**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум» (ГБПОУИО «ИАТ»)

КР.09.02.07.23.202.02 ПЗ

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

«ЗАДАЧНИК»

|  |  |  |
| --- | --- | --- |
| Председатель ВЦК: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Н.Р. Карпова) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.С. Александрова) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | (К.А. Анужис) |

(подпись, дата)

Иркутск 2023

## **Содержание**

[**Введение 3**](#_Toc153754493)

[**1 Предпроектное исследование 4**](#_Toc153754494)

[**1.1 Описание предметной области 4**](#_Toc153754495)

[**2 Проектирование мобильного приложения 11**](#_Toc153754496)

[**2.1 Структурная схема приложения 11**](#_Toc153754497)

[2.2 Проектирование базы данных 12](#_Toc153754498)

[2.3 Проектирование пользовательского интерфейса 15](#_Toc153754499)

[**3 Разработка мобильного приложения 18**](#_Toc153754500)

[3.2 Разработка базы данных 18](#_Toc153754501)

[3.3 Разработка мобильного приложения 20](#_Toc153754502)

[Технологическая документация мобильного приложения 22](#_Toc153754503)

[4.1 Руководство пользователя мобильного приложения 22](#_Toc153754504)

[Заключение 26](#_Toc153754505)

[Список используемых источников 27](#_Toc153754506)

[Приложение А Листинг Подключение к базе данных 28](#_Toc153754507)

## **Введение**

Задачник (сборник задач) - собрание задач и упражнений по определенному учебному предмету, предназначенный для усвоения, закрепления и углубления знаний, выработки навыков. Сборник задач представляет собой собрание заданий и упражнений по определённой теме или предмету. Задачник является полезным инструментом для практического применения полученных знаний.

Основное преимущество сборника задач заключается в том, что он предоставляет доступ ко множеству практических заданий, позволяющие закрепить теоретический материал. Такая тренировка позволяет лучше освоить изучаемый материал и быть более увереннее, применяя его на практике. Более того, задачник способствует развитию аналитического мышления. Решение таких задач требует логического и системного подхода. Таким образом, прохождение такого рода тестов способствует развитию интеллектуальных способностей и нахождения новых подходов к решению проблем. Таким образом, все задания и решения будут храниться в базе данных мобильного приложения, который при определенном запросе будет сравнивать решения и выдавать результаты.

Мобильное приложение «Задачник» будет предоставлять тестовые задания по всему курсу предмета «История искусств». В сборнике задач будет предоставлен доступ к тестам по различным темам данного предмета.

Целью данной курсовой работы является создание мобильного приложения сборника тестов.

Основными задачами данной работы являются:

* Изучить программу AndroidStudio.
* Проанализировать информацию по курсу предмета «История искусств».
* Исходя из собранной информации составить задачник.
* Проектирование пользовательского интерфейса.
* Разработка базы данных.
* Разработка мобильного приложения.

# 1 Предпроектное исследование

# Описание предметной области

Задачником является учебно-практическое издание, содержащее задачи, сборники тестов и методические рекомендации к их выполнению в рамках определенного курса, способствующих усвоению и закреплению пройденного материала, и проверке знаний. Сборники задач используются во многих школьных программах, потому что они могут помочь учащимся научиться критически мыслить, выявлять закономерности и разрабатывать творческие решения проблем.

Главная цель сборника задач и упражнений – помочь разобраться в определенной теме и научиться применять полученные знания на практике. Он предоставляет возможность непосредственной тренировки и закрепления материала, а также развивает логическое мышление, аналитические навыки и способность к решению проблем.

Назначение сборника задач и упражнений включает:

* Повышение навыков и практической подготовки: При решении задач и выполнения упражнений из сборника, можно получить дополнительную практику, которая способствует совершенствованию своих навыков и развитию практической подготовки в области, связанной с выбранной темой или предметом.
* Углубление в тему и закрепление знаний: Сборник задач и упражнений часто содержит различные типы задач, позволяющие более углубленно погрузиться в изучаемую тему, усвоить основные понятия и закрепить полученные знания.
* Подготовка к экзаменам и проверка знаний: сборник задач и упражнений часто используется для подготовки к предстоящим экзаменам или проверки уровня знаний. Решение задач и выполнение упражнений позволяет оценить свои знания и узнать, на каких аспектах необходимо сосредоточиться больше.
* Развитие аналитического мышления и умений решать проблемы: Задачи и упражнения в сборнике способствуют развитию аналитического мышления, логического и абстрактного мышления, а также умению решать проблемы.

Они требуют анализа, выбора оптимального способа решения, аргументации и логического обоснования принятых решений. Существуют справочники по различным предметным областям и разной сложности, в зависимости от темы или предмета.

# Обзор инструментальных средств для разработки

Инструменты разработки программного продукта определяют будущий результат.

Проектировать структуру мобильного приложения удобно через MySQL Workbench и Draw.io, а его дизайн - через онлайн-сервис Figma.

MySQL Workbench - инструмент для визуального проектирования [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных [MySQL](https://ru.wikipedia.org/wiki/MySQL). В проекте используется для создания ER-диаграммы БД.

Draw.io – это удобное бесплатное онлайн-приложение для создания диаграмм для рабочих процессов, BPM, организационных, сетевых диаграмм, блок-схемм (флоучарты), UML и принципиальных электросхем. В проекте используется для создания прототипа страниц.

Figma – бесплатный удобный онлайн-сервис для дизайнеров, веб-разработчиков и маркетологов. Он предназначен для создания прототипов сайтов или приложений, иллюстраций и векторной графики. В редакторе можно настроить совместную работу, вносить и обсуждать правки, причём как в браузере, так и через приложение на компьютере. Популярен, в частности, для разработки прототипа и дизайна сайта или приложения. В проекте используется для создания наглядного ожидаемого дизайна проекта, а также уникальных SVG-элементов в едином стиле.

Мобильное-приложение будет содержать в себе информацию - её необходимо хранить, изменять, структурировать и использовать. Это реализуется благодаря базе данных. Были рассмотрены следующие варианты реализации СУБД:

1. MySQL;
2. SQLite;
3. PostgreSQL.

MySQL - свободная реляционная система хранения и управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

На сегодняшний день является самой популярной серверной базы данных (далее - БД), за счёт своей простоты, скорости работы и внушительного функционала. Поддерживаются такие основные движки MyISAM, InnoDB, MEMORY, Berkeley DB. Реализация всех новых возможностей стандарта SQL отсутствует в пользу простоты использования.

PostgreSQL - свободная объектно-реляционная система хранения и управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

Работает только на одном движке - Storage Engine. Все таблицы представлены в виде объектов, они могут наследоваться, а все действия с таблицами выполняются с помощью объектно-ориентированных функций. Обладает открытым исходным кодом, разрабатывается командой энтузиастов, при этом старается максимально соответствовать стандарту SQL. Реализует все самые новые стандарты, что приводит к ущербу простоты, из-за чего PostgreSQL очень сложный и уступает в популярности MySQL.

SQLite – компактная встраиваемая СУБД с исходным кодом. В 2005 году проект получил награду Google-O’Reilly Open Source Awards. SQLite поддерживает динамическое типизирование данных. Возможные типы значений: INTEGER, REAL, TEXT и BLOB. Также поддерживается специальное значение NULL. Размеры значений типа TEXT и BLOB не ограничены ничем, кроме константы SQLITE\_MAX\_LENGTH в исходниках sqlite, равной миллиарду.

SQLite напрямую хранит информацию в одном файле, что облегчает его копирование. Большая популярность в мобильной разработке и небольших автономных приложениях, поскольку она занимает меньше места на дисковом пространстве, имеет высокую скорость работы и не требует в отличии от MySQL не требует наличие сервера для запуска. Минусы: ограничения на запись, всего 5 типов данных, отсутствие встроенного механизма аутентификации.

Для наглядности сравнения вариантов реализации базы данных была составлена таблица 1.

Таблица 1 – Сравнение вариантов СУБД.

|  |  |  |  |
| --- | --- | --- | --- |
| Название БД | MySQL | SQLite | PostgreSQL |
| Большое кол-во типов данных | + | - | + |
| Популярность | + | + | - |
| Отказоустойчивость | - | - | + |
| Не требует удаленного сервера | - | + | - |
| Простота использования | - | + | - |
| Портативность | - | + | - |

Таким образом, в качестве базы данных для будущего продукта была выбрана SQLite, так как она предоставляет весь необходимый функционал для разработки продукта, и при этом она проста в изучении и овладении, достаточно быстрая, предоставляет возможность гибкой настройки, легко переносимая.

Для разработки программного продукта рассмотрены следующие инструментальные средства разработки программных продуктов:

1. Android Studio.
2. Back4App.
3. Firebase.

Аndroid Studio – программа, являющаяся средой разработки приложений для мобильной платформы Android. Прямой конкурент самой популярной утилиты для создания софта под Android – Eclipse. Android studio превосходит конкурентов по многим параметрам, к ним можно отнести:

* Гибкость среды разработки.
* Большой набор функций.
* Процесс разработки, который подстраивается под разработчика.

Это универсальная среда разработки, так как позволяет оптимизировать работу будущих приложений для работы на смартфонах, но и на планшетах, портативных ПК, которые работают на основе рассматриваемой операционной системы.

Back4App – это инструмент разработки бэкенда для мобильных приложений с минимальным количеством кода, обладающий невероятной масштабируемостью и являющийся популярным вариантом программного обеспечения для многих разработчиков.

В отличие от других вариантов программного обеспечения для разработки бэкенда, Back4Appиспользует более простой для понимания код (также требует меньшего написания кода). Это делает его более доступным и отличным вариантом для предприятий, пытающихся самостоятельно разработать свое мобильное приложение.

Преимущества разработки мобильных приложений Back4App многочисленны, но включают в себя масштабируемый хостинг, защиту вашего приложения, живые запросы для более эффективного взаимодействия с данными в реальном времени и общую доступность, благодаря минимальному барьеру для входа, когда речь заходит о написании кода.

Второй инструмент разработки мобильных приложений, который мы предлагаем вашему вниманию – Firebase, платформа, приобретенная и доработанная Google в 2014 году. Firebase позволяет ускорить разработку приложений при полном управлении инфраструктурой бэкенда.

Кроме того, вы можете отслеживать производительность и стабильность с помощью их инновационных инструментов, одновременно повышая вовлеченность с помощью впечатляющих кампаний и богатой аналитики.

Как и Back4App, Firebase от Google также содержит масштабируемую инфраструктуру. Однако Firebase выделяется среди других систем благодаря многочисленным интеграциям, включая GoogleAds, Play Store, Slack, data studio и другие.

Кроме того, вы также получаете преимущества встроенной расширенной системы безопасности и дополнительного хранилища файлов, все из которых резервируются в облаке на случай ошибки.

Для наглядности представлена таблица вариантов приложений для создания android приложений таблица 2.

Таблица 2 – Сравнение вариантов приложений для создания android приложений.

|  |  |  |  |
| --- | --- | --- | --- |
| Название БД | Androidstudio | Back4App | Firebase |
| Большое кол-во типов данных | + | - | + |
| Популярность | + | + | - |
| Отказоустойчивость | - | - | + |
| Не требует удаленного сервера | - | + | - |
| Простота использования | - | + | - |
| Портативность | - | + | - |

Обоснование выбора.

Для создания программного продукта было решено использовать средства:

1. Для создания структурных схем, контекстной и диаграмм декомпозиции использовались CASE-средства - Draw.io.
2. Для наглядного составления структуры базы данных использовался инструмент для визуального проектирования [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) - MySQL Workbench.
3. Для разработки дизайна web-приложения использовался онлайн-сервис для разработки дизайна и прототипа сайта или приложения - Figma.
4. На этапе разработки программного продукта использовались инструменты и среды для разработки:язык программирования Java, IDE Android Studio, СУБД SqLite.
5. Для структурирования, чтения, изменения и удаления информации использовалась база данных SQLite.

# Проектирование мобильного приложения

# Структурная схема приложения

Одним из важных этапов разработки мобильного приложения является проектирование диаграмм, которое способны помочь нам лучше понять структуру и работу приложения.

Диаграмма вариантов использования, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент – возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

На рисунке изображена Use Case View, отражает структурную схему информационной системы «Задачник» для «Пользователя».

Данная таблица отражает действия, которые доступны пользователю в данном приложении. «Пользователь» - является актером. Также таблица отображает то, что пользователь может как просматривать тесты, так и решать их. Выполнив предложенные задания, далее предоставляется доступ к просмотру ответов, и также проверки решенного пользователем задания. На рисунке 1 представлена диаграмма претендентов.

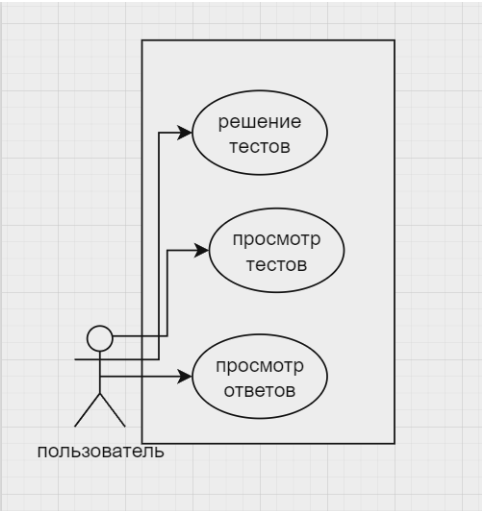


Рисунок 1 – Диаграмма претендентов

Диаграмма деятельности - UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов - вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений. На рисунке изображена диаграмма деятельности. На рисунке 2 представлена диаграмма деятельности.

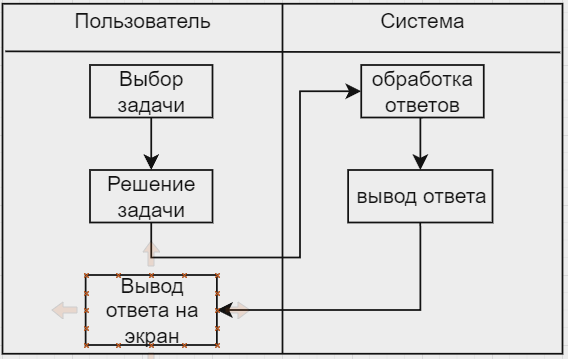


Рисунок 2 – Диаграмма деятельности

В итоге проектирования диаграммы деятельности были выделены основные возможные действия пользователя с программным продуктом.

## **2.2 Проектирование базы данных**

Прежде чем приступить к разработке программного обеспечения, была спроектирована база данных. Процесс проектирования заключается в том, чтобы понять с какими данными работают участники системы, и как эти данные связаны между собой.

Целью инфологического моделирования заключается в том, чтобы обеспечить наиболее естественных способов сбора и представления информации, которая будет храниться в данной базе данных.

Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты). На рисунке 3 представлена инфологическая модель базы данных.

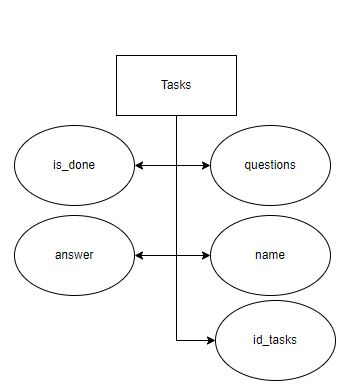


Рисунок 3 – Инфологическая модель базы данных

Модель представляет одну сущность, имеющая свои индивидуальные атрибуты. В сумме получается структура, называемая инфологической моделью.

Проектирование базы данных не ограничивается только этим этапом разработки, а проходит на протяжении всей разработки, до того момента пока в системе не появятся данные, которые нельзя потерять. Результатом проектирования базы данных является ER-модель.

На рисунке 5 представлена ER-модель базы данных, а также на рисунке 4 даталогическая модель БД.

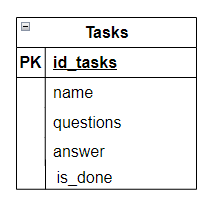


Рисунок 4 – Даталогическая модель БД

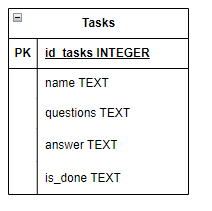


Рисунок 5 – ER-модель базы данных

В таблице 3 представлено то, что содержит таблица Questions».

Таблица 3 – «Questions»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| id\_questions | integer | Код вопросов |
| text | Text | Текст |
| answers | text | ответы |

## **2.3 Проектирование пользовательского интерфейса**

Пользовательский интерфейс – это средства взаимодействия между человеком и компьютером. Говоря простыми словами, интерфейс – внешняя часть программы или устройства, с которыми работает пользователь.

Интерфейсы являются основой взаимодействия всех современных информационных систем. Если интерфейс какого-либо объекта не изменяется (стабилен, стандартизирован), это даёт возможность модифицировать сам объект, не перестраивая принципы его взаимодействия с другими объектами.

Данный прототип интерфейса был построен в онлайн конструкторе figma.

На рисунке 5 представлена главная страница, где будут представлены вопросы, которые пользователь сможет решить. Отображено меню и основные блоки.



Рисунок 5 – Страница «Вопросы»

На рисунке 6 представлена отображение активности. Основная зона, где и будет проводиться основная работа пользователем – решение и проверка задач.

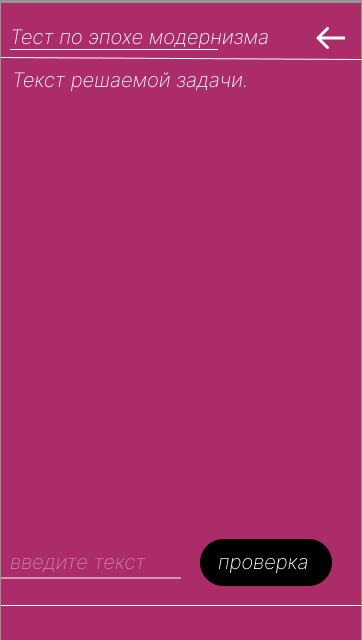


Рисунок 6 Страница для решения задач

На рисунке 7 представлена страница, где будут выводиться ответы.



Рисунок 7 – Страница ответов

Во время разработки прототипа интерфейса был учтен проведенный анализ аналогов, что помогло реализовать представленный вид прототипа информационной системы «Задачник».

Заканчивая работу над дизайном, можно выделить то, что интерфейс разрабатывался с упором на удобства при его использовании и эстетических побуждений, к тому же сам функционал приложения довольно легок и интуитивно понятен.

# Разработка мобильного приложения

**3.1 Разработка интерфейса мобильного приложения**

Одним из важнейших этапов в процессе создания мобильного приложения «Задачник» – это разработка пользовательского интерфейса. Благодаря Android Studio были созданы формы приложения, таблицы, а также кнопки. Фрагмент кода изображён на рисунке 8.



Рисунок 8 – Фрагмент кода интерфейса

## **Разработка базы данных**

Разработка БД мобильного приложения «Задачник» создавалось в SQLiteStudio. База данных мобильного приложения состоит из 1 таблицы.

На рисунке 9 изображен таблица «Tasks». В таблице имеется первичный ключ, а также всё необходимое для создания программируемого задачника.

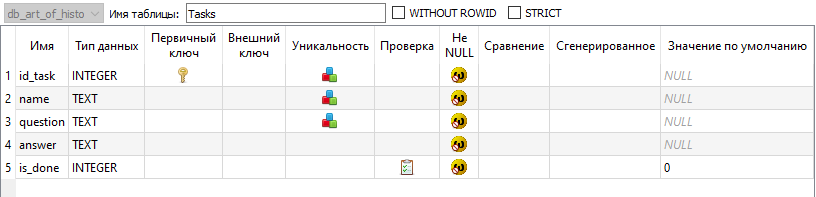


Рисунок 9 – Таблица «Задачи»

## **Разработка мобильного приложения**

В соответствии с выполняемым заданием, необходимо реализовать следующие функции:

* Чтение данных.
* Добавление данных
* Вывод информации
* Просмотр данных
* Обновление данных

Разработка мобильного приложения было сделано в Android Studio. Реализация подключение к базе данных показана на рисунке 10.



Рисунок 10 – Фрагмент кода подключения к БД

Полный код подключения к базе данных представлен в приложении А.



Рисунок 11 – Фрагмент кода подключения к базе данных

На рисунке 12 продемонстрирован фрагмент кода вывода информации из базы данных, полный код можно посмотреть в Приложении.



Рисунок 12 – Фрагмент кода вывода информации из базы данных

## **Технологическая документация мобильного приложения**

## **4.1 Руководство пользователя мобильного приложения**

Заходя в приложение пользователь может заметить заставку мобильного приложения «Задачник», на ней изображена сама иконка мобильного приложения. Заставка приложения изображена на рисунке 13.



Рисунок 13 – Заставка приложения

После того, как приложение запустилось, пользователю показывается главный экран мобильного приложения, где и расположены все тесты, которые можно открыть и решить. Главный экран мобильного приложения расположен на рисунке 14.

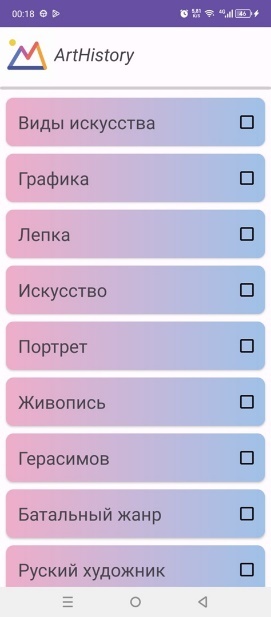


Рисунок 14 – Главный экран мобильного приложения

Заходя в любой из перечисленных в списке тестов, появляется страница с вопросом по данной теме, который также можно решить и проверить. Страница с тестовым вопросом изображена на рисунке 15.



Рисунок 15 – Страница тестового вопроса

При условии, если пользователь отвечает неверно на указанный в тесте вопрос, на экране всплывает окно с фразой «Неправильный ответ!». Страница с окном неправильного ответа изображена на рисунке 16.



Рисунок 16 – страница неправильного ответа

Также при неверном ответе на указанный вопрос в тесте, в общем списке вопросов, будет указан тот ответ, который пользователь ввёл последним. Страница неверного ответа в общем списке указана на рисунке 17.

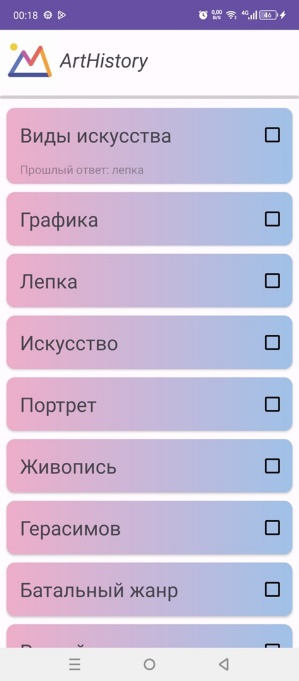


Рисунок 17 – Страница неверного ответа в общем списке

Также неверный ответ будет показываться в строке ввода ответа при следующем решении тестового вопроса. Страница с неверным ответом в строке ввода указана на рисунке 18.

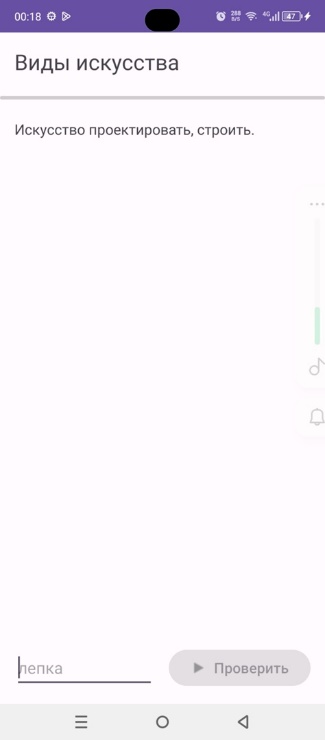


Рисунок 18 – Страница с неверным ответом в строке ввода

Решив тестовый вопрос, в списке предложенных тестов он отобразиться как решенный и рядом с ним на кнопке появится галочка, а также снизу кнопки теста будет отображаться верный ответ, предоставленный пользователем. Решая все последующие тесты, с ними будут происходить аналогичные действия, пока всё приложение с тестами не станет решено. Рисунок с верно решённым вопросом показан на странице 19.

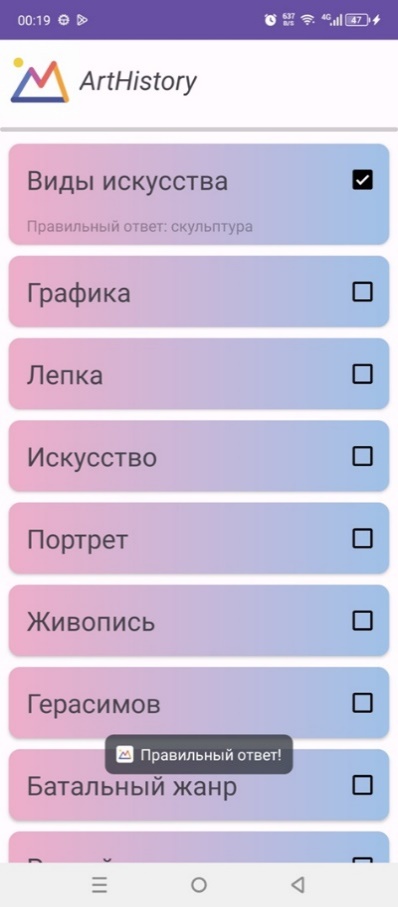


Рисунок 19 – Страница с верно решённым тестом в общем списке вопросов

## **Заключение**

Мобильное приложение «Задачник» предоставляет тестовые задания по всему курсу предмета «История искусств». В сборнике задач предоставлен доступ к тестам по различным темам данного предмета.

Целью данной курсовой работы являлось создание мобильного приложения сборника тестов.

Также были составлены и выполнены задачи данной работы:

* Изучена программа AndroidStudio.
* Проанализирована информация по курсу предмета «История искусств».
* Исходя из собранной информации составлен задачник.
* Спроектирован пользовательский интерфейс.
* Разработана база данных.
* Разработано мобильное приложение.

Мобильное приложение успешно закончено и достигнуты ранее поставленные в работе цели. Мобильное приложение обладает удобным функционалом для решения задач по курсу «Истории искусств», а также будет очень эффективным для подготовки к каким либо проверочным или контрольным работам по нужным темам.

## **Список используемых источников**

1. [Harrix](https://harrix.dev/) / Подключение существующей БД SQLite в Android Studio – URL: <https://harrix.dev/blog/2017/exist-sqlite-android-studio/> (дата обращения: 06.12.2023). – Текст: электронный.
2. Ruseller / Основные команды SQLite – URL: <https://ruseller.com/lessons.php?id=2277> (дата обращения: 06.12.2023). – Текст: электронный.
3. Unetway / SQLite Синтаксис – URL: <https://unetway.com/tutorial/sqlite-syntax?ysclid=la0uq3qfnr275255001> (дата обращения: 06.12.2023). – Текст: электронный.
4. Developer android studio / android studio – URL: <https://developer.android.com/studio> (дата обращения: 06.12.2023). – Текст: электронный.
5. SQLiteStudio / sqlitestudio download – URL: <https://sqlitestudio.pl> (дата обращения: 06.12.2023). – Текст: электронный.
6. Как пользоваться Android Studio / timeweb – URL: <https://timeweb.com/ru/community/articles/kak-polzovatsya-android-studio> (дата обращения: 06.12.2023). – Текст: электронный.
7. SQLite HOME / sqlite – URL: [SQLite Home Page](https://www.sqlite.org/index.html) (дата обращения: 06.12.2023). – Текст: электронный.
8. Android Wikipedia / Android studio Wikipedia – URL: [Android Studio - Wikipedia](https://en.wikipedia.org/wiki/Android_Studio) (дата обращения: 06.12.2023). – Текст: электронный.
9. Blog.skillfactory / Как пользоваться Android studio – URL: [Android Studio: как пользоваться - начало работы и настройка (skillfactory.ru)](https://blog.skillfactory.ru/kak-polzovatsya-android-studio/?ysclid=lbbvjrnuy5880863937) (дата обращения: 06.12.2023). – Текст: электронный.

## **Приложение А Листинг Подключение к базе данных**

package com.example.artofhistory.utils;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import androidx.annotation.Nullable;  
import com.example.artofhistory.models.Task;  
import java.util.ArrayList;  
import java.util.List;  
public class DatabaseHelper extends SQLiteOpenHelper {  
 private SQLiteDatabase db;  
 private final Context context;  
 private static final String  
 DATABASE\_NAME = "db\_art\_of\_history\_tasks",  
 TABLE\_NAME = "Tasks",  
 COLUMN\_ID = "id\_task",  
 COLUMN\_NAME = "name",  
 COLUMN\_QUESTION = "question",  
 COLUMN\_ANSWER = "answer",  
 COLUMN\_IS\_DONE = "is\_done",  
 COLUMN\_LAST\_ATTEMPT = "last\_attempt";  
 public DatabaseHelper(@Nullable Context context) {  
 super(context, DATABASE\_NAME, null, 3);  
 this.context = context;  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("CREATE TABLE IF NOT EXISTS " + TABLE\_NAME + " (" +  
 COLUMN\_ID + " INTEGER NOT NULL UNIQUE PRIMARY KEY AUTOINCREMENT," +  
 COLUMN\_NAME + " TEXT NOT NULL UNIQUE," +  
 COLUMN\_QUESTION + " TEXT NOT NULL UNIQUE," +  
 COLUMN\_ANSWER + " TEXT NOT NULL," +  
 COLUMN\_LAST\_ATTEMPT+" TEXT," +  
 COLUMN\_IS\_DONE + " INTEGER NOT NULL DEFAULT 0 CHECK (is\_done IN (0, 1)))"  
 );  
 db.execSQL("INSERT INTO 'Tasks' ('name', 'question', 'answer')" +  
 "VALUES" +  
 "('Виды искусства', 'Искусство проектировать, строить.', 'скульптура')," +  
 "('Графика', 'Произведение, в котором изображение нанесено на бумагу карандашом или тушью.', 'графика')," +  
 "('Лепка', 'Изделия из глины, закрепленные обжигом.', 'керамика')," +  
 "('Искусство', 'Объёмное художественное произведение, созданное путём резьбы, лепки, высекания, отливки.', 'скульптура')," +  
 "('Портрет', 'Погрудное изображение человека в скульптуре.', 'бюст')," +  
 "('Живопись', 'Картина, на которой изображено море, морской вид.', 'марина')," +  
 "('Герасимов', 'На картине Герасимова \"После дождя\" изображена беседка, на которой стоит стол, а на нём - вааза с сиренью. К какому жанру относится эта картина.', 'натюрморт')," +  
 "('Батальный жанр', 'На картинах батального жанра может быть изображено...', 'сражение')," +  
 "('Руский художник', 'Его называли \"певцом русского леса\".', 'шишкин')," +  
 "('Возрождение', 'Город - родина Возрождения.', 'флоренция')"  
 );  
 }  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);  
 onCreate(db);  
 }  
 public List<Task> readTasks() {  
 String query = "SELECT \* FROM " + TABLE\_NAME;  
 db = this.getReadableDatabase();  
 Cursor cursor = null;  
 if(db != null) {  
 cursor = db.rawQuery(query, null);  
 }  
 List<Task> tasksList = new ArrayList<>();  
 if (cursor == null) return tasksList;  
 if (cursor.moveToFirst()) {  
 do {  
 Task task = new Task();  
 task.setIdTask(cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_ID)));  
 task.setName(cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_NAME)));  
 task.setQuestion(cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_QUESTION)));  
 task.setAnswer(cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_ANSWER)));  
 task.setLastAttempt(cursor.getString(cursor.getColumnIndexOrThrow(COLUMN\_LAST\_ATTEMPT)));  
 task.setIsDone(cursor.getInt(cursor.getColumnIndexOrThrow(COLUMN\_IS\_DONE)) > 0);  
 tasksList.add(task);  
 } while (cursor.moveToNext());  
 }  
 cursor.close();  
 db.close();  
 return tasksList;  
 }  
 public void updateTaskIsDone(int id, boolean isDone) {  
 db = this.getWritableDatabase();  
 ContentValues cv = new ContentValues();  
 cv.put(COLUMN\_IS\_DONE, isDone);  
 db.update(TABLE\_NAME, cv, COLUMN\_ID + "=?", new String[]{String.valueOf(id)});  
 db.close();  
 }  
 public void updateTaskLastAttempt(int id, String lastAttempt) {  
 db = this.getWritableDatabase();  
 ContentValues cv = new ContentValues();  
 cv.put(COLUMN\_LAST\_ATTEMPT, lastAttempt);  
 db.update(TABLE\_NAME, cv, COLUMN\_ID + "=?", new String[]{String.valueOf(id)});  
 db.close();  
 }  
}