Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

Кафедра информационных систем и программной инженерии

**КУРСОВОЙ ПРОЕКТ**

Мобильное приложение «Мобильная ведомость»

по дисциплине

«Технологии разработки мобильных приложений»

Выполнил: студент гр. ИСТ-114

Аксенов Н.А.

Принял: доцент

Салех Х.М.

Владимир, 2017

**АННОТАЦИЯ**

Тема: Мобильное приложение «Мобильная ведомость».

В данной работе представлена реализация мобильной программной системы для учета знаний учащихся с базовым набором функций для системы такого рода. Прототип программной системы обеспечивает взаимодействие между сотрудниками деканата, преподавателями и студентами.

Курсовой проект представлен на 26 страницах, рисунков – 28, использованных источников – 11, приложений – 1.

Ключевые слова: Мобильная Ведомость, Андроид приложение, Клиент-Сервер, Java, PHP, MySQL, JSON, API.

**ABSTRACT**

Subject: Development of a software system "Students knowledge accounting".

This work presents the implementation of a mobile system to accommodate students knowledge with a basic set of features for such system. The mobile system prototype provides an interaction between the staff of a dean’s office, teachers and students.

The course project is represented in 26 pages, images - 28, used sources – 11, applications – 1.

Keywords: Mobile Statement, Android Application, Client-Server, Java, PHP, MySQL, JSON, API.

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc501322121)

[ВВЕДЕНИЕ 3](#_Toc501322122)

[1 ТРЕБОВАНИЯ 4](#_Toc501322123)

[1.1 Определения, акронимы и сокращения 4](#_Toc501322124)

[1.2 Обзор аналогов 4](#_Toc501322125)

[1.3 Сценарий работы приложения 4](#_Toc501322126)

[1.4 Функциональные требования 5](#_Toc501322127)

[1.5 Нефункциональные требования 7](#_Toc501322128)

[1.6 Допущения и зависимости 8](#_Toc501322129)

[2 ПРОЕКТИРОВАНИЕ 9](#_Toc501322130)

[2.1 Общая организация системы 9](#_Toc501322131)

[2.2 Проектирование приложения 9](#_Toc501322132)

[3 РЕАЛИЗАЦИЯ 12](#_Toc501322133)

[3.1 Диаграмма развертывания 12](#_Toc501322134)

[3.2 Пользовательский интерфейс. 12](#_Toc501322135)

[3.3 Структура БД. 20](#_Toc501322136)

[3.4 Структура проекта на сервере. 21](#_Toc501322137)

[3.4 Структура проекта на клиента. 21](#_Toc501322138)

[3.5 Тестирование приложения. 23](#_Toc501322139)

[ЗАКЛЮЧЕНИЕ 24](#_Toc501322140)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc501322141)

[ПРИЛОЖЕНИЕ А ЛИСТИНГ ПРОГРАММНОЙ СИСТЕМЫ 26](#_Toc501322142)

# ВВЕДЕНИЕ

В наше время сложно представить жизнь без мобильных устройств. Все больше компаний, имеющих свои информационные системы, стараются также обзавестись мобильными приложениями, т.к. практически у каждого человека имеется мобильное устройство, а соответственно, сфера мобильных приложений охватывает огромное количество людей.

Целью данного курсового проекта является разработка мобильного приложения для устройств, работающих на ОС Android.

Назначение разработки данного мобильного приложения - создать удобное мобильное приложение, работающее на ОС Android, позволяющее облегчить работу преподавателей и сотрудников деканата с ведомостями, а также обеспечить учащихся возможностью следить за своей успеваемостью.

Область применения разрабатываемого приложения: деканат университета.

Программа рассчитана для использования в университете. Её функционал позволит облегчить учет успеваемости студентов для преподавателей и сотрудников деканата, а также увеличит точность процесса и облегчит их работу. Студентам данное приложение позволит в любое время и удобным образом контролировать свою успеваемость.

Для реализации поставленной задачи требуется изучить предметную область, спроектировать систему, реализовать серверную часть на хостинге «000webhost», а также реализовать клиентскую часть приложения при помощи IDE Android Studio.

# ****1 ТРЕБОВАНИЯ****

## 1.1 Определения, акронимы и сокращения

Мобильное приложение – программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах.

Клиент – это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.

Сервер – программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определённым ресурсам или услугам.

Ведомость – документ, в котором хранятся оценки студентов.

Оценка – значение, определяющее уровень знаний студента (5 - балльная).

Пользователь – действующее лицо, наделенное определенными правами.

## 1.2 Обзор аналогов

Аналогов для данного приложения не найдено.

## 1.3 Сценарий работы приложения

Взаимодействие пользователей с системой осуществляется с помощью Android приложения. Каждый пользователь имеет определенную роль, которая определяет его возможности. Можно выделить следующие роли и их возможности:

Сотрудник деканата:

* Посмотреть список всех ведомостей
* Посмотреть список сданных ведомостей
* Посмотреть отдельную ведомость
* Создать ведомость
* Открыть ведомость
* Закрыть ведомость
* Задать пользователю роль
* Определить студента в группу

Преподаватель:

* Посмотреть список несданных ведомостей
* Посмотреть список сданных ведомостей (с графиком успеваемости)
* Поставить оценку в ведомость
* Удалить оценку из ведомости
* Сдать ведомость

Студент:

* Посмотреть список всех оценок
* Посмотреть список неудовлетворительных оценок
* Посмотреть детали об отдельной оценке

Незарегистрированный пользователь:

* Войти в систему
* Зарегистрироваться

Также, каждый зарегистрированный пользователь имеет возможность авторизоваться в системе и выйти из своего аккаунта.

## 1.4 Функциональные требования

Qr1 «Вход пользователя в систему»

Пользователь может войти в систему, заполнив поля «e-mail» и «пароль», а затем нажав кнопку «Войти». Если такой пользователь существует и введен правильный пароль, происходит предоставление пользователю функционала, присущему его роли в системе. При отсутствии такого пользователя в базе или неправильно введенном пароле, происходит показ сообщения об ошибки.

Qr2 «Регистрация пользователя в системе»

Незарегистрированный пользователь может зарегистрироваться, нажав на кнопку «Зарегистрироваться» на странице входа в систему, будет произведен переход на страницу регистрации, где необходимо заполнить поля «ФИО», «EMail», «Пароль» и нажать кнопку «Зарегистрироваться». Система автоматически присвоит новому пользователю роль «neizv» (неизвестный пользователь). Пользователь с данной ролью временно не имеет доступа к функционалу системы до тех пор, пока сотрудник деканата не присвоит данному пользователю другую роль.

Qr3 «Просмотр списка ведомостей»

Данная функция доступна пользователям с ролью «сотрудник деканата», или «преподаватель». Сотрудник деканата имеет право просматривать все ведомости. Преподаватель имеет право просматривать только ведомости, в котором в качестве преподавателя указан именно он, а также данные ведомости должны быть открыты. Для просмотра списка ведомостей необходимо перейти по ссылке в боковом меню приложения «Все ведомости». Преподаватель также наделен правом просматривать уже сданные ведомости. Однако в них будет отсутствовать функционал редактирования ведомости, но будет доступен график успеваемости студентов.

Qr4 «Изменение роли пользователя»

Данная функция доступна для пользователей с ролью «сотрудник деканата». Для того, чтобы изменить или добавить роль пользователю, сотруднику деканата необходимо перейти по ссылке «Пользователи» и в списке пользователей выбрать нужного пользователя, а на странице конкретного пользователя выбрать подходящую для него роль и нажать на кнопку «Уст.».

Qr5 «Добавление или изменение группы студента»

Данная функция доступна для пользователей с ролью «сотрудник деканата». Для того, чтобы добавить или изменить группу студента, пользователю необходимо перейти по ссылке «Пользователи», затем выбрать пользователя с ролью студента и задать ему нужную группу студента.

Qr6 «Создать ведомость»

Данная функция доступна для пользователя с ролью «сотрудник деканата». Для того, чтобы создать новую ведомость, сотруднику деканата необходимо перейти по ссылке «Создать ведомость» в боковом меню, выбрать в выпадающих меню предмет, преподавателя, группу студентов, указать название ведомости и нажать кнопку «Создать ведомость». При создании ведомости система автоматически присвоит статус ведомости «открыта» и она станет доступна преподавателю.

Qr7 «Редактирование ведомости»

Данная функция доступна для пользователя с ролью «преподаватель». Для того, чтобы отредактировать ведомость, преподавателю необходимо выбрать ведомость со страницы списка ведомостей, и нажать на интересующую его ведомость. Затем преподавателю станет доступна форма добавления оценки и список уже существующих в ведомости оценок, а также информация о ведомости.

Qr8 «Изменить статус ведомости»

Данная функция доступна пользователям с ролью «сотрудник деканата» и «преподаватель». Сотруднику деканата доступна функция открытия и закрытия ведомости. Для этого нужно перейти на страницу нужной ведомости, установить нужный статус в выпадающем списке и нажать на кнопку «Уст.». Преподавателю доступна функция сдачи ведомости в деканат. Для этого необходимо перейти на страницу нужной ведомости и нажать на кнопку «Сдать ведомость».

Qr9 «Завершение работы с системой»

Для завершения работы с системой пользователь выполнивший вход, может выйти из системы, перейдя на страницу своего профиля и нажав на кнопку «Выйти из учетной записи».

## 1.5 Нефункциональные требования

* Все данные хранятся в базе данных последней версии.
* Время отклика программы и элементов управления должно быть минимальное.
* Программа должна иметь понятный и лаконичные дизайн и навигацию.
* Приложение должно исправно работать на Android 4.4 и старше.

Исходные данные:

1. Архитектура: Android
2. Язык программирования: Java
3. СУБД: MySQL 5.5
4. Среда разработки: Android Studio
5. Язык моделирования: UML
6. Среда моделирования: Visual Paradigm

## 1.6 Допущения и зависимости

Система должна состоять из следующих программных компонентов:

клиент, сервер.

Характеристики клиента:

OS: Android начиная с версии 4.4

CPU: 1.20 Ггц

RAM: 1 Гб

ROM: 2 Гб

Network: 3G

Характеристики сервера:

OS: Linux

CPU: Intel Core i3-7100U 2.20 Ггц

RAM: 4 Гб

ROM: 100 Гб

LAN: 60 Мбит/с

На сервере должны быть установлены:

* Apache HTTP Server
* PHP 7
* MySQL 5.5

# ****2 ПРОЕКТИРОВАНИЕ****

## ****2.1 Общая организация системы****

Model-View-Presenter (MVP) — шаблон проектирования, производный от MVC, который используется в основном для построения пользовательского интерфейса.

Model (Модель) предоставляет данные и реагирует на команды контроллера, изменяя свое состояние.

View (Представление) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Представитель (Presenter) в данном шаблоне берёт на себя функциональность посредника (аналогично контроллеру в MVC) и отвечает за управление событиями пользовательского интерфейса так же, как в других шаблонах обычно отвечает представление.

Обычно экземпляр представления создаёт экземпляр представителя, передавая ему ссылку на себя. При этом представитель работает с представлением в абстрактном виде, через его интерфейс. Когда вызывается событие представления, оно вызывает конкретный метод представителя, не имеющего ни параметров, ни возвращаемого значения. Представитель получает необходимые для работы метода данные о состоянии пользовательского интерфейса через интерфейс представления и через него же передаёт в представление данные из модели и другие результаты своей работы.

## ****2.2 Проектирование приложения****

Алгоритм работы приложения представлен на рисунке 1.

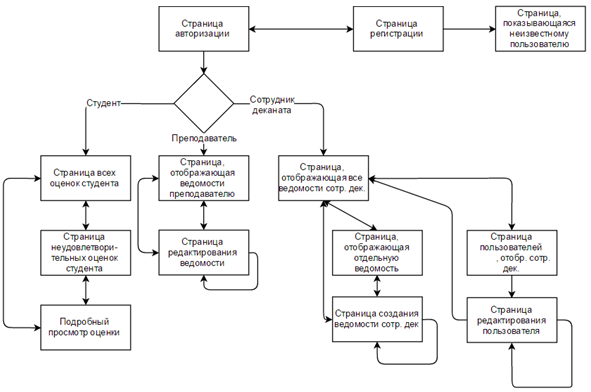


Рисунок 1. Алгоритм работы приложения

Принцип создания ведомости можно наблюдать на диаграмме последовательности, представленной на рисунке 2.

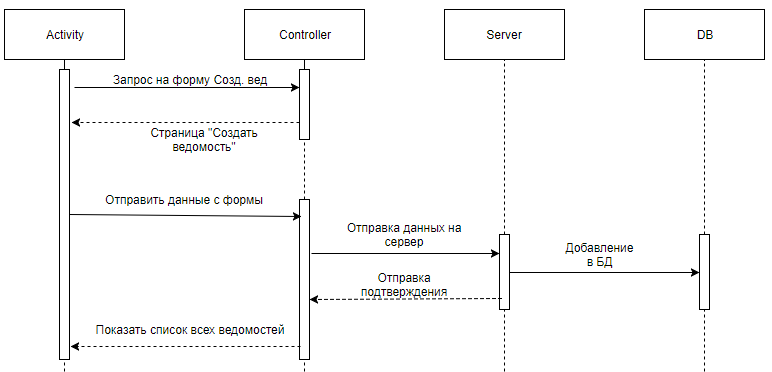


Рисунок 2. Диаграмма последовательности

Функции приложения можно наблюдать на диаграмме прецедентов, представленной на рисунке 3.

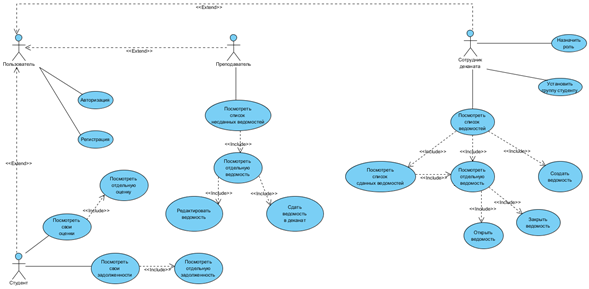


Рисунок 3. Диаграмма прецедентов

Диаграмма состояний ведомости представлена на рисунке 4.

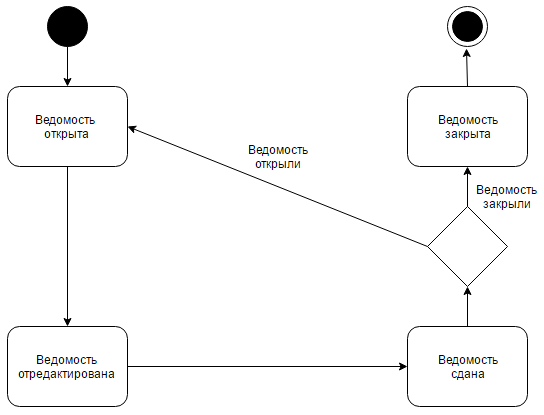


Рисунок 4. Диаграмма состояний ведомости

# ****3 РЕАЛИЗАЦИЯ****

## ****3.1 Диаграмма развертывания****

Наиболее наглядно работу приложения можно наблюдать на диаграмме развертывания, представленной на рисунке 5.

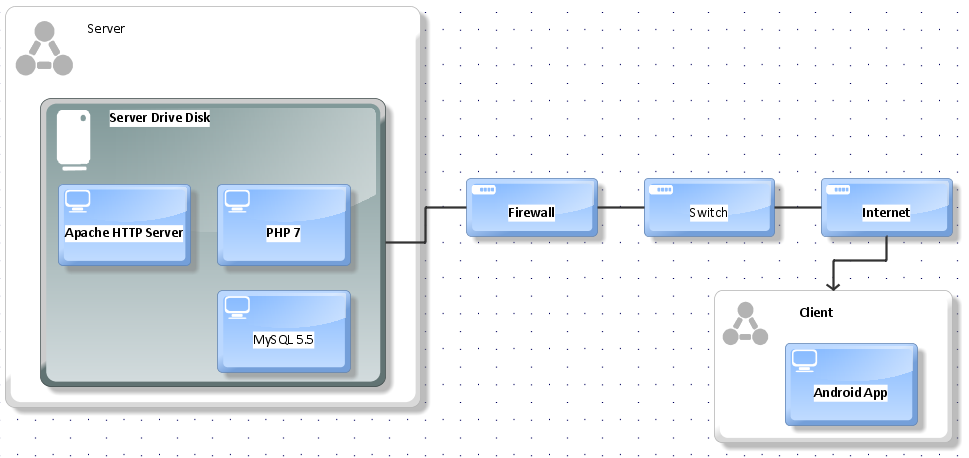


Рисунок 5. Диаграмма развертывания.

На сервере функционирует серверное ПО «Apache HTTP Server». Данные хранятся в базе данных MySQL. В Качестве контроллера выступает серверная часть, написанная на PHP.

В качестве клиента выступает Android приложение, написанное на Java.

## ****3.2 Пользовательский интерфейс.****

Для удобного взаимодействия пользователя с системой был разработан пользовательский интерфейс на клиенте, представленный на рисунках 6 - 19.

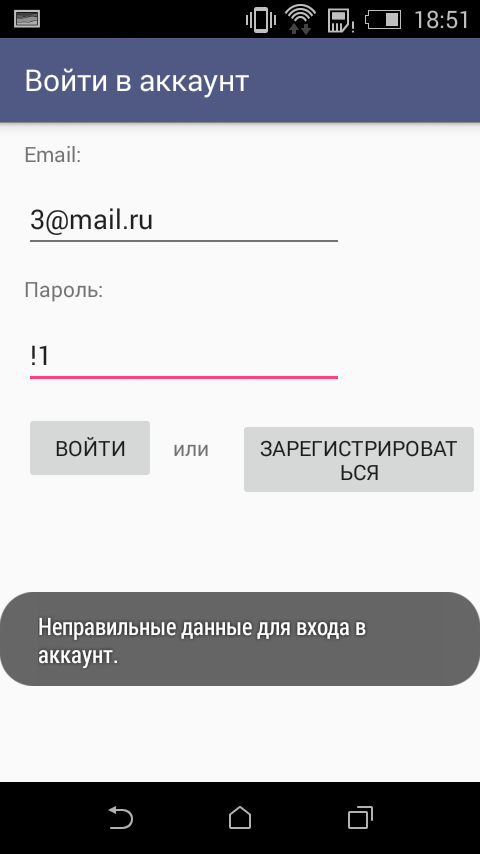


Рисунок 6. Страница входа с сообщением об ошибке.

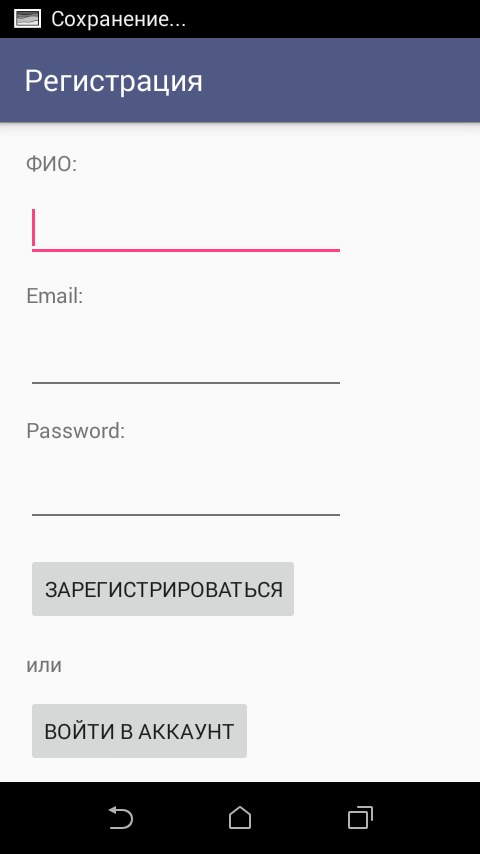


Рисунок 7. Страница регистрации.

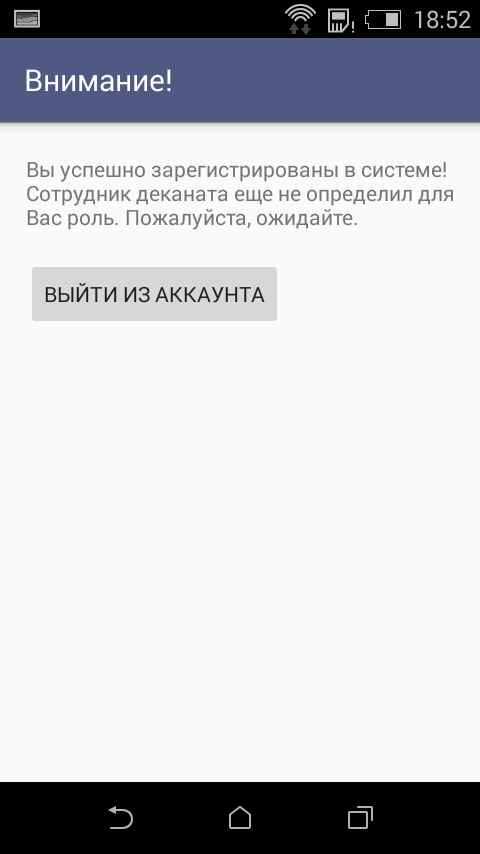


Рисунок 8. Страница пользователя с ролью неизвестного

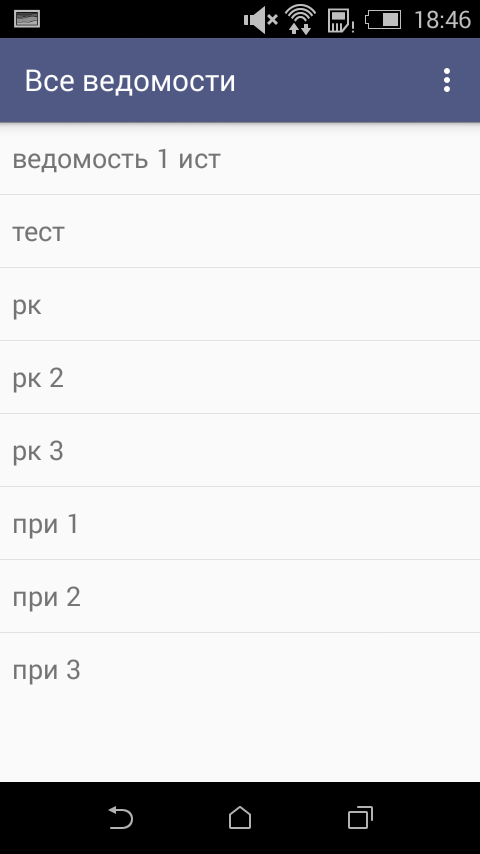


Рисунок 9. Страница списка всех ведомостей.

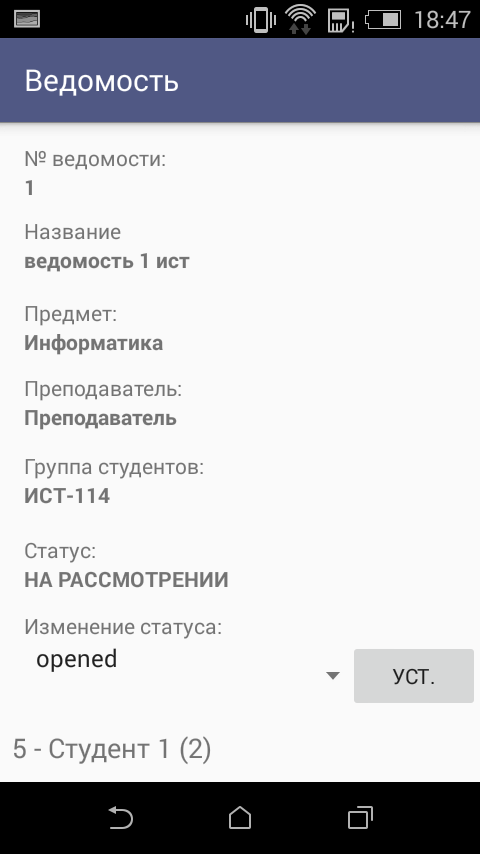


Рисунок 10. Страница отдельной ведомости для сотрудника деканата.

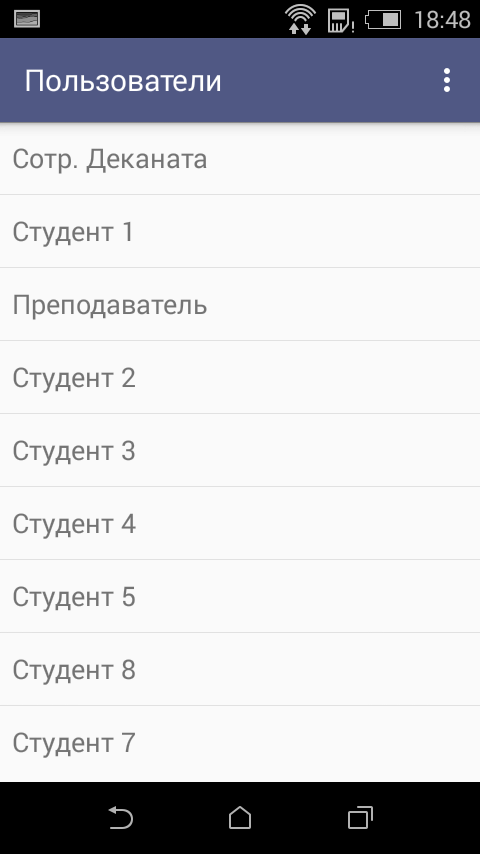


Рисунок 11. Список всех пользователей.

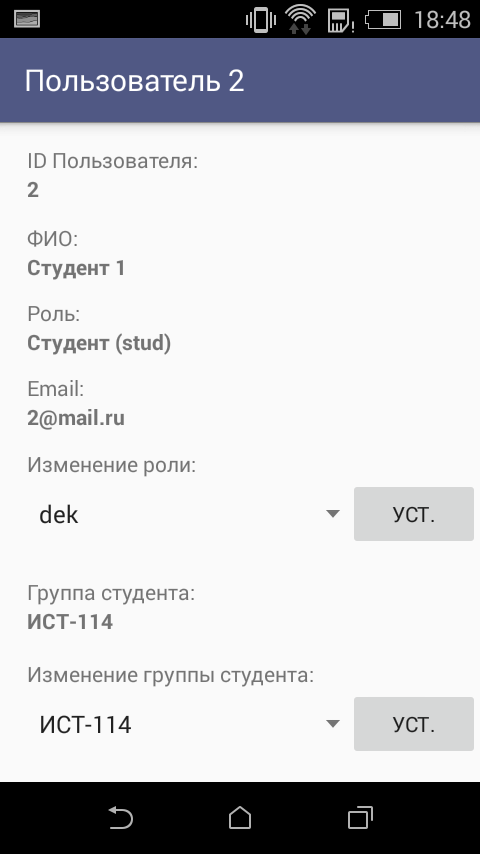


Рисунок 12. Страница отдельного пользователя.

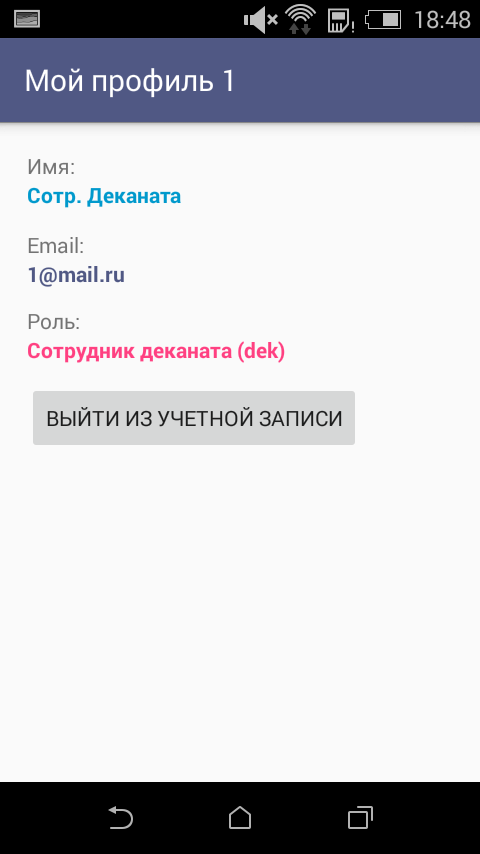


Рисунок 13. Страница профиля пользователя.

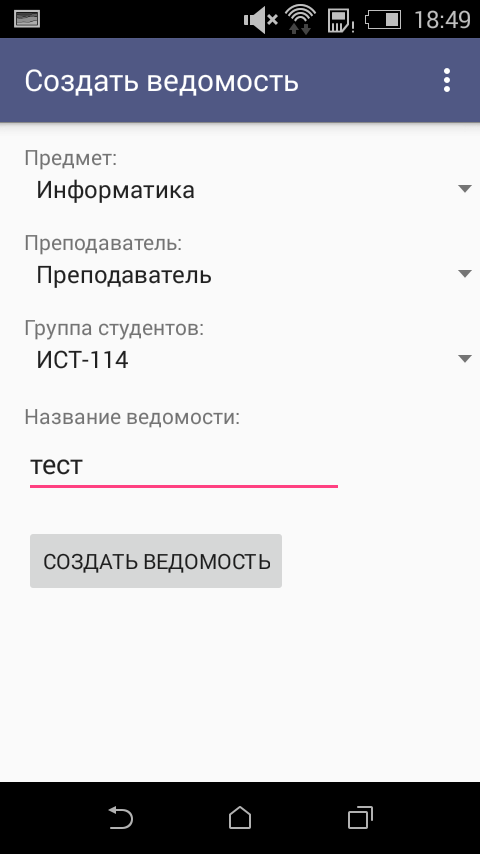


Рисунок 14. Страница создания ведомости.

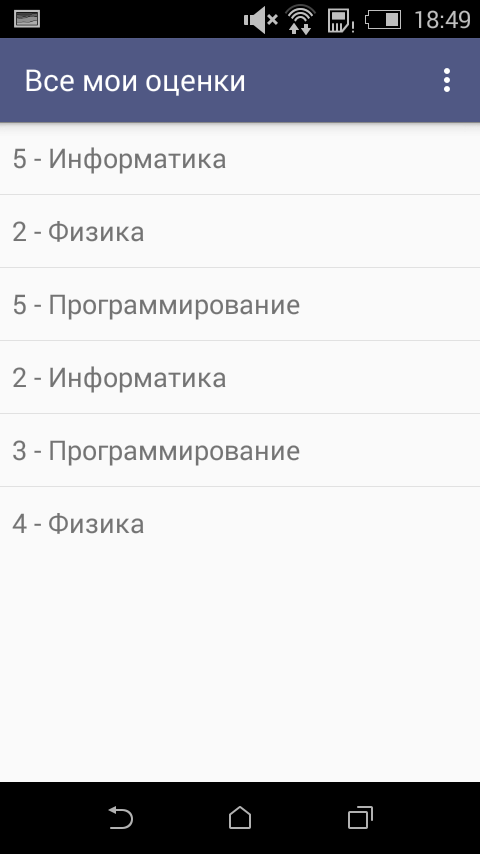


Рисунок 15. Страница списка оценок студента.

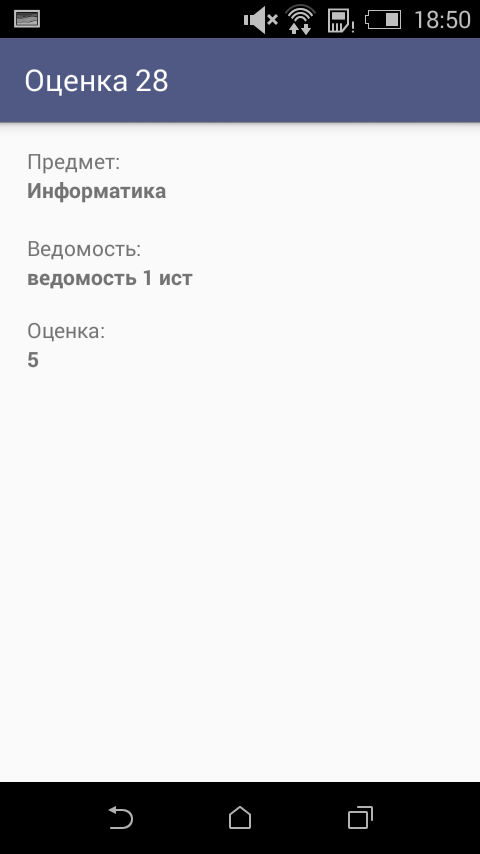


Рисунок 16. Страница отдельной оценки студента.

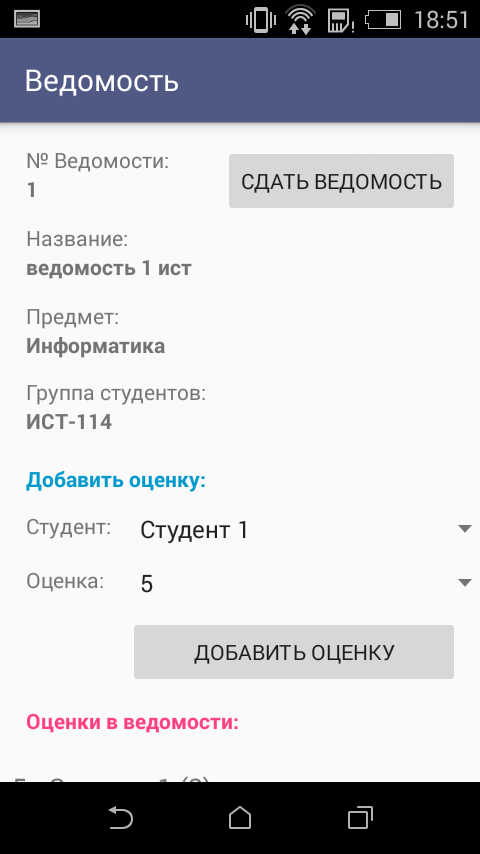


Рисунок 17. Страница редактирования преподавателем ведомости.



Рисунок 18. Страница закрытых преподавателем ведомостей.

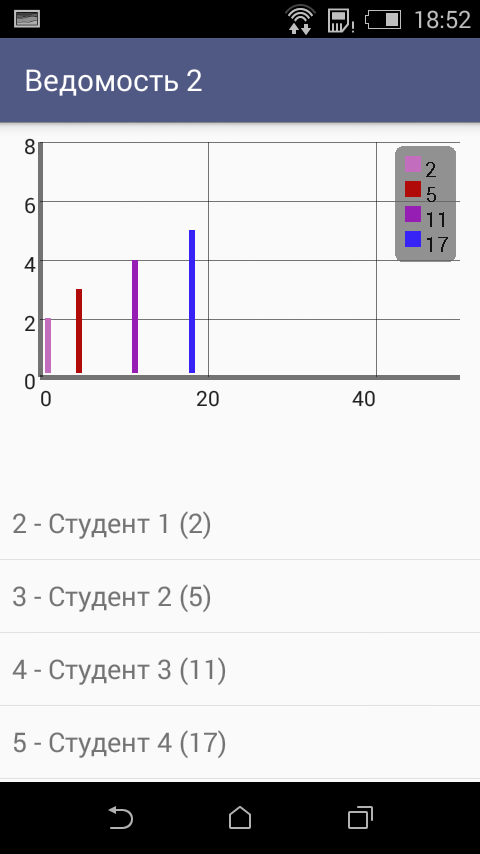


Рисунок 19. Отдельная закрытая преподавателем ведомость.

## ****3.3 Структура БД.****

Схема базы данных представлена на рисунке 20.

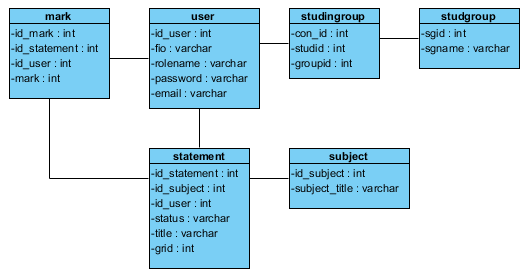


Рисунок 20. Схема базы данных

Структура таблиц базы данных представлена на рисунках 21-26.



Рис. 21 – структура таблицы Mark.

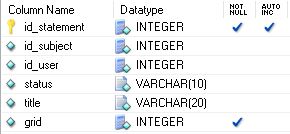


Рис. 22 – структура таблицы Statement

Снимок

Рис. 23 – структура таблицы studgroup



Рис. 24 – структура таблицы studingroup

Снимок

Рис. 25 – структура таблицы subject

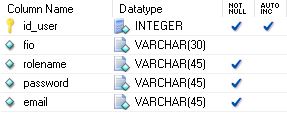


Рис. 26 – структура таблицы user

## ****3.4 Структура проекта на сервере.****

В ходе курсового проекта была написана серверная часть, состоящая из трех файлов (представлено на рисунке 27), написанная на PHP. Для удобного взаимодействия с Android приложением, файлы, находящиеся на сервере, выдают ответ в формате JSON (текстовый формат обмена данными, основанный на JavaScript).

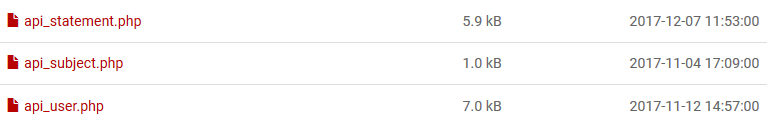


Рисунок 27. Структура проекта на сервере.

## ****3.4 Структура проекта на клиента.****

В ходе разработки клиентского приложения было создано несколько типов классов: активити, сущности и дополнительные классы.

Активити: DekActivity, DekActivityStatementItem, DekActivityUserItem, LoginActivity, MainActivity, PrepActivity, PrepActivityStatementItem, PrepGraph, PrepGraphItem, PrepMark, RegActivity, StudActivity, StudMark, UserProfile.

Сущности: entityMark, entityStatement, entityStudGroup, entitySubject, entityUser.

Дополнительные классы: FetchData, MySpinnerData.

Класс FetchData служит для обмена данными с сервером и обработки полученных от сервера данных.

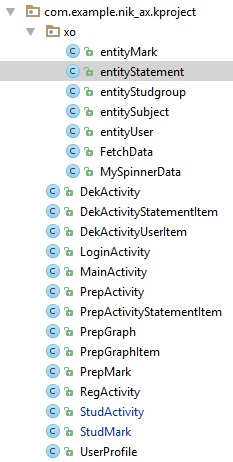


Рисунок 28. Структура проекта клиента.

Для обмена данными по интернету в файле AndroidManifest.xml было прописано соответствующее разрешение.

## ****3.5 Тестирование приложения.****

В ходе разработки приложения было проведено функциональное тестирование на устройстве HTC Dual Sim, работающего на операционной системе Android 4.4 KitKat.

Каждая из написанных функций исправно работает и, в случае, введения неверных данных, пользователю выдается сообщение об ошибке. К примеру, данное сообщение можно наблюдать на странице входа при неверно введенных данных (представлено на рисунке 6).

# **ЗАКЛЮЧЕНИЕ**

В ходе курсового проекта была изучена предметная область, спроектировано и разработано нативное мобильное приложение «Мобильная ведомость» для устройств, работающих на операционной системе Android.

Данное приложение существенно сократит время сотрудников деканата и преподавателей на обработку ведомостей, исключит вероятность ошибочной интерпретации данных, а также предоставит студентам удобный доступ к их оценкам, что является важным моментом в учете успеваемости студентов.

Разработанное мобильное приложение является актуальным на сегодняшний день и имеет большую практическую значимость. Оно может помочь в работе учебного заведения.

Данное приложение является многопользовательским, и у каждого типа пользователя имеются свои функции. В ходе реализации приложения были решены все задачи, поставленные на стадии проектирования.

В перспективе в приложении «Мобильная ведомость» можно улучшить дизайн, расширить функционал до масштабов целого университета, добавить учет посещения студентами занятий, добавить нативные версии для других операционных систем, либо сделать кроссплатформенное приложение.

При дальнейшем развитии проекта можно будет ввести электронную систему пропусков при помощи QR-кодов, что позволит сократить расходы на пропуска. Также возможно добавление в систему расписания занятий, и предстоящих событий, таких как: рейтинги, зачеты и экзамены.

В ходе выполнения курсового проекта мною был получен опыт разработки Android приложений, написания сервера на языке PHP, работы с JSON, а также распознавания результатов, полученных в формате JSON. Данный опыт является очень полезным и значимым.

# ****СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ****

1) Android NDK: руководство для начинающих [Электронный ресурс] / Ретабоуил Сильвен - М. : ДМК Пресс, 2016. - http://www.studentlibrary.ru/book/ISBN9785970603949.html

2) Android архитектура клиент-серверного приложения [Электронный ресурс] - https://habrahabr.ru/company/e-Legion/blog/265405/

3) Connect to MYSQL remote server database using PHP and display ListView [Электронный ресурс] - http://adblogcat.com/connect-mysql-remote-server-php/

4) Sending JSON Data to Server using Async Thread [Электронный ресурс] - http://blog.appliedinformaticsinc.com/sending-json-data-to-server-using-async-thread/

5) The Busy Coder's Guide to Android Development – CommonsWare : Version 8.8, 2017 - https://commonsware.com/Android/

6) Голощапов А.Л. - Google Android. Программирование для мобильных устройств

7) Хашими С., Коматинени С., Маклин Д. - Разработка приложений для Android – 2011

8) Рето Майер - Android 4. Программирование приложений

9) Android Recipes A Problem-Solution Approach By Dave Smith , Jeff Friesen.

10) П. Дейтел, Х. Дейтел, А. Уолд. Android для разработчиков. 3-е издание - 2016 [Электронный ресурс] - https://vk.com/doc147275941\_440987495

11) Брайан Харди, Билл Филлипс,Крис Стюарт, Кристин Марсикано.Программирование под Android. 2-е издание – 2016 [Электронный ресурс] - https://vk.com/doc291466009\_441195688

# ПРИЛОЖЕНИЕ А ЛИСТИНГ ПРОГРАММНОЙ СИСТЕМЫ

Доступно по ссылке <https://github.com/nikaxen/ACP>