Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

Допускаю к защите

|  |  |
| --- | --- |
| Руководитель |  |
|  | подпись |
|  | М.Д. Ярков |
|  | И.О. Фамилия |

Разработка службы для добавления расписания в календарь

|  |
| --- |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

|  |
| --- |
| Объектно-ориентированное программирование |

|  |
| --- |
| 1.023.00.00 - ПЗ |
| обозначение документа |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | ИСТб-20-3 |  |  |  | | М.Д. Ярков |
|  |  | шифр |  | подпись |  | | И.О. Фамилия |
| Нормоконтроль |  |  |  |  |  | | Т.В. Маланова |
|  |  |  |  | подпись |  | | И.О. Фамилия |
| Курсовой проект защищен с оценкой | | | |  | |  | |

Иркутск 2022 г.

**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ЗАДАНИЕ**

**НА КУРСОВОЙ ПРОЕКТ**

|  |  |  |
| --- | --- | --- |
| По курсу | Объектно-ориентированное программирование | |
| Студенту | Яркову М.Д. | |
|  | (фамилия, инициалы) | |
| Тема проекта: | Разработка службы для добавления расписания в календарь | | |
|  | | | |
| Исходные данные: | | Разработать приложение для ОС Android, позволяющее | |
| автоматически сохранять расписание в стандартный календарь Android и | | | |
| просматривать расписание с сортировкой в приложении. | | | |
| Рекомендуемая литература: | | | |
| 1. Фримен Э. Head First. Паттерны проектирования. Обновленное юбилейное издание. – СПб.: Питер, 2018. – 656 с. 2. Эккель Б. Философия Java. 4-е полное изд. — СПб.: Питер, 2018. — 1168 с. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Дата выдачи задания |  | 1 марта 2022 г. | |
| Задание получил студент | | |  | |  | Ярков М.Д. |
|  | | | подпись | |  | Фамилия И.О. |

|  |  |  |  |
| --- | --- | --- | --- |
| Дата представления проекта руководителю | | 23 мая 2022 г. | |
| Руководитель курсового проекта | |  | |  | | Маланова Т.В. | |
|  | подпись | |  | | Фамилия И.О. | |

Содержание

[Содержание 3](#_Toc104801226)

[Введение 4](#_Toc104801227)

[1 Анализ задания и описание предметной области 5](#_Toc104801228)

[2 Проектирование интерфейса 6](#_Toc104801229)

[3 Проектирование приложения 7](#_Toc104801230)

[3.1 Архитектура приложения 7](#_Toc104801231)

[3.2 Реализация хранения данных 7](#_Toc104801232)

[3.3 Архитектура классов 8](#_Toc104801233)

[4 Тестирование 10](#_Toc104801234)

[5 Результаты тестирования 10](#_Toc104801235)

[Заключение 14](#_Toc104801236)

[Список использованных источников 15](#_Toc104801237)

Введение

Современные ОС ушли далеко от задачи простого запуска программ: сегодня они предоставляют множество различных возможностей, направленных на улучшение взаимодействия с пользователем и дружбу приложений между собой.

Одна из таких – возможность добавления событий календаря приложениями. Достаточно предоставить разрешение, и программа добавит полезную информацию в календарь, общий для всех приложений, а также носимых устройств вроде умных часов.

Цель работы – разработка приложения для ОС Android, позволяющее автоматически сохранять расписание в стандартный календарь Android и просматривать расписание с сортировкой в приложении.

Задачи:

* Исследование предметной области;
* Обзор программных средств разработки;
* Проектирование и написание кода;
* Разработка пользовательского интерфейса.

1. Анализ задания и описание предметной области

В рамках предметной области – получение расписания в удобном виде – требуется разработать службу (приложение) для ОС Android, позволяющее:

* Автоматически сохранять расписание в стандартный календарь Android;
* Просматривать расписание с сортировкой в приложении;
* Устанавливать ссылку на расписание;
* Выбирать подгруппу для сортировки;
* Автоматически синхронизировать расписание.

1. Проектирование интерфейса

Приложение состоит из одного окна, в котором содержится расписание на одну неделю. При запуске отображается текущая неделя.

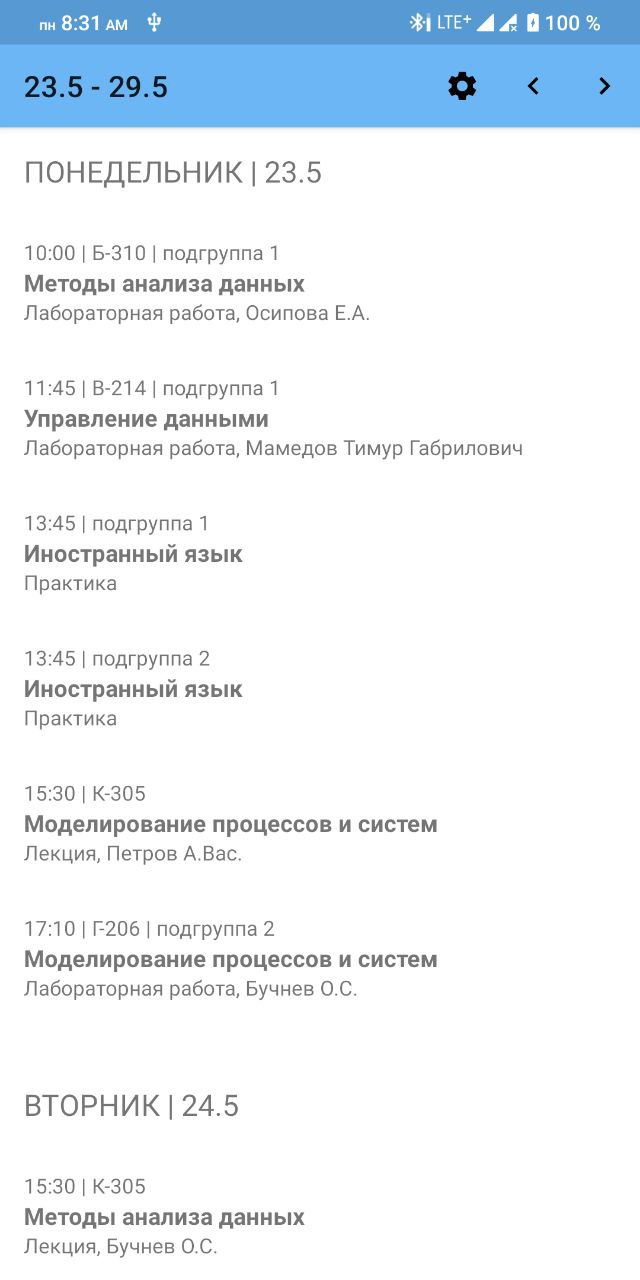


Рисунок 1 – Главное окно приложения

Кнопки навигации сверху (в виде стрелок) позволяют переходить на неделю назад и вперёд. Кнопка настроек открывает диалоговое окно, позволяющее задать группу и выбрать подгруппу для сортировки внутри приложения.

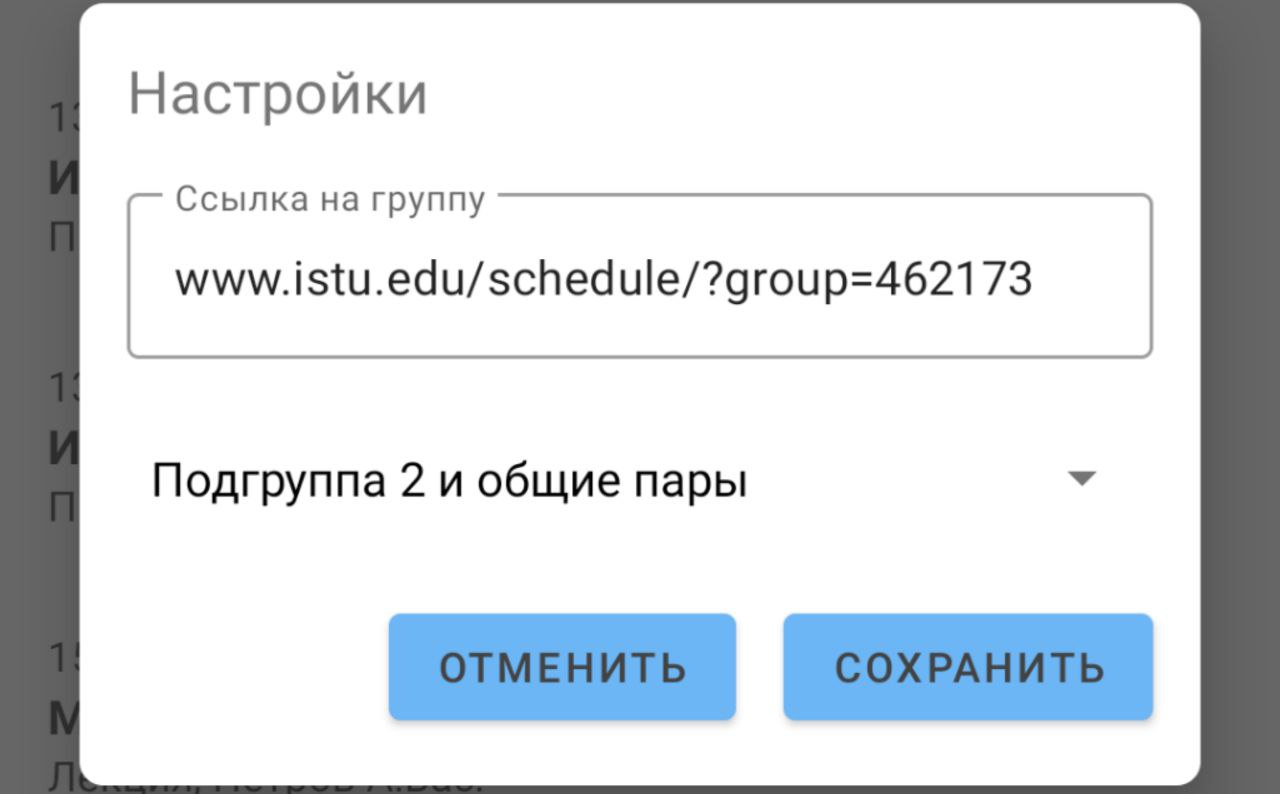


Рисунок 2 – Диалоговое окно настроек

1. Проектирование приложения

3.1 Архитектура приложения

Код приложения разделён на три части: view (работа с отображением и взаимодействие с данными и логикой), datamanager (хранение данных и их обработка), logic (основная логика приложения, а также реализация автообновления и добавления в календарь).

3.2 Реализация хранения данных

Для хранения данных приложение использует решения Android Room (база данных) и Jetpack Datastore (хранение данных ключ-значение).

База данных хранит расписание занятий, для этого используется одна таблица «lesson\_table» со следующими столбцами:

* internalWeek: Int - внутренний номер недели;
* dayOfWeek: Int - день недели, начало с 0;
* time: Int - время в минутах (8:15 соответствует 495);
* name: String - название предмета;
* type: String - тип пары (лекция/лаба/т.п.);
* subgroup: Int - подгруппа (если 0, то пара общая);
* classroomName: String - номер аудитории;
* classroomLink: String - ссылка на аудиторию;
* teacherName: String - ФИО преподавателя;
* teacherLink: String - ссылка на преподавателя.

Помимо хранения занятий, требуется хранить некоторые пары ключ-значение:

* currentWeek: Int - неделя в последний запуск проверки устаревания данных в БД;
* subgroup: Int - выбранная подгруппа;
* link: String - сохранённая ссылка на группу.

3.3 Архитектура классов

Были созданы следующие классы:

* view (работа с отображением и взаимодействие с данными и логикой)
  + MainActivity – работа с основным Activity;
  + WeekAdapter – отображение недели в RecyclerView;
  + LessonAdapter – отображение пар в RecyclerView;
* datamanager (хранение данных и их обработка):
  + ScheduleHtmlData – класс данных, предоставляющий обработанную информацию;
  + DataManager – инкапсулирует StoredDatabase и StoredVariables и предоставляет удобные методы для работы с БД и хранилищем переменных;
  + StoredVariables – реализует хранилище данных ключ-значение;
  + StoredDatabase – реализует подключение к БД при помощи LessonDao и Lesson;
  + LessonDao – реализует получение данных к БД;
  + Lesson – класс данных, предоставляющий необходимые поля таблицы БД;
* logic (основная логика приложения, а также реализация автообновления и добавления в календарь):
  + AutoUpdater – реализация автоматического обновления данных в фоне;
  + CalendarWorker – работа с календарём Android;
  + Time – работа со временем и датой;
  + Parser – получение и обработка данных извне.

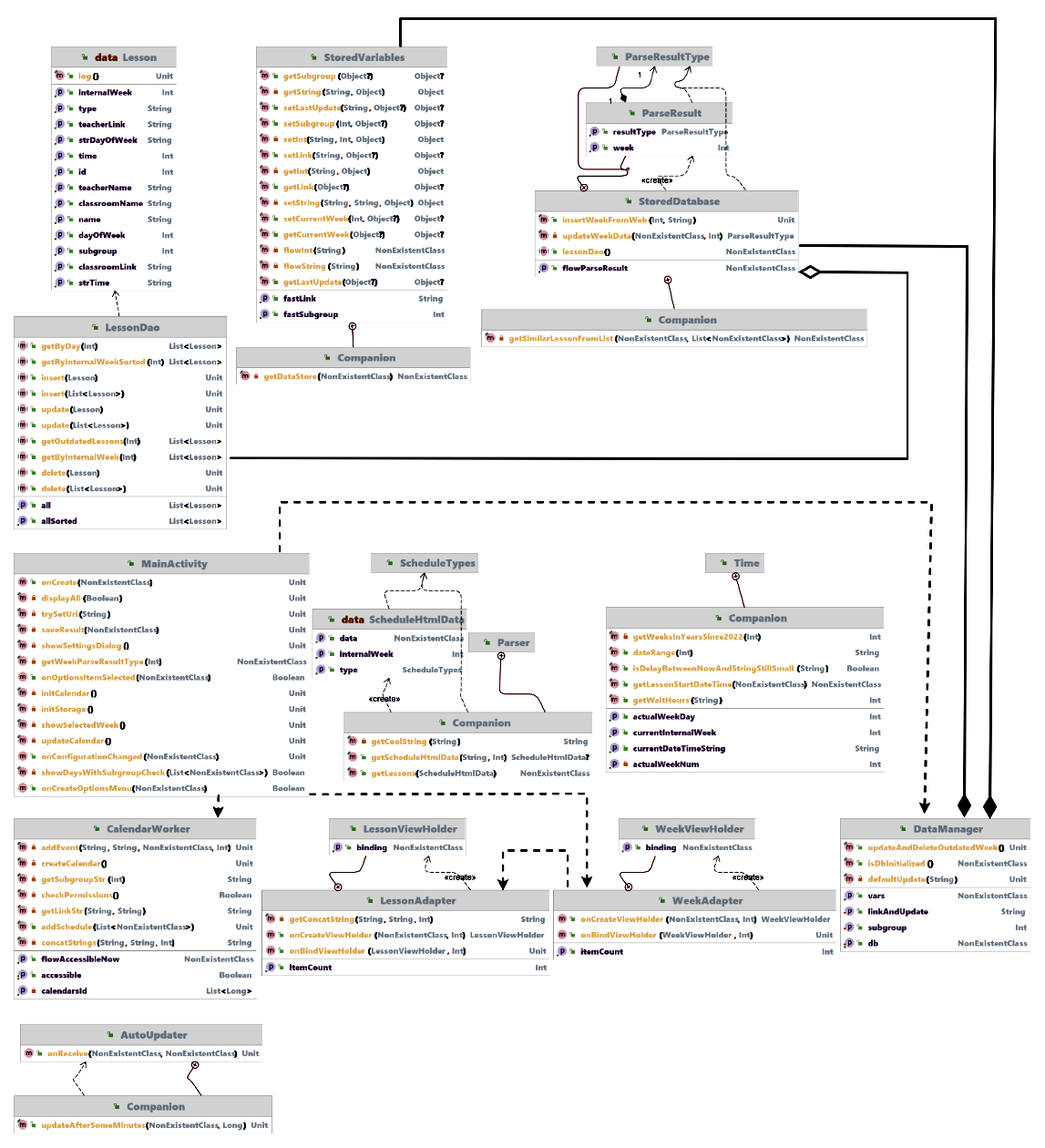


Рисунок 3 – Диаграмма классов

Начало работы программы происходит через автоматический вызов класса MainActivity, который инкапсулирует DataManager и CalendarWorker. DataManager инкапсулирует StoredDatabase и StoredVariables. В свою очередь StoredDatabase реализует LessonDao и Lesson. Остальные классы абстрагируют логику от других классов и имеют исключительно статические методы, вызываемые из разных частей по необходимости. Небольшое исключение – класс AutoUpdater, содержащий нестатический метод onReceive, вызываемой системой Android автоматически каждые 12 часов для обновления данных в БД.

1. Тестирование

Таблица 1 - Проектирование тестов

|  |  |  |
| --- | --- | --- |
| **№** | **Действие или событие** | **Результат** |
| 1 | Запуск приложения | Открытие текущей недели, добавления расписания в системный календарь |
| 2 | Отсутствие данных недели | Получение данных |
| 3 | Данные получены или данные были в БД | Вывод расписания |
| 4 | Данные не получены (расписание недоступно) | Вывод сообщения о том, что расписания нет |
| 5 | Данные не получены (возникла проблема) | Вывод сообщения с ошибкой и возможными решениями |
| 6 | Изменение недели нажатием кнопки | Изменение промежутка чисел и вывод информации |
| 7 | Установка ссылки на расписание | Вывод расписания |
| 8 | Выбор подгруппы | Вывод расписания с сортировкой |

1. Результаты тестирования

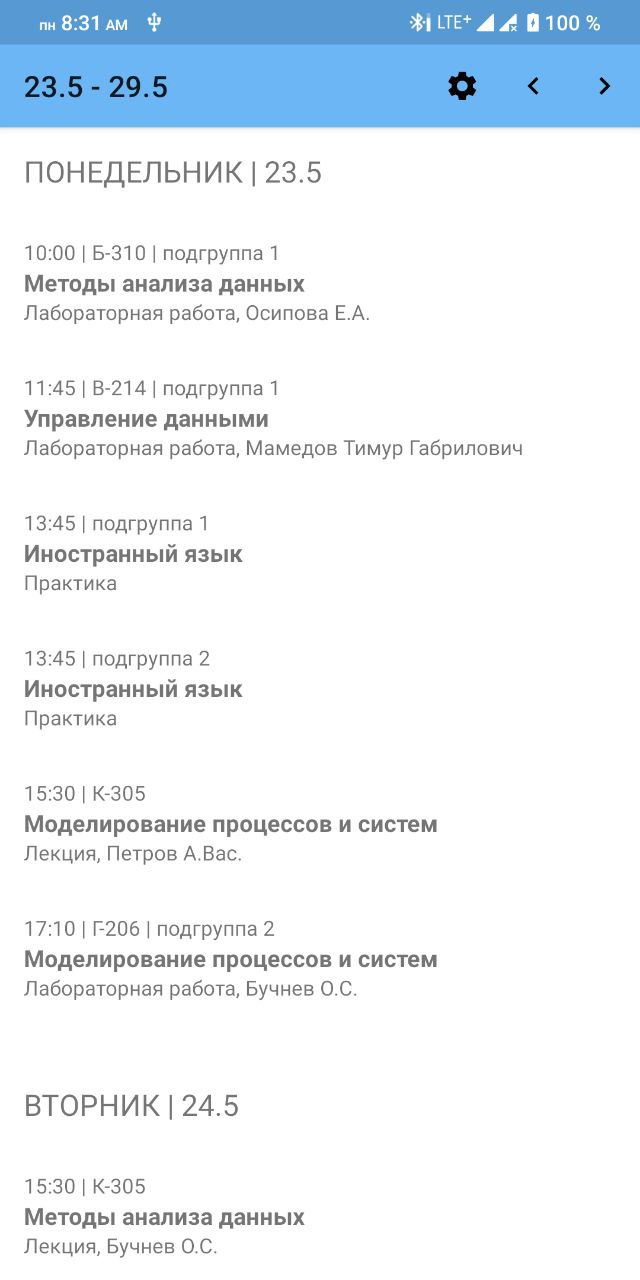


Рисунок 4 – Результаты тестов 1, 3, 7

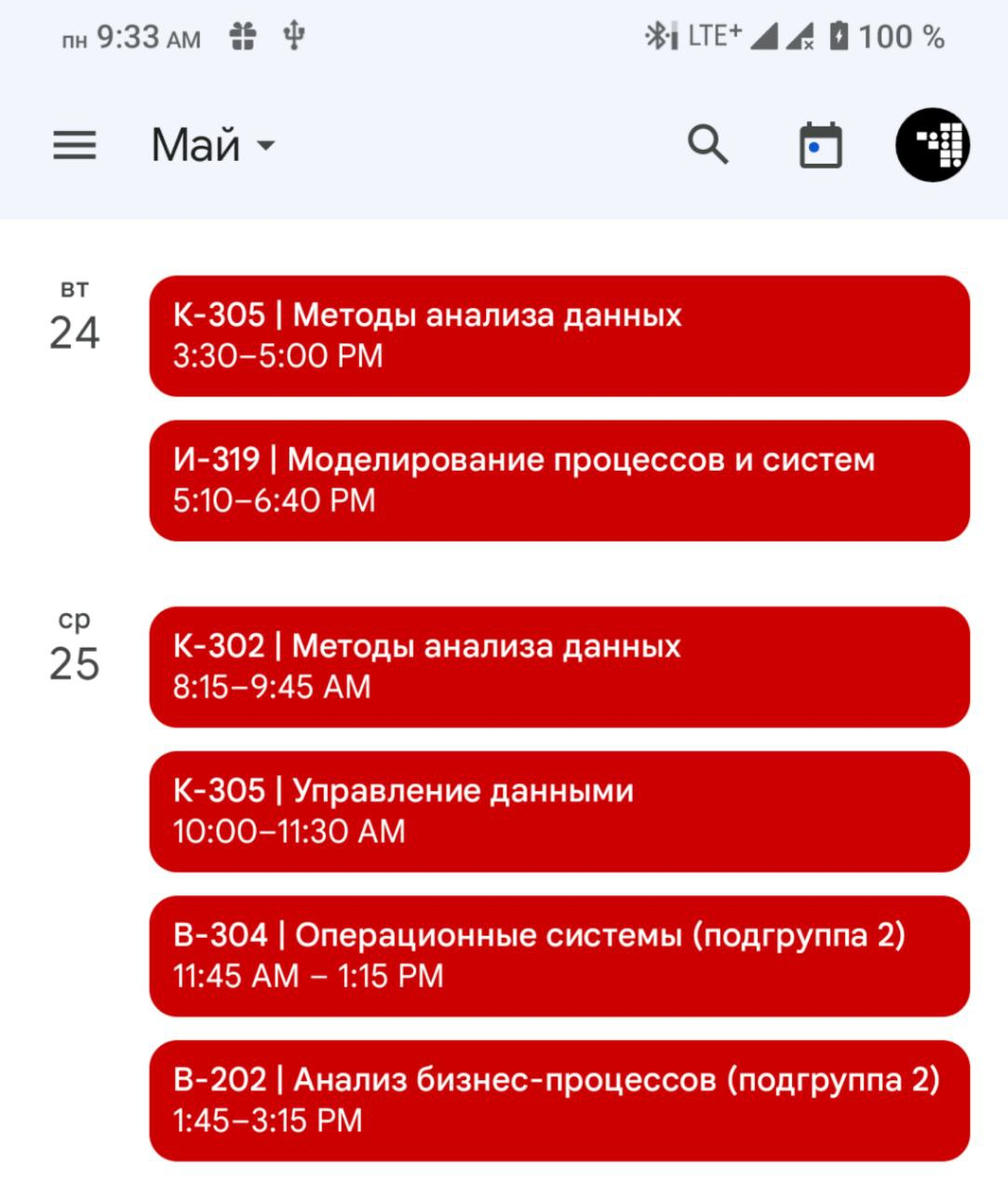


Рисунок 5 – Результаты теста 1

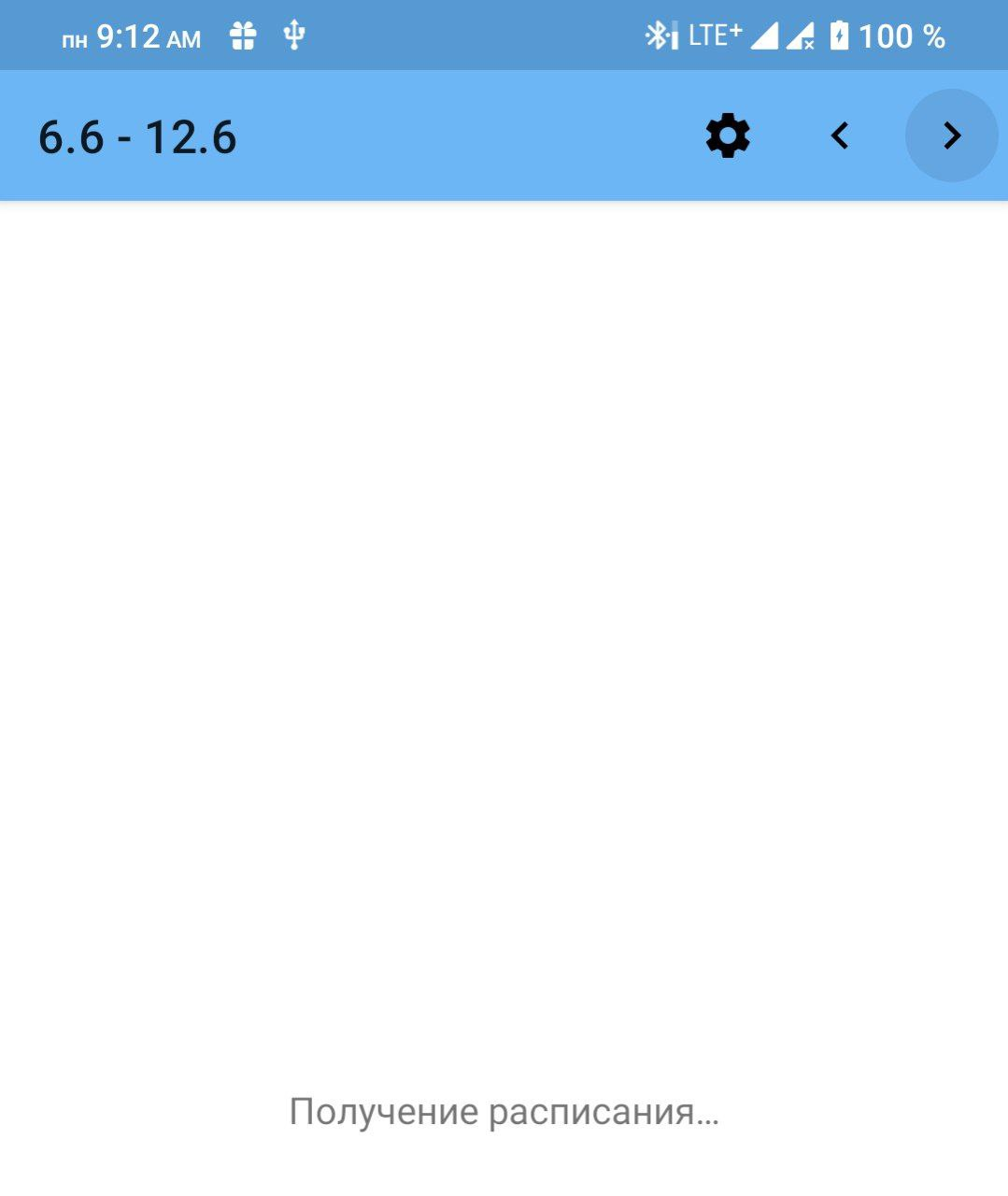


Рисунок 6 – Результаты теста 2

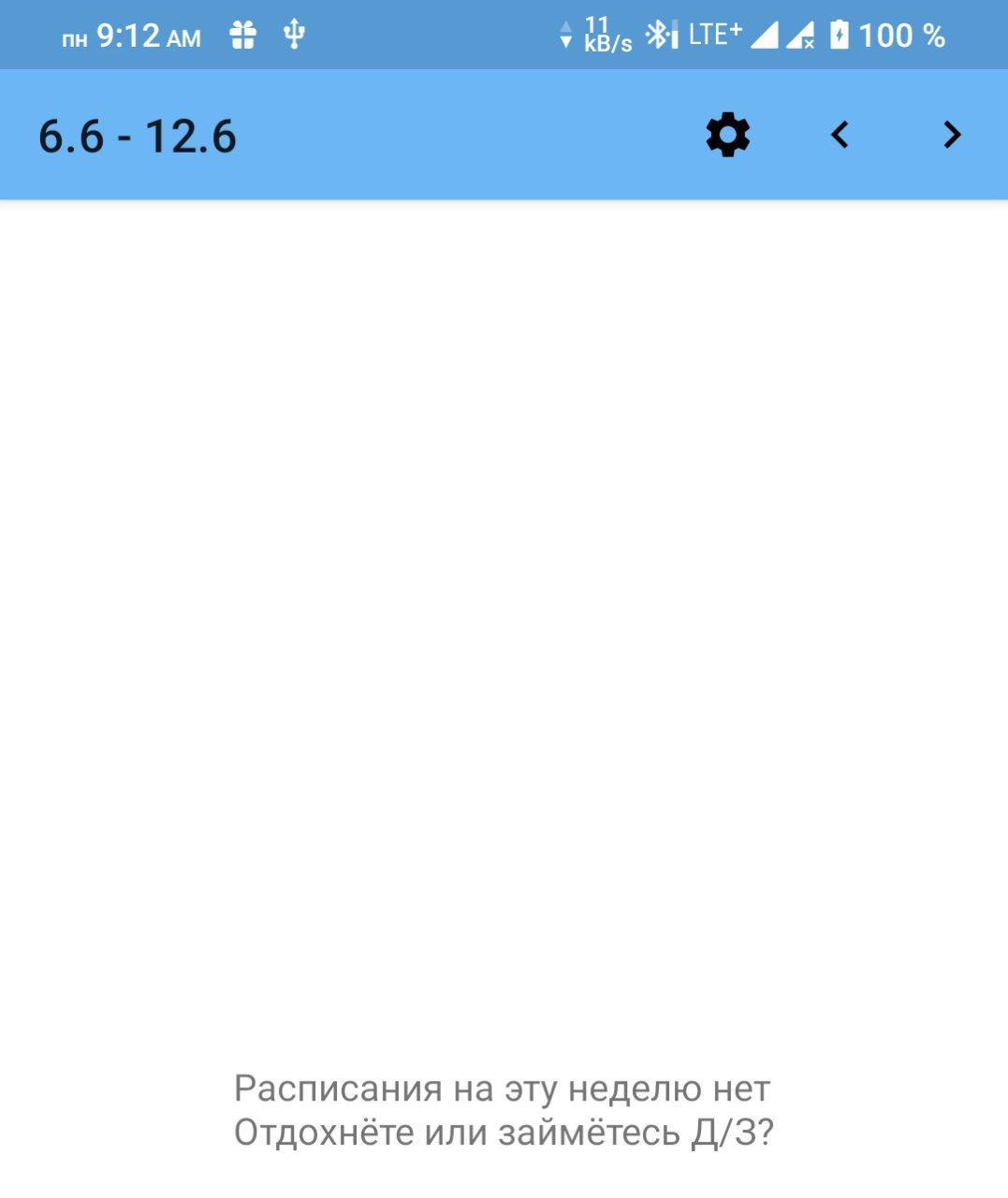


Рисунок 7 – Результаты теста 4

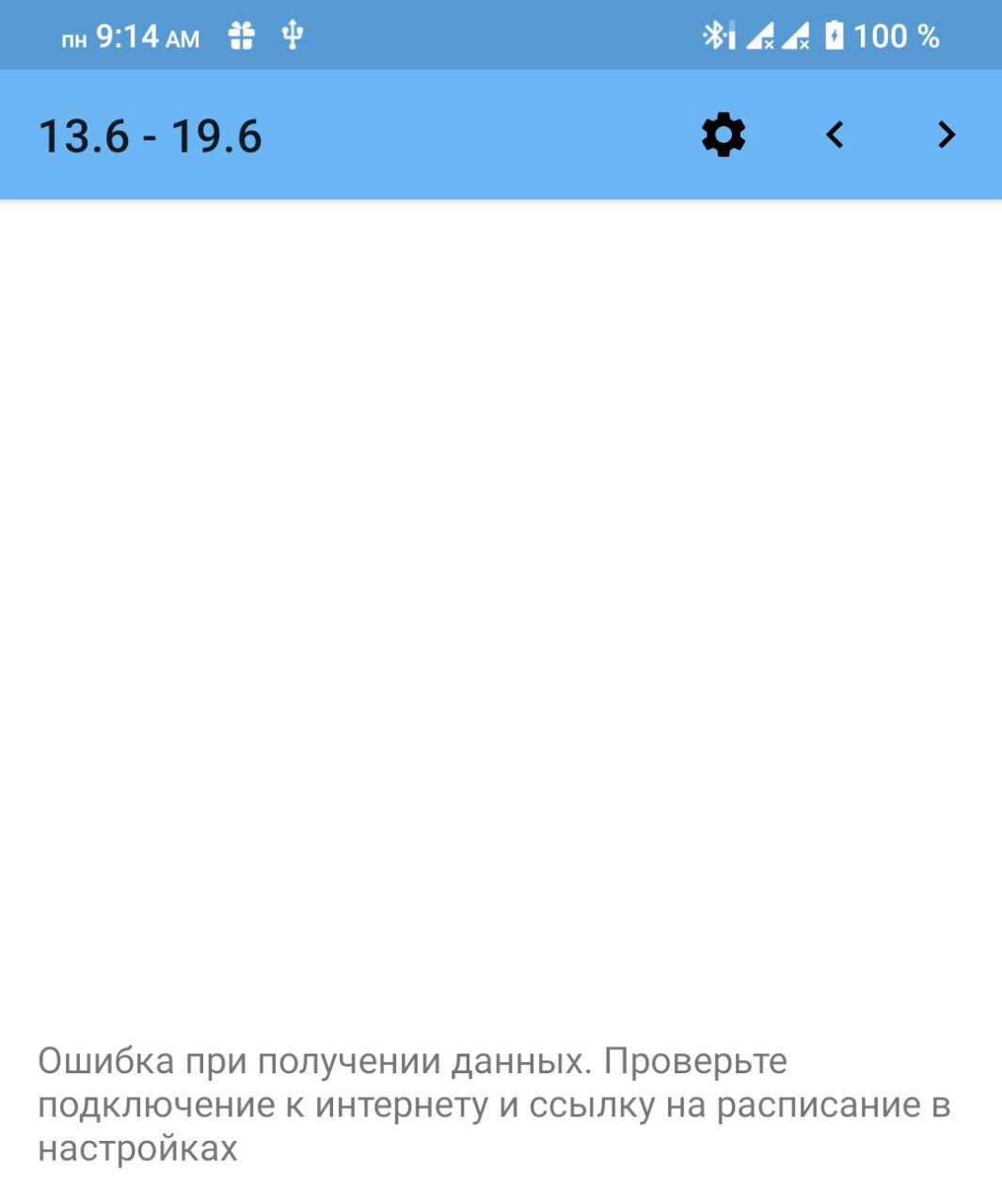


Рисунок 8 – Результаты теста 5

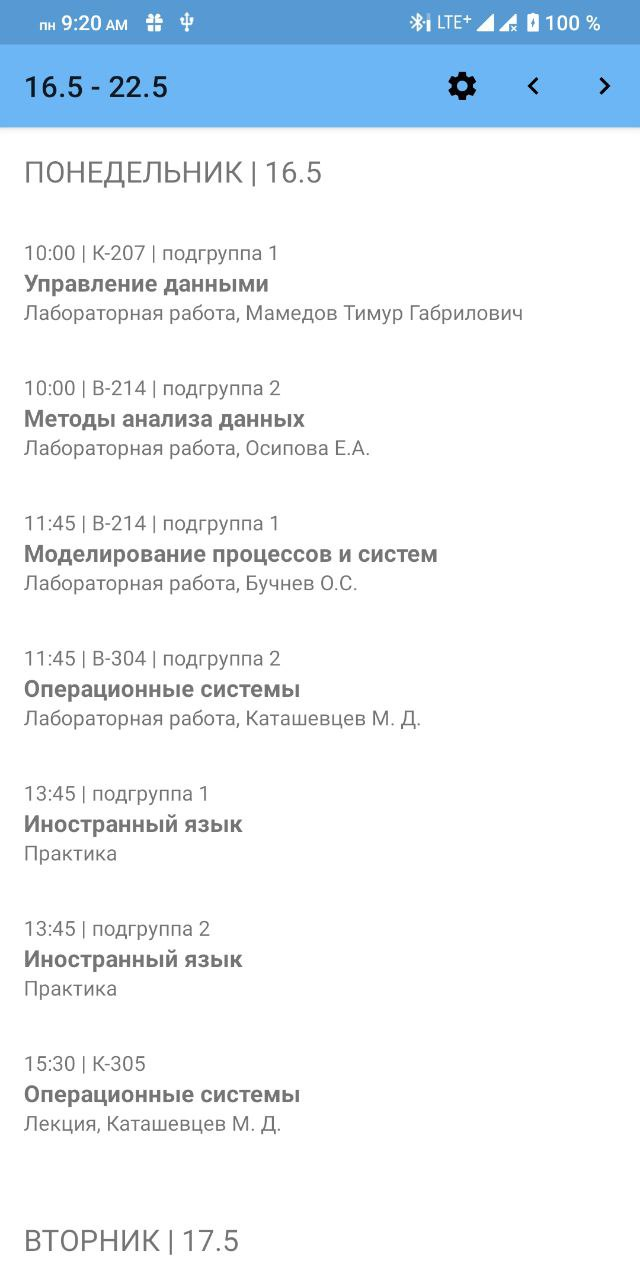


Рисунок 9 - Результаты теста 6

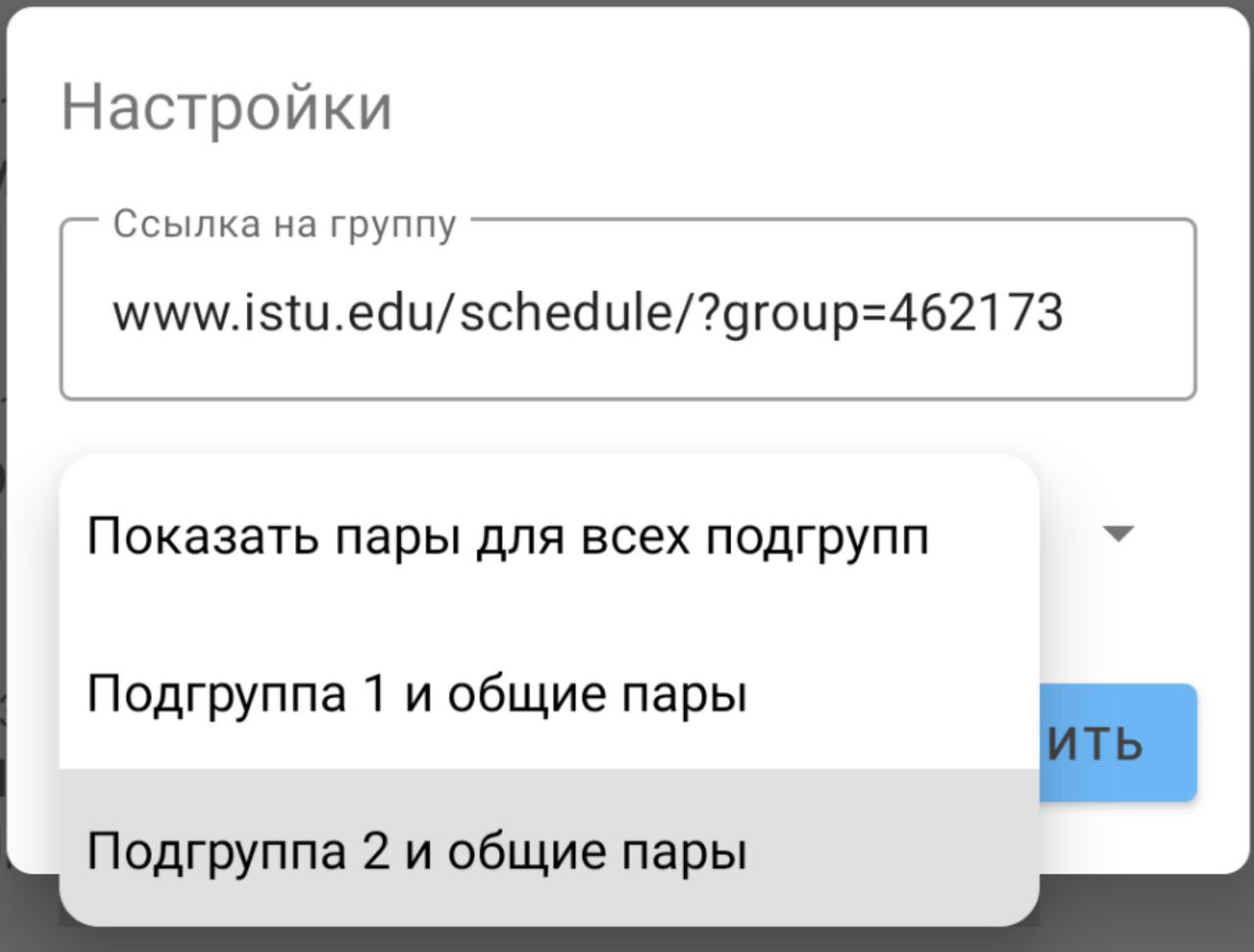


Рисунок 10 – Действие для теста 7, 8

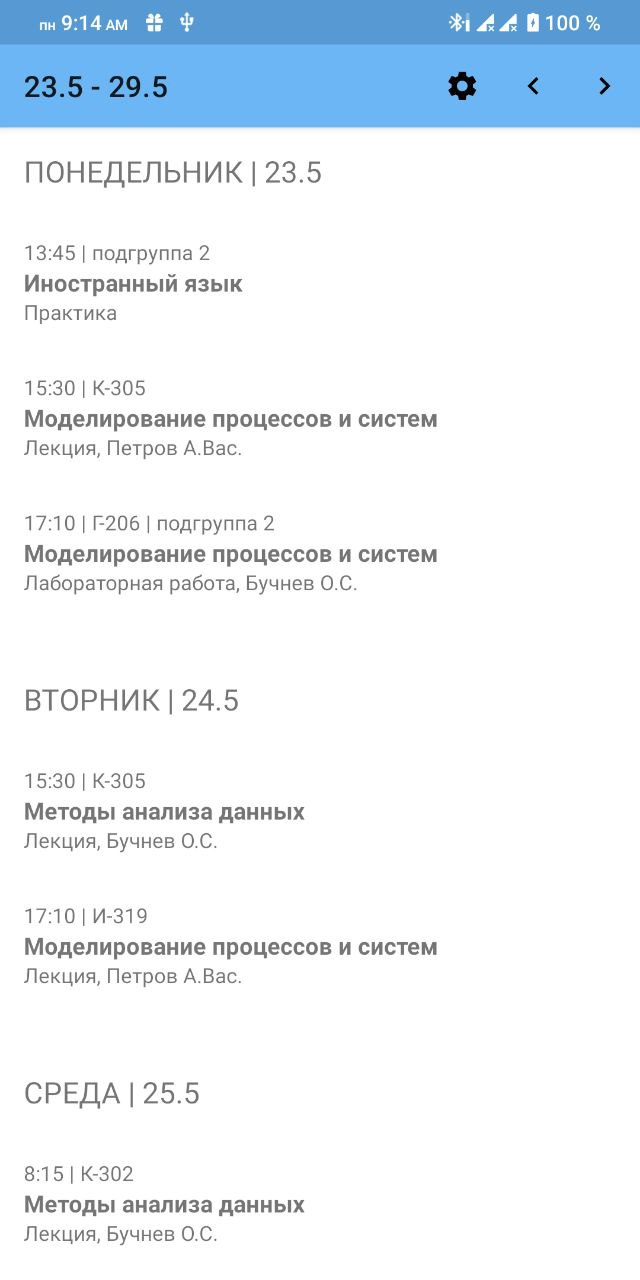


Рисунок 11 - Результаты теста 8

Заключение

В ходе выполнения курсовой работы были получены основные навыки разработки приложений для ОС Android на языке Kotlin, также были получены навыки работы с базами данных и данными ключ-значение через библиотеки Android Room и Jetpack Datastore, работы с Calendar API и возможностями BroadcastReceiever.

Приложение полностью выполняет изначально поставленные задачи и позволяет видеть расписание как внутри самого приложения, так и через системный календарь и носимые устройста.

Исходный код доступен по адресу

<https://github.com/mxkmn/AndroidSchedule>.

Список использованных источников

1. Документация Android [Электронный ресурс] // developer.android.com: Business Studio, 2022. URL: https://developer.android.com/docs (дата обращения: 27.05.2022):
2. Habr [Электронный ресурс] // habr.com: Habr. URL: <https://habr.com> (дата обращения 27.05.2022);
3. Android: Освой программирование играючи [Электронный ресурс] // developer.alexanderklimov.ru: Освой программирование играючи. URL: <http://developer.alexanderklimov.ru/android> (дата обращения 27.05.2022).