|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА***

***К КУРСОВОЙ РАБОТЕ***

***НА ТЕМУ:***

***Прототип ANDROID-приложения «Помощник учителя»***

Студент \_\_ИУ6-52Б\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_И.С.Марчук\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Фетисов М.В.\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

*2021 г.*

**Реферат**

Расчетно-пояснительная записка 44 страницы, 3 части, 36 рисунков, 3 таблицы, 6 источников, 2 приложения.

ANDROID-ПРИЛОЖЕНИЕ, УЧЕБНЫЙ ПРОЦЕСС, ПРЕПОДАВАТЕЛЬ, ЗАМЕТКИ, УСПЕВАЕМОСТЬ, ОЦЕНКИ.

Объектом разработки является android-приложение для учета учителем успеваемости учеников, «Помощник учителя».

Цель работы – проектирование и реализация android-приложения для фиксации количества ответов учащегося, при помощи виртуальных классов, создания расписания, а также для просмотра и сравнения результатов вне урока.

В результате работы был спроектировано приложение, позволяющее преподавателю во время урока быстро, не отвлекаясь на лишние действия отмечать ответы учащихся с помощью выводящихся на экран парт и учеников. А после урока, просматривать результаты работы учеников и выставлять соответствующие оценки. А также было проведено тестирование программного продукта.

Пользователями данного приложения могут быть как учителя начального и среднего, так и высшего уровней образования.

**Содержание**

[**Введение** 5](#_Toc90601770)

[**1.** **Анализ требований и уточнение спецификаций** 6](#_Toc90601771)

[**1.1.** **Анализ задания и выбор технологии, языка и среды разработки** 6](#_Toc90601772)

[**1.2.** **Разработка диаграммы вариантов использования** 7](#_Toc90601773)

[**1.3** **Анализ хранимой информации и выбор способа ее хранения** 10](#_Toc90601774)

[**2** **Проектирование структуры и компонентов программного продукта** 13](#_Toc90601775)

[**2.1** **Разработка структуры и компонентов приложения** 13](#_Toc90601776)

[**2.1.1** **Разработка структурной схемы приложения** 13](#_Toc90601777)

[**2.1.2** **Разработка структурной карты константайна** 14](#_Toc90601778)

[**2.2** **Разработка интерфейса пользователя** 15](#_Toc90601779)

[**2.2.1** **Построение диаграммы состояний интерфейса** 15](#_Toc90601780)

[**2.2.2** **Разработка форм интерфейса** 17](#_Toc90601781)

[**2.3** **Локализация** 36](#_Toc90601782)

[**3.** **Выбор стратегии тестирования и разработка тестов** 37](#_Toc90601783)

[**3.1 Функциональное тестирование** 37](#_Toc90601784)

[**3.2 Оценочное тестирование** 40](#_Toc90601785)

[**Заключение** 41](#_Toc90601786)

[**Список используемых источников** 42](#_Toc90601787)

[**Приложение А.** 43](#_Toc90601788)

[**Приложение Б.** 44](#_Toc90601789)

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ТЗ – техническое задание;

ООП – объектно-ориентированное программирование;

БД – база данных;

СУБД – система управления базами данных;

SQLite3 – Встраиваемая СУБД;

Java – строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Oracle;

Android – операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, фитнес-браслетов, игровых приставок, ноутбуков, нетбуков, смартбуков, очков Google Glass, телевизоров, проекторов и других устройств;

Android SDK – универсальное средство разработки мобильных приложений для операционной системы Android;

ANDROID-приложение – рограммное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы (в данном случае android);

Активность (англ. activity) – совокупность программной логики и XML разметки, относящийся к одному экрану в системе android;

\_id – уникальный идентификатор записи в БД;

Shared preferences – настройки хранящиеся в xml формате внутри приложения.

XML – расширяемый язык разметки.;

Виртуальный класс – оцифрованные в андроид приложении данные об классе, учениках в нем и их оценках;

Виртуальный кабинет – Графическое представление на экране реального кабинета с партами.

# **Введение**

Данная работа посвящена проектированию и разработке прототипа android-приложения «Помощник учителя» (помощник учителя). Прототип предназначен для школьных учителей с целью упростить процесс ведения заметок об успеваемости учеников на уроке.

Актуальность проекта оправдана, поскольку обычные заметки отнимают у учителя слишком много времени и внимания от урока, а система android стоящая на большинстве мобильных устройств позволяет делать достаточно гибкий интерфейс, превосходящий по функционалу обычные заметки, и не отнимающий много внимания пользователя.

Из аналогов на данный момент существует испанское android-приложение «Учительский Журнал» [1], которое реализует большую часть необходимых функций, однако имеет высокий порог вхождения и плохую реализацию интерфейса некоторых функций. Исходя из этого было принято решение создать замещающий аналог с более понятным интерфейсом.

# **Анализ требований и уточнение спецификаций**

## **Анализ задания и выбор технологии, языка и среды разработки**

Методологией программирования было выбрано объектно-ориентированное программирование. ООП является основой всех современных приложений и имеет удобное и практическое применение. Она основана на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса.

Для разработки приложения была выбран комплект средств разработки Android SDK, среда Android Studio и язык Java 8. Java – объектно-ориентированный язык программирования, который составляет основную часть разработки под OS Android. Так, как программа должна иметь привычный интерфейс для диалога с пользователем, выбор остановился на среде разработки Android Studio. Компания Google, являющаяся владельцем платформы Android, непрерывно поддерживает эту среду [2], выпуская новые библиотеки и плагины. Библиотеки Android SDK обладают огромным потенциалом и предоставляют разработчику широкий спектр инструментов для разработки. Основным достоинством Android Studio является то, что она используется в качестве посредника между пользователем, консолью ОС и файлами проекта, переводя это в обработку событий, что сильно облегчает работу разработчика, освобождая его от прямого контакта с сообщениями системы. Ещё одним достоинством Android Studio является возможность визуализированной работы с пользовательским интерфейсом, наличие удобных средств уведомления и логирования что позволяет увидеть результаты ещё на стадии разработки.

При проектировании использовался объектно-ориентированный подход, это позволяет разделить части приложения на отдельные составляющие со своими характеристиками и методами, особенно это касается разработки под Android, там каждая активность представляет собой отдельную часть приложения и для их создания и связи между собой необходимо использование ООП.

В качестве хранилища данных приложения была выбрана база данных. А в качестве СУБД я решил использовать легкую встраиваемую библиотеку SQLite3 [3]. Эта библиотека входит в стандартный пакет инструментов разработки, а это значит, что она уже настроена и готова для использования в коде.

## **Разработка диаграммы вариантов использования**

Первый шаг проектирования android-приложения – определение основных вариантов его использования. Для этого была разработана диаграмма вариантов использования, отображающая основные варианты взаимодействия приложения и пользователя.

После анализа ТЗ были получены следующие варианты взаимодействия пользователя и приложения:

* ввод данных об учениках, которых будет оценивать преподаватель (Название класса, Фамилия имя отчество);
* ввод данных о рассадке учеников в классе (расположение парт в кабинете и количество мест на них, расположение виртуальных учеников в виртуальном кабинете);
* ввод данных об успеваемости учеников (отсутствие учеников, успеваемость учеников на уроке);
* просмотр данных об успеваемости с сортировкой учеников по фамилии;
* редактирование введенных данных (изменение и удаление);
* расчет и вывод средних оценок учеников и количества пропусков за период.

Рассмотрим самый распространенный вариант использования приложения.

Таблица 1 – описание вариантов использования ***Выставление оценок учеников на уроке***

|  |  |
| --- | --- |
| Название варианта | Выставление оценок учеников на уроке |
| Цель | Выставление оценок ученикам по количеству ответов на уроке |
| Действующие лица | Пользователь |
| Краткое описание | Пользователь открывает урок в расписании, заходит в него, отмечает присутствующих учеников.  Во время урока, если ученик отвечает, учитель добавляет ему один ответ в приложении.  После урока учитель заменяет проставленные баллы на оценку |
| Тип | Основной |

Таблица 2 – Вариант использования ***Выставление оценок учеников на уроке***

|  |  |
| --- | --- |
| Действие пользователя | Отклик системы |
| 1. Пользователь открывает расписание в приложении   3. Пользователь выбирает нужный урок.  5. Во время урока пользователь проставляет отсутствующих учеников и баллы для присутствующих.  7. Пользователь нажимает кнопку «закончить урок»  9. Пользователь меняет проставленные баллы на оценки  11. пользователь нажимает кнопку «сохранить» | 1. Приложение загружает из БД сведения об уроках за отображаемый период и выводит их   4. Приложение загружает из БД сведения о классе и кабинете входящих в урок, данные об учениках и партах, а также о рассадке учеников за партами. А затем вывод всю информацию на экран.  6. Приложение запоминает состояние учеников и их баллов, а также обновляет состояние интерфейса  8. Приложение выводит учеников и их баллы списком с возможностью редактирования баллов.  10. Приложение запоминает состояние учеников и их баллов, а также обновляет состояние интерфейса  12. Приложение сохраняет все оценки в БД, а также закрывает активность урока. |

Разработанная диаграмма вариантов использования представлена на рисунке 1.



Рисунок 1 – Диаграмма вариантов использования

## **Анализ хранимой информации и выбор способа ее хранения**

При анализе предметной области и технического задания была выявлена следующая информация для хранения:

— Класс:

— \_id;

— название.

— Ученик:

— \_id;

— Имя;

— Фамилия;

— \_id класса.

— Предмет:

— \_id;

— Название;

— \_id класса.

— Ученик-оценка:

— \_id;

— Оценка;

— \_id предмета;

— \_id ученика;

— время.

— Кабинет:

— \_id;

— Название.

— Предмет-кабинет-время(Урок):

— \_id;

— \_id Кабинета;

— \_id предмета;

— дата начала урока;

— дата конца урока;

— повторения.

— Парты:

— \_id;

— положение парты по оси x;

— положение парты по оси y;

— число мест;

— \_id кабинета.

— Места:

— \_id;

— \_id парты;

— порядковый номер места.

— Ученик-место:

— \_id;

— \_id ученика;

— \_id места.

— Профиль настроек:

— \_id;

— название;

— размер интерфейса.

В данном списке информация уже сгруппирована по смысловым группам, на основе которых будут строиться таблицы в базе данных.

Схема базы данных приведена на рисунке номер 2.



Рисунок 2 – Инфологическая модель базы данных

# **Проектирование структуры и компонентов программного продукта**

## **Разработка структуры и компонентов приложения**

### **Разработка структурной схемы приложения**

Основными сущностями в android-приложениях являются активности – модуль включающий в себя всю логику одного экрана. Поэтому при проектировании приложение было разбито на составляющие его экраны, каждый из которых выполнял свою часть строго определенного функционала.

Например:

* модуль расписания – включает в себя просмотр расписания уроков с возможностью редактирования и проведения каждого из этих уроков;
* модуль управления классами – содержит всю информацию об ученических классах, самих учениках и их оценках;
* модуль управления кабинетами – позволяет просматривать и редактировать кабинеты, в том числе редактирование расстановки парт в виртуальном кабинете;
* Редактор настроек – создание этого модуля позволило объединить все общие настройки приложения в одной сущности.

Также каждый из этих модулей можно разделить на отдельные подструктуры, что показано на структурной схеме приложения, представленной на рисунке 3.



Рисунок 3 – Структурная схема приложения

### **Разработка структурной карты константайна**

Методика структурных карт используется чтобы продемонстрировать, каким образом программный продукт выполняет системные требования. Для описания отношений между модулями я использовал структурную карту Константайна (рисунок 4).



Рисунок 4 – Структурная карта Константайна

Особо хочется отметить, что благодаря карте было сделано решение, что хранение общих настроек приложения можно вынести отдельно от базы данных во встроенное хранилище shared preferences, которое работает быстрее, чем СУБД, однако не позволяет содержать в себе слишком сложные структуры данных.

## **Разработка интерфейса пользователя**

### **Построение диаграммы состояний интерфейса**

Исходя из требований к пользовательскому интерфейсу и проектных решений, принятых выше, составим диаграмму состоряний интерфейса (рисунок 5).

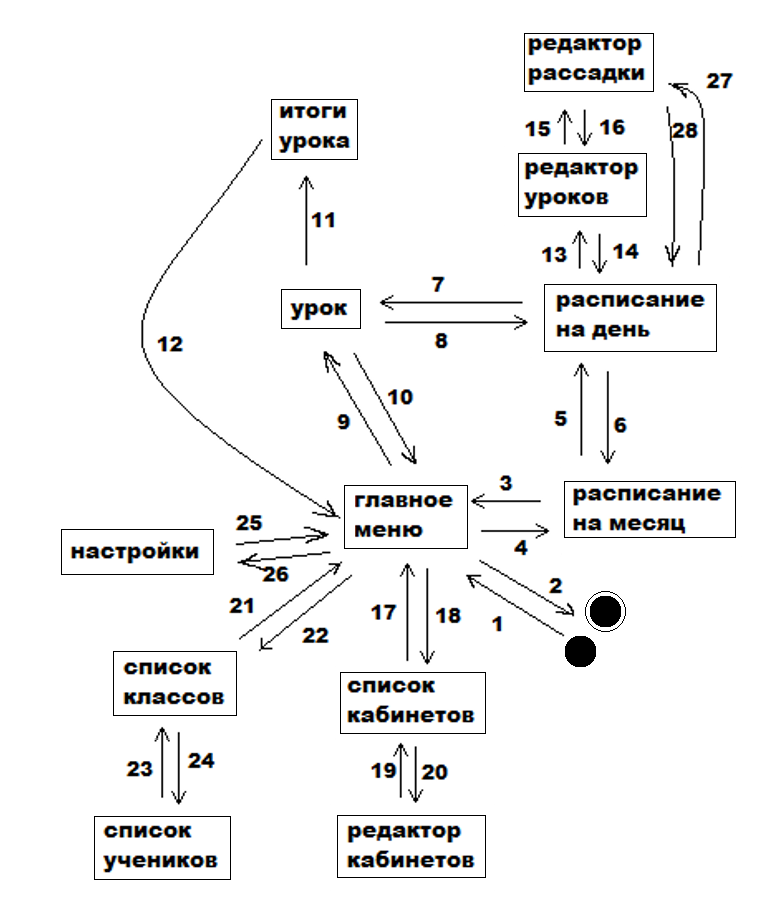
****

Рисунок 5 – Диаграмма состояний интерфейса приложения

На диаграмме приняты следующие обозначения:

1. – запуск приложения;
2. – нажатие кнопки «назад»;
3. – нажатие кнопки «назад»;
4. – выбор пункта «моё расписание»;
5. – выбор дня;
6. - нажатие кнопки «назад»;
7. – выбор урока;
8. - нажатие кнопки «назад»;
9. - выбор пункта «текущий урок» (при наличии урока на текущее время);
10. - нажатие кнопки «назад»;
11. - нажатие кнопки «закончить»;
12. - нажатие кнопки «сохранить» или «назад» (без сохранения результатов);
13. – нажатие на пункт списка без урока или долгое нажатие на урок;
14. - нажатие кнопки «назад»;
15. - нажатие кнопки «редактировать рассадку»;
16. - нажатие кнопки «назад»;
17. - нажатие кнопки «назад»;
18. - выбор пункта «мои кабинеты»;
19. - нажатие кнопки «назад»;
20. - выбор кабинета из списка;
21. - нажатие кнопки «назад»;
22. - выбор пункта «мои ученики»;
23. - нажатие кнопки «назад»;
24. - выбор класса из списка;
25. - нажатие кнопки «назад»;
26. - выбор пункта «настройки»;
27. - выбор урока (если ученики не рассажены в классе где проходит урок);
28. - нажатие кнопки «назад»;

На основе этой диаграммы в дальнейшем был спроектирован пользовательский интерфейс.

### **Разработка форм интерфейса**

Так как приложение разрабатывалось для интеграции прямо в рабочий процесс учителя, основной задачей стало создать максимально удобный интерфейс.

При входе в приложение пользователю предлагается главное меню с навигацией по всему приложению:

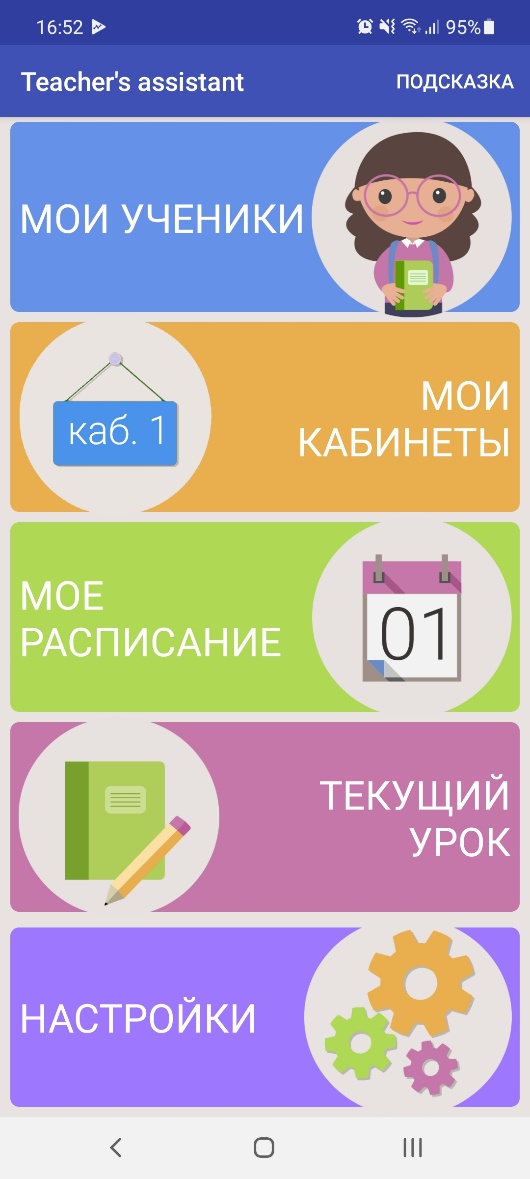


Рисунок 6 – главное меню

Из главного меню пользователю доступны «текущий урок», «расписание», «список кабинетов», «список классов», «настройки». Рассмотрим «список кабинетов»:

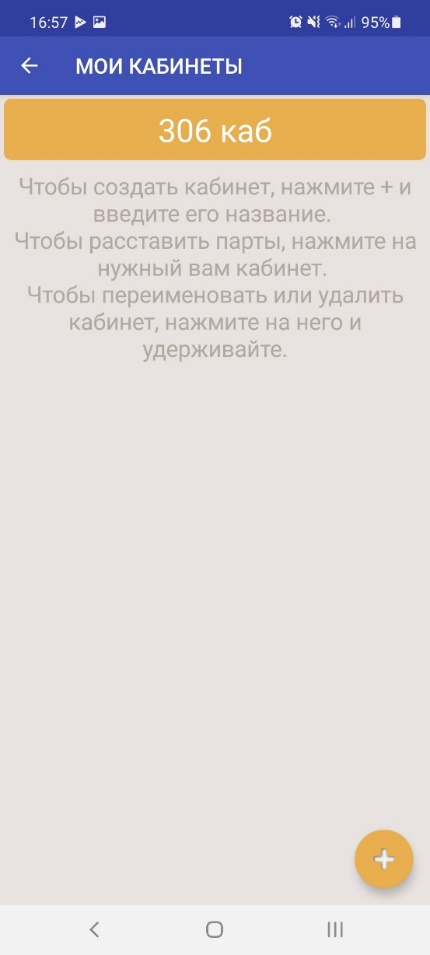


Рисунок 7 – список кабинетов

Список построен на ListView. В правом нижнем углу находится кнопка для добавления кабинета, при нажатии на неё открывается диалог «создание кабинета» в котором пользователю нужно ввести название нового кабинета.

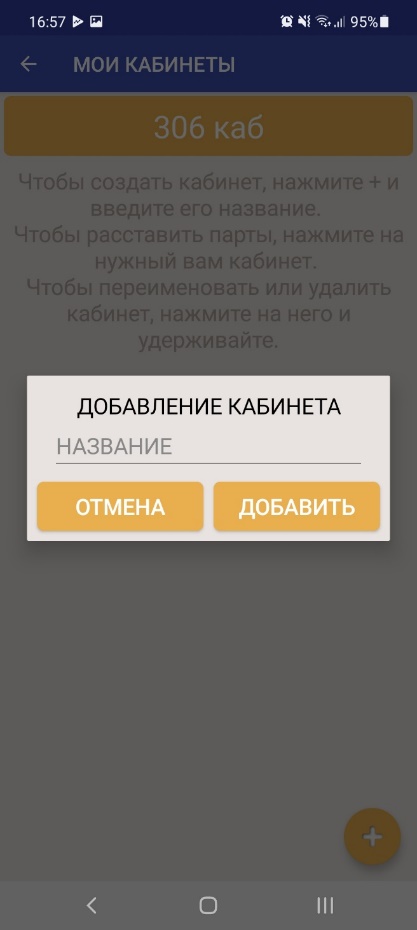


Рисунок 8 – список кабинетов – диалог создания

При длинном нажатии на пункт списка открывается дополнительное меню. Кнопка «редактировать» открывает диалог редактирования имени кабинета.



Рисунок 9 – список кабинетов – диалог редактирования имени

Нажатие на кнопку «удалить», удаляет выбранные кабинеты.

Нажатие на кабинет из списка, открывает активность редактирования кабинета.



Рисунок 10 – редактирование кабинета

При нажатии на кнопку «добавить» в центре экрана появляется новая парта.

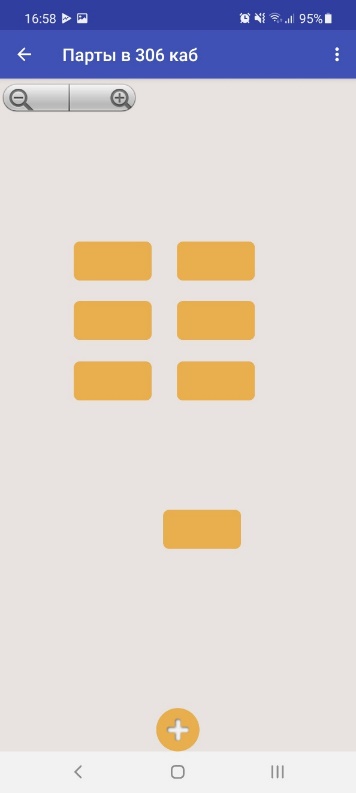


Рисунок 11 – редактирование кабинета – новая парта

При нажатии удерживании парты, её можно перемещать по экрану, при этом кнопка «добавить» будет меняться на кнопку «удалить».



Рисунок 12 – редактирование кабинета – перемещение парты

Если переместить парту к кнопке «удалить», парта удаляется с экрана и из БД, а кнопка «удалить» заменится на кнопку «добавить».

Рассмотрим список классов.

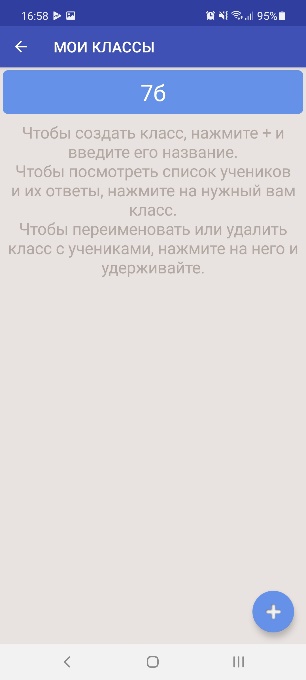


Рисунок 13 – список классов

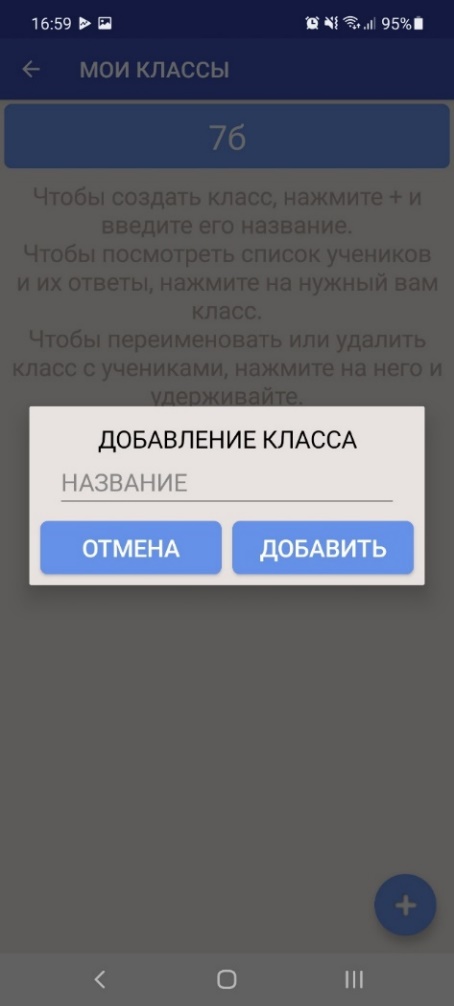


Рисунок 14 – список классов (диалог создания такой-же что и в предыдущем примере)

Интерфейс аналогичен интерфейсу списка кабинетов, но при нажатии на пункт списка мы попадаем в список учеников в выбранном классе.

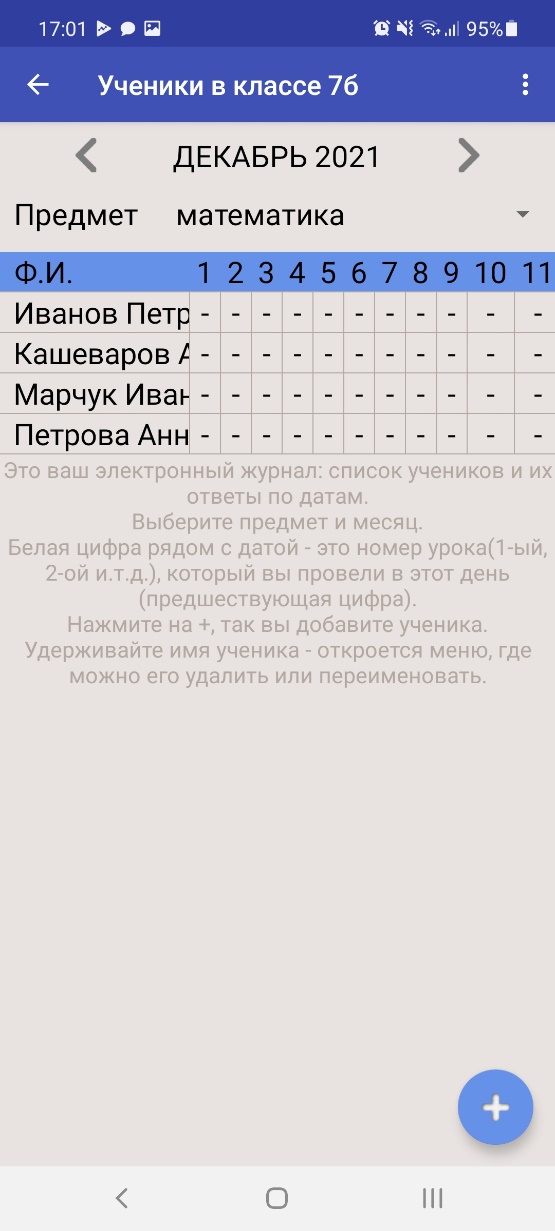


Рисунок 15 – список учеников

В этом списке интерфейс схож с двумя предыдущими, но в диалоге теперь два поля: для имени и для фамилии.

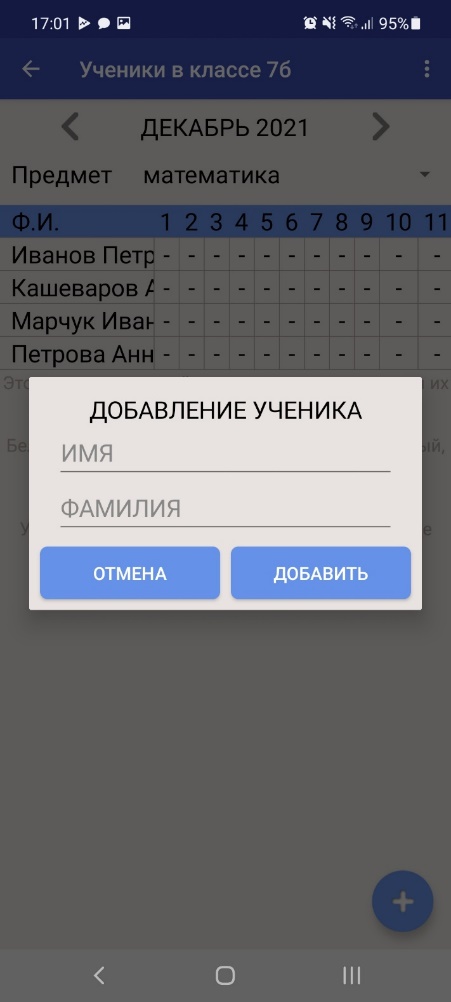


Рисунок 16 – список учеников – диалог создания

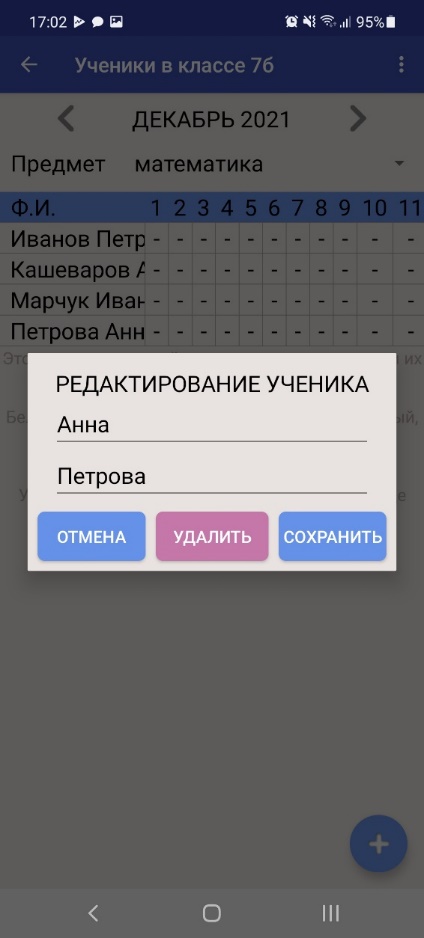


Рисунок 17 – список учеников – диалог редактирования

Рассмотрим раздел «моё расписание».

В нем выводится календарь на текущий месяц, красным цветом обозначается текущий день, а тёмно-зелёным, день когда есть уроки.

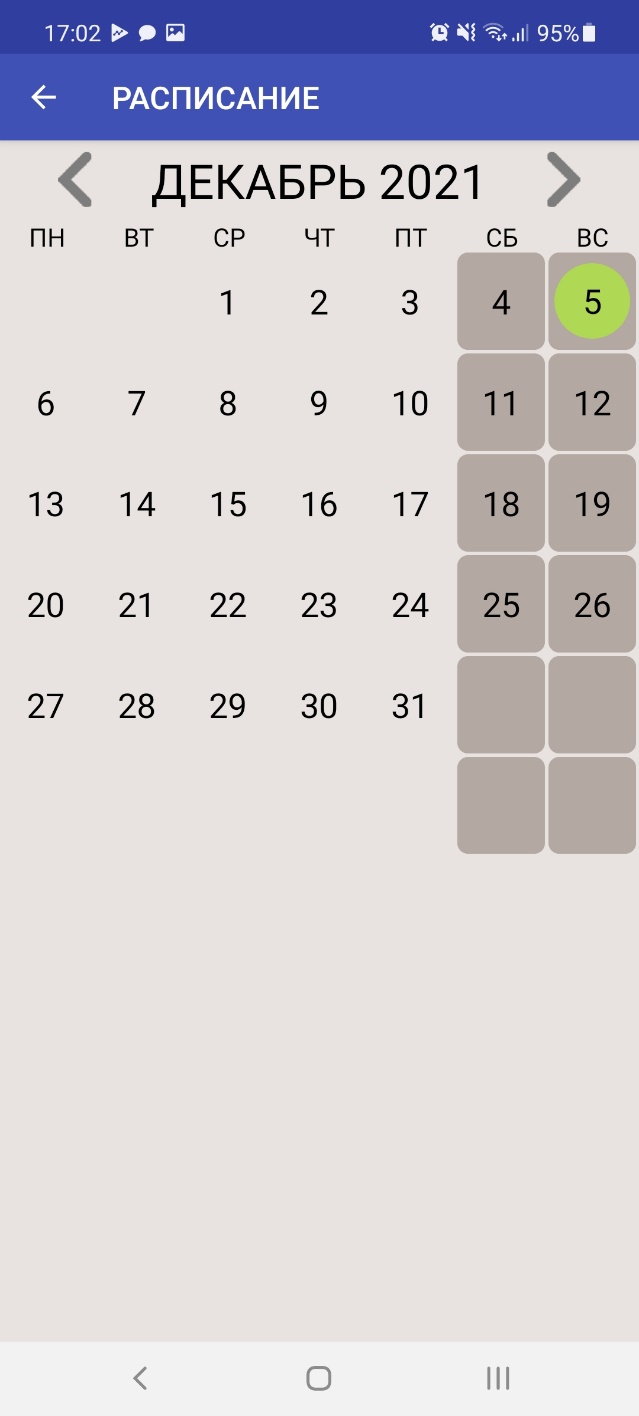


Рисунок 18 – расписание на месяц

При помощи кнопок «назад» и «вперёд» или при помощи перелистывания, можно перемещаться между месяцами.

При нажатии на день открывается расписание на этот день.

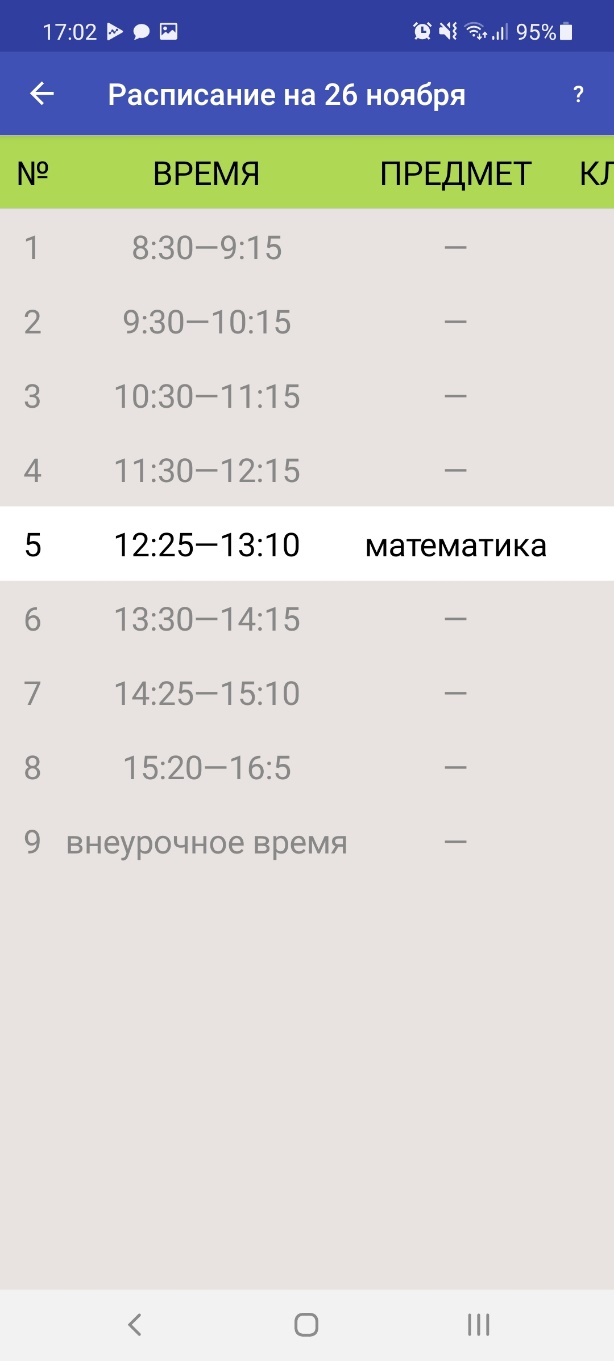


Рисунок 19 - расписание на день

При нажатии на пункт без урока или удерживании пункта с уроком открывается редактор урока.

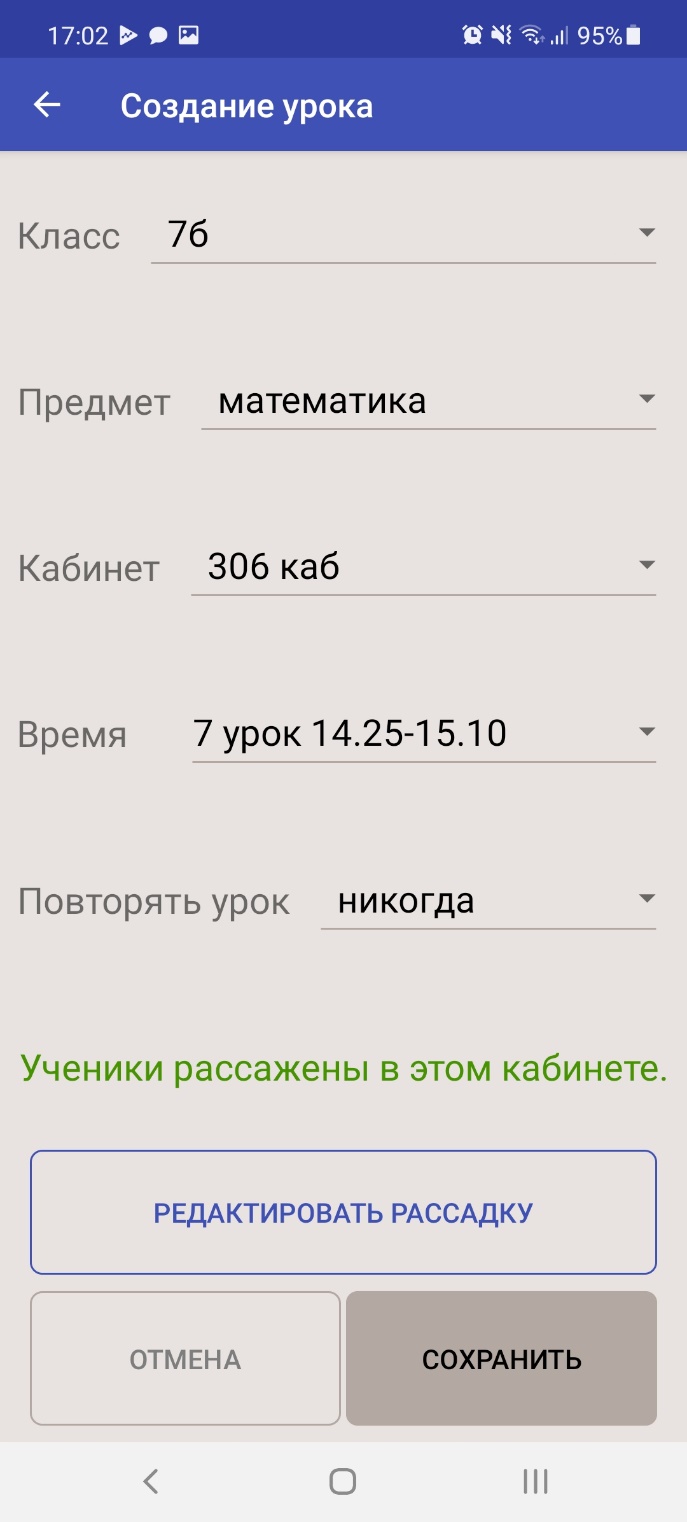


Рисунок 20 – редактор урока

В выпадающих списках можно настроить параметры урока,

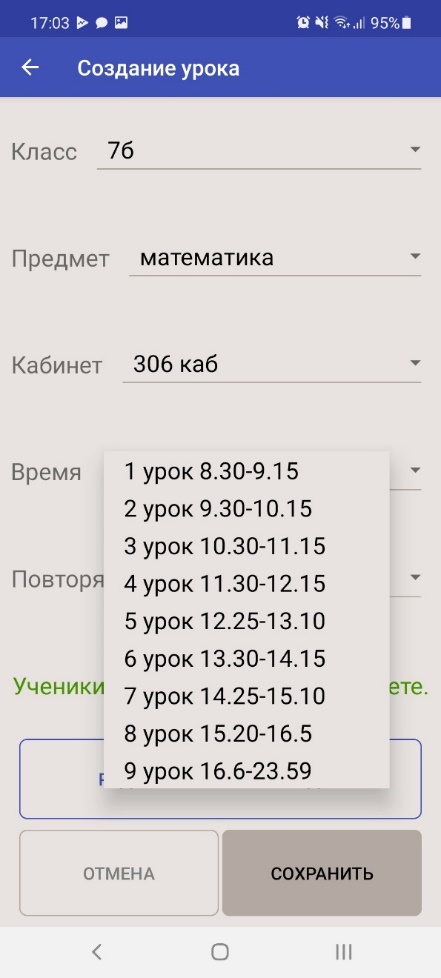


Рисунок 21 – редактор урока – список времени

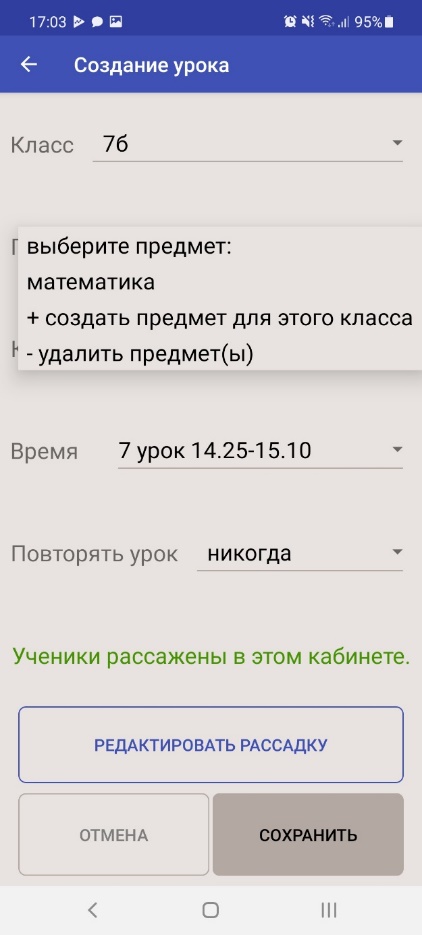


Рисунок 22 – редактор урока – список предметов

В списке «предметы» пункт «создать предмет» открывает диалог, в котором надо ввести имя предмета.

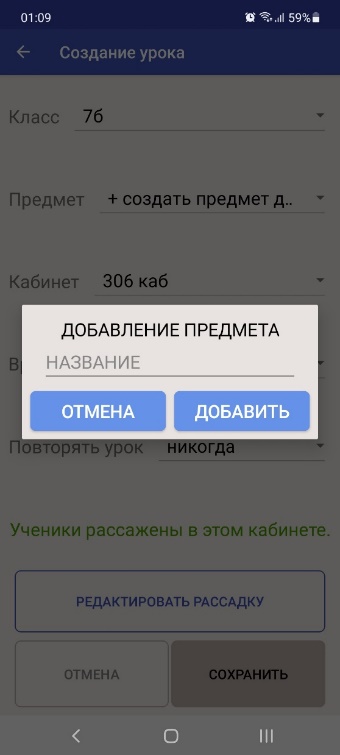


Рисунок 23 – редактор урока – список предметов – создание

А пункт «удалить предметы» (доступен если есть предметы) открывает диалог со списком доступных предметов и флажками выбора. При нажатии кнопки «удалить» удаляются все выбранные предметы.

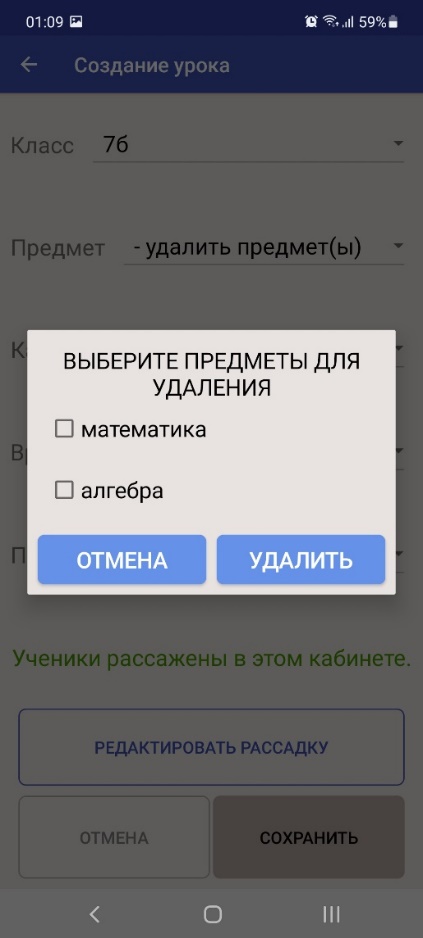


Рисунок 24 – редактор урока – список предметов - удаление

Ниже находится текстовое поле с информацией о том рассажены ли ученики выбранного класса в выбранном кабинете.

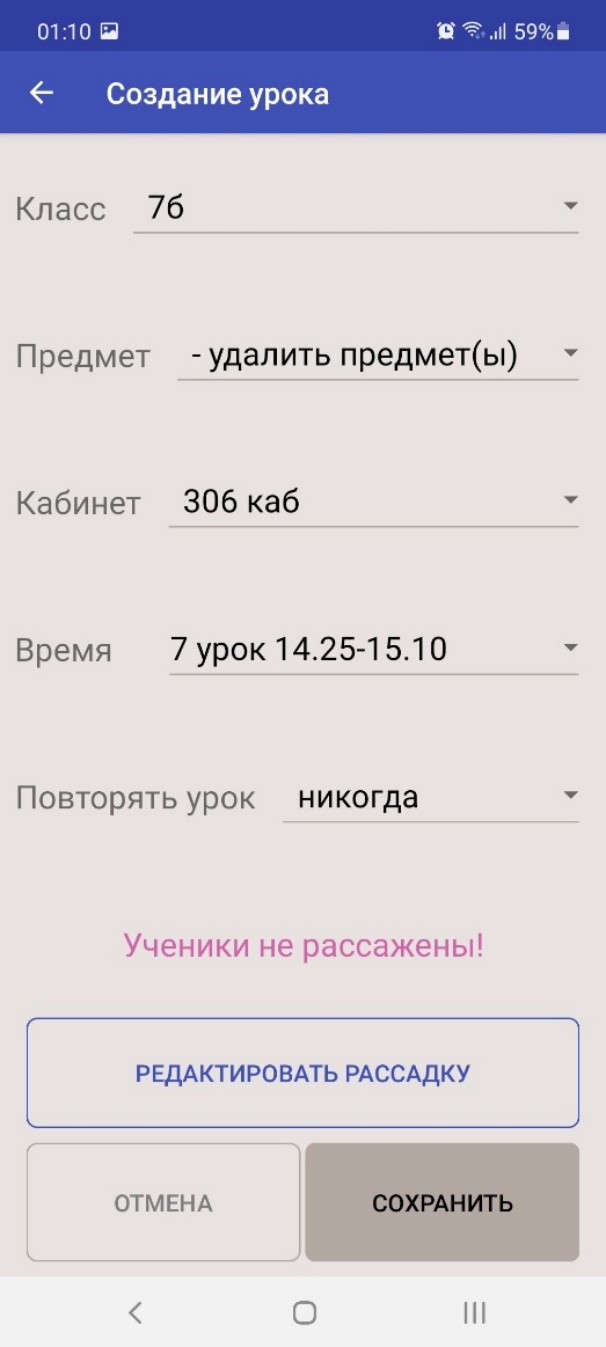


Рисунок 25 – редактор урока – ученики не рассажены

Под ним располагается кнопка перехода на редактор рассадки учеников выбранного класса в выбранном кабинете.

В самом низу есть кнопки «сохранить» и «отмена» (если в расписании на день выбран не пустой урок, то вместо «отмены» будет кнопка «удалить урок» ).

«Сохранить» - сохраняет все изменения, «удалить урок» - удаляет его из базы, «отмена» - тоже что и «назад».

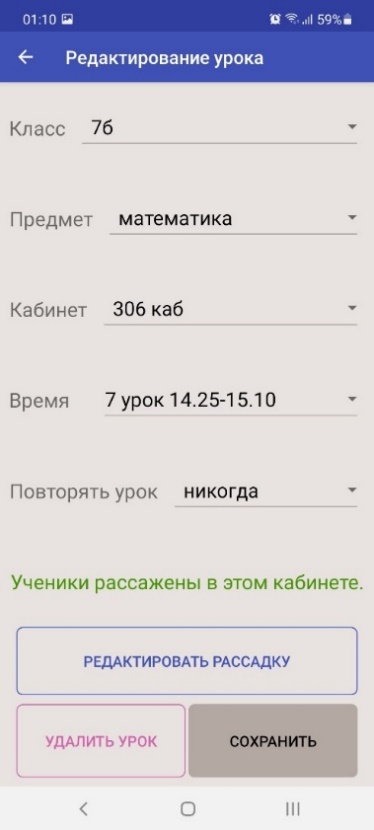


Рисунок 26 – редактор урока – редактирование уже созданного

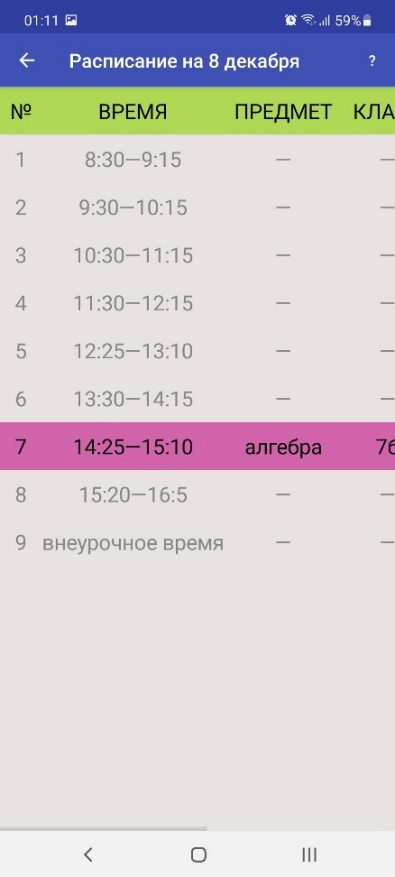


Рисунок 27 – расписание на день – не рассаженные ученики

Рассмотрим редактор рассадки.

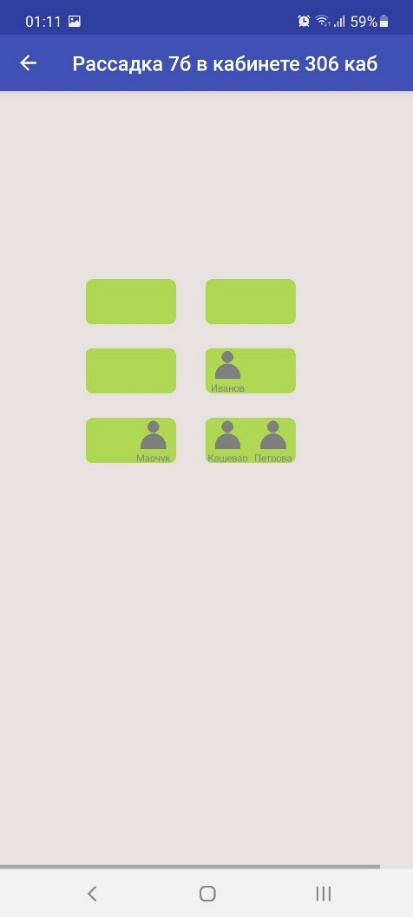


Рисунок 28 – редактор рассадки

Приложение выводит парты в выбранном кабинете и рассаженных учеников выбранного класса. Если не все ученики рассажены, то на всех свободных местах появляется кнопка «добавить».

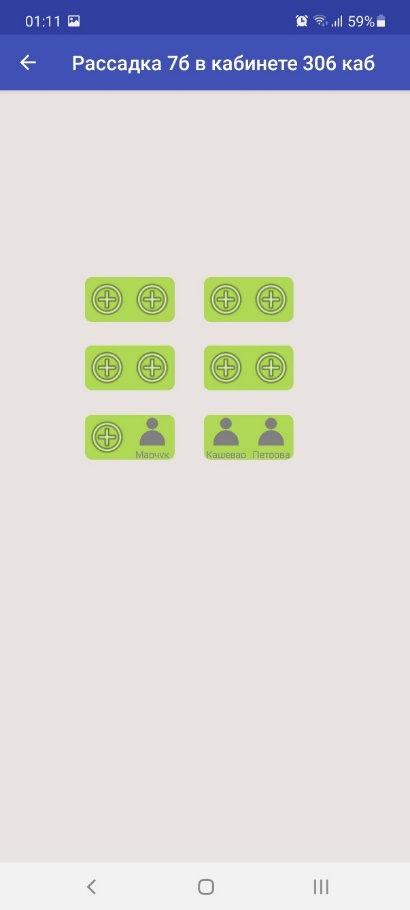


Рисунок 29 – редактор рассадки – «Добавить»

При нажатии на кнопку «добавить» выводится диалог со списком не рассаженных учеников, а кнопка «добавить» на выбранном месте исчезает.

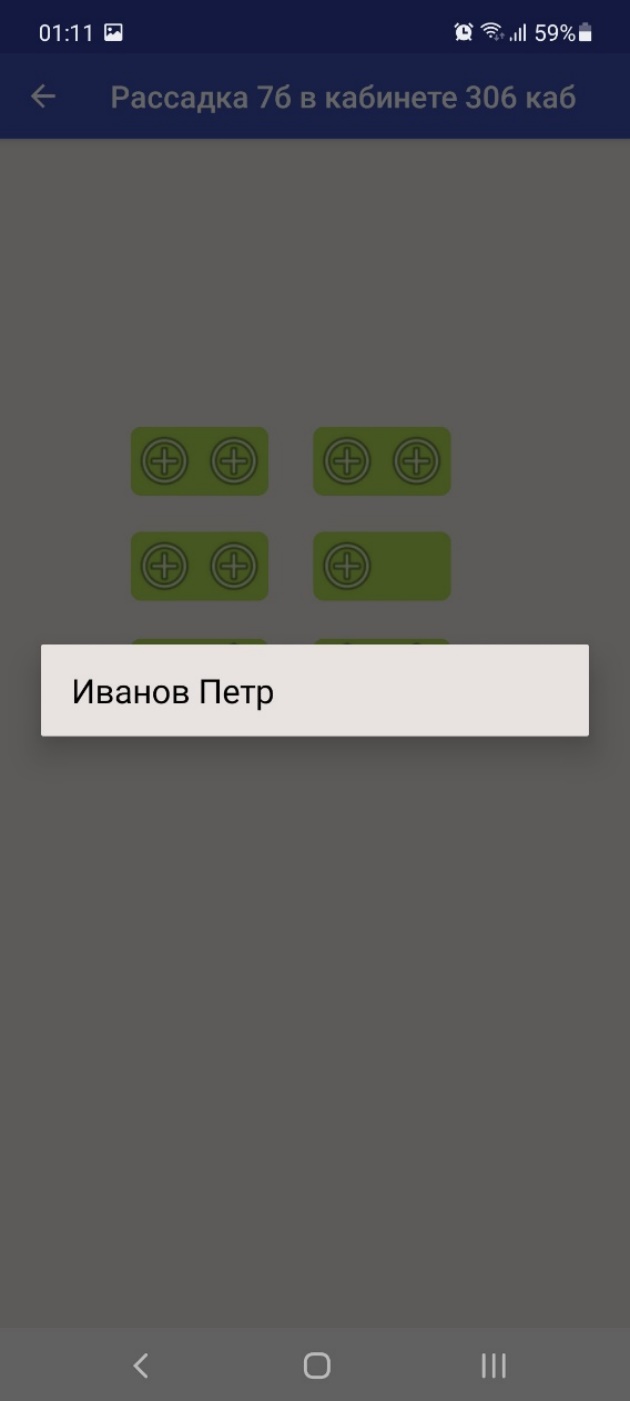


Рисунок 30 – редактор рассадки – список не рассаженных

После выбора ученика диалог закрывается, а ученик добавляется на выбранное место и сохраняется в базу данных.

Рассмотрим урок.

При нажатии на урок в расписании дня открывается активность урока с переданными в неё учениками и классом.

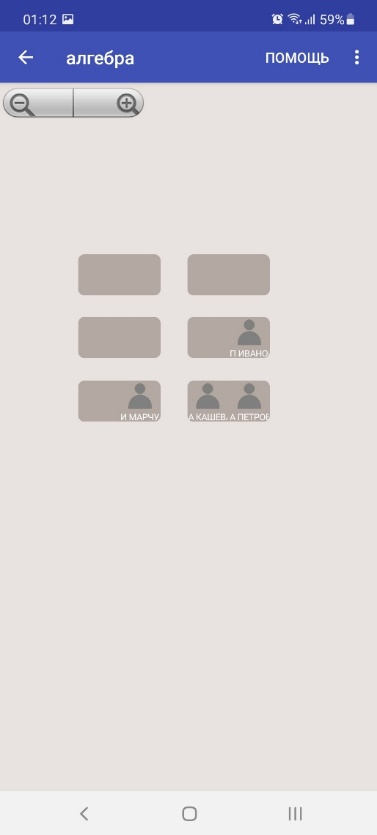


Рисунок 31 – урок

При нажатии на ученика к его оценке добавляется один балл.

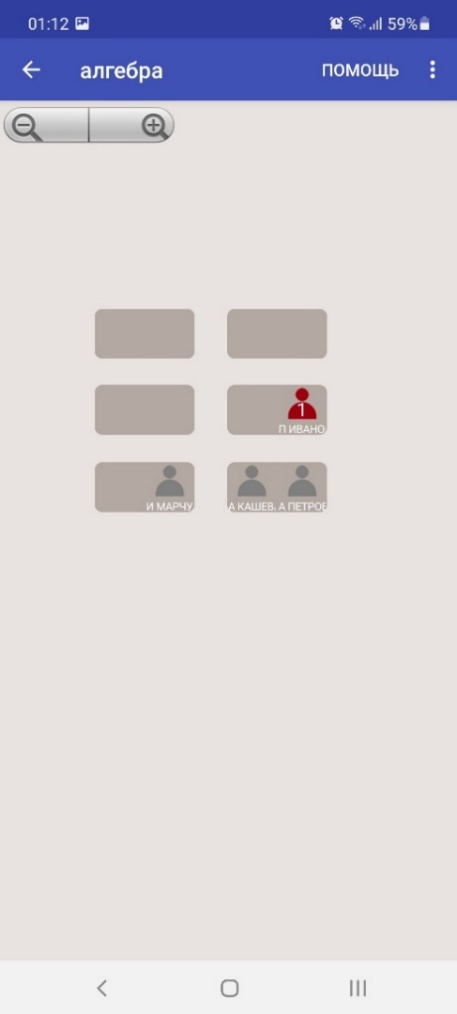


Рисунок 32 – урок – 1 балл

При долгом нажатии открывается контекстное меню с выбором оценки (нельзя поставить оценку которая уже стоит).

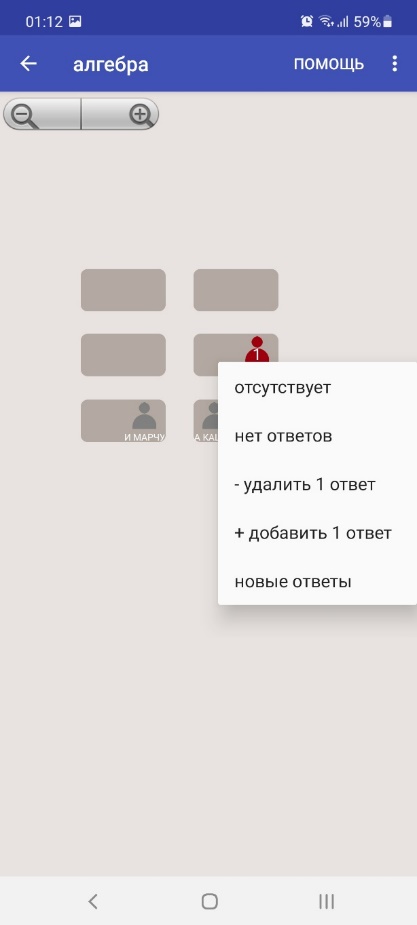


Рисунок 33 – урок – выбор оценки

Пункт «новая оценка» сохраняет оценку у ученика и даёт возможность поставить другую оценку.

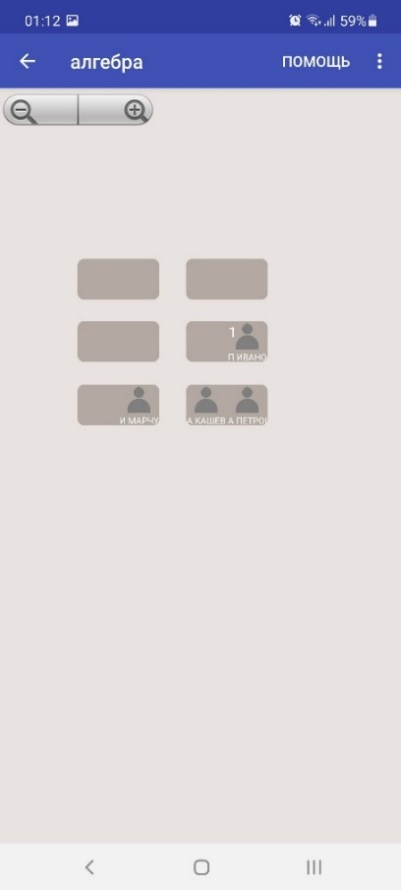


Рисунок 34 – урок – оценка сохранена

Таким образом у ученика на уроке может быть максимум 3 оценки.

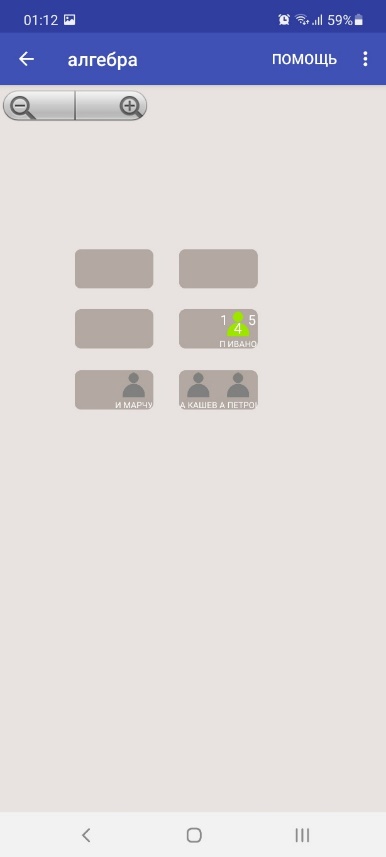


Рисунок 35 – урок – 3 оценки

После выставления оценок, пользователь нажимает кнопку «закончить урок». Происходит переход на активность итогов урока. Приложение выводит список учеников с оценками.



Рисунок 36 – Итоги урока

Пользователь нажимает кнопку «сохранить» оценки сохраняются в базу данных.

## **Локализация**

Система андроид поддерживает работу приложений с несколькими языками. Для того чтобы встроить перевод в приложение достаточно лишь создать один текстовый файл, содержащий все константные строки нужной локализации.

# **Выбор стратегии тестирования и разработка тестов**

Опираясь на знания полученные в курсе теории разработки программных систем [4][5], для тестирования программы был выбран ряд стратегий.

Функциональное тестирование, так как большая часть взаимодействий с пользователем происходит посредством интерфейса и небольшого количества входных данных.

Также, поскольку приложение должно быть интуитивно понятным, чтобы пользователь (учитель) мог использовать его во время уроков, необходимо протестировать приложение на удобство использования. Такая стратегия реализована в оценочном тестировании.

## **3.1 Функциональное тестирование**

Данный вид тестирования производится вручную и часто называется тестированием с управлением по данным, т.е. при тестировании неизвестно, как работает программа, но важно, чтобы входные данные при этом корректно обрабатывались и приводили к ожидаемому результату.

Функциональное тестирование помогает выявлять ошибки для неочевидных вариантов использования приложения, поскольку, например, неправильная работа с некорректными данными может испортить первый опыт использования приложения.

Воспользуемся методом причинно-следственной связи, т.к. методы эквивалентного разбиения и анализа граничных значений предназначены для программ, занимающихся, например, расчетами и вычислениями или отрисовкой сложных графических объектов. Разработанная программа выполняет более простые функции. Разработанные методом причинно-следственной связи тесты приведены в таблице 3.

Таблица 3 – Тестирование методом анализа причинно-следственных связей

|  |  |  |  |
| --- | --- | --- | --- |
| Тест | Ожидаемый результат | Полученный результат | Вывод |
| Выставление оценки в таблице оценок и учеников без выбора предмета | Оценка не добавляется, так как нужно выбрать предмет | Если в классе нет ни одного предмета, то приложение не даст пользователю доступ к таблице оценок | Успешно |
| Изменение интервала оценок на 100, выставление оценки «100», а затем изменение интервала обратно | Оценка должна отображаться вместе с остальными оценками и не препятствовать работе приложения | Оценка действительно отображается, при попытке ее изменить, мы можем выставить только значения из текущего диапазона | Успешно |
| Расстановка парт с проставленной галочкой «выравнивание» и без неё | Если выравнивание выбрано, то парту можно поставить только на линиях сетки. Если нет, то пользователь может перемещать парту в любое место активности | При выбранном пункте, левый верхний край притягивается к углам сетки. Если пункт не выбран, сетка выравнивания становится в 4 раза меньше | Успешно |
| Попытка создать урок при невыбранном предмете | Приложение должно предупредить о том, что пользователь не выбрал у данного класса предмет, урок создать невозможно | Приложение действительно информирует пользователя о том, что предмет не выбран, а также рассказывает, как можно создать этот предмет. | Успешно |
| Создание повторяющихся каждую неделю уроков | После создания урок должен повторяться в календаре каждую неделю в выбранный день | Действительно, урок повторяется. каждый из повторов можно провести, оценки сохраняются на выбранную дату | Успешно |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| Выставление нескольких оценок на уроке | Оценки должны редактироваться и сохраняться, на уроке, в результатах урока и в таблице оценок | Оценки редактируются и сохраняются, на уроке, в результатах урока и в таблице оценок | Успешно |
| Открытие текущего урока через кнопку на главной активности | Если урок на это время есть, то он открывается. Если его нет, то выводится предупреждение об отсутствии урока. | Если урок на это время есть, то он открывается. Если его нет, то выводится предупреждение об отсутствии урока. | Успешно |
| Изменение максимальной оценки в настройках | Во всех меню диапазон оценок меняется на установленный | Во всех меню диапазон оценок меняется на установленный | Успешно |
| Удаление всех данных приложения | При вводе правильного числа в настройках данные удаляются. При вводе неправильного данные остаются на месте. | При вводе правильного числа в настройках данные удаляются. При вводе неправильного данные остаются на месте. | Успешно |

Все тесты, разработанные с помощью метода причинно-следственной связи, прошли успешно.

## **3.2 Оценочное тестирование**

После проведения тестов на работоспособность, приложение можно протестировать с помощью конечных пользователей, которые дадут свою оценку разработанному приложению. В процессе оценочного тестирования участвовали 5 преподавателей из общеобразовательной школы. Участники тестирования оценивали программу по четырем критериям: удобство интерфейса, работоспособность приложения, правильность работы и универсальность. Результаты опроса показано в таблице 11.

Таблица 11 – Оценочное тестирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № пользователя | Удобство интерфейса | Работоспособность приложения | Правильность работы | Универсальность |
| 1 | 7 | 9 | 8 | 9 |
| 2 | 8 | 9 | 10 | 9 |
| 3 | 7 | 10 | 9 | 8 |
| 4 | 7 | 6 | 8 | 8 |
| 5 | 9 | 10 | 9 | 9 |
| Средняя оценка: | 7.6 | 8.8 | 8.8 | 8,6 |

После тестирования была получена информация об удобстве использовании приложения, а также о корректности реализованных функций (то есть на сколько функции подходили под реальные потребности пользователей), что отражено в таблице. Также тестирование помогло найти ряд незначительных ошибок в работе приложения, которые в последствии были исправлены. В дальнейших версиях приложения надо уделить внимание доработке уже существующих функций и добавлению новых, обязательно с сохранением интуитивно понятного интерфейса.

# **Заключение**

В результате выполнения курсовой работы был получен опыт в разработке на языке Java для платформы Android, а также опыт проектирования баз данных. Как итог – разработан прототип приложения «Помощник учителя», позволяющее учителю эффективнее отмечать активность учеников на уроке.

В ходе разработки была использована система контроля версий Git и платформа для хранения репозиториев Github, на которой хранится исходный код программы [6].

Также в процессе разработки были созданы различные диаграммы и схемы с целью получения более продуманного продукта и понимания всех выкладок его работы.

По окончании разработки приложение было протестировано потенциальными пользователями, благодаря чему был обнаружен ряд недочетов, которые были устранены, а также намечен план по добавлению новых функций.

Итоговый сервис соответствует заявленному ТЗ и выполняет все описанные в ТЗ требования.

# **Список используемых источников**

1. Android-приложение «Учительский Журнал» [Электронный ресурс]. URL: <https://play.google.com/store/apps/details?id=com.apolosoft.cuadernoprofesor> (Дата обращения: 19.09.2021).
2. Официальный сайт разработчиков android, [Электронный ресурс]. – URL: <https://developer.android.com> (дата обращения: 01.10.2021).
3. Официальная документация СУБД SQLite3, [Электронный ресурс]. – URL: <https://www.sqlite.org/index.html> (дата обращения: 01.10.2021).
4. Иванова Г.С., Ничушкина Т.Н. Тестирование программного обеспечения: Методические указания по выполнению лабораторной работы по дисциплине "Технология разработки программных систем" [Текст]. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2019. – 13 с.
5. Методы обработки данных и оценки программ : учебное пособие / Г. С. Иванова, Т. Н. Ничушкина, Е. К. Пугачев. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2020. — 72, [2] с. : ил.
6. Репозиторий проекта, [Электронный ресурс]. – URL: <https://github.com/Texnar13/BmstuCourseProject> (дата обращения: 01.10.2021).

# **Приложение А.**

Техническое задание.

Листов 7

# **Приложение Б.**

Руководство пользователя

Листов 20

Исходный код android-приложения «Помощник учителя» находится на Github.com: <https://github.com/Texnar13/BmstuCourseProject>