

Инженерная школа информационных технологий и робототехники

Направление: 09.03.01 «Информатика и вычислительная техника»

Отделение информационных технологий

Отчёт по теме

**«Автоматизация контроля посещаемости занятий студентами и преподавателями»**

по дисциплине «Творческий проект»

Выполнили:

студенты гр. 8В12 \_\_\_\_\_\_\_\_\_\_\_\_\_ А.И. Самохвалов

\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Береговенко

\_\_\_\_\_\_\_\_\_\_\_\_\_ И.Д. Казенков

\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.П. Гынгазов

\_\_\_.\_\_\_.\_\_\_\_\_\_

Проверил:

доцент ОИТ \_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Мыцко

оценка (до 30 б.): \_\_\_\_\_\_\_\_\_\_\_\_\_

к защите допускаю \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_.\_\_\_.\_\_\_\_\_\_

Оглавление

[**Введение** 3](#_Toc138315088)

[**Цель и задачи** 4](#_Toc138315089)

[**Список требований к проекту** 5](#_Toc138315090)

[**Обзор технологий** 6](#_Toc138315091)

[**Реализованное приложение** 7](#_Toc138315092)

[**Алгоритм** 10](#_Toc138315093)

[**Реализованный сайт** 10](#_Toc138315094)

[**Заключение** 11](#_Toc138315095)

[**Список литературы** 13](#_Toc138315096)

# **Введение**

Руководство институтов заинтересован в непрерывном учете и контроле за посещаемостью студентов, так как посещение студентами занятий напрямую влияет на их успеваемость, что в свою очередь влияет на рейтинг и популярность институтов. Основным способом контроля посещаемости является ведение журнала, и чтобы его заполнение не занимало много времени, мы решили создать систему, позволяющую с помощью генерации и считывания QR-кодов отмечать студентов и преподавателей.

# **Цель и задачи**

Цель проекта:

Создать систему, позволяющую с помощью генерации и считывания QR-кодов отмечать студентов и преподавателей.

Задачи проекта на этот семестр:

1. Спроектировать схему базы данных
2. Реализовать вывод расписания
3. Реализовать отображения факта посещения занятия

# **Список требований к проекту**

В нашем проекте мы создает приложение, с помощью которого можно отметиться на занятии. При этом оно должно быть удобно в использовании.

Поэтому основные требования к нашему приложению:

* Автозаполнение email и пароля при авторизации
* Интуитивный интерфейс
* Отображение того, что студент был отмечен

# **Обзор технологий**

Для создания приложения использовались XML и Kotlin в среде разработки Android Studio. А для создания базы данных использовалась СУБД SQLite

XML — это язык разметки, который используется для обмена данными. Он используется для создания и хранения структурированной информации. XML позволяет создавать пользовательские теги, что делает его более гибким и удобным для использования.

Kotlin — это статически типизированный язык программирования, который в основном используется для создания мобильных приложений. Kotlin является очень популярным языком, так как он имеет множество преимуществ перед другими языками программирования. Он облегчает и ускоряет процесс разработки приложений благодаря более короткому и эффективному коду.

# **Реализованное приложение**

Приложение состоит из четырех форм. Первая — это форма входа (рисунок 1). На ней расположены: заголовок с призывом к действию, кнопка для входа в аккаунт и перехода на следующую форму, две EditView для ввода email и пароля

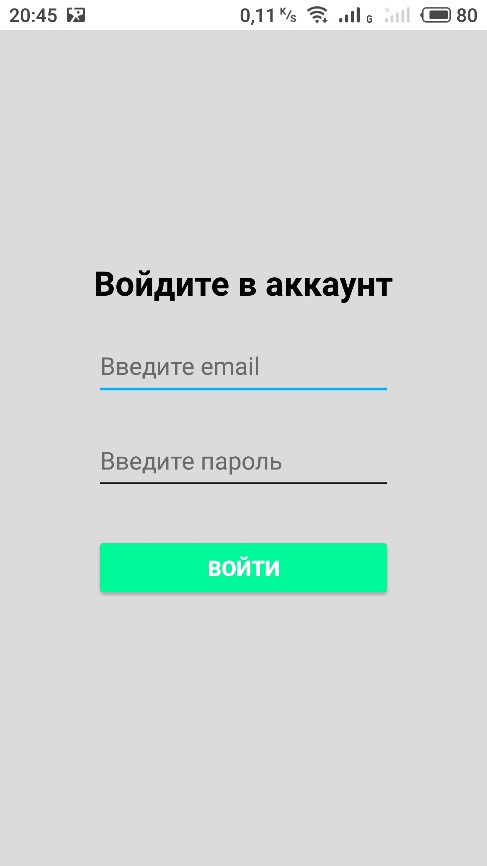


Рис. 1 – Форма входа

Через кнопку “Войти” осуществляется переход на форму, на которой осуществляется сканирование Qr-кода (рисунок 2). На данной форме открыта камера с областью для Qr-кода.

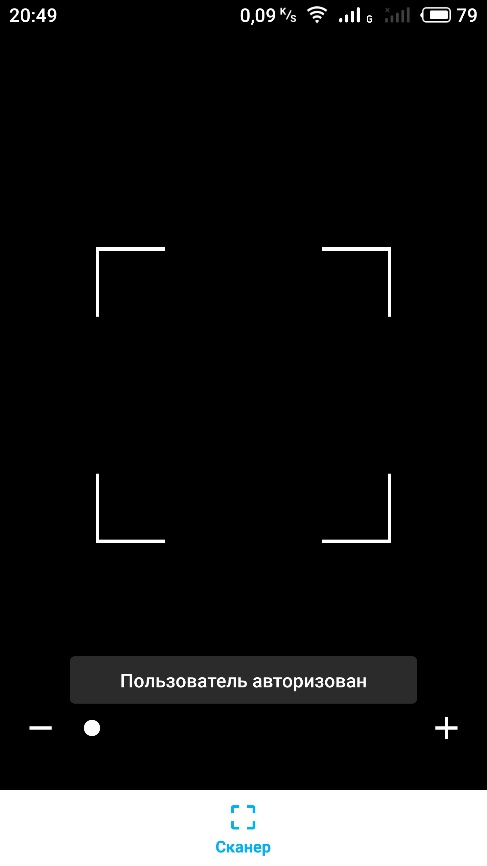


Рис. 2 – Сканер Qr-кода

При сканировании Qr-кода открывается следующая форма с ссылкой на расписание и с отображением того, что студент отмечен (рисунок 3).

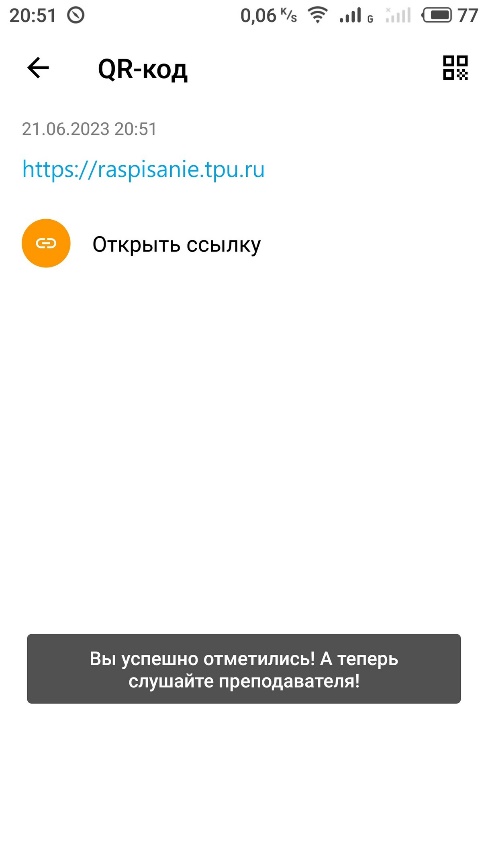


Рис. 3 – Форма с ссылкой

При нажатии на иконку с Qr-кодом (в правом верхнем углу), пользователю открывается форма с Qr-кодом (рисунок 4).



Рис. 4 – Форма с Qr-кодом

Также была спроектирована и создана база данных (рисунок 5)

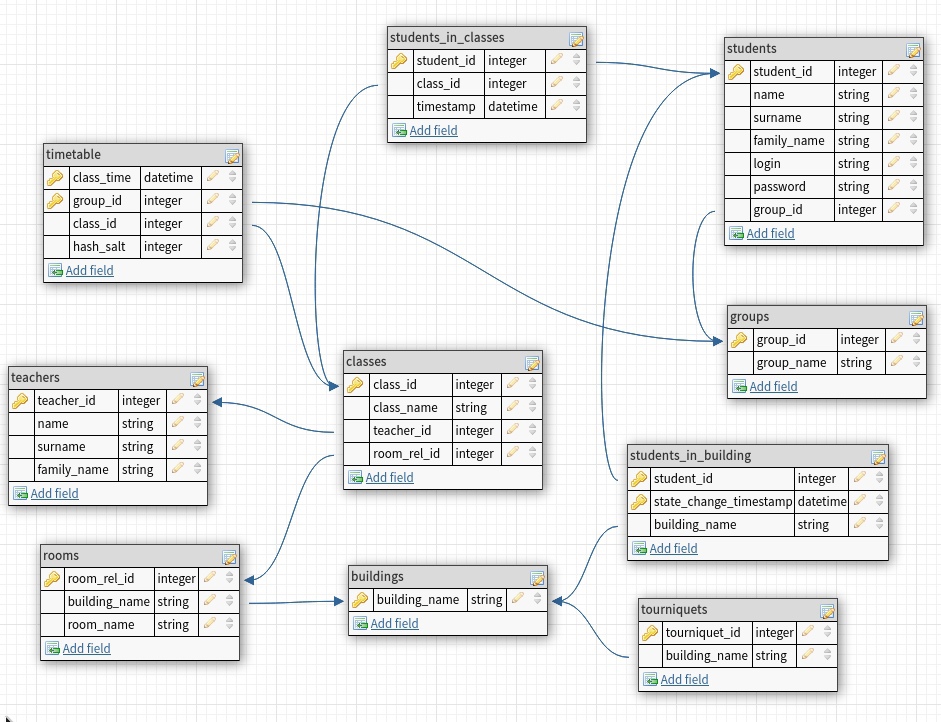


Рис. 5 – UML-диаграмма базы данных

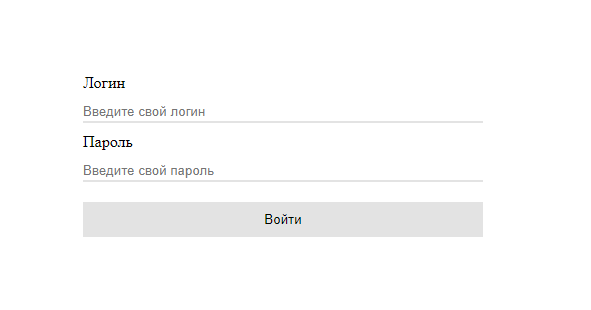
# **Алгоритм**

Для авторизации пользователя были созданы классы: User и DatabaseHelper. User – нужен для присвоения email и пароля из EditView соответствующим полям объекта. Класс DatabaseHelper – нужен для обращения к базе данных. Также использовался уже готовый класс SQLiteDatabase.

Алгоритм работы: пользователь вводит Email и пароль, они добавляются в поля объекта User при помощи конструктора класса, приложение находит адрес до базы данных, открывает ее и проверяет есть ли пользователь с таким email и паролем.

# **Реализованный сайт**

Сайт состоит из четырех 3 страниц. Первая — это форма авторизации преподавателя (рисунок 6). На ней расположены кнопка для входа в аккаунт и перехода на следующую страницу, две Input для ввода email и пароля

Рис. 6 – Форма с Qr-кодом

На следующей странице, после входа у нас генерируется QR-код для каждого предмета из полученного ключа из БД, чтобы ученики смогли отметиться. Так же у нас присутствует шапка, благодаря которой можно посмотреть расписание преподавателя, группы студентов, и выйти из аккаунта



Рис. 7 – Форма с Qr-кодом

На странице «Расписание» можно увидеть какие вообще есть предметы в расписании. Оно содержит время занятия, место, где проходит занятие, предмет, преподавателя и группу

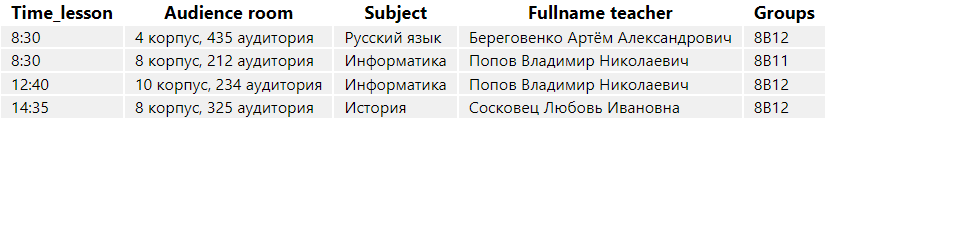


Рис. 8 – Расписание

# **Заключение**

В течении этого семестра при работе над творческим проектом были выполнены все поставленные задачи. Среди выполненных задач:

Основные:

1. Спроектировать схему базы данных
2. Реализовать вывод расписания
3. Реализовать отображения факта посещения занятия

Дополнительные:

1. Создана база данных
2. Реализована авторизация с автозаполнением при повторном входе и выход из аккаунта

Также уже сейчас были поставлены цели на следующий семестр:

1. Интеграция приложения с базой данных;
2. Установка связи между приложением и веб-сайтом;
3. Реализация систем уведомлений, которая будет информировать студентов, преподавателей и администраторов о непосещенных занятиях или пропусках.
4. Вывод расписания по группам и по преподавателям
5. Доработка логики сайта и приложения

# **Список литературы**

1. K.P. Naveen Reddy, Alekhya T., Sushma Manjula T., Rashmi Krishnappa. AI-Based Attendance Monitoring System. Journal of Innovative Technology and Exploring Engineering, 2019, vol. 9, no. 2S, pp. 592-597. <https://www.doi.org/10.35940/ijitee.B1057.1292S19>
2. Использование простой базы данных SQLite в Android-приложении. https://www.internet-technologies.ru/articles/ispolzovanie-prostoy-bazy-dannyh-sqlite-v-android-prilozhenii.html
3. Sogbaike C.O., Ahoro V.O. Design and Implementation of Class Attendance Management System Using Fingerprint Recognition. International Journal of Scientific and Research Publications, 2019, vol. 9, no. 5, pp. 882-897. <https://dx.doi.org/10.29322/IJSRP.9.05.2019.p89110>
4. Xiong Wei, Anupam Manori, Nandgopal Devnath, Nitin Pasi, Vivek Kumar. QR Code Based Smart Attendance System. International Journal of Smart Business and Technology, 2017, vol. 5, no. 1, pp. 1-10. <https://www.doi.org/10.21742/ijsbt.2017.5.1.01>
5. S. Saraswathi, Sindhuja M., Y. Salini, M. Venkatesh. Student attendance system using bar code scanner. Materials Today: Proceedings, 2021, <https://doi.org/10.1016/j.matpr.2020.12.898>
6. GitHub <https://github.com/topics/qrcode-scanner?l=kotlin>
7. Робин Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5
8. PHP | Сессии – Metanit   
   <https://metanit.com/php/tutorial/4.3.php>
9. Open Server — профессиональный инструмент веб-разработчика  
   <https://habr.com/ru/articles/137388/>