МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

ФИЛИАЛ «МИНСКИЙ РАДИОТЕХНИЧЕСКИЙ

КОЛЛЕДЖ»

**ОТЧЕТ**

**по преддипломной практике**

Дрозд Максим Алексеевич

(фамилия, имя, отчество)

Специальность   2-40-0101 «Программное обеспечение информационных технологий»

(код и название специальности)

Отделение Компьютерных технологий

|  |  |
| --- | --- |
| Выполнил учащийся гр. 62493 | Дрозд М.А. |
| Руководитель практики от колледжа | Бельчик М.А |
| Руководитель практики от предприятия | Толстиков А.А. |

МИНСК 2020

Содержание

[Введение 3](#_Toc40060187)

[1 Календарный план-график 4](#_Toc40060188)

[2 Постановка задачи 6](#_Toc40060189)

[2.1 Описание предметной области 6](#_Toc40060190)

[2.2 Обзор существующих аналогов 7](#_Toc40060191)

[2.3 Функциональное назначение 10](#_Toc40060192)

[3 Программная реализация 11](#_Toc40060193)

[3.1 Физическая структура 11](#_Toc40060194)

[3.2 Описание разработанных модулей 13](#_Toc40060195)

[4 Тестирование 20](#_Toc40060196)

[5 Применение 30](#_Toc40060197)

[5.1 Описание процесса установки и запуска приложения 30](#_Toc40060198)

[5.2 Руководство пользователя 30](#_Toc40060199)

[Заключение 39](#_Toc40060200)

[Список использованных источников 40](#_Toc40060201)

[Приложение А 41](#_Toc40060202)

## Введение

Преддипломная практика является важным этапом в процессе подготовки специалиста к самостоятельной производственной деятельности. Она является видом учебно-вспомогательного процесса, в ходе которого закрепляется теоретические знания на производстве.

Основным содержанием практики является выполнение практических учебных, творческих заданий, соответствующих характеру будущей профессиональной деятельности обучающихся.

Цель преддипломной практики: закрепление теоретических знаний и получение навыков их практического применения.

Задачи преддипломной практики:

* развитие профессионального мышления, повышение уровня квалификации по профессии;
* изучение технологии и организации производства;
* приобретение практического опыта, навыков по организаторской деятельности, способствующих формированию профессиональной компетентности будущего специалиста;
* формирование профессиональных знаний и навыков работы в коллективе.

Заданием на преддипломной практику является разработка, сопровождение и использование программного обеспечения для решения производственных задач в ООО «ЯндексБел» и индивидуальное задание на тему: «Мобильное приложение «Расписание учебного заведения»».

В период прохождения преддипломной практики учащийся может привлекаться к различным видам работ, соответствующим профилю образования и программе практики, участвовать в проектно-изыскательской, опытной, творческой работе, получить квалификационный разряд (класс, категорию) по профессии или получить дополнительную профессию в соответствии со специальностью.

# Календарный план-график

Календарный план-график прохождения преддипломной практики представляет собой ежедневное планирование этапов прохождение практики и состоит из даты и содержания этапа. Основными этапами преддипломной практики являются ознакомление с технологическим процессом обработки информации в IT-отделах, сопровождение производственных задач, разработку программных средств индивидуального задания.

Таблица 1.1 – Календарный план-график

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Дата | Выполняемая работа | Подпись руководителя |
| 1. | 02.04.2020 | Определение целей и задач практики. Инструктаж по охране труда. Выдача индивидуального задания |  |
| 2. | 03.04.2020 | Организационно-функциональная структура предприятия. Используемое ПО. Должностные обязанности техника-программиста |  |
| 3. | 04.04.2020 | Технологический процесс выполнения производственных задач. Анализ индивидуального задания |  |
| 4. | 06.04.2020 | Особенности организации при решении производственных задач |  |
| 5. | 07.04.2020 | Выбор и обоснование инструментов разработки |  |
| 6. | 08.04.2020 | Инсталляция программного обеспечения |  |
| 7. | 09.04.2020 | Настройка программного обеспечения |  |
| 8. | 10.04.2020 | Сопровождение мобильного приложения Яндекс.Толока. |  |
| 9. | 11.04.2020 | Сопровождение мобильного приложения Яндекс.Толока. |  |
| 10. | 13.04.2020 | Сопровождение мобильного приложения Яндекс.Толока. |  |
| 11. | 14.04.2020 | Сопровождение мобильного приложения Яндекс.Толока. |  |
| 12. | 15.04.2020 | Сопровождение мобильного приложения Яндекс.Толока. |  |
| 13. | 16.04.2020 | Кодирование, отладка мобильного приложения Яндекс.Толока. |  |
| Продолжение таблицы 1.1 | | | |
| № п/п | Дата | Выполняемая работа | Подпись руководителя |
| 14. | 17.04.2020 | Кодирование, отладка мобильного приложения Яндекс.Толока. |  |
| 15. | 18.04.2020 | Тестирование мобильного приложения Яндекс.Толока. |  |
| 16. | 20.04.2020 | Тестирование. Функционально-стоимостной анализ. |  |
| 17. | 21.04.2020 | Опытная эксплуатация и анализ качества мобильного приложения Яндекс.Толока. |  |
| 17. | 22.04.2020 | Разработка разделов отчета. |  |
| 18. | 23.04.2020 | Анализ и обобщение материалов по практике. Оформление отчета. |  |
| 19. | 24.04.2020 | Анализ и обобщение материалов по практике. Оформление отчета. |  |
| 20. | 29.04.2020 | Подведение итогов практики. |  |

Ознакомлен

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

от предприятия (организации) Подпись Расшифровка подписи

# Постановка задачи

## Описание предметной области

Учебное заведение – общее название школ, гимназий, техникумов, колледжей, университетов и других мест получения знаний.

Расписание — вид календаря, для которого указана информация о предстоящих (планируемых или потом произошедших) событиях. Оформляется обычно в виде таблицы.

У всех учебных заведений есть какое-либо расписание, будь то расписание занятий или расписание работы столовой. Расписание занятий составляется из пар или уроков, в зависимости от учебного заведения.

Урок – временной промежуток длительностью 45 минут – 1 академический час (длительность может изменяться по различным причинам, например, сокращенные уроки в честь праздничных дней), во время которого учащиеся должны овладеть изучаемым материалом. На уроке, помимо учащихся так же присутствует учитель. Его основная роль – воспитание учащихся и помощь в освоении материала.

Учащийся – тот, кто учится в учебном заведении.

Пара – урок, длительность которого увеличение в 2 раза, т.е. или 1 час 35 минут (5 минут выделены на перерыв межу первыми и вторыми 45 минутами) – 2 академических часа. Пары ведут преподаватели.

Слова преподаватель и учитель – синонимы, но исторически сложилось, что для различных учебных заведений принято использовать различные термины. Так, например, в вузах университетах и колледжах пары ведут преподаватели, а в школах и гимназиях уроки ведут учителя.

Современные технологии проникли во все сферы человеческой жизни, образование не является исключением. Сейчас трудно представить жизнь современного человека без таких устройств как смартфон, планшет или таких технологий как интернет, мессенджеры, электронная почта и социальные сети.

На фоне всех этих технических инноваций, прочно закрепившихся в нашей жизни, традиционное расписание, представленное в форме доски, кажется недостаточным.

Каждая современная организация зачастую имеет своё информационное пространство, в которое входит сайт образовательного учреждения, электронно-образовательный портал, электронно-библиотечная система, внутренняя система взаимодействия сотрудников (корпоративный портал, мессенджер, электронная почта). На фоне этого невозможность узнать своё расписание с применением современных гаджетов вызывает достаточно сильный дискомфорт.

Создание мобильного приложения для просмотра расписания позволило бы любому студенту, находясь в любом месте, узнать какие занятия у него будут, в какое время и в какой именно аудитории, помогут найти преподавателя в случае необходимости. Так же появляется возможность мгновенно отслеживать изменения в графике учебного процесса.

* 1. Обзор существующих аналогов

Приложений для расписания бывает два вида.

Первый - общего назначения, когда пользователь сам создает расписание и обновляет его. Оно может составляться для любой предметной области.

Второй – создается специально для конкретной цели. Само расписание скачивается из внешних источников, и пользователь не может его редактировать, только просматривать.

Поскольку мое программное средство относится ко 2 категории, рассматривать аналоги мы будем также 2 категории.

Первый аналог – приложение «Расписание БГУИР», рабочее окно которого изображено на рисунке 2.1. С его помощью можно просматривать расписание сразу для нескольких групп, читать информацию о преподавателях, менять шрифт и язык.

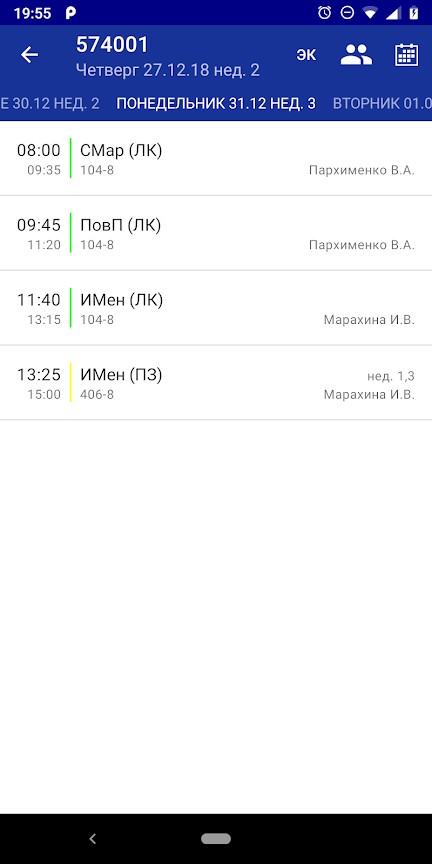


Рисунок 2.1 – Мобильное приложение «Расписание БГУИР»

Из недостатков этого приложения можно отметить неудобный интерфейс. Если вам необходимо посмотреть расписание на пятницу, придется делать четыре свайпа влево. Свайп влево – достаточно неудобный жест, особенно если пользоваться телефоном одной рукой. А тут их целых четыре. Намного удобнее было бы сделать навигацию по дням недели списком, или поместить все расписание в один столбец, так, чтобы пролистать вниз, к пятнице, достаточно было бы сделать всего один сильный свайп вверх.

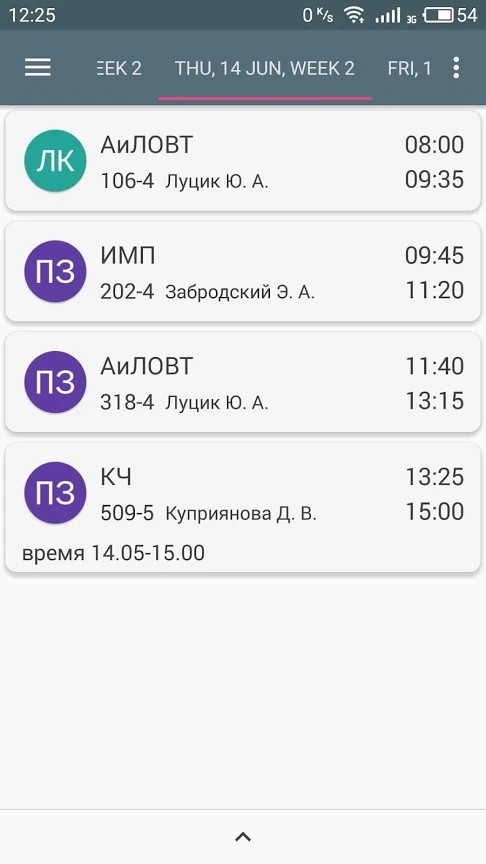


Рисунок 2.2 – Мобильное приложение «BSUIR Schedule»

Следующий аналог – мобильное приложение «BSUIR Schedule», рабочее окно которого изображено на рисунке 2.2. Оно обладает намного более широким функционалом. Помимо чтения расписания для нескольких групп, мобильное приложение «BSUIR Schedule» позволяет изменять локальную версию расписания, просматривать место нахождения преподавателей, имеет виджет на рабочий стол, и самое главное, темную тему.

Сразу же можно отметить такой же недостаток, как и у приложения «Расписание БГУИР»: неудобный интерфейс. Затем если присмотреться, то большинство функциональных кнопок спрятались за кнопку меню, что увеличивает время доступа к функциям этих кнопок. Это еще один недостаток интерфейса.

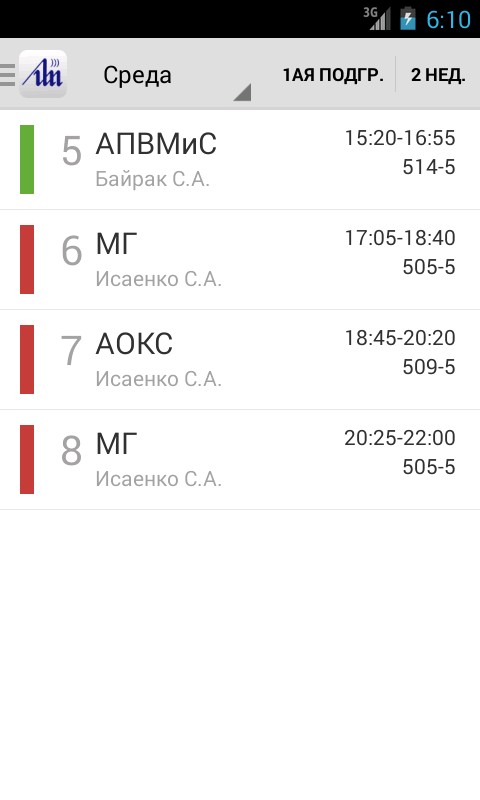


Рисунок 2.3 – Мобильное приложение «БГУИР Расписание»

И последний аналог – приложение «БГУИР Расписание», рабочее окно которого изображено на рисунке 2.3. Функционал этого приложения похож на функционал «BSUIR Schedule»: он позволяет изменять локальную версию расписания, имеет виджет на рабочий стол, темную тему, но отсутствует возможность посмотреть, где находится преподаватель.

В отличие от своих предшественников, это приложение добавило выпадающее меню для выбора для недели, что является достаточно неплохим решением по сравнению с решениями предыдущих приложений. У приложения минималистичный дизайн, все нужные кнопки находятся под рукой, нет ничего лишнего, или чего-нибудь, что мешало бы просмотру расписания.

На основе преимуществ и недостатков уже существующих решений можно понять, как сделать свое приложение лучше, на что следует обратить внимание, а от чего следует избавиться, чтобы получить качественный продукт.

## Функциональное назначение

У каждого учебного заведения есть расписание. Однако не каждое учебное заведение может его автоматизировать. Автоматизация процесса создания и распространения расписания позволит существенно облегчить и ускорить работу учебных заведений. Облегчение внедрения процессов автоматизации процессов создания и распространения расписания является целью написания программного обеспечения.

Задачами разрабатываемого программного обеспечения являются:

* добавление, редактирование, удаление преподавателей;
* добавление, редактирование, удаление аудиторий;
* добавление, редактирование, удаление групп;
* добавление, редактирование, удаление предметов;
* создание расписания на основе существующих преподавателей, аудиторий, групп и предметов;
* получение и отображение расписания на мобильном устройстве;
* получение и отображение информации о преподавателях на мобильном устройстве;
* поиск по преподавателям на мобильном устройстве.

Таким образом разработанное программное средство позволит любому учебному заведению без особых проблем внедрить процессы автоматизации создания и распространения программного обеспечения.

1. Программная реализация

## Физическая структура

Разработка приложения состоит из 3 частей:

* разработка серверной части приложения;
* разработка админ-панели;
* разработка клиентского мобильного приложения.

Все вышеупомянутые части приложения должны между собой как-то взаимодействовать. Для этого была выбрана классическая клиент-серверная архитектура. На рисунке 3.1 представлена схема работы разработанной архитектуры.

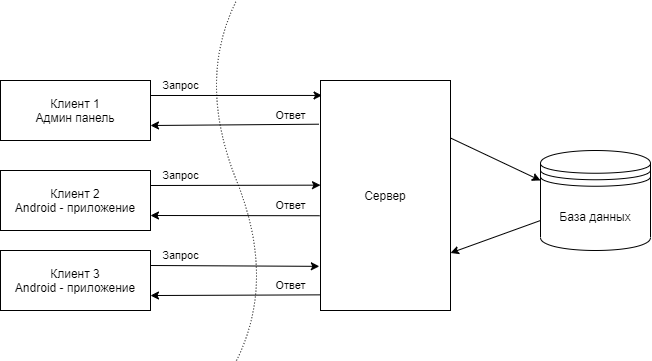


Рисунок 3.1 – Схема клиент-серверной архитектуры

Самая главная часть приложения – сервер. Только через него можно добавлять, изменять и читать расписания.

Общение с сервером происходит по протоколу HTTP. Для получения данных особых разрешений или авторизации не надо. Однако для запросов, которые подразумевают редактирование, добавление или удаление данных, необходимо быть авторизованным.

При обращении к серверу, сервер делает запрос в базу данных, представляет данные в стандартизированном виде и отправляет их в ответ на запрос.

К серверу будут обращаться 2 вида клиентов – админ-панель и Android-приложения. Только админ панель будет располагать возможностями для редактирования расписания, следовательно, в админ-панели авторизация обязательна. Для получения данных авторизация не нужна, следовательно, в Android-предложении авторизации нет и им может пользоваться любой желающий без ограничений.

Сначала был разработан веб-сервер. Веб сервер содержит следующие папки:

* common – папка, содержащая классы, логика которых используется в других классах;
* login – папка, содержащая логику авторизации;
* response – папка, содержащая логику обработки ошибок;
* static – папка, содержащая логику отправки статических html страниц;
* utils – папка с утилитарными классами.

Папки в остальных папках group, lesson, office, teacher, discipline лежат файлы, отвечающие за обработку запросов на получение, добавление, редактирование и удаление групп, пар, аудиторий, преподавателей и предметов соответственно.

Для хранения данных, веб-сервер использует базу данных. Схема спроектированной базы данных представлена на рисунке 3.2.

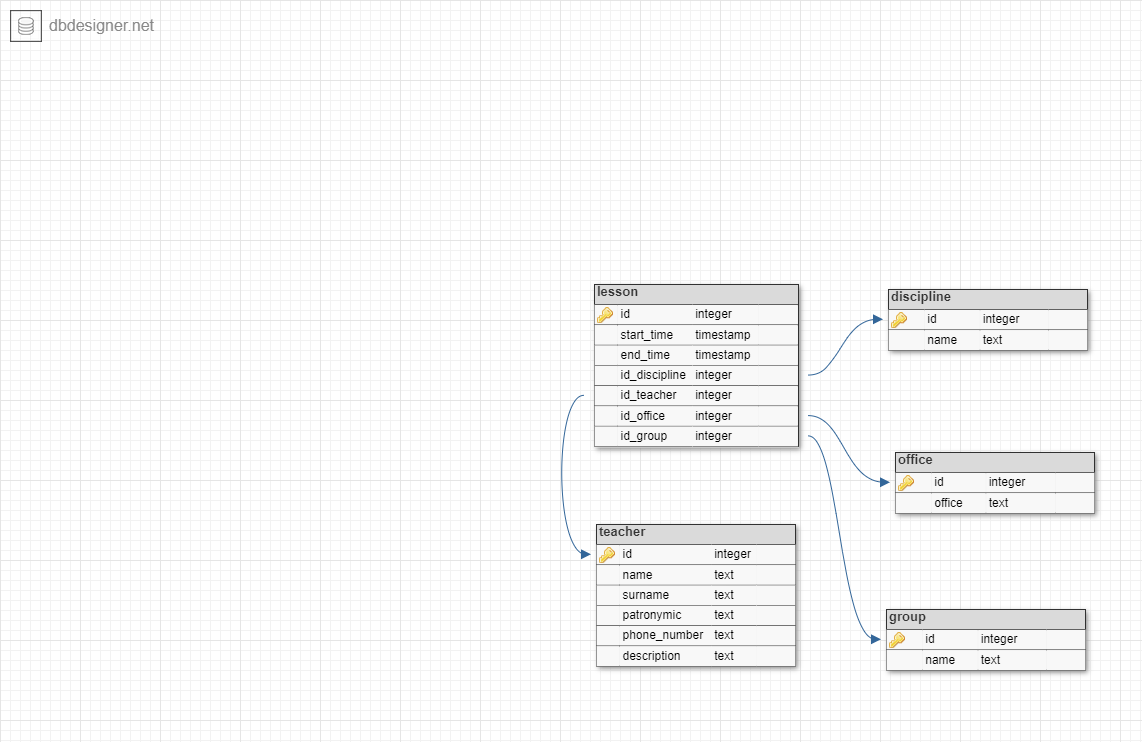


Рисунок 3.2 – Схема базы данных, используемая на сервере

В базе данных имеется основная таблица, которая хранит непосредственно сами расписания и 4 остальныеместо для ния ая часть андроид приложений состоит мых важных из них.лицасписаниярограммного обеспечения.олв о оаооо двлаовгакоаа таблицы, которые необходимы для отсутствия дубликатов данных, в главной таблице используются только ссылки на индексы дополнительных таблиц.

Следующим этапом разработки стало написание админ-панели. Структура админ-панели состоит из 2-х папок: public и src. В папке public содержатся статические файлы, такие как html-файлы, картинки, текстовые файлы. В папке src содержатся сами исполняемые исходные файлы. Именно в этой папке лежат скрипты разработанной админ-панели.

Заключающим и самым важным этапом разработки стало написание клиентского мобильного приложения. Исходные файлы приложения распределены по 2 папкам: res и src. В корне проекта лежит файл manifest.xml в котором указаны основные компоненты приложения, чтобы Android знал, каким функционалом обладает приложение. В папке src содержатся 7 папок – teachers, di, schedule, settings, tools и view, где лежат логика для работы с преподавателями, файлы dependency injection, логика для работы с расписанием, логика настроек, инструментальные классы и кастомные view-элементы соответственно. В папке res лежат файлы, содержащие все ресурсы приложения, а также разметки всех экранов.

Таким образом была разработана система ПО, решающая проблему невозможности посмотреть расписание

## Описание разработанных модулей

Разработка приложения состоит из 3 частей:

* разработка серверной части приложения;
* разработка админ-панели;
* разработка клиентского мобильного приложения.

Сначала был разработан веб-сервер. Его эндпоиты для работы с админ-панелью реализуют собой полноценный REST API.

В текущей реализации REST API имеются следующие возможности: получение всех элементов выборки, добавление элемента в выборку, получение данных и конкретном элементе выборки по номеру, изменение данных конкретного элемента в выборке по номеру, удаление элемента из выборки по номеру.

За получение всех элементов отвечает HTTP метод GET на эндпоинте с названием типа объекта. В теле ответа будет находится JSON массив запрошенных объектов.

За добавление элемента отвечает HTTP метод POST на эндпоинте с названием типа объекта. В тело запроса необходимо поместить JSON объект, который необходимо добавить в выборку. В теле ответа будет находится JSON объект добавленного элемента. Это необходимо на случай, если сервер произведет какую-либо обработку данных и изменит их. Клиент должен об этом знать.

За получение элемента по id отвечает HTTP метод GET на эндпоинте с названием типа объекта и id элемента. В теле ответа будет находится JSON объект найденного элемента с индексом id.

За редактирование элемента по id отвечает HTTP метод PUT на эндпоинте с названием типа объекта и id элемента. В тело запроса необходимо поместить JSON объект, измененного элемента. В теле ответа будет находится JSON объект отредактированного элемента.

За удаление элемента по id отвечает HTTP метод DELTE на эндпоинте с названием типа объекта и id элемента. В теле ответа будет находится JSON объект удаленного элемента.

На рисунке 3.3 находится схематичное представление REST API преподавателей. Для других объектов, таких как предметы, группы, аудитории и пары схема идентичная, за исключением названия объекта в пути к эндпоинту. Так, для получения всех групп, путь будет «/groups», для получения всех аудиторий, путь будет «/offices» и т.д.

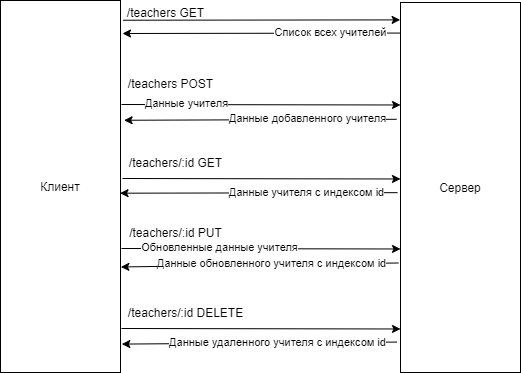


Рисунок 3.3 – Схематичное представление REST API преподавателей

Веб-сервер состоит из файлов, сгруппированных по логике работы, которую они исполняют.

Так, в папке common содержатся следующие файлы CanDeleteUtils.kt – файл с функциями-расширениями, Filter.kt – файл с общей логикой фильтрации, Interactor.kt – файл с абстрактным классом, содержащий логику работы с БД.

В папке login расположился файл LoginController,kt, отвечающий за авторизацию пользователей.

В папке response – файл ResponseController.kt, отвечающий за обработку ошибок.

В папке static – файл StaticController.kt, отвечающий за отправку статических html страниц.

В папке utils содержатся утилитарные классы AddHeader.kt, Enums.kt, GetAllParams.kt и Path.kt, которые используются по всему проекту.

В папке discipline лежит файл Discipline.kt, который выражает собой структуру объекта «предмет», файл DisciplineController.kt, который содержит маппинг эндпоинтов сервера на функции работы с БД и файл DisciplineInteractor.kt, который является реализацией абстрактного класса Interactor и служит для работы с БД.

В папке group лежит файл Group.kt, который выражает собой структуру объекта «группа», файл GroupController.kt, который содержит маппинг эндпоинтов сервера на функции работы с БД и файл GroupInteractor.kt, который является реализацией абстрактного класса Interactor и служит для работы с БД.

В папке lesson лежит файл Lesson.kt, который выражает собой структуру объекта «пара», файл LessonController.kt, который содержит маппинг эндпоинтов сервера на функции работы с БД и файл LessonInteractor.kt, который является реализацией абстрактного класса Interactor и служит для работы с БД.

В папке office лежит файл Office.kt, который выражает собой структуру объекта «аудитория», файл OfficeController.kt, который содержит маппинг эндпоинтов сервера на функции работы с БД и файл OfficeInteractor.kt, который является реализацией абстрактного класса Interactor и служит для работы с БД.

В папке teacher лежит файл Teacher.kt, который выражает собой структуру объекта «преподаватель», файл TeacherController.kt, который содержит маппинг эндпоинтов сервера на функции работы с БД и файл TeacherInteractor.kt, который является реализацией абстрактного класса Interactor и служит для работы с БД.

В корне проекта лежат 2 файла – Main.kt, содержащий логику инициализации веб-сервера и DB.kt, содержащий логику соединения с базой данных.

Следующим этапом разработки стало написание админ-панели.

Файл index.html является главной точкой входа в приложение. Файл favicon.ico представляет собой картинку, отображаемую на вкладке в браузере.

После загрузки html страницы первым делом загружается и запускается файл App.js. Он содержит в себе координирующую логику приложения. Он отвечает за то, что, как и где показывать.

Файл index.css содержит стили страницы.

Файл customBottomToolbar.js содержит кастомный компонент нижней панели на экране редактирования.

Файл customDeleteToolbar.js содержит кастомный компонент контекстного меню.

Файл customTextField.tsx содержит кастомное поля для отображения текста.

Файл deleteButton.tsx содержит компонент кнопки удаления.

Файл deleteButtonEdit.tsx содержит компонент кнопки удаления в контекстном меню.

Файл discipline.js содержит разметку для экранов просмотра, создания, изменения и редактирования предметов.

Файл group.js содержит разметку для экранов просмотра, создания, изменения и редактирования групп.

Файл lesson.js содержит разметку для экранов просмотра, создания, изменения и редактирования пар.

Файл office.js содержит разметку для экранов просмотра, создания, изменения и редактирования аудиторий.

Файл teacher.js содержит разметку для экранов просмотра, создания, изменения и редактирования преподавателей.

Файл url.js хранит в себе ip адрес бэкэнд-сервера.

Заключающим и самым важным этапом разработки стало написание клиентского мобильного приложения.

Все экраны приложения выполнены на архитектуре MVP. MVP расшифровывается как Model-View-Presenter.

В разработанном приложении роль View выполняет Android компонент Fragment. View сообщает Presenter о пользовательских взаимодействиях, таких как нажатия кнопки, ввод текста.

Presenter слушает пользовательские взаимодействия и на их основе решает, как изменить Model. Model – это данные в приложении. Например, номер текущей группы.

Presenter также слушает обновления модели и на их основе решает, как обновить View.

Схема архитектуры MVP представлена на рисунке 3.4.

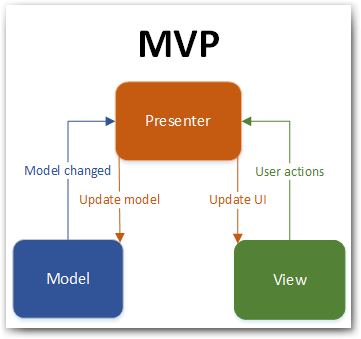


Рисунок 3.4 – Схема архитектуры MVP

В корне проекта лежат файлы MainActivity.kt, MainPresenter.kt, MainView.kt и MainDialogs.kt, которые в совокупности образуют главный экран приложения. MainView представляет из себя интерфейс, который создает контракт, по которому можно изменять визуальные компоненты. Класс MainPresenter.kt содержит логику работы главного экрана и через интерфейс MainView общается с классом MainActivity, который его реализует. Класс MainActivity, реализуя интерфейс MainView представляет из себя прослойку между Android и абстрактным экраном, который представлен через интерфейс MainView. MainDialogs.kt также является частью этой связки и отвечает за отображение диалогов.

Также в корне проекта лежит файл ScheduleApplication.kt, в котором настраивается DI во время запуска приложения.

Класс ApplicationModule, находящийся в папке di, отвечает за внедрение в DI зависимостей, которые зависимы от Android.

В папке tools находится интерфейс файл Extesnsions.kt, который содержит функции расширения, в частности для перевода «Density-independent Pixels» в обычные экранные пиксели.

В папке view находится файл CustomPagerTitleStrip.java, который представляет из себя кастомный компонент карусели.

В папке settings находятся файлы SettingsViewImpl.kt, SettingsPresenter.kt, SettingsView.kt, SettignsInretactor.kt и SettingsRepository.kt, которые в совокупности образуют нижнюю выдвигалку с настройками. SettingsView представляет из себя интерфейс, который создает контракт, по которому можно изменять визуальные компоненты. Класс SettingsPresenter.kt содержит логику работы настроек и через интерфейс SettingsView общается с классом SettingsViewImpl, который его реализует. Класс SettingsViewImpl, реализуя интерфейс SettingsView представляет из себя прослойку между Android и абстрактными настройками, которые представлены через интерфейс SettingsView. Классы SettignsInretactor и SettingsRepository служат для получения и сохранения настроек во внутреннее хранилище Android.

В папке schedule находятся файлы ScheduleFragment.kt, SchedulePresenter.kt, ScheduleView.kt и ScheduleRepository.kt, которые в совокупности образуют экран с расписанием. ScheduleView представляет из себя интерфейс который создает контракт по которому можно изменять визуальные компоненты. Класс SchedulePresenter.kt содержит логику работы экрана расписания и через интерфейс ScheduleView общается с классом ScheduleFragment, который его реализует. Класс ScheduleFragment, реализуя интерфейс ScheduleView представляет из себя прослойку между Android и абстрактным экраном, который представлен через интерфейс ScheduleView. Класс ScheduleRepository служит для получения расписания из интернета и кэша. Помимо этих классов тут так же лежит файл Schedule.kt, который представляет из себя данные о расписании, файлы ScheduleAdapter.kt и ScheduleViewHolder.kt, которые служат специальным адаптером для android-компонента RecyclerView, файл DayView, который предоставляет из себя часть экрана, показывающую один день из расписания, файлы SchedulePagerAdapter.kt и ScheduleViewHolder.kt, которые служат специальным адаптером для еще одного android-компонента ViewPager2.

В папке teacher находятся файлы TeacherFragment.kt, TeacherPresenter.kt, TeacherView.kt и TeacherRepository.kt, которые в совокупности образуют экран с преподавателями. TeacherView представляет из себя интерфейс, который создает контракт, по которому можно изменять визуальные компоненты. Класс TeacherPresenter.kt содержит логику работы экрана с преподавателями и через интерфейс TeacherView общается с классом TeacherFragment, который его реализует. Класс TeacherFragment, реализуя интерфейс TeacherView представляет из себя прослойку между Android и абстрактным экраном, который представлен через интерфейс TeacherView. Класс TeacherRepository служит для получения информации о преподавателях из интернета и кэша. Помимо этих классов тут так же лежит файл Teacher.kt, который представляет из себя данные о преподавателе, файлы TeacherAdapter.kt и TeacherViewHolder.kt, которые служат специальным адаптером для android-компонента RecyclerView.

В папке res находятся статические, неисполняемые файлы ресурсы приложения.

Файл activity\_main.xml содержит разметку экрана главного окна.

Файл bottom\_sheet\_settings.xml содержит разметку нижней выдвигающейся панели и настроек.

Файл fragment\_schedule.xml содержит разметку экрана с расписанием.

Файл fragment\_teachers.xml содержит разметку экрана с преподавателями.

Файл item\_lesson.xml содержит разметку одного дня расписания.

Файл item\_teacher.xml содержит разметку одного преподавателя.

Файл search.xml содержит разметку для строки поиска преподавателей.

Таким образом были рассмотрены структуры файлов веб-сервера, админ-панели и мобильного приложения, а также архитектура веб-сервера и мобильного приложения. Архитектура разработанных компонентов позволит их легко изменять, модифицировать, добавлять новые функции, а также поддерживать существующие.

1. Тестирование

Тестирование – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом.

Статическое тестирование производится без запуска программного кода продукта. Тестирование осуществляется путем анализа программного кода или скомпилированного кода. Анализ может производиться как вручную, так и с помощью специальных инструментальных средств. Целью анализа является раннее выявление ошибок и потенциальных проблем в продукте.

Примерами ошибок, которые потенциально можно выявить с помощью автоматического статического тестирования, могут быть:

* утечки ресурсов (утечки памяти, не освобождаемые файловые дескрипторы и так далее);
* возможность переполнения буфера;
* ситуации частичной (неполной) обработки ошибок.

В отличие от статического, динамическое тестирование производится путем запуска продукта и проверки его функционала. Проверка осуществляется с помощью ручного или автоматического выполнения заранее подготовленного набора тестов. Примеры динамического тестирования:

* модульное тестирование;
* интеграционное тестирование;
* приемочное тестирование.

Чтобы удостовериться в правильной работе программного продукта был проведен тест-кейс.

Тест-кейс — это профессиональная документация тестировщика, последовательность действий, направленная на проверку какого-либо функционала, описывающая как прийти к фактическому результату.

В таблице 4.1 представлено тестирование программы.

Таблица 4.1 – Тест-кейсы программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Шаги | Ожидаемый результат | Результат |
| Запуск мобильного приложения | 1. Тап по иконке приложения | 1. Приложение открылось | Пройдено |
| Продолжение таблицы 4.1 | |  |  |
| Название | Шаги | Ожидаемый результат | Результат |
| Первое указание группы в мобильном приложении | 1. Начать вводить в единственное доступное поле номер группы | 1. В поле ввода появляются введенные символы | Пройдено |
| 2. Дождаться появления подсказки с доступными номерами групп | 2. Появляется выпадающая подсказка с номерами групп |
| 3. Нажать на номер группы | 3. В поле ввода отображается полный номер группы |
| 4. Нажать на кнопку «Готово» | 4. Окно закрылось, на главном экране отображается расписание для введенной группы |
| Изменение группы в мобильном приложении | 1. Нажать на кнопку «Настройки» | 1. Снизу выдвигается окно с настройками | Пройдено |
| 2. В появившемся окне нажать на кнопку «Изменить» напротив номера группы | 2. Открывается диалог с вводом номера группы |
| 3. Начать вводить номер группы в поле ввода | 3. В поле ввода появляются введенные символы |
| 4. Дождаться появления подсказки с доступными номерами групп | 4. Появляется выпадающая подсказка с номерами групп |
| 5. Нажать на номер группы | 5. В поле ввода отображается полный номер группы |
| 6. Нажать на кнопку «Готово» | 6. Окно закрылось, на главном экране отображается расписание для введенной группы |
| Продолжение таблицы 4.1 | |  |  |
| Название | Шаги | Ожидаемый результат | Результат |
| Просмотр расписания в мобильном приложении | 1. Ввести номер группы, расписание для которой заведомо известно | 1. Группа выбрана | Пройдено |
| 2. Просмотреть расписание этой группы | 2. Расписание совпадает с заранее известным расписанием |
| Просмотр преподавателя в мобильном приложении | 1. Перейти на экран с преподавателями | 1. Отображается страница с преподавателями | Пройдено |
| 2. Найти преподавателя, информация о котором заранее известна и тапнуть по нему. | 2. Станица с преподавателями открыта на преподавателе, информация о котором известна. |
| 2. Просмотреть информацию о преподавателе | 2. Информация о преподавателе совпадает с заранее известной информацией |
| Поиск преподавателя в мобильном приложении | 1. Перейти на экран с преподавателями | 1. Отображается страница с преподавателями | Пройдено |
| 2. Ввести часть имени, фамилии, отчества, номера телефона или описания преподавателя | 2. В списке преподавателей остались только те преподаватели, в имени, фамилии, отчестве, номере телефона или описании содержатся введённые символы |
| Продолжение таблицы 4.1 | |  |  |
| Название | Шаги | Ожидаемый результат | Результат |
| Ввод номера несущес- твующей группы в мобильном приложении | 1. Начать вводить в поле ввода номера группы номер несуществующей группы | 1. В поле ввода появляются введенные символы | Пройдено |
| 2. Убедится, что подсказок не появляется | 2. Подсказок не появляется |
| 3. Нажать на кнопку «Готово» | 3. Окно закрылось, на главном экране отображается надпись, что для текущей группы расписания нет |
| Добавление группы в админ-панели | 1. Перейти на страницу с группами | 1. Открывается страница с группами | Пройдено |
| 2. Нажать на кнопку «Создать» | 2. Открывается страница с добавлением групп |
| 3. Ввести данные группы | 3. Введенные данные группы отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Добавленная группа видна в списке всех групп |
| Редактирование группы в админ-панели | 1. Перейти на страницу с группами | 1. Открывается страница с группами | Пройдено |
| 2. Нажать на кнопку «Редактировать» напротив группы, которую необходимо отредактировать | 2. Открывается страница редактирования выбранной группы |
| 3. Ввести новые данные группы | 3. Введенные данные группы отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Отредактированная группа видна в списке всех групп |
| Продолжение таблицы 4.1 | | | |
| Название | Шаги | Ожидаемый результат | Результат |
| Удаление группы в админ-панели | 1. Перейти на страницу с группами | 1. Открывается страница с группами | Пройдено |
| 2. Выбрать группы, которые необходимо удалить | 2. Напротив выбранных групп появляется галочка |
| 3. Нажать на кнопку «Удалить» | 3. Выбранные группы больше не отображаются в списке всех групп |
| Поиск по группам в админ-панели | 1. Перейти на страницу с группами | 1. Открывается страница с группами | Пройдено |
| 2. Ввести часть номера группы в поле поиска | 2. В списке отображаются только те группы, в которых содержится введенная в поле поиска строка. |
| Добавление преподавателей в админ-панели | 1. Перейти на страницу с преподавателями | 1. Открывается страница с преподавателями | Пройдено |
| 2. Нажать на кнопку «Создать» | 2. Открывается страница с добавлением преподавателей |
| 3. Ввести данные преподавателя | 3. Введенные данные преподавателя отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Добавленный преподаватель виден в списке всех преподавателей |
| Продолжение таблицы 4.1 | | | |
| Название | Шаги | Ожидаемый результат | Результат |
| Редактирование преподавателей в админ-панели | 1. Перейти на страницу с преподавателями | 1. Открывается страница с преподавателями | Пройдено |
| 2. Нажать на кнопку «Редактировать» напротив преподавателя, которого необходимо отредактировать | 2. Открывается страница редактирования выбранного преподавателя |
| 3. Ввести новые данные преподавателя | 3. Введенные данные преподавателя отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Отредактированный преподаватель виден в списке всех преподавателей |
| Удаление преподавателей в админ-панели | 1. Перейти на страницу с преподавателями | 1. Открывается страница с преподавателями | Пройдено |
| 2. Выбрать преподавателей, которые необходимо удалить | 2. Напротив выбранных преподавателей появляется галочка |
| 3. Нажать на кнопку «Удалить» | 3. Выбранные преподаватели больше не отображаются в списке всех преподавателей |
| Поиск по преподавателям в админ-панели | 1. Перейти на страницу с преподавателями | 1. Открывается страница с преподавателями |  |
| Продолжение таблицы 4.1 | | | |
| Название | Шаги | Ожидаемый результат | Результат |
| Поиск по преподавателям в админ-панели | 2. Ввести часть ФИО преподавателя в поле поиска | 2. В списке отображаются только те преподаватели, в которых содержится введенная в поле поиска строка. | Пройдено |
| Добавление предметов в админ-панели | 1. Перейти на страницу с предметами | 1. Открывается страница с предметами | Пройдено |
| 2. Нажать на кнопку «Создать» | 2. Открывается страница с добавлением предметов |
| 3. Ввести данные предмета | 3. Введенные данные предмета отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Добавленный предмет виден в списке всех предметов |
| Редактирование предметов в админ-панели | 1. Перейти на страницу с предметами | 1. Открывается страница с предметами | Пройдено |
| 2. Нажать на кнопку «Редактировать» напротив предмета, который необходимо отредактировать | 2. Открывается страница редактирования выбранного предмета |
| 3. Ввести новые данные предмета | 3. Введенные данные предмета отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Отредактированный предмета виден в списке всех предметов |
| Удаление предметов в админ-панели | 1. Перейти на страницу с предметами | 1. Открывается страница с предметами |  |
| 2. Выбрать предметы, которые необходимо удалить | 2. Напротив выбранных предметов появляется галочка |
| Продолжение таблицы 4.1 | | | |
| Название | Шаги | Ожидаемый результат | Результат |
| Удаление предметов в админ-панели | 3. Нажать на кнопку «Удалить» | 3. Выбранные предметы больше не отображаются в списке всех предметов | Пройдено |
| Поиск по предметам в админ-панели | 1. Перейти на страницу с предметами | 1. Открывается страница с предметами | Пройдено |
| 2. Ввести часть номера предмета в поле поиска | 2. В списке отображаются только те предметы, в которых содержится введенная в поле поиска строка. |
| Добавление аудиторий в админ-панели | 1. Перейти на страницу с аудиториями | 1. Открывается страница с аудиториями | Пройдено |
| 2. Нажать на кнопку «Создать» | 2. Открывается страница с добавлением аудиторий |
| 3. Ввести данные аудитории | 3. Введенные данные аудитории отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Добавленная аудитория видна в списке всех аудиторий |
| Редактирование аудиторий в админ-панели | 1. Перейти на страницу с аудиториями | 1. Открывается страница с аудиториями | Пройдено |
| 2. Нажать на кнопку «Редактировать» напротив аудитории, которую необходимо отредактировать | 2. Открывается страница редактирования выбранной аудитории |
| 3. Ввести новые данные аудитории | 3. Введенные данные аудитории отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Отредактированная аудитория видна в списке всех аудиторий |
| Продолжение таблицы 4.1 | | | |
| Название | Шаги | Ожидаемый результат | Результат |
| Удаление аудиторий в админ-панели | 1. Перейти на страницу с аудиториями | 1. Открывается страница с аудиториями | Пройдено |
| 2. Выбрать аудитории, которые необходимо удалить | 2. Напротив выбранных аудиторий появляется галочка |
| 3. Нажать на кнопку «Удалить» | 3. Выбранные аудитории больше не отображаются в списке всех предметов |
| Поиск по аудиториям в админ-панели | 1. Перейти на страницу с аудиториями | 1. Открывается страница с аудиториями | Пройдено |
| 2. Ввести часть номера аудитории в поле поиска | 2. В списке отображаются только те аудитории, в которых содержится введенная в поле поиска строка. |
| Добавление пар в админ-панели | 1. Перейти на страницу с парами | 1. Открывается страница с парами | Пройдено |
| 2. Нажать на кнопку «Создать» | 2. Открывается страница с добавлением пар |
| 3. Ввести данные пары | 3. Введенные данные пары отображаются |
| 4. Нажать на кнопку «Сохранить» | 4. Добавленная пара видна в списке всех пар |
| Редактирование пар в админ-панели | 1. Перейти на страницу с парами | 1. Открывается страница с парами |  |
| 2. Нажать на кнопку «Редактировать» напротив пары, которую необходимо отредактировать | 2. Открывается страница редактирования выбранной пары |
| 3. Ввести новые данные пары | 3. Введенные данные пары отображаются |
| Продолжение таблицы 4.1 | | | |
| Название | Шаги | Ожидаемый результат | Результат |
| Редактирование пар в админ-панели | 4. Нажать на кнопку «Сохранить» | 4. Отредактированная пара видна в списке всех аудиторий | Пройдено |
| Удаление пар в админ-панели | 1. Перейти на страницу с парами | 1. Открывается страница с парами | Пройдено |
| 2. Выбрать пары, которые необходимо удалить | 2. Напротив выбранных пар появляется галочка |
| 3. Нажать на кнопку «Удалить» | 3. Выбранные пары больше не отображаются в списке всех пар |
| Фильтрация по парам в админ-панели | 1. Перейти на страницу с парами | 1. Открывается страница с парами | Пройдено |
| 2. Нажать на кнопку фильтрации | 2. Из кнопки выпадает список критериев фильтрации |
| 3. Выбрать критерий фильтрации | 3. Появляется поле, в котором можно выбрать, объект фильтрации |
| 4. Выбрать объект фильтрации | 4. В списке отображается только записи с объектом фильтрации |

В ходе тестирования программного средства было проверено качество ПС. Дефектов у программного средства выявлено не было.

1. Применение

## Описание процесса установки и запуска приложения

Админ панель установке не подлежит, поскольку это сайт.

Для запуска админ панели необходимо перейти на сайт, на котором она расположена.

Веб-сервер в установке не нуждается, он представляет из себя один jar файл. Для его запуска необходимо, чтобы на компьютере была уставлена java 9 или новее. Для непосредственно запуска необходимо использовать команду «java –jar {имя jar-файла}».

Для установки приложения на Android устройство, необходимо скачать apk файл, полученный в результате сборки проекта на целевое устройство. В настройках телефона необходимо включить установку из неизвестных источников. После этого достаточно запустить apk файл. Откроется экран установки, в котором надо согласится с установкой приложения.

Для запуска приложения необходимо найти его в списке установленных приложений и нажать на него.

## Руководство пользователя

Веб-сервер не имеет графического интерфейса, поэтому пользоваться им напрямую невозможно. Однако мобильное приложение и админ-панель не будут без него работать. Так что даже несмотря на то, что мы не видим работу сервера, он должен быть запущен в обязательном порядке.

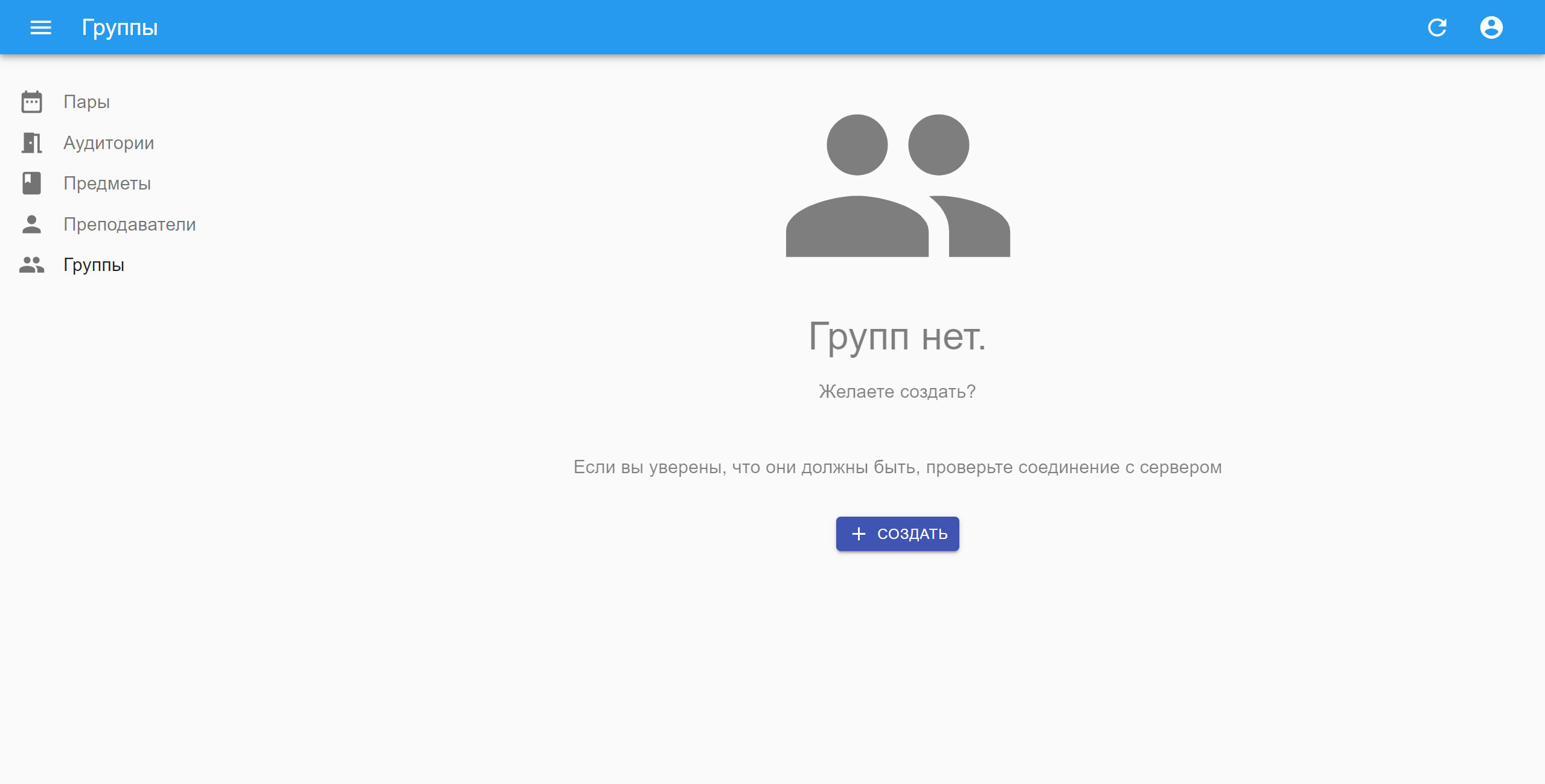
Админ-панель представляет из себя инструмент для работников учебных заведений, ответственных за составление расписания. В ней они могут редактировать, добавлять, удалять преподавателей, предметы, группы и наконец составлять расписания.

При первом запуске админ-панели после установки сервера пользователя будет встречать сообщение о том, что расписания нет. Так как это только что установленный сервер, его там быть и не должно.

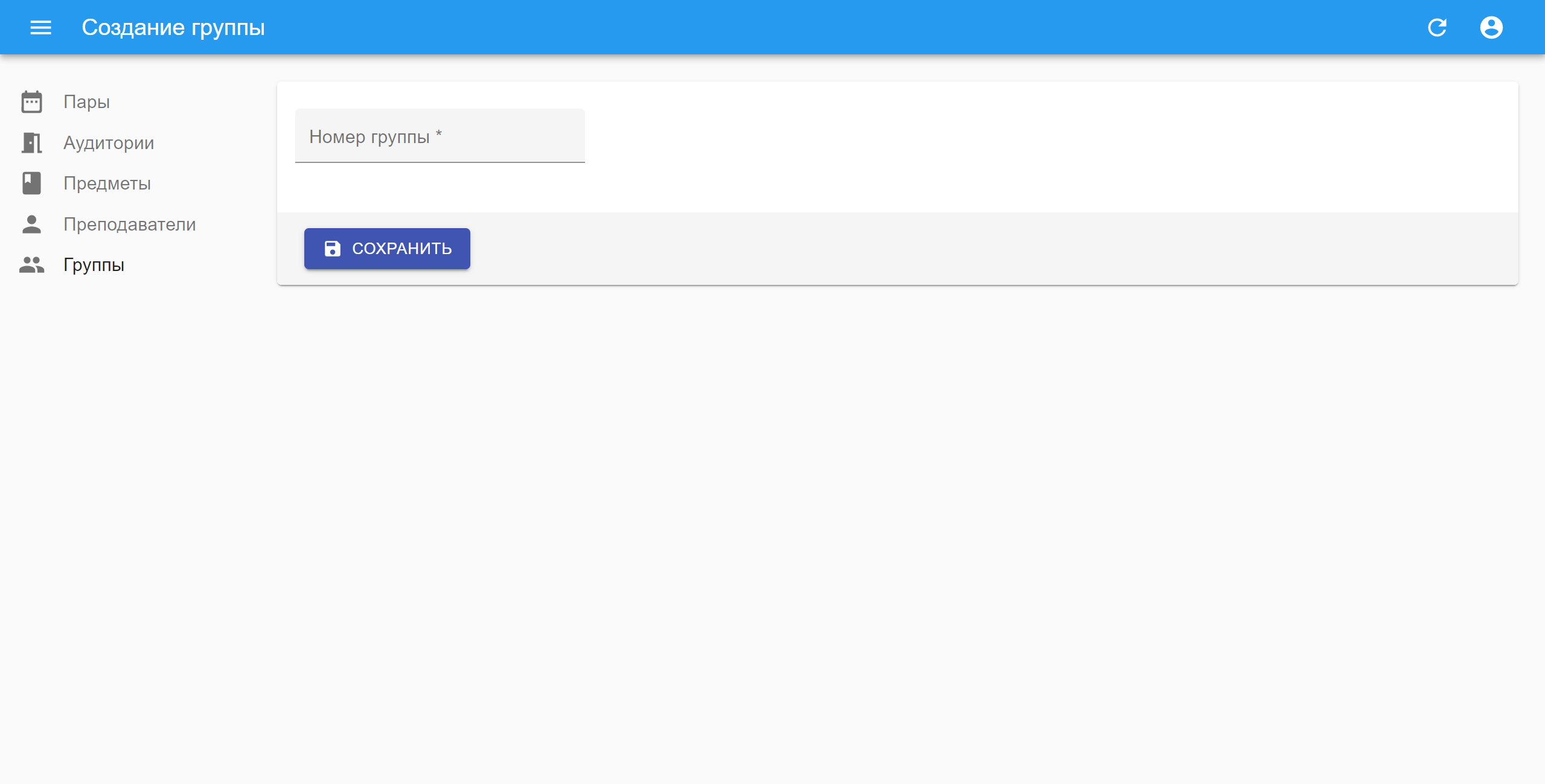
Однако перед тем, чтобы создать расписание сначала необходимо добавить преподавателей, группы, предметы и аудитории.

Без преподавателей, групп, предметов и аудиторий расписание составить просто невозможно, так как расписание состоит из всех этих компонентов в обязательном порядке.

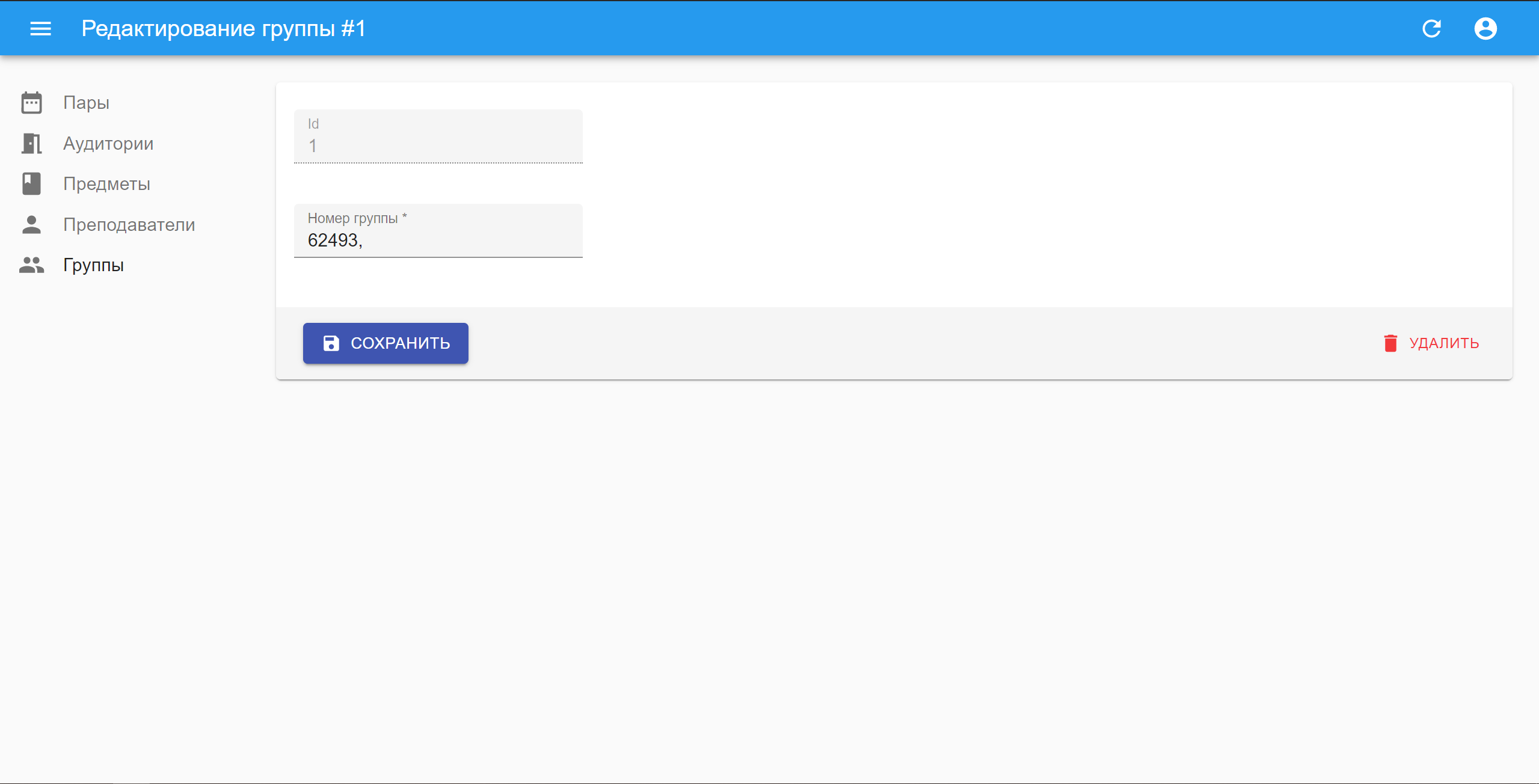
Для начала заполнения всех необходимых данных, необходимо выбрать тип данных, которые мы хотим заполнить. Начнем с групп. Окно с пустым списком групп представлено на рисунке 5.1.

Рисунок 5.1 – Страница групп с пустым списком

Для добавления группы, необходимо нажать на кнопку «Создать». Тогда мы перейдем на страницу добавления групп. Эта страница представлена на рисунке 5.2.

Рисунок 5.2 – Страница добавления групп

В данном окне нам необходимо только ввести номер группы и жать на кнопку «Сохранить». Нас перебросит на страницу просмотра группы, там мы сможем отредактировать нашу группу. Окно просмотра группы представлено на рисунке 5.3.

Рисунок 5.3 – Страница редактирования группы

Как мы можем заметить, в номере группы была совершена опечатка. Для исправления этой опечатки, необходимо отредактировать номер группы и нажать на кнопку «Сохранить».

После добавления нескольких групп, на странице просмотра групп мы увидим их все. Пример приведен на рисунке 5.4.

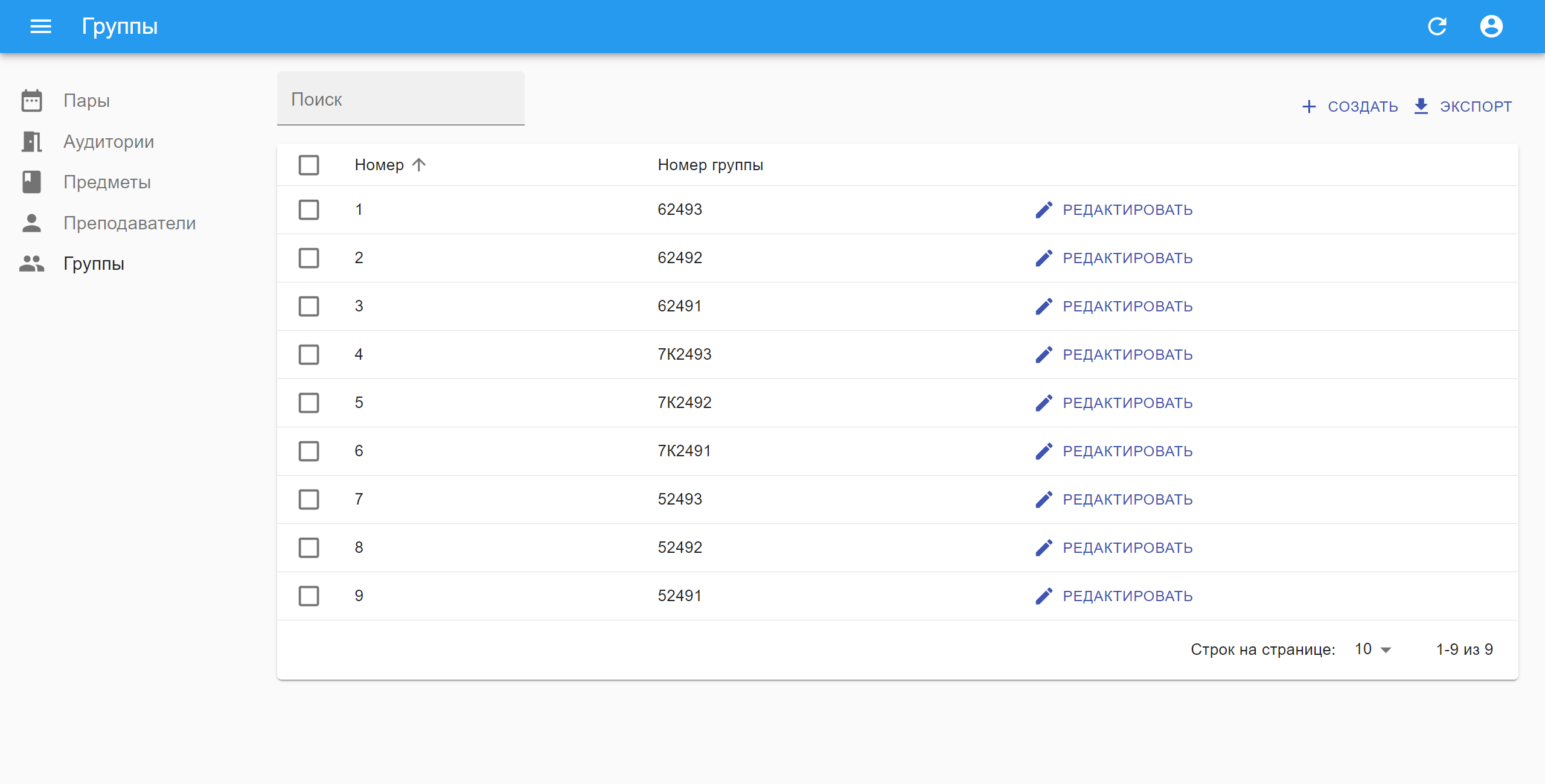


Рисунок 5.4 – Пример заполнения страницы с группами

Далее заполним преподавателей. Сейчас их нет, поэтому зайдем на страницу преподавателей и нажмем на кнопку «Создать». Перед нами появится окно, идентичное тому, что мы видели при добавлении групп, однако полей будет побольше. Экран добавления преподавателей представлен на рисунке 5.5.

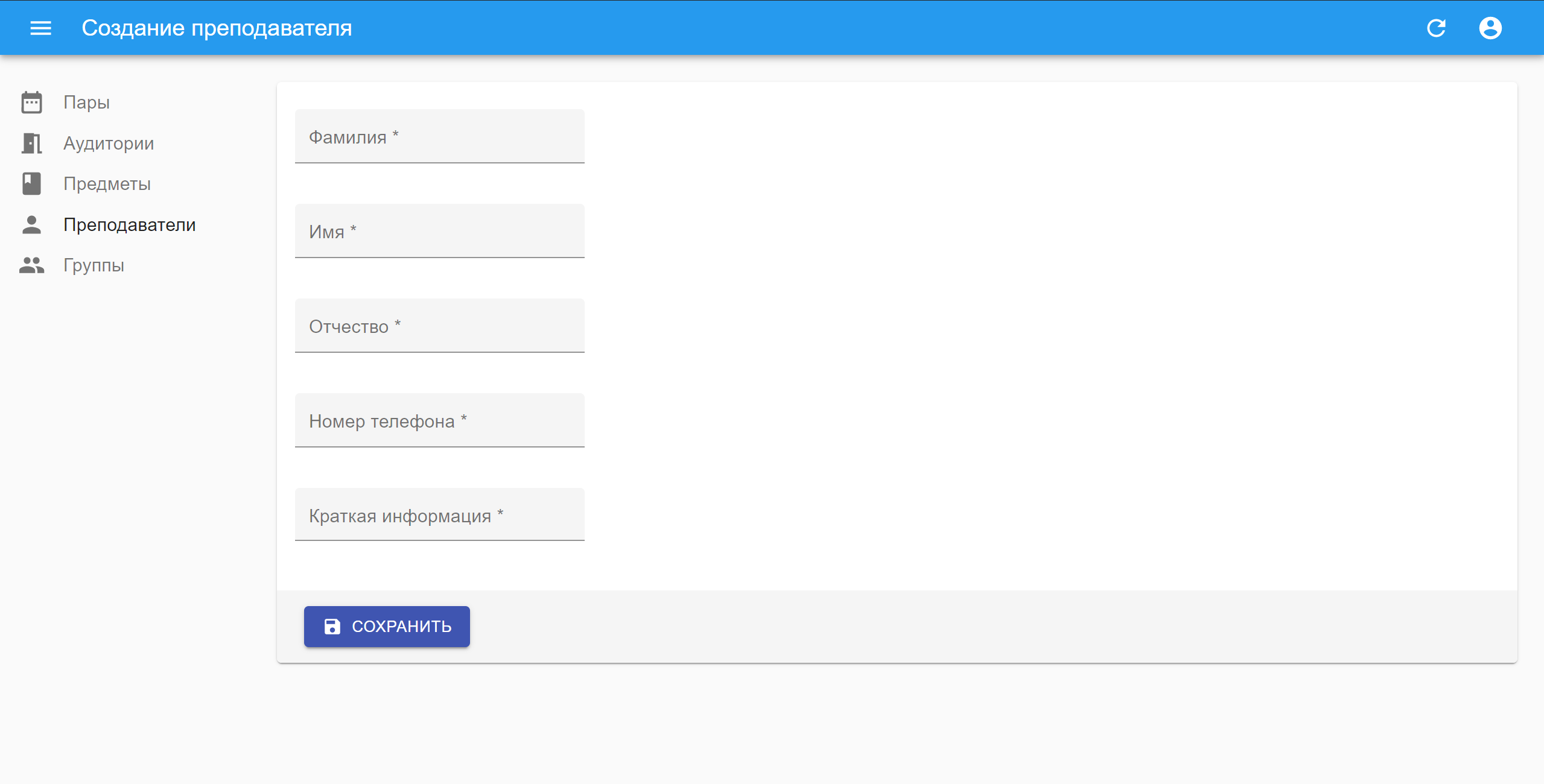


Рисунок 5.5 – Экран добавления преподавателей.

С именем, фамилией, отчеством и номером телефона все и так понятно. Поле краткой информации может содержать в себе краткую биографию преподавателя, которая может быть полезна учащимся, а также дополнительные контакты, такие как Telegram, Viber, VK, WhatsApp, второй номер телефона и другие. Какие именно, решает сам преподаватель. Он может написать сюда все что угодно.

Пример заполнения страницы преподавателей представлен на рисунке 5.6.

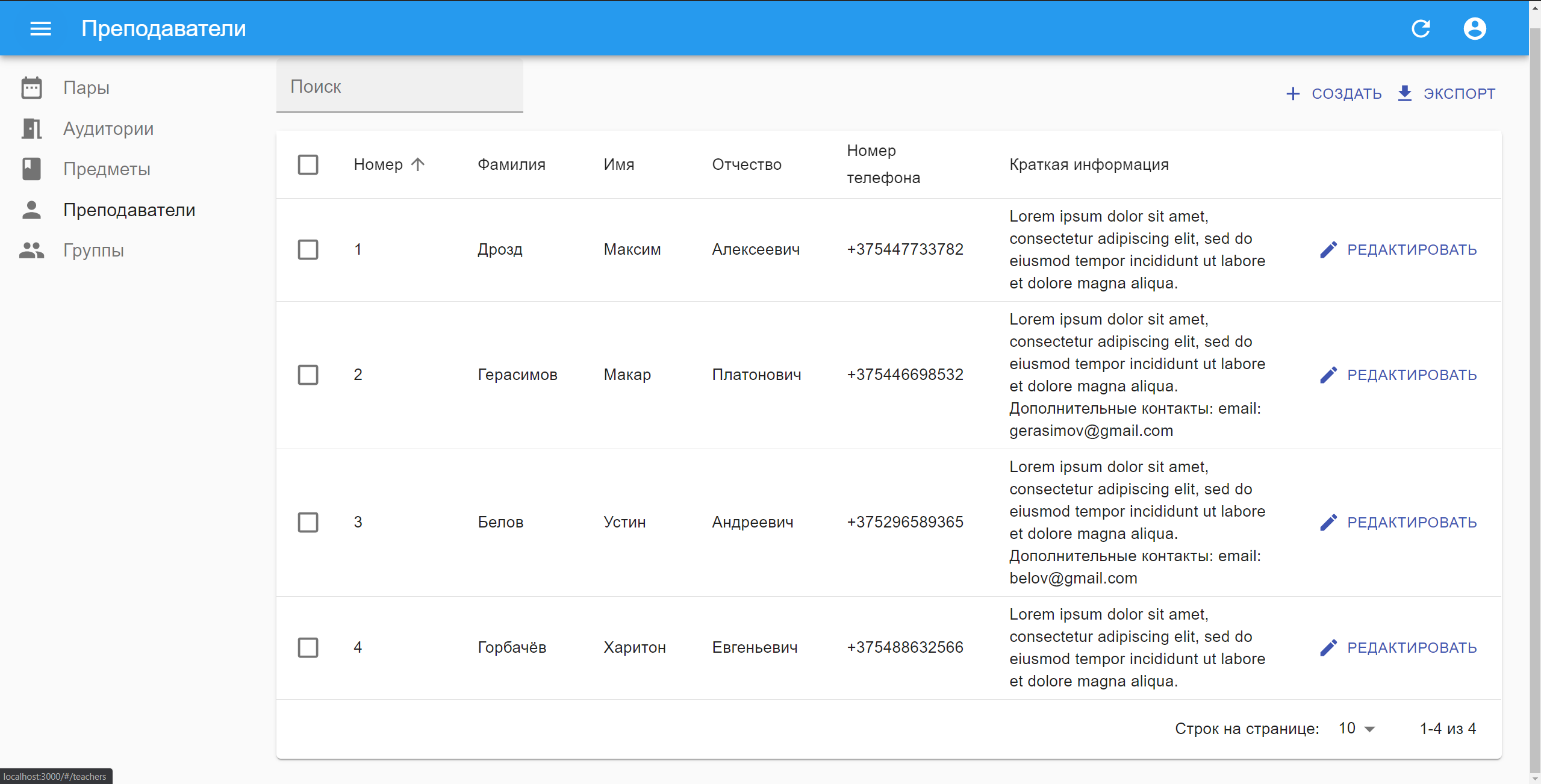


Рисунок 5.6 – Экран с преподавателями.

Преподавателей, так же, как и группы можно редактировать. Для этого необходимо открыть преподавателя, отредактировать его и нажать на кнопку «Сохранить». Экран редактирования преподавателей представлен на рисунке 5.7.

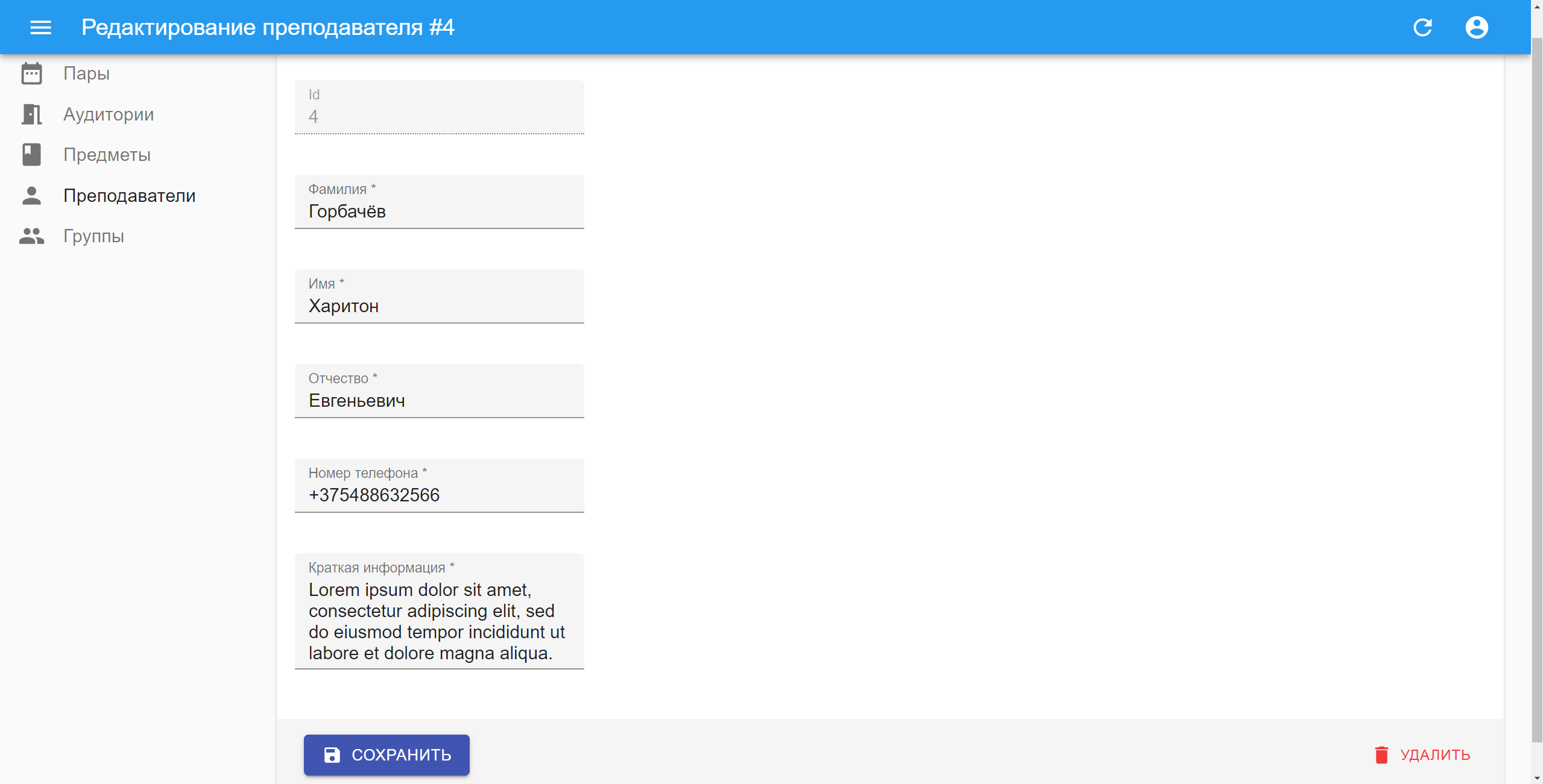


Рисунок 5.7 – Экран редактирования преподавателей

Экран добавления предметов и аудиторий точно такой же, как и групп. Пользоваться ими можно точно так же, как и экраном добавления групп. Экран добавления предметов представлен на рисунке 5.8.

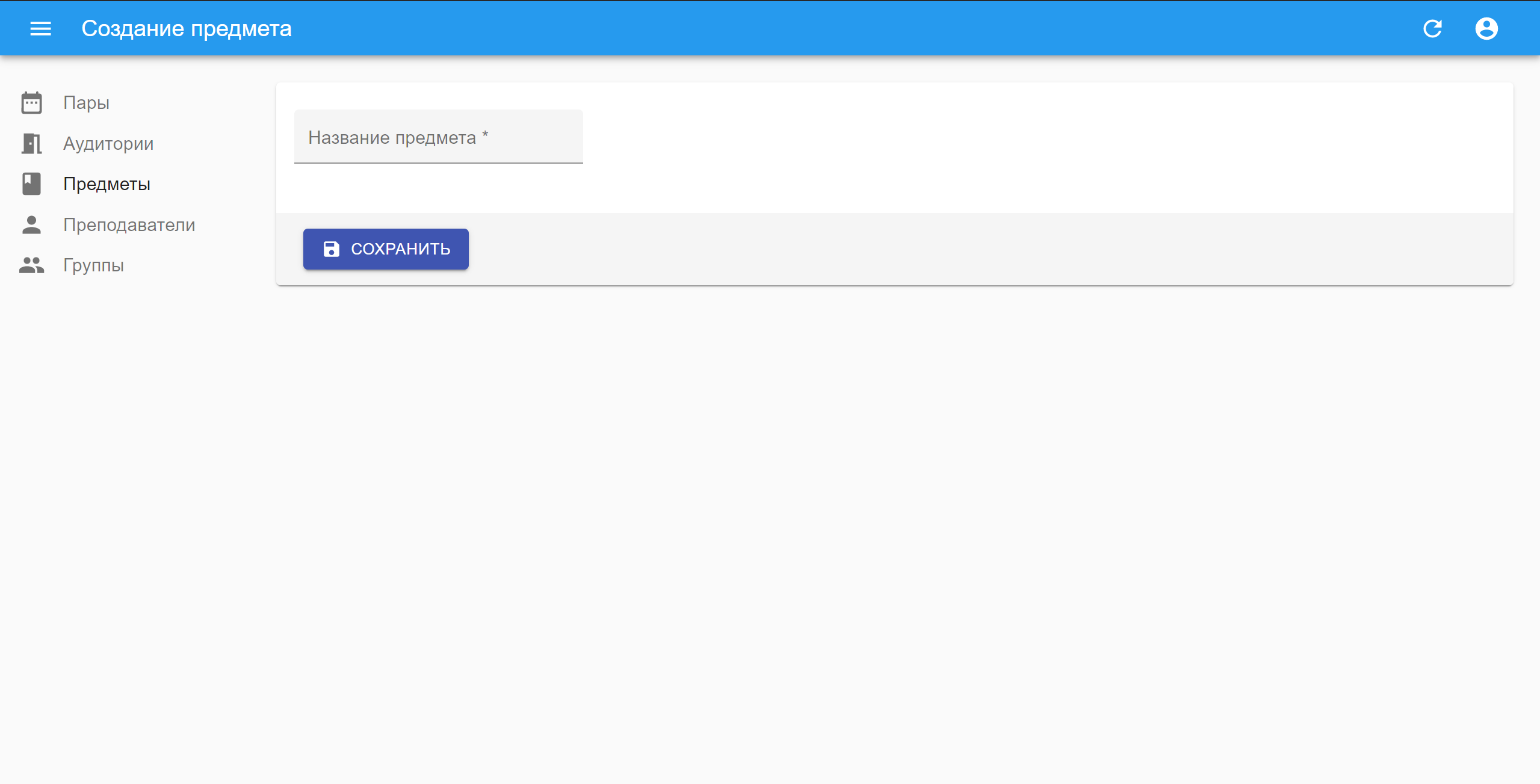


Рисунок 5.8 – Экран добавления предметов

Экран добавления аудиторий представлен на рисунке 5.9.

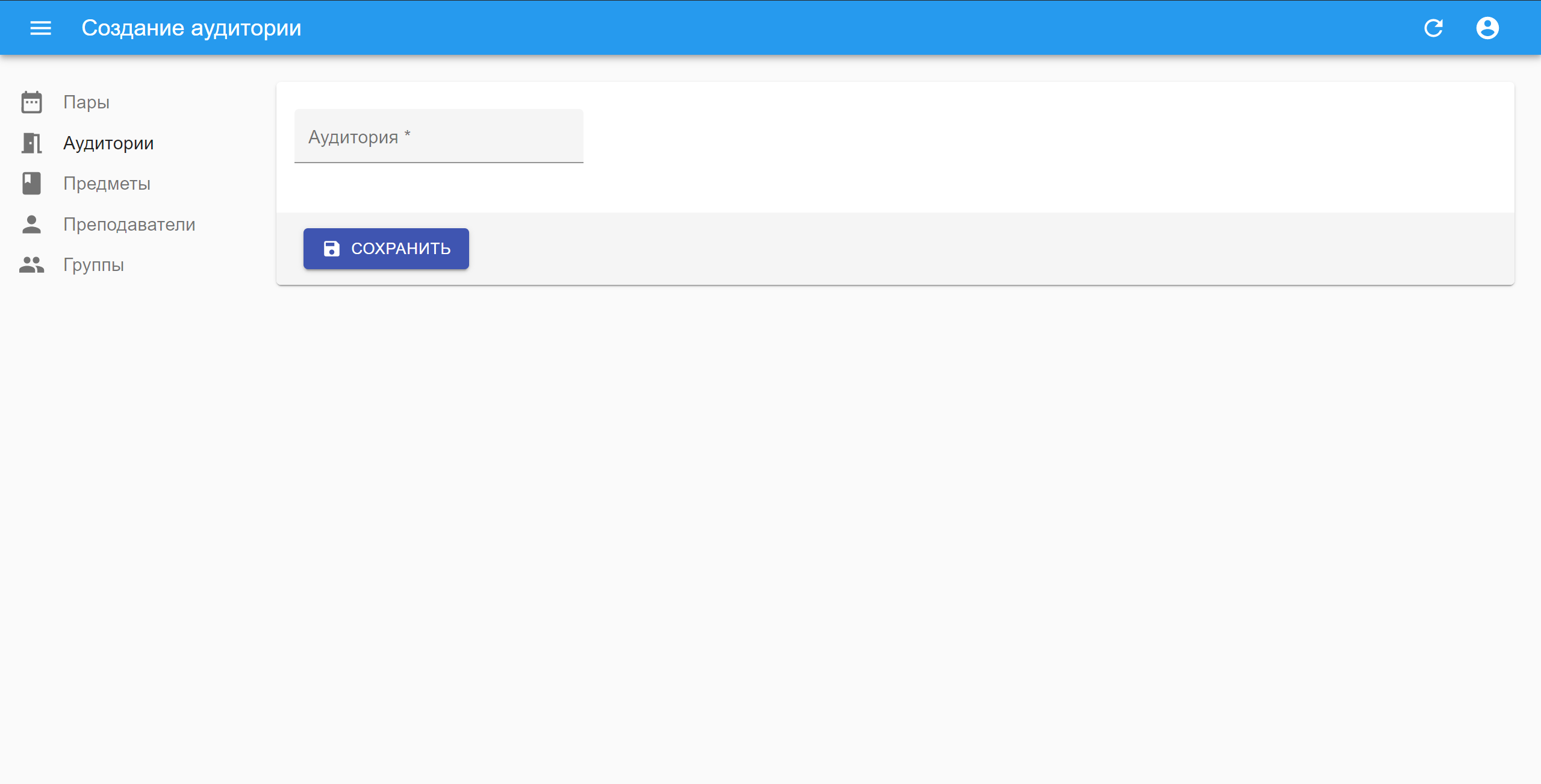


Рисунок 5.8 – Экран добавления аудиторий

Редактировать аудитории и предметы также, как и группы – выбрать аудиторию или предмет, изменить данные и нажать на кнопку «Сохранить».

Пример экрана со списком с добавленными предметами представлен на рисунке 5.9.

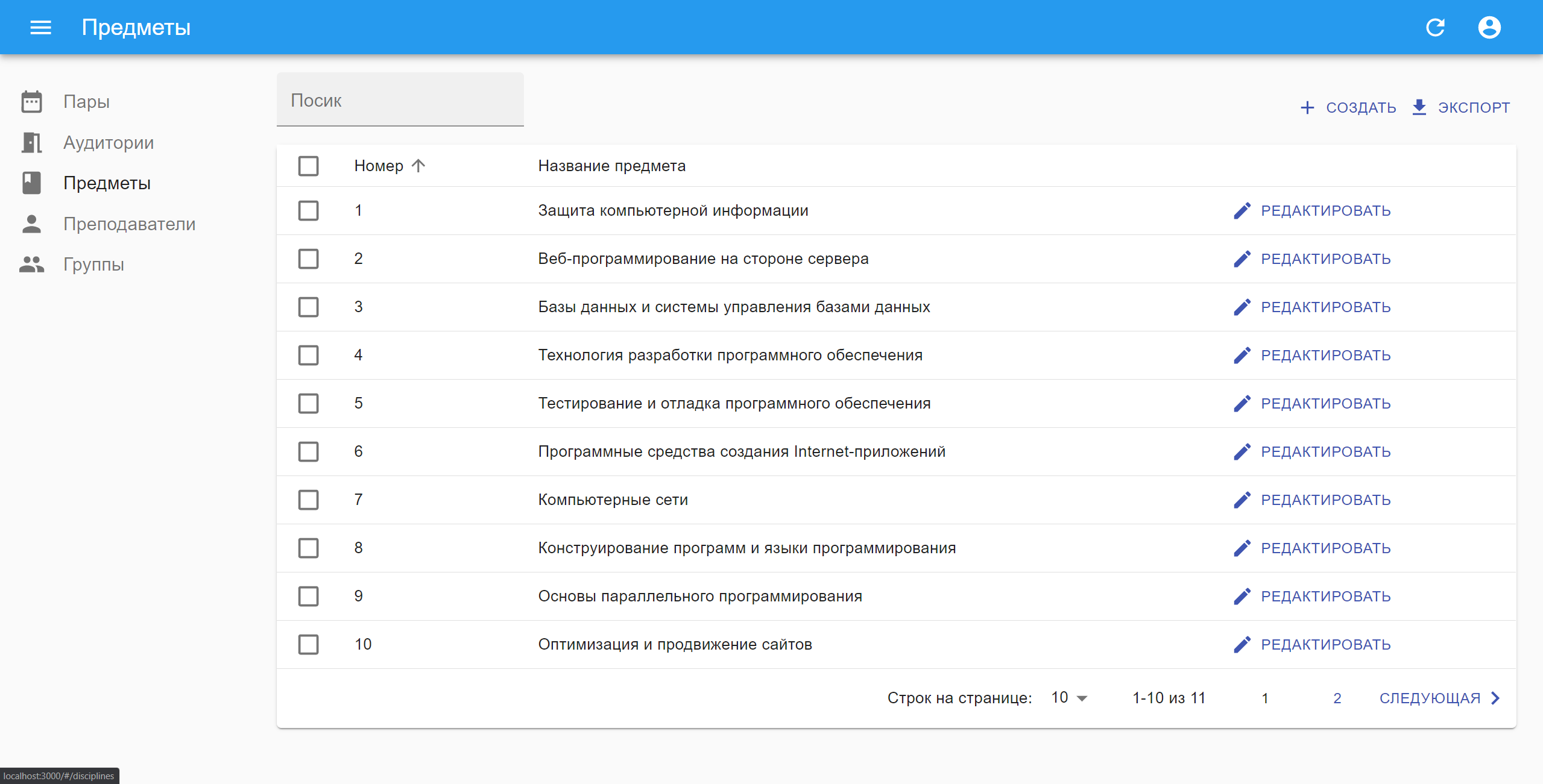


Рисунок 5.8 – Пример заполнения экрана с предметами

Пример списка с добавленными предметами представлен на рисунке 5.9.

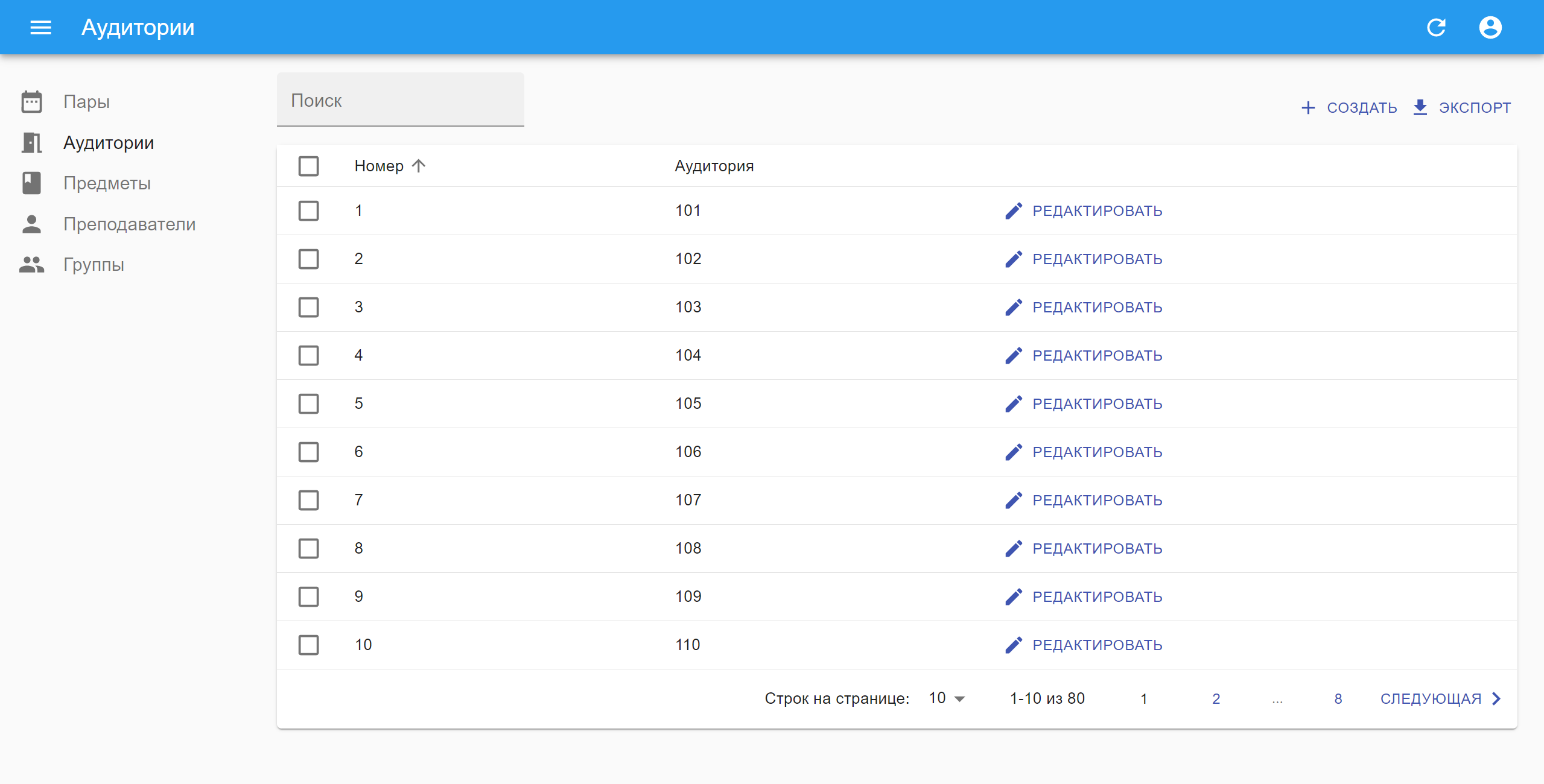


Рисунок 5.9 – Пример заполнения экрана с аудиториями.

Теперь можно начать составлять расписания. Экран с добавлением пары в расписание представлен на рисунке 5.10.

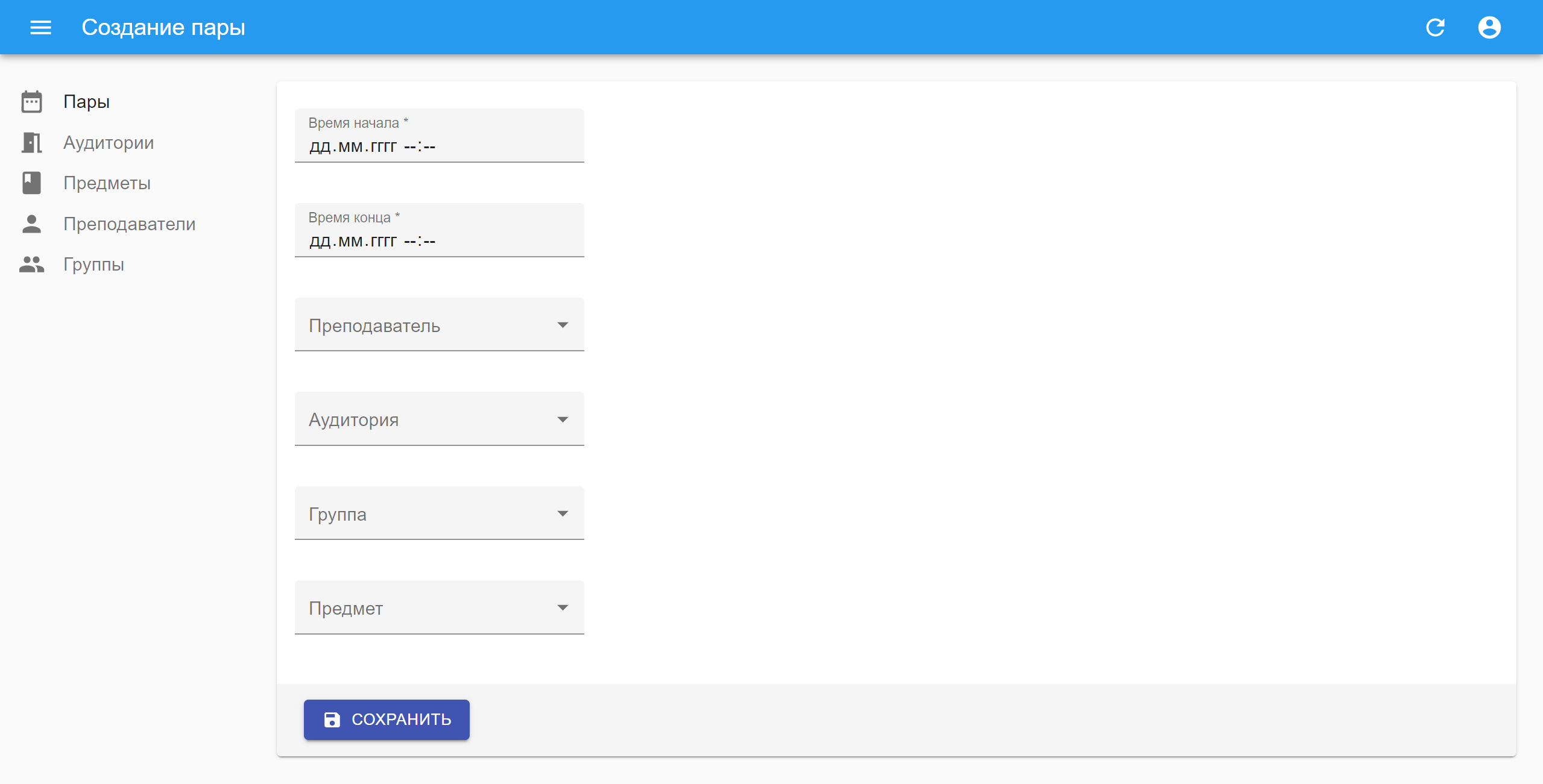


Рисунок 5.10 – Экран с добавлением пар в расписание.

Для добавления пары в расписание требуется указать время ее начала и время ее конца. Ввод производится с клавиатуры или с помощью мыши. Затем следует выбрать преподавателя, аудиторию, группу и предмет. Если нажать на эти поля, то можно заметить, что это не просто поле ввода текста, это поле выбора из уже существующих вариантов. Так, например, если мы нажмем на список преподавателей

При первом запуске приложения необходимо ввести номер группы в окно, представленное на рисунке 5.2.1. По ходу ввода номера группы выпадет подсказка с номерами групп, доступными на данный момент.

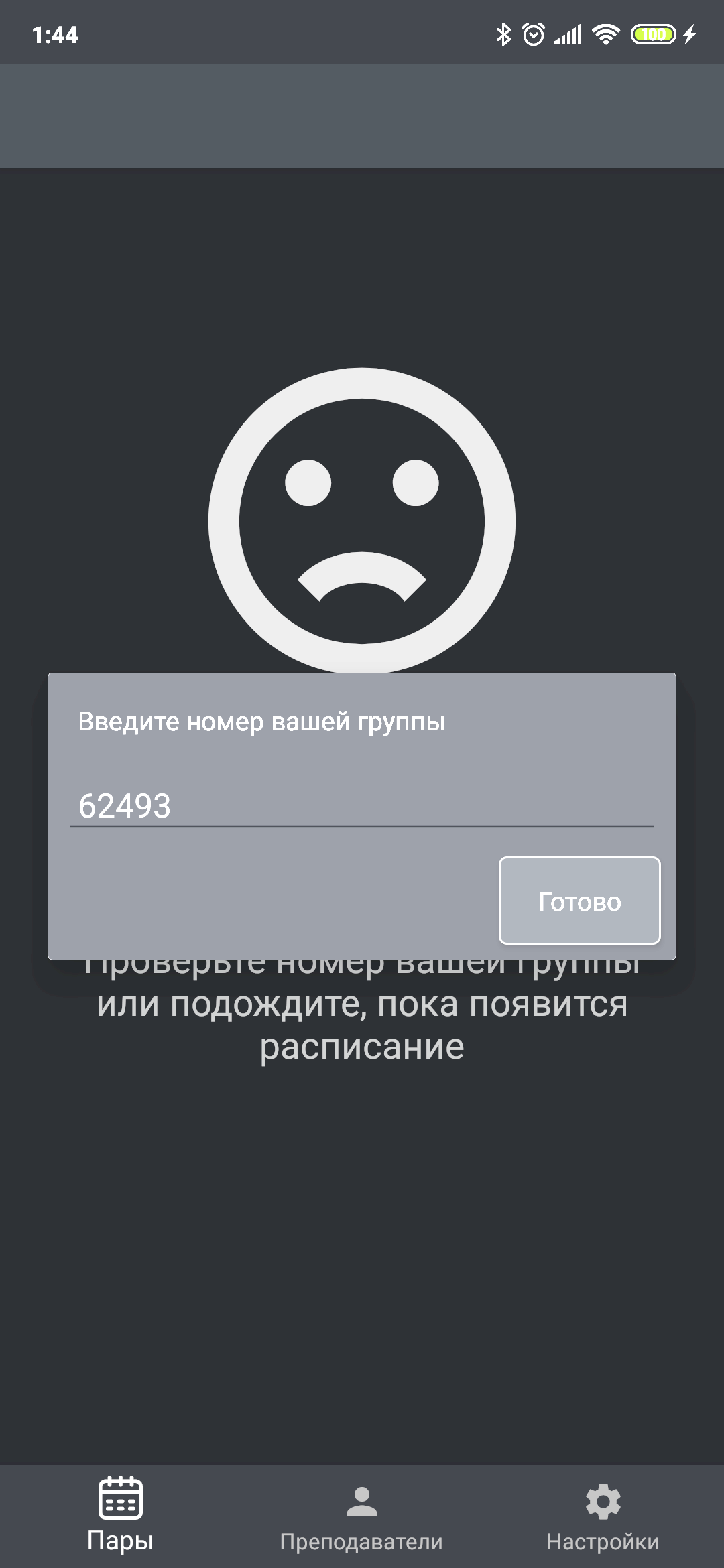


Рисунок 5.2.1 – Окно ввода номера группы

После ввода номера группы откроется окно с расписанием на ближайший день. Свайпами влево-вправо можно переключаться межу расписаниями на разные дни. Окно представлено на рисунке 5.2.2.

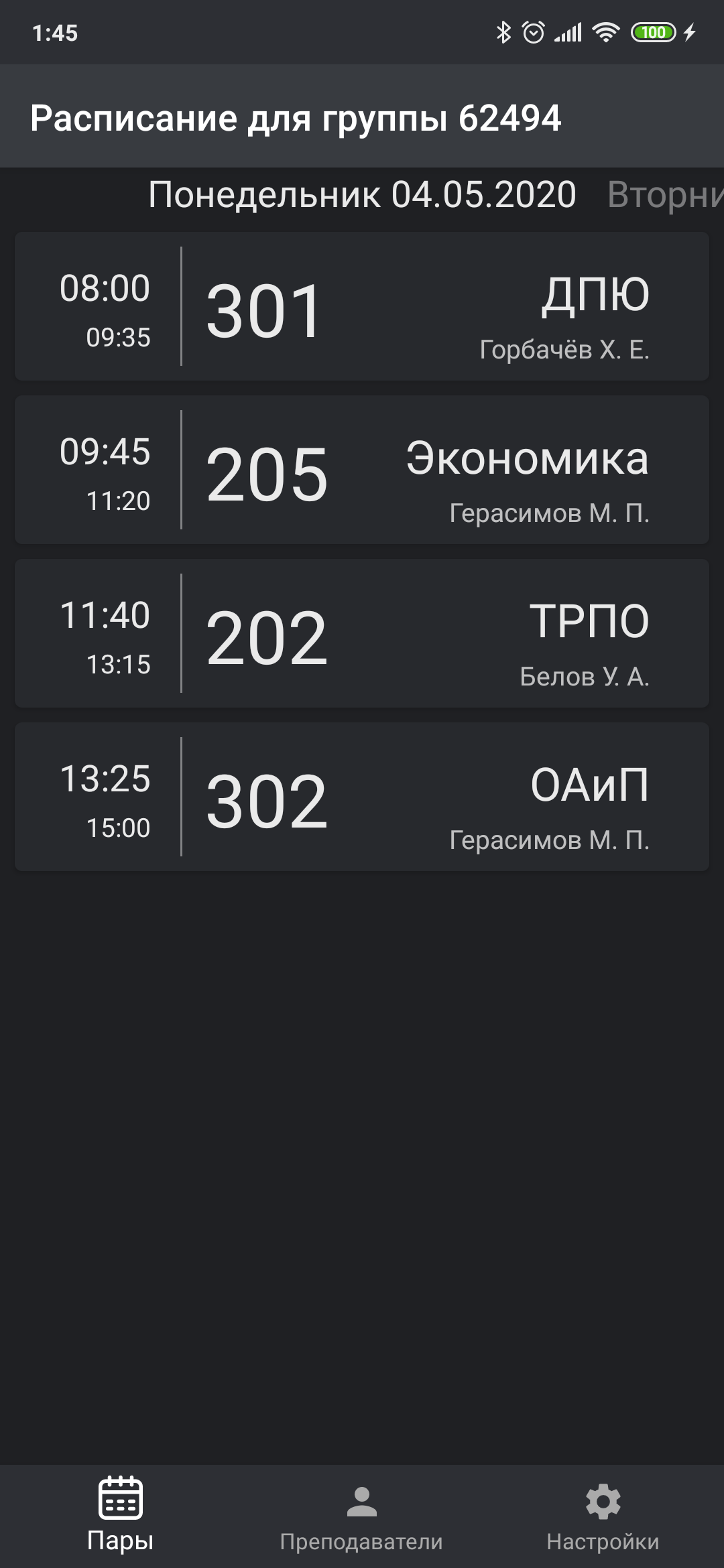


Рисунок 5.2.2 – Окно с расписанием

В случае если пользователь ошибся при вводе номера группы, или если он захочет поменять ее, но может это сделать нажатием по кнопке настроек. После нажатия на кнопку, выедет окно, в котором можно посмотреть текущий номер группы и нажать на кнопку изменения расписания. Экран с окном настроек представлен на рисунке 5.2.3.

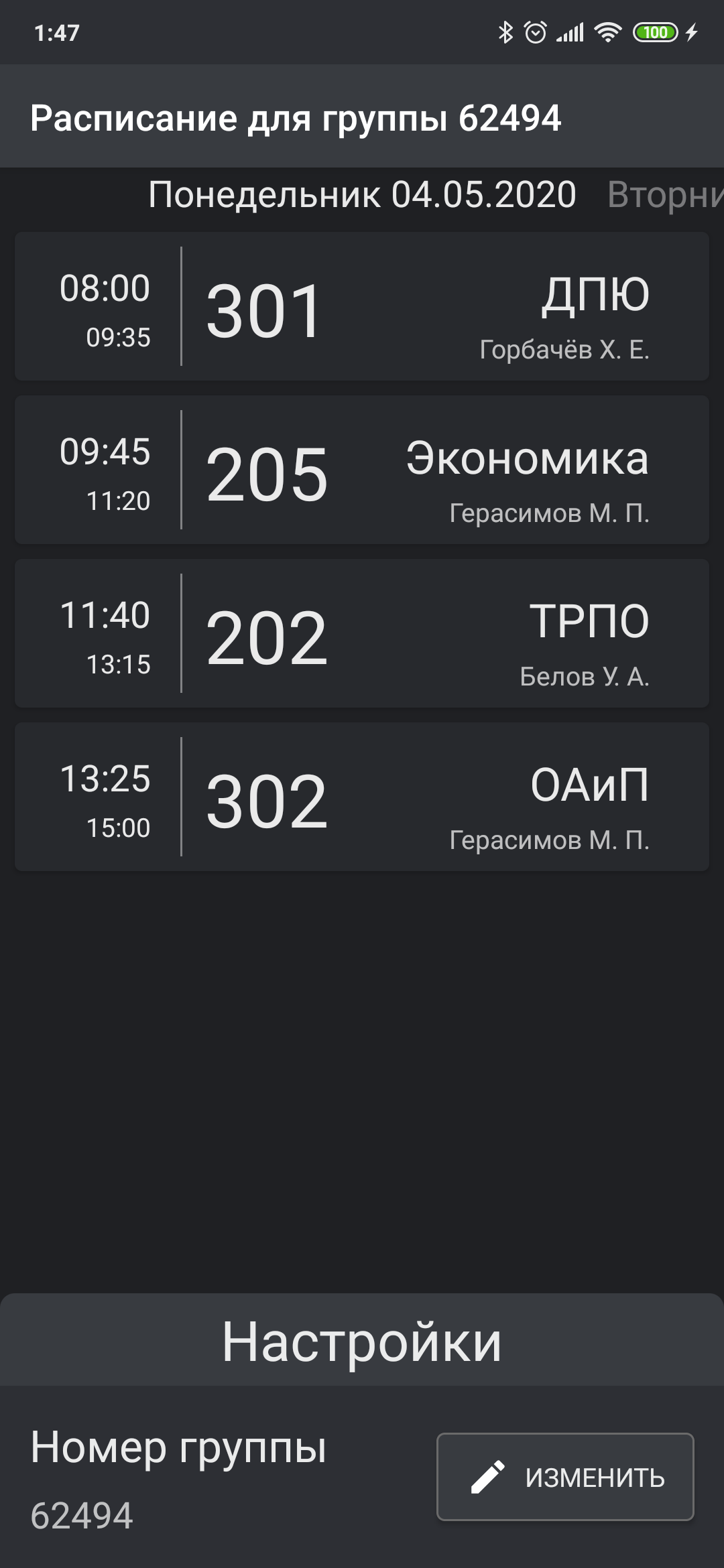


Рисунок 5.2.3 – Экран с окном настроек

При нажатии на кнопку редактирования появляется окно ввода номера группы. При заполнении нового номера группы будут появляться подсказки возможных номеров групп. Они формируются из списка всех групп. Окно редактирования с подсказами представлено на рисунке 5.2.4.

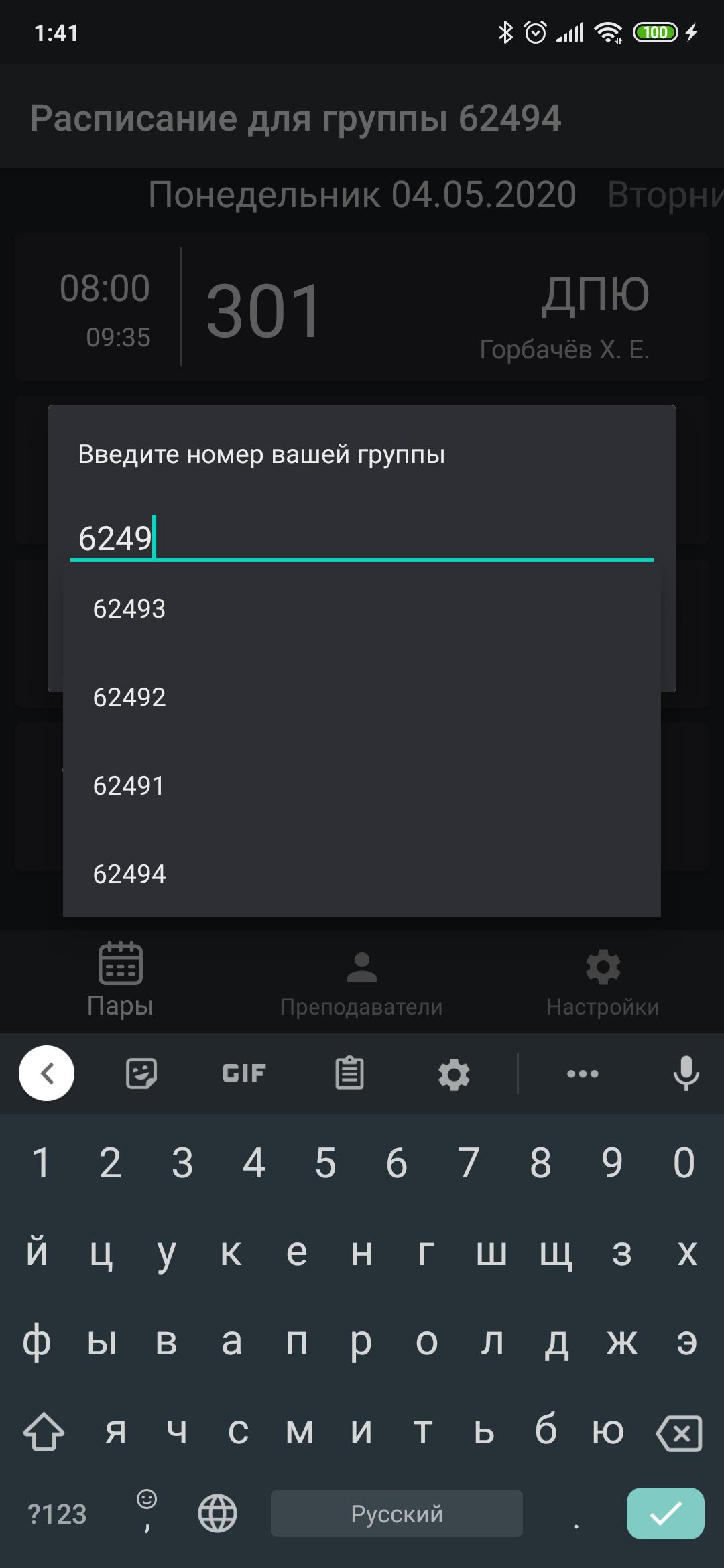


Рисунок 5.2.4 – Окно ввода номера группы с подсказками

После ввода нового номера группы, расписание обновится и будет актуальным. В процессе обучения может понадобится необходимость в личном общении с преподавателями. Для этого необходимо знать их контактные данные. Специально для этого существует вкладка преподавателей. В ней расположен список всех преподавателей, их номер телефона и краткая информация с альтернативными средствами связи. Список преподавателей изображен на рисунке 5.2.5.

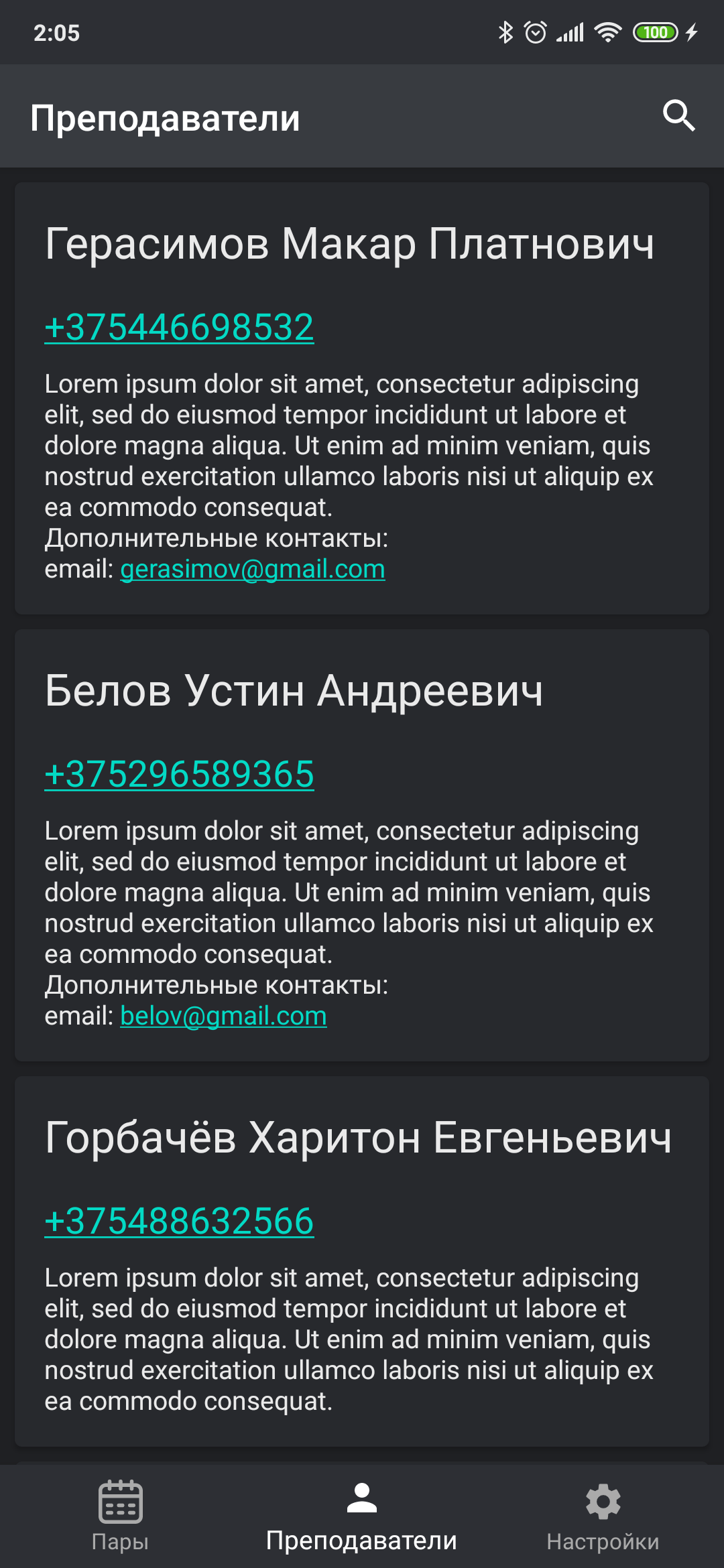


Рисунок 5.2.5 – Окно со списком преподавателей

Если необходимо быстро найти конкретного преподавателя, можно воспользоваться поиском. Для этого необходимо нажать на иконку лупы в правом верхнем углу экрана. В результате появится строка поиска в которую можно водить имеющуюся информацию. Пример пользования поиском представлен на рисунке 5.2.6.

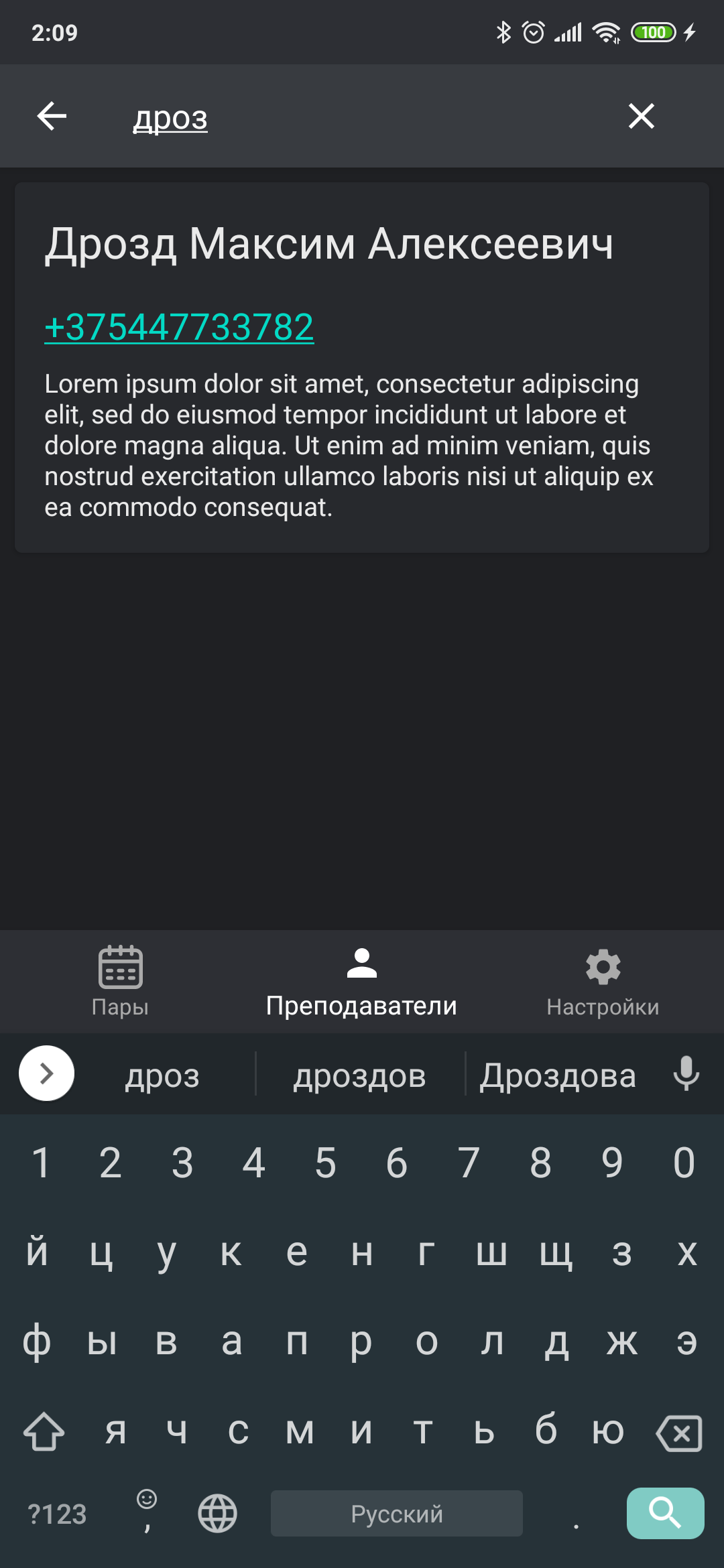


Рисунок 5.2.6 – Пример пользования поиском

Таким образом был рассмотрен весь функционал разрабатываемого приложения.

# Заключение

Преддипломная практика проходила в ООО «ЯндексБел» под руководством представителя от предприятия в течение 20 календарных дней.

По завершению практики были выполнены все поставленные задачи преддипломной практики, а также достигнуты следующие цели:

* развитие профессионального мышления, повышение уровня квалификации по профессии;
* изучение технологии и организации производства;
* приобретение практического опыта, навыков по организаторской деятельности, способствующих формированию профессиональной компетентности будущего специалиста;
* формирование профессиональных знаний и навыков работы в коллективе.

В результате преддипломной практики было разработано мобильное приложение «Расписание учебного заведения», проанализированы аналоги этого приложения, описана предметная область, подробно описана физическая структура проекта, а также подробно описаны разработанные модули.

В конце разработки было проведено тестирование приложения.

В результате преддипломной практики был также проведён экономический расчёт, в результате которого было определено, что разрабатываемый ПП является прибыльным и окупится уже на третий год.

Исходя из проведенных расчетов и полученных данных был сделан вывод, что использование данного программного средства является целесообразным.

В ходе практики были изучены должностные обязанности техника-программиста, были закреплены знания и умения, полученные в результате обучения в колледже

, выполнено индивидуальное задание.

# Список использованных источников

Литература

1. Медведев, В.И. M 42 Особенности объектно-ориентированного программирования на C++/CLI, C# и Java/ В.И. Медведев, 2-е изд., испр. и доп. - Казань: РИЦ «Школа», 2010

Электронный ресурс

1. Android documentation – [Электронный ресурс]. – Режим доступа: https://developer.android.com/
2. Javalin tutorials – [Электронный ресурс]. – Режим доступа: https://javalin.io/tutorials/
3. Расписание БГУИР – [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.bakan.universchedule
4. BSUIR Schedule – [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.bsuir.pechuro.bsuirschedule
5. БГУИР Расписание– [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=by.bsuir.schedule
6. Расписание – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Расписание

# Приложение А

План-проспект

|  |  |  |
| --- | --- | --- |
| 1. | Тема проекта | Мобильное приложение «Расписание учебного заведения» |
| 2. | Актуальность проекта | На данный момент на рынке нет программного обеспечения для ведения расписания, которое можно легко внедрить в любое учебное заведение. |
| 3. | Цель проекта | Целью дипломного проекта является разработка мобильного приложения «Расписание учебного заведения». Для достижения этой цели, помимо мобильного приложения так же придется разработать бэкэнд часть и админ-панель. В дальнейшем этот продукт можно будет внедрить в любое учебное заведение. |
| 4. | Задачи проекта | Имеются следующие задачи дипломного проектирования в рамках описанной темы:  - проектирование схемы базы данных, предназначенной для хранения всех необходимых для расписания данных;  - описание алгоритмов, работы программы. Кроме того, подробное описание логической модели и физической структуры программы, в частности, реализация классов и основных методов;  - проведение тестирования программы, которое будет показывать, что программа работает корректно и готова к использованию;  - анализ существующих средств разработки программных средств. |
| 5. | Содержание проекта | Введение  1 Постановка задачи (цели, задали и функциональное назначение разрабатываемого ПС)  1.1 Описание предметной области  1.2 Обзор существующих аналогов  1.3 Функциональное назначение  2 Проектирование (определение структуры проекта, способы манипулирования данными, проектирование внешнего вида ПС)  2.1 Логическое моделирование  2.2 Проектирование интерфейса  2.3 Выбор и обоснование инструментов разработки  3 Программная реализация  3.1 Физическая структура  3.2 Описание разработанных модулей  4 Тестирование (тест-кейс, результаты тестирования)  5 Применение (инструкция по установке и использованию)  5.1 Описание процесса установки и запуска приложения 5.2 Руководство пользователя  6 Экономический раздел  7 Охрана труда  Заключение  Список использованных источников  Графическая часть:  – Диаграмма сущность-связь  – Диаграмма деятельности |
| 6. | Источники информации по теме проекта | 1. Android documentation – [Электронный ресурс]. – Режим доступа: https://developer.android.com/ 2. Javalin tutorials – [Электронный ресурс]. – Режим доступа: https://javalin.io/tutorials/ 3. Расписание БГУИР – [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.bakan.universchedule 4. BSUIR Schedule – [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=com.bsuir.pechuro.bsuirschedule 5. БГУИР Расписание– [Электронный ресурс]. – Режим доступа: https://play.google.com/store/apps/details?id=by.bsuir.schedule 6. Расписание – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Расписание |