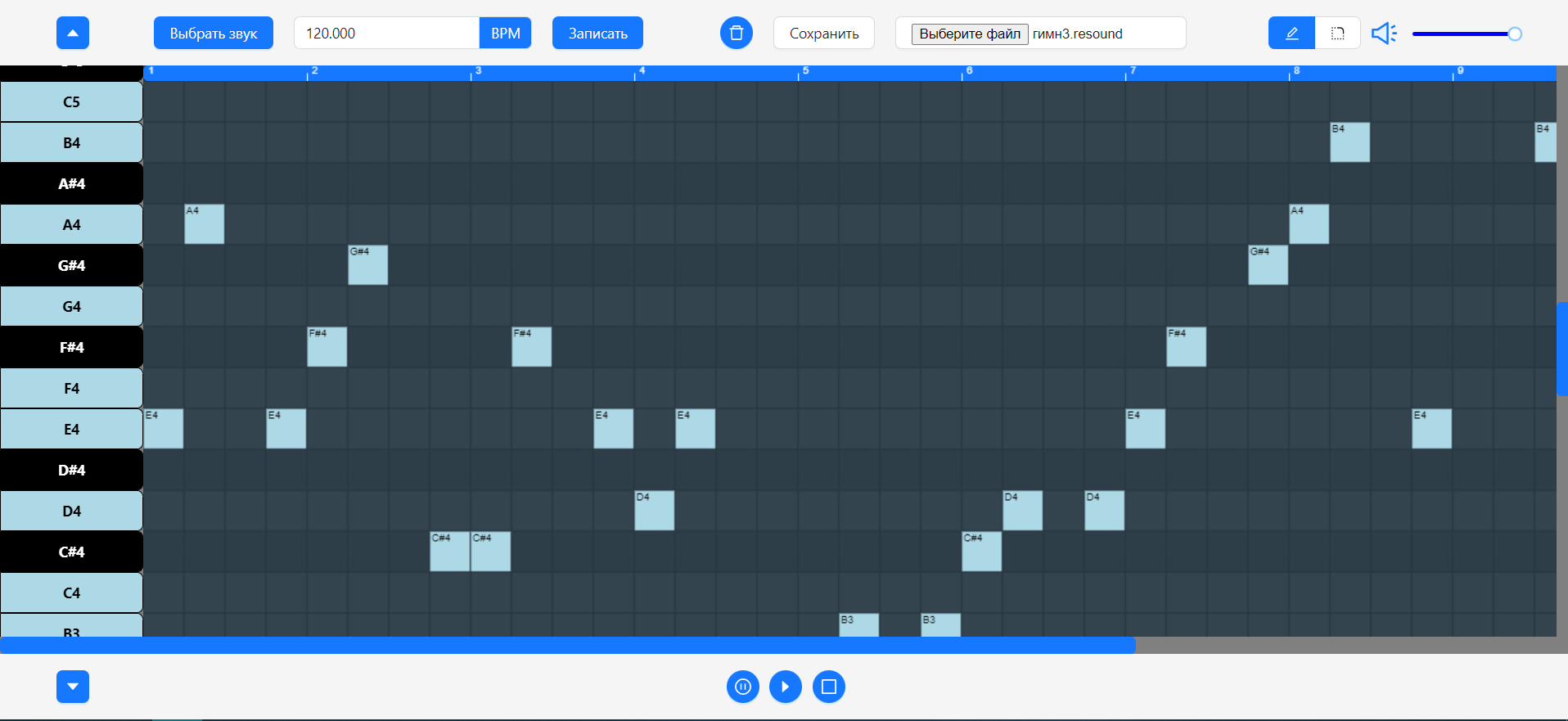


Рисунок 11.1.4.5 – Страницы шаблона с размещенными нотами

С помощью поля BPM можно изменить скорость воспроизведения музыки. Кнопка «Очистить удаляет все ноты с сетки». Изменить громкость можно при помощи слайдера. Для того чтобы добавить новые столбцы у сетки нот необходимо пролистать до края сетки и нажать на кнопку «+», после чего добавить еще 48 столбцов.

Для того чтобы сменить звук проигрывателя необходимо нажать на кнопку «Выбрать звук», после чего откроется модальное окно, где можно выбрать звук. Модальное окно выбора звука представлено на рисунке 11.1.4.6.

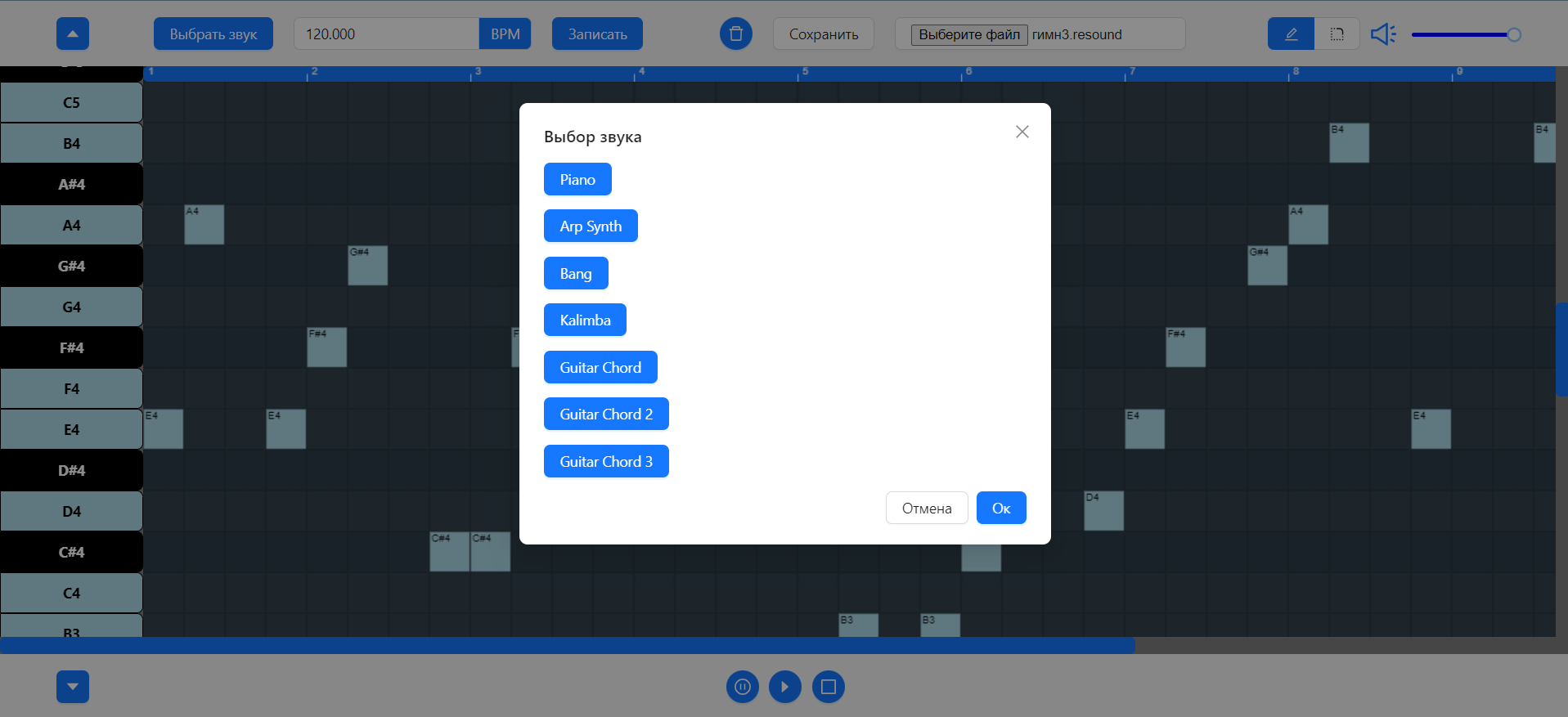


Рисунок 11.1.4.6 – Модальное окно выбора звука

Пользователь может записать свою мелодии при помощи кнопки «Записать». После нажатия на эту кнопку будет воспроизводиться вся мелодия, а после она сохранится.

Чтобы пользователь мог продолжить свою работу спустя какое-то время, он может сохранить свой шаблон при помощи кнопки «Сохранить». После того, как пользователь нажмет на эту кнопку сверху появится сообщение что нужно дать название файлу и поле ввода. Как только пользователь введет название файла и нажмет «Ок», шаблон сохраниться.

А чтобы продолжить работу необходимо воспользоваться кнопкой «Выбрать файл». После её нажатие пользователю будет необходимо выбрать файл, который он сохранял ранее. После выбора файла на сетке нот появятся ноты, которые были в сохраненном файле шаблона.

11.1.5 Аварийные ситуации

При недоступности веб-сайта необходимо:

* убедиться в том, что соединение компьютера с Интернетом установлено;
* убедиться в правильности ввода доменной части адреса сайта.

11.1.6 Рекомендации по освоению

Изучите интерфейс сайта. Познакомьтесь с основными функциями, такими как размещение нот и сохранение шаблонов.

Начинайте с простых проектов. Не пытайтесь сразу создавать сложные композиции. Начните с простых проектов, которые помогут вам освоить основные принципы создания музыки. Вы можете использовать готовые шаблоны или лупы, чтобы сэкономить время и силы. Постепенно усложняйте свои проекты, добавляя новые инструменты.

Не бойтесь делиться своей музыкой с другими людьми и получать обратную связь.

11.2 Руководство программиста

11.2.1 Назначение и условия применения программ

Онлайн секвенсор «ReSound» предназначен для создания музыкальных произведений. Обеспечивает возможность создания, редактирования и публикации музыкальных произведений. Предоставляет пользователям инструмент для начала музыкальной карьеры.

Условия применения:

* поддерживаемые операционные системы: Windows, Linux, Mac;
* поддерживаемые браузеры: все современные браузеры;
* языки программирования: ASP .NET CORE C# (backend), React TypeScript (frontend);
* cреды разработки: Microsoft Visual Studio 2022 Community, Visual Sudio Code;
* СУБД: PostgreSQL;
* среда для визуального проектирования баз данных: pgAdmin 4 v7.

11.2.2 Характеристика программы

Функциональные возможности:

* создание музыкальных произведений с указанием названия, описания и приватности;
* редактирование основных сведений музыкальных произведений;
* регистрация и авторизация пользователей;
* работа с содержимым музыкального произведения (шаблон, скорость, громкость, главная дорожка);
* создание и редактирование шаблонов (ноты, эффекты, громкость, звук);
* редактирование нот в шаблонах (название, расположение, громкость, длина);
* редактирование главной дорожки (добавление, удаление, перемещение шаблонов);
* оставление комментариев и оценок к опубликованным музыкальным произведениям;
* просмотр опубликованных музыкальных произведений пользователей (поиск, сортировка, комментарии, оценки);
* просмотр профиля пользователей (музыкальные произведения, поиск, данные пользователя);
* просмотр собственного профиля (музыкальные произведения, поиск, статистика, данные пользователя);
* отслеживание статистики своих произведений (количество прослушиваний, популярные произведения).

Нефункциональные требования:

* кроссплатформенность;
* безопасность;
* удобство использования;
* производительность.

11.2.3 Обращение к программе

Для запуска веб сайта необходимо перейти в папку с React проектом и ввести в терминал команду npm start.

11.2.4 Входные и выходные данные

Входной информацией является:

* аудио-семплы: Звуковые файлы, которые пользователь может загружать в секвенсор для использования в создании музыкальных композиций;
* шаблоны: Готовые музыкальные шаблоны или лупы, которые пользователь может использовать в своих композициях;
* настройки проекта: Общая скорость воспроизведения, темп, количество тактов и другие параметры проекта;
* список пользователей;
* опубликованные произведения
* комментарии пользователей под опубликованными произведениями
* оценки пользователей под опубликованными произведениями

Выходной информацией является:

* аудио-файлы: Созданные пользователем музыкальные композиции, которые могут быть экспортированы в формате WAV, MP3 или других аудио-форматах;
* проектные файлы: Файлы проекта, содержащие информацию о композиции, расположении треков, настройках эффектов и других параметрах для возможности дальнейшего редактирования;
* ссылка на онлайн-проигрыватель: Пользователь может поделиться своей композицией, отправив другим уникальную ссылку на онлайн-проигрыватель, где можно прослушать результат;
* список топ опубликованных произведений.

11.2.5 Сообщения

Информационные сообщения:

* сообщения о успешной регистрации, авторизации, создании, редактировании и удалении данных;
* сообщения о поиске, сортировке и фильтрации данных.

Предупреждающие сообщения:

* сообщения о некорректных данных (логин, пароль, электронная почта);
* сообщения о дублировании данных (логин, название музыкального произведения);
* сообщения о превышении лимитов (длина комментария, количество символов в названии).

Ошибки:

* сообщения о некорректных данных (недопустимые символы, пустые поля);
* сообщения о проблемах с соединением с базой данных;
* сообщения о внутренних ошибках сервера.

ЗАКЛЮЧЕНИЕ

В ходе практики в АО «Башкирский регистр социальных карт» была разработана клиентская часть веб-приложения, состоящего из фронтенда на React. Приложение представляет собой платформу для создания музыкальных произведений, разработанную с использованием современных технологий и с учетом актуальных трендов в веб-дизайне.

Фронтенд на React был разработан с использованием компонентного подхода, что позволило создать модульный и переиспользуемый код. Интерфейс приложения был разработан с учетом современных трендов в веб-дизайне, обеспечивая удобство использования и отзывчивость.

Реализована основная функциональность для создания музыкальных произведений. Пользователи могут создавать мелодии, добавлять различные инструменты, изменять темп и ритм, а также сохранять и делиться созданными произведениями. Приложение предоставляет широкий выбор инструментов и звуков, что позволяет создавать разнообразные музыкальные произведения. Интуитивно понятный интерфейс и гибкие настройки позволяют пользователям с разным уровнем музыкальных навыков создавать интересные и уникальные композиции.

В ходе практики были получены следующие знания и навыки:

* разработка веб-приложений на React. Получены знания и навыки по созданию современных веб-приложений с использованием библиотеки React;
* понимание принципов создания музыкальных произведений. Получены базовые знания о теории музыки, что позволяет создавать более интересные и сложные музыкальные произведения.

Практика в АО «Башкирский регистр социальных карт» позволила получить ценный опыт в разработке веб-приложений на React, а также в разработке платформы для создания музыкальных произведений. Полученные знания и навыки будут полезны для дальнейшей работы в сфере веб-разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вандеркам, Д. Эффективный TypeScript: 62 способа улучшить код. — СПб.: Питер, 2021. — 288 с.: ил. — (Серия «Бестселлеры O’Reilly»).
2. Васильев, А.Н. Программирование на C# для начинающих. Основные сведения: / А.Н. Васильев. — Москва: Бомбора, 2023. — 592 с. — ISBN 978-5-04-092519-3. — Текст: непосредственный.
3. Моисеев, А. TypeScript быстро: / А. Моисеев, Я. Файн. — Санкт-Петербург: Питер, 2021. — 528 с. — ISBN 978-5-4461-1725-3. — Текст: непосредственный.
4. Прайс, М. C# 10 и .NET 6. Современная кросс-платформенная разработка: / М. Прайс. — Санкт-Петербург: Питер, 2023. — 848 с. — ISBN 978-5-4461-2249-3. — Текст: непосредственный.
5. Ришкуция, В. Программируй & типизируй: / В. Ришкуция. — Санкт-Петербург: Питер, 2021. — 352 с. — ISBN 978-5-4461-1692-8. — Текст: непосредственный.
6. Стефанов, С. React. Быстрый старт: / С. Стефанов. — 2-е изд. — Санкт-Петербург: Питер, 2023. — 304 с. — ISBN 978-5-4461-2115-1. — Текст: непосредственный.
7. Флэнаган, Д. JavaScript. Полное руководство: / Д. Флэнаган. — 7-е изд. — Москва: Диалектика Вильямс, 2021. — 720 с. — ISBN 978-5-907203-79-2. — Текст: непосредственный.
8. Полное руководство по языку программирования C# 12 и платформе .NET 8 — metanit.com — URL: https://metanit.com/sharp/tutorial/ — Текст: электронный
9. Руководство по TypeScript — metanit.com — URL: https://metanit.com/web/typescript/ — Текст: электронный
10. Руководство по ASP.NET Core 8 — metanit.com — URL: https://metanit.com/sharp/aspnet6/ — Текст: электронный
11. Руководство по JavaScript — metanit.com — URL: https://metanit.com/web/javascript/ — Текст: электронный
12. Руководство по HTML 5 и CSS 3 — metanit.com — URL: https://metanit.com/web/html5/ — Текст: электронный
13. About MobX — mobx.js.org — URL: https://mobx.js.org/ — Текст: электронный
14. Components Overview — ant.design — URL: https://ant.design/components/overview — Текст: электронный
15. Docker Docs — docs.docker.com — URL: https://docs.docker.com/ — Текст: электронный
16. PostgreSQL 16.1 Documentation — postgresql.org — URL: https://www.postgresql.org/docs/current/ — Текст: электронный
17. React Документация — reactjs.org — URL: https://ru.legacy.reactjs.org/docs/getting-started.html — Текст: электронный
18. Tone.js — tonejs.github.io— URL: https://tonejs.github.io/ — Текст: электронный
19. TypeScript Documentation — typescriptlang.org — URL: https://www.typescriptlang.org/docs/ — Текст: электронный
20. Using TypeScript — react.dev — URL: https://react.dev/learn/typescript — Текст: электронный