

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХ-
СТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. Гумарбека Даукеева»
Кафедра IT-инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»

Зав. кафедрой _____

(ученая степень, звание, Ф.И.О.)

_____ «___» _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Разработка мобильного приложения «AUES Mobile». Подсистема
«Электронный журнал»

Специальность 5В070300 - Информационные системы

Выполнил Азизов М.М.

Группа ИС-16-2

Научный руководитель к.т.н., доцент Тусупова Б.Б.

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р.

(ученая степень, звание,

Ф.И.О.)

_____ «___» _____ 20__ г.
(подпись)

по безопасности жизнедеятельности: доцент Приходько Н.Г.

(ученая степень, звание,

Ф.И.О.)

_____ «___» _____ 20__ г.
(подпись)

по применению вычислительной техники: ст.преп. Майкотов М.Н.

(ученая степень, звание,

Ф.И.О.)

_____ «___» _____ 20__ г.
(подпись)

Нормоконтролер: ст.преп. Абсатарова Б.Р.

(ученая степень, звание,

Ф.И.О.)

_____ «___» _____ 20__ г.
(подпись)

Рецензент: _____

Ф.И.О.)

(ученая степень, звание,

_____ « _____ » _____ 20 ____ г.
(подпись)

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХ-
СТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. Гумарбека Даукеева»

Институт систем управления и информационных технологий
Кафедра IT-инжиниринг
Специальность 5В070300 - Информационные системы

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Азизову М. М.

Тема проекта «Разработка мобильного приложения «AUES Mobile».
Подсистема «Электронный журнал»»

Срок сдачи законченного проекта «___» _____ 20__ г.

Исходные данные к проекту (требуемые параметры результатов исследова-
ния (проектирования) и исходные данные объекта): постановленные за-
дачи, информационная система «Platonus», расписания групп университета.

Перечень вопросов, подлежащих разработке в дипломном проекте, или
краткое содержание дипломного проекта:

- а) определение основных целей и требований к ИС, разработка техни-
ческого задания проекта;
- б) проведение анализа существующих систем;
- б) разработка информационного обеспечения ИС;
- в) разработка интерфейса и программная реализация системы;
- г) расчет экономических показателей;
- д) расчет показателей по обеспечению безопасности жизнедеятельно-
сти.

Основная рекомендуемая литература:

1 Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программи-
рование для профессионалов. 2-е изд. — СПб.: Питер, 2016. — 640 с.: ил. —
(Серия «Для профессионалов»).

2 Шилдт, Герберт. Java 8. Полное руководство; 9-е изд.: Пер. с англ. -
М.: ООО «И.Д. Вильямс», 2015. - 1376 с.: ил. - Парал. тит. англ.

3 Гвоздева Т.В. Проектирование информационных систем. - Ростов-на-
Дону: Феникс, 2009.

4 Грекул В.И. и др. Проектирование информационных систем. - М.:
Интернет Университет Информационных Технологий, 2010.

5 Поляков Е. В. PHP на примерах. 2-е изд. - «Наука и техника», 2017. -
256с.

6 Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие - М.:
Postgress Professional, 2017. - 256 с.

7 Кузнецов М.В., Симдянов И.В. MySQL 5 - СПб.: БХВ-Петербург,
2010. - 1024 с.

8 Сьерра К., Бейтс Б. Изучаем Java. 2-е изд. — СПб.: Питер, 2012. — 708 с.: ил. — (Серия «Head First O'Reilly»).

9 Официальная документация Android на сайте <https://developer.android.com/>

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Нормоконтроль	Абсатарова Б.Б.	13.05.2020 - 18.05.2020	
Программное обеспечение	Майкотов М.Н.	14.05.2020 - 15.05.2020	
Экономическая часть	Габелашвили К.Р.	02.03.2020 - 10.04.2020	
Охрана труда и БЖД	Приходько Г.Н.	02.03.2020 - 13.04.2020	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Обзор литературы и анализ современного состояния вопроса	13.01.2020 - 13.02.2020	
Разработка информационного обеспечения системы	13.01.2020 - 10.03.2020	
Разработка программного обеспечения	13.01.2020 - 30.04.2020	
Экономическая часть	02.03.2020 - 10.04.2020	
Охрана труда и БЖД	02.03.2020 - 13.04.2020	

Дата выдачи задания «13» января 2020 г.

Заведующий кафедрой _____ Досжанова А.А.
(подпись)

Научный руководитель
проекта _____ Тусупова Б.Б.
(подпись)

Задание принял к
исполнению студент _____ Азизов М.М.
(подпись)

Аннотация

Дипломный проект отражает основные тенденции развития информационных технологий. Основной такой тенденцией в этой сфере является нарастающая популярность создания мобильных приложений для операционных систем Android. Целью проектирования является создание приложения «AUES Mobile» с функциями подсистем «Электронный журнал» и «Новости и объявления». Новизна заключается в том, что для университета не имеются аналоги мобильных приложений. Основное преимущество системы - быстрое получение актуальной информации о новостях и успеваемости студентов. Приложение проектировалось при помощи современных средств информационных технологий: объектно-ориентированное программирование, модернизированный UI/UX дизайн, СУБД и средства аналитики от ведущей технологической компании Google. По своей архитектуре программный продукт относится к клиент-серверным информационным системам. В результате проектирования была получена многопользовательская информационно-аналитическая, информационно-поисковая, информационно-справочная система.

Андатпа

Дипломдық жоба ақпараттық технологиялар дамуының негізгі бағыттарын көрсетеді. Бұл бағыттағы негізгі үрдіс - Android операциялық жүйелеріне арналған мобильді қосымшаларды жасаудың өсіп келе жатқан танымалдығы. Жобаның мақсаты: «Электрондық журнал» мен «Жаңалықтар мен Хабарландырулар» ішкі жүйелерінің функциялары бар AUES мобильді қосымшасын құру. Жаңалық - университет үшін мобильді қосымшалардың аналогтары жоқ. Жүйенің басты артықшылығы - жаңалықтар мен студенттердің үлгерімі туралы тиісті ақпаратты жедел алу. Қосымша заманауи ақпараттық технологиялардың көмегімен жасалған: объектіге бағытталған бағдарламалау, заманауи UI/UX дизайны, DBMS және Google жетекші технологиялық компаниясының аналитика әдістері. Өзінің архитектурасы бойынша бағдарламалық өнім клиент-сервердің ақпараттық жүйелеріне жатады. Жобалау нәтижесінде көп қолданушыларға арналған ақпараттық-аналитикалық, ақпараттық-ізвестіру, ақпараттық-анықтамалық жүйе құрылды.

The Summary

The graduation project reflects the main trends in the development of information technology. The main trend in this area is the growing popularity of creating mobile applications for Android operating systems. The design goal is to create the «AUES Mobile» application with the functions of the subsystems «Electronic Journal» and «News and Announcements». The novelty is that for the university there are no analogues of mobile applications. The main advantage of the system is the rapid receipt of relevant information about the news and student's academic performance. The application was designed using modern remedies of information technology: object-oriented programming, modernized UI/UX design, DBMS and analytics from the leading technology company Google. By its architecture, the software product relates to client-server information systems. As a result of the design, a multi-user information-analytical, information-retrieval, information-reference system was obtained.

Содержание

Введение	11
1 Обзор литературы и анализ современного состояния вопроса.....	12
1.1 Обзор литературы.....	12
1.2 Анализ существующих систем	13
1.3 Постановка задачи.....	19
2 Разработка информационного обеспечения системы	22
2.1 Проектирование функциональной структуры	22
2.2 Разработка бизнес модели ИС	26
2.3 Описание информационного обеспечения	33
2.4 Firebase Analytics	58
3 Разработка программного обеспечения	60
3.1 Обоснование выбора среды программирования.....	60
3.2 Построение макета пользовательского интерфейса	62
3.3 Структура программного обеспечения	82
3.4 Клиентская часть	83
3.5 Серверная часть	86
3.6 Тестирование и отладка программного обеспечения.....	90
4 Экономическое обоснование	100
4.1 Определение объема и трудоемкости разработки программного обеспечения	100
4.2 Расчет затрат на разработку информационных технологий	102
4.3 Расчет динамических показателей эффективности капитальных вложений	107
5 Охрана труда и БЖД	111
5.1 Анализ потенциально опасных и вредных факторов, воздействующих на обслуживающий персонал в процессе труда.....	111
5.2 Расчетная часть.....	115
Заключение	126
Список использованной литературы.....	127
ПРИЛОЖЕНИЕ А	
Техническое задание	129
ПРИЛОЖЕНИЕ Б	
Листинг программы	134
ПРИЛОЖЕНИЕ В	
Авторское свидетельство.....	169

Введение

В современном обществе все больше растет число пользователей смартфонов и люди меньше тратят время на использование ноутбуков и ПК для досуга. Кроме того, некоторые сайты, которые открываются на смартфонах не адаптированы под них и бывают трудно читабельными и плохо влияют на глаза.

Пользователю не придётся постоянно открывать браузер и запоминать уйму DNS серверов или вбивать нужный им сайт в браузере, достаточно открыть мобильное приложение, в котором все интуитивно понятно и просто. Также современные инструменты позволяют сделать мобильные приложения поистине более функциональными нежели сайты. Удобство и простота для пользователя являются самыми важными аспектами. В данной работе анализируется и разрабатывается приложение «AUES Mobile» для студентов и преподавателей Алматинского университета энергетики и связи им. Г. Даукеева по заказу самого АУЭС. Как было указано выше, данное приложение будет обладать всем достоинствами современных мобильных приложений. Основными задачами является обеспечение пользователей электронным журналом и актуальными новостями.

Целью данной работы является разработка мобильного приложения «AUES Mobile», а именно подсистемы «Электронный журнал» основным функционалом которого является представление информации об оценках студентов и выставление оценок преподавателями. Студентам доступна функция просмотра транскрипта, календарного плана, а также новостей и объявлений. В приложении должно быть осуществлено ролевое разделение (студент, сотрудник, преподаватель, администратор).

Основными задачами дипломного проектирования являются:

- определение основных целей и требований к ИС (описание постановки задачи и разработка технического задания);
- разработка информационного обеспечения системы (спроектировать функциональную структуру системы, разработать бизнес-модели ИС, описать информационное обеспечение, обосновать выбор СУБД);
- разработка программного обеспечения (описать язык разметки, построить макеты пользовательского интерфейса, описать структуру программного обеспечения, обосновать выбор инструментального ПО, описать клиентскую и серверную части, произвести тестирование и отладку ПО с целью выявления ошибок и их дальнейшего предотвращения);
- рассчитать экономическое обоснование проекта и рассмотреть вопросы охраны труда.

1 Обзор литературы и анализ современного состояния вопроса

1.1 Обзор литературы

В ходе работы были использованы научная литература, статьи и ГОС-Ты. В качестве основной литературы используется книга «Android программирование для профессионалов» авторов Б.Филиппс, К. Стюарт, К.Марсикано. Данная книга содержит все исчерпывающие знания, которые необходимы для написания мобильного приложения под операционную систему Android. Книга описывает основные концепции разработки приложений, знакомит читателя с интегрированной средой разработки Android Studio. Описаны правила построения макетов на языке XML. Удобство данного источника заключается в том, что в данной книге используются примеры и рецепты с использованием языка Java. В основном книга используется как подручный инструмент, поскольку наиболее исчерпывающим источником для Android-программистов является официальная документация, размещенная на сайте <https://develo-per.android.com/>. [1]

Однако, прежде чем приступить к ознакомлению вышеназванных источников, был изучен язык программирования Java. В качестве литературы для изучения данного языка является книга Герберта Шилдта «Java 8. Полное руководство». Данная книга является исчерпывающим и незаменимым учебным пособием по написанию, компилированию и выполнению современных программ на Java. В ней приводятся основные особенности и достоинства языка. Кроме того, четко разъяснены основные понятия объектно-ориентированного программирования, такие как классы и объекты, а также три кита, на которых лежит ООП: инкапсуляция, полиморфизм и наследование. Все три концепции имеют место при создании мобильного приложения. Рассмотрены такие основные понятия, как типы данных, операции, управляющие операторы, классы, наследование, модификаторы доступа, обработка исключений. При изучении языка Java использовалась IDE IntelliJ IDEA 2018. [2]

Любая информационная система не пишется без должных знаний о базах данных и систем управления базами данных. В качестве подручных средств при дипломном проектировании были использованы несколько книг по базам данных и СУБД. Это книга Е.П.Моргунова «Язык SQL», М.В.Кузнецова и И.В.Симдянова «MySQL 5». Книга «Язык SQL» служит для получения практических навыков и начальных знаний по языку структурированных запросов. Описываются основные и довольно исчерпывающие понятия о структуре простых и сложных запросов. В мобильном приложении в основном используются сложные запросы с соединением. Основным недостатком данной книги является то, что при написании примеров используется СУБД PostgreSQL. Второй источник же имеет яркое представление об использовании СУБД MySQL, которая используется в дипломном проектировании. В книги описываются вложенные запросы, хранимые процедуры,

функции, триггеры, также имеются примеры с интеграцией MySQL с серверным языком программирования PHP. В книге имеются не только навыки по созданию функций, а также готовые - математические, строковые, функции агрегирования, которые широко используются при написании запросов в данном проекте. [3]

В качестве серверного языка программирования в ходе разработки серверной части мобильного приложения был использован PHP. Навыки по этому языку черпались из книги «PHP на примерах» Е.В. Полякова. Особенность данного источника состоит в том, что он наполнен больше практическими мини проектами, чем теорией о языке, такими как: подключение других файлов как части сценария, включение волшебных кавычек, предотвращение атак типа SQL Injection. [4]

Также важной частью работы является правильное проектирование информационной системы мобильного приложения. Учебное пособие Т.В. Гвоздева «Проектирование информационных систем» наглядно показывает, как правильно описывать функциональную структуры системы, бизнес-процессы.

Помимо вышеназванных источников в ходе проектирования были использованы множество статей с тематических форумов по программированию, ссылки которых указаны в списке литературы (знания, касающиеся различных библиотек Android, UML-диаграммы, библиотеки для передачи и принятия данных между сервером и клиентом и др.). [5]

1.2 Анализ существующих систем

В данном разделе рассматриваются приложения таких университетов, как КазНУ, ИТМО, Нархоз, НИЯУ МИФИ, МГИМО, СФУ и Массачусетский технологический институт, для выявления их возможностей, достоинств и недостатков, что может помочь в улучшении функциональной части приложения «AUES Mobile».

КазНУ

Официальное приложение КазНУ им. аль-Фараби - Univer. «Univer» - это приложение для быстрого и удобного доступа к информационным ресурсам университета. Основные функции данного приложения:

- получение и просмотр последних новостей, связанных с учебным процессом;
- просмотр расписания занятий и последние баллы в журнале успеваемости;
- получение информацию по текущей аттестации
- скачивание учебных материалов.

Дата выхода: 9 сентября 2015.

Приложение включает в себя новости, интерфейс данной подсистемы показан на рисунке 1.1, расписание - интерфейс подсистемы предоставлен на рисунке 1.2, текущую аттестацию - интерфейс показан на рисунке 1.3, жур-

нал посещений и успеваемости - рисунок 1.4, рубежный контроль - рисунок 1.5, УМКД - рисунок 1.6.



Рисунок 1.1 - Новости

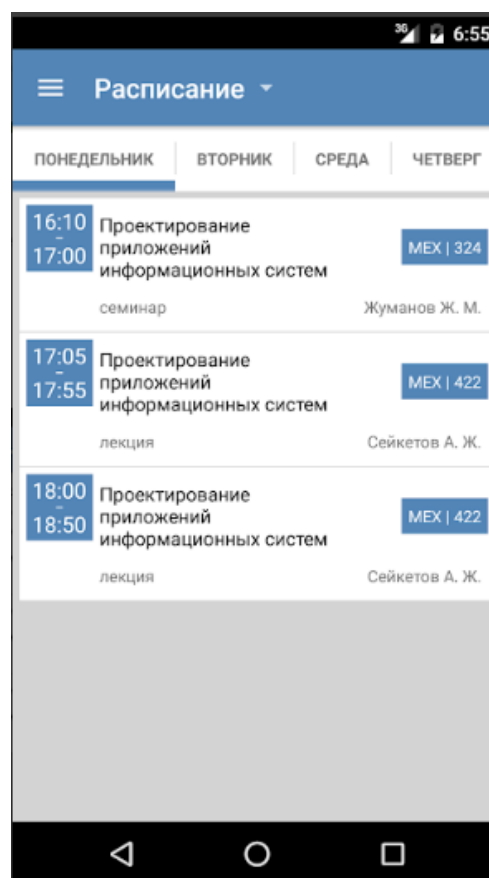


Рисунок 1.2 - Расписание

Общая оценка приложения составляет 4.1 на основе 893 мнений.

В целом приложением довольны, но нет функции уведомления об оценках и изменении в расписании. Сбои в работе «Текущая аттестация», множество незначительных сбоев в работе приложения. Приложение было оптимизировано до Android 9.0, что способствовало увеличению охвата аудитории среди пользователей. На некоторых устройствах не скачиваются файлы с «УМКД», хотя доступ на скачивание разрешен. Пользователями обнаружены ошибки с сетью, сервер приложения был недоступен: ошибка - «Нет соединения». Некоторых пользователей не устраивает интерфейс из-за его архаичности, хотя он простой, понятный и дружелюбный. Данное приложение включает в себя 12 университетов помимо КазНУ (список можно увидеть на рисунке 1.7), однако корректная работа приложения наблюдалась только при выборе КазНУ. [6]



Рисунок 1.3 - Текущая аттестация

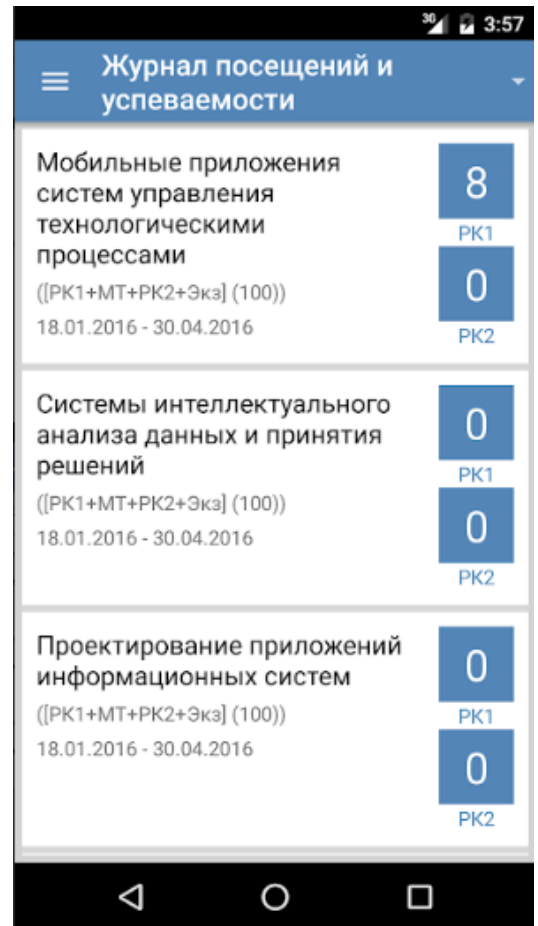


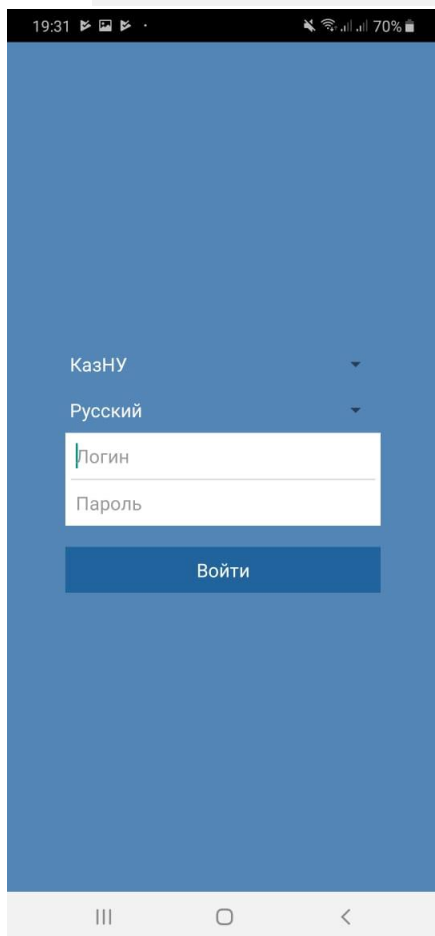
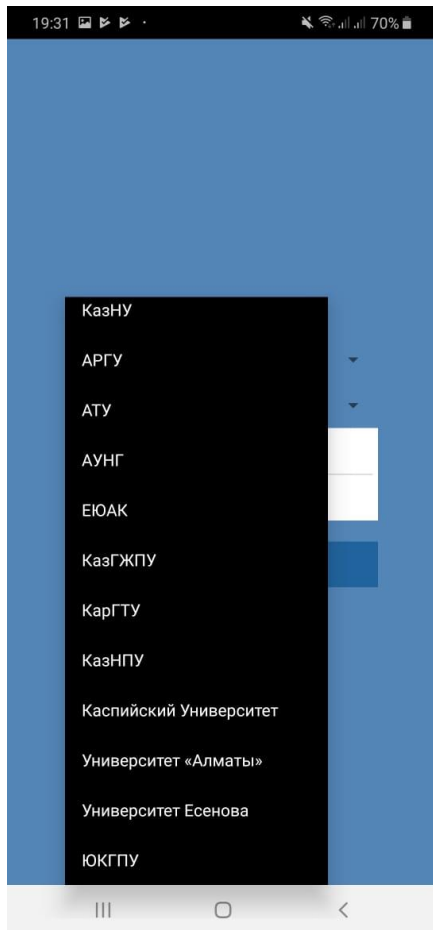
Рисунок 1.4 - Журнал



Рисунок 1.5 - РК



Рисунок 1.6 - УМКД



ИТМО

Официальное приложение Университета ИТМО включает в себя следующие возможности [7]:

- таймлайн с расписанием занятий и персональными новостями;
- стипендия и зарплата;
- рейтинг и баллы ЦДО;
- новости и мероприятия университета;
- расписание занятий групп, преподавателей и аудиторий;
- информация о сотрудниках и подразделениях университета;
- карта учебных корпусов и общежитий.

Общая оценка приложения - 3,5 на основании 297 голосов. Интерфейсом пользователи довольны, однако при работе наблюдаются множество ошибок: вылеты из приложения при нажатии на «Расписание», при попытке входа в учётную запись ИСУ неизвестная ошибка, временами не отображается стипендия. Пожелания: добавить больше функций и просмотр расписания в режиме оффлайн, хотелось бы еще видеть вкладку «Справочник телефонов и электронных адресов» как на главном сайте вуза. На рисунке 1.9 изображено меню приложения, новости и информация об университете. На рисунке 1.10 изображены информация о преподавателях и подразделениях университета.

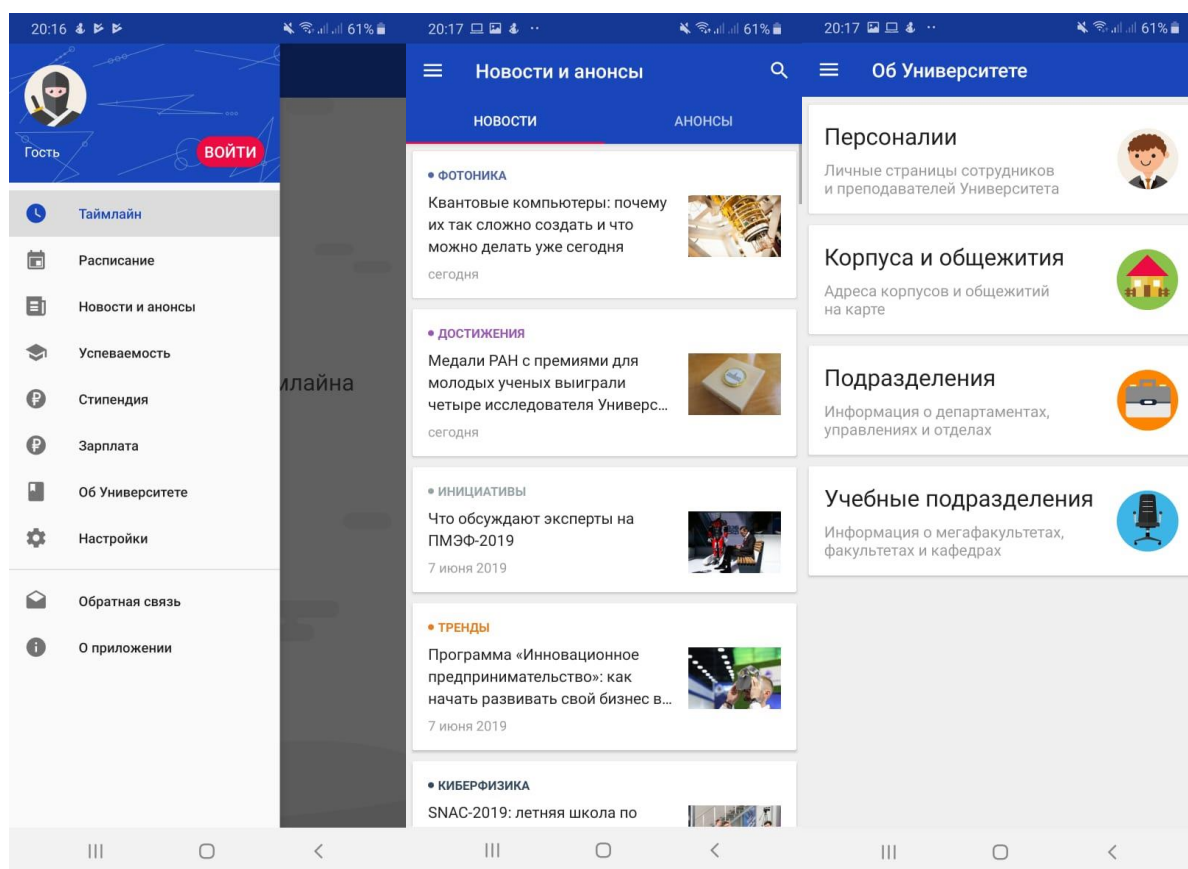


Рисунок 1.9 - Страница входа, новости и информация об ИТМО

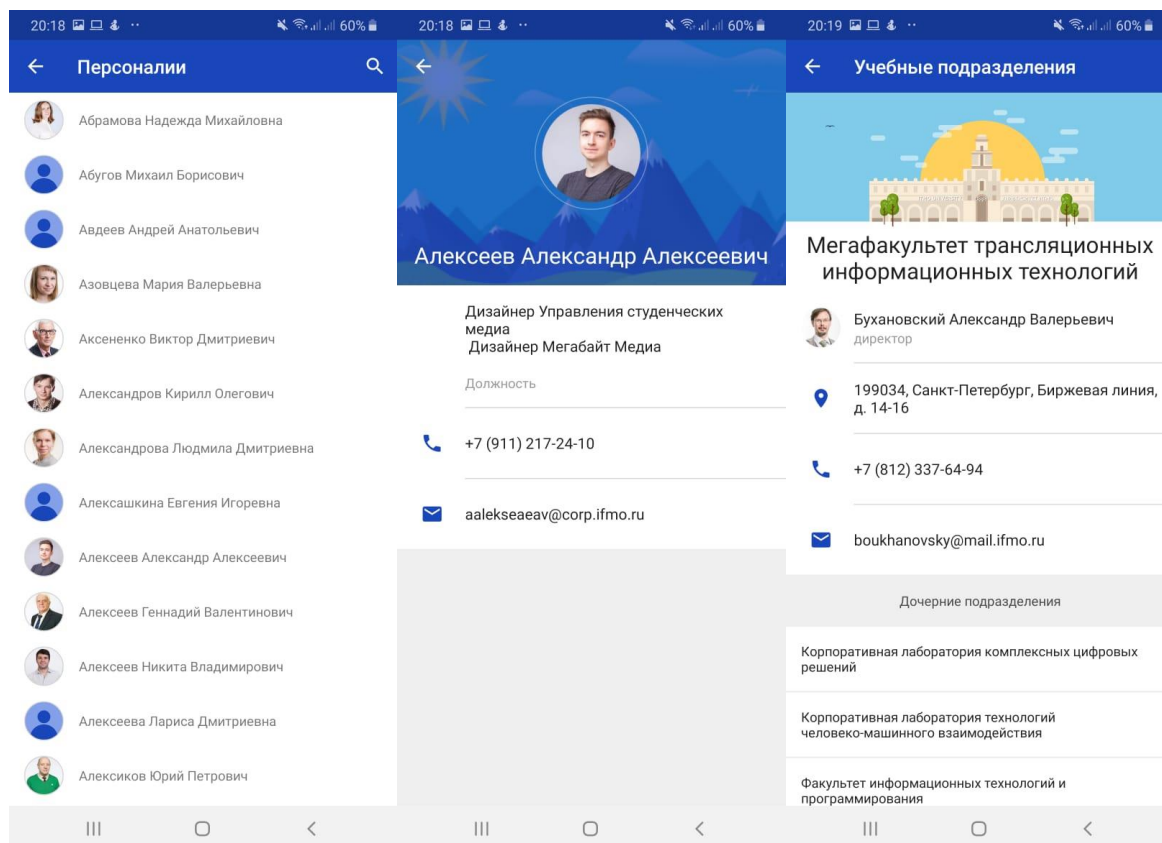


Рисунок 1.10 - Список преподавателей, информация о преподавателе и учебные подразделения

Нархоз

Мобильное приложение Университета НАРХОЗ для доступа к информационным ресурсам Университета и к информационной системе Ellucian Banner Student. Общая оценка: 3,3 на основе 78 голосов. В общем отзывы о приложении отрицательные. Интерфейс на Galaxy S8 плохо выглядит, всё громоздкое (увеличенное). Чтобы узнать номер аудитории нужно каждый раз нажимать на дисциплину в расписании. При просмотре текущих оценок на LG G7 не отражается страница. При каждом запуске нужно проходить аутентификацию заново, нет возможности сохранить данные для автоматического входа. Никакие работы не были проведены со студентами по использованию приложения. Основные разделы приложения закрыты от неавторизованных пользователей. Интерфейс показан на рисунке 1.11. [8]

Таким образом, проведя аналитику существующих приложений, учитывая их недостатки и достоинства, а также согласно ТЗ, будущее приложение для Алматинского Университета Энергетики и Связи имени Г. Даукеева будет предоставлять итоговые, промежуточные и текущие оценки для студента. Приложение будет включать в себя также новости и объявления, которыми будет управлять администратор. Для него будет создана функция добавления новостей и объявлений прямо из приложения, доступ к которой будут иметь только пользователи с полномочиями администратора. Для каждого пользователя будет создан личный кабинет, в котором будут реализованы

функции конкретно под роль пользователя. Например, транскрипт, академический календарь для студента. В правом меню будут находиться основные вкладки и изображение пользователя с его идентифицирующими данными. Для того, чтобы выявить функции, которыми больше всего пользуются, будет использована аналитическая система от Google Firebase Analytics. Кроме того, она поможет выявить функции, которые пользователи не используют. В дальнейшем это приведет к удалению или модернизации функций.

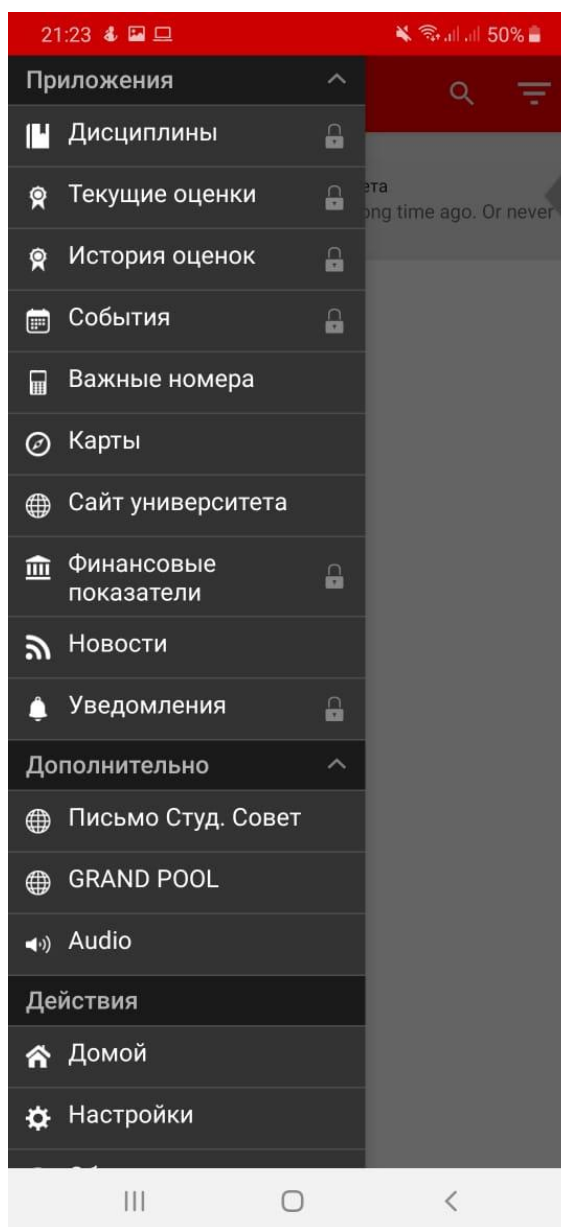


Рисунок 1.11 - Главная страница

1.3 Постановка задачи

Исходя из проведенного выше анализа было определено, что необходимо разработать мобильное приложение, которое решало бы следующие задачи:

- предоставить информацию об оценках студента, в которой будут отображаться даты и полученная за эту дату оценка, дисциплина;
- предоставить информацию о транскрипте, в которой будут отображаться дисциплины и полученные за данную дисциплину итоговые оценки в буквенном, традиционном и числовом эквиваленте;
- обеспечить получение информации о новостях и объявлениях, касающихся мероприятий и событий университета;
- обеспечить администраторов функцией добавления новостей и объявлений;
- предоставить преподавателям функцию, посредством которой будут выставляться итоговые и текущие оценки;
- организовать сбор данных при помощи современного средства Firebase Analytics, то есть проследить все статистические события пользователей;
- предоставить информацию об академическом календаре текущего учебного года;
- обеспечить студентов информацией о рубежных и экзаменационных оценках.

Требования к информационному обеспечению:

- необходимо изучить учебный процесс, систему «Platonus» и официальный сайт АУЭС им. Г.Даукеева в качестве источников информации. Далее на основе полученной и анализированной информации приступить к выделению сущностей, а затем к построению логической модели базы данных, которая должна будет отвечать следующим требованиям:
 - название сущностей и атрибутов необходимо писать в именительном падеже и на русском языке, исключая таким образом противоречивость;
 - в каждой сущности должны быть определены первичные ключи, которые будут однозначно идентифицировать запись той или иной сущности;
 - в проектируемой модели необходимо исключить лишние связи без потери данных;
 - необходимо привести базу данных в первую нормальную форму (исключить атрибуты с списком значений);
 - база данных должна быть приведена ко второй нормальной форме (каждый неключевой атрибут должен зависеть только от потенциального ключа);
 - привести базу данных в третью нормальную форму (исключить транзитивные зависимости неключевых атрибутов от потенциального ключа);

- выбрать СУБД которая будет обеспечивать безопасность данных, их целостность и надежное хранение. СУБД необходима для средней базы данных, кроме того СУБД должна быть с бесплатным лицензированием, поскольку финансирование проекта не определено;

- на основании выбранной СУБД каждому столбцу каждой таблицы присвоить типы данных и размер, где необходимо, поддерживаемые выбранной системой управления базами данных;

- исключить возможность использования следующего типа данных: CHAR (использовать VARCHAR2).

На основе логической модели БД построить реляционную модель, которая будет отвечать следующим ниженазванным требованиям:

- исключение связи многие-ко-многим, поскольку такая связь на данном уровне не допустима;

- отношения и столбцы должны быть написаны на английском языке;

- настроить в ссылочных таблицах опцию CASCADE.

Требования к программному обеспечению:

- версия Android 4.2 Jelly Bean и выше;

- оперативная память 1 ГБ и выше;

- минимальная частота процессора 1 ГГц;

- наличие свободного места на устройстве: 100 МБ и выше;

- поддержка GPS и ГЛОНАСС;

- наличие интернет-соединения;

- двухъядерный процессор (Dual Core) и лучше.

2 Разработка информационного обеспечения системы

2.1 Проектирование функциональной структуры

Организационная структура НАО «АУЭС» им. Г. Даукеева показана на рисунке 2.1.

Исходя из организационной структуры НАО «АУЭС» им. Г. Даукеева в данной информационной системе (ИС) можно выделить следующие подсистемы, касающиеся обучающихся и преподавателей:

1) подсистема «Электронный журнал». Основные функции: хранение и вывод информации об оценках студента, транскрипте, академическом календаре, индивидуальном учебном плане, оплате за обучение, задолженности и стипендии. Подробные привилегии того или иного пользователя описаны на схеме 1 в подсистеме «Личный кабинет». Подсистема разделена на модули в зависимости от типа пользователя. Модуль «Сотрудник» не связан с подсистемой «Электронный журнал». Кроме того имеется функция выставления оценок преподавателями. Преподаватель может выставить как текущие оценки, так и оценки промежуточных итогов (экзамен, рубежные контроли, расчетно-графические работы, курсовые работы). Каждая дисциплина имеет свое описание, в котором четко указан вес каждой работы. Преподаватель сам решает какие работы оцениваются больше, какие меньше;

2) подсистема «Новости и объявления». Основные функции: хранение и вывод информации о новостях и объявлениях. Имеется функция специально для администратора, суть которой заключается в добавлении новостей и объявлений. Пользователь может просматривать каждую новость отдельно. Причем при отображении поддерживается формат разметки HTML. Также приложение имеет возможность считывать PDF-файлы;

3) дополнительно к приложению подключена аналитическая система от Google - Firebase Analytics. Firebase представляет собой набор различных инструментов, которые помогает разработчикам мобильных приложений разрабатывать, развивать и зарабатывать больше денег со своими проектами. Аналитика важна для создания успешного приложения, поэтому разработчикам приложений так много инструментов для аналитики. В приложении есть поведенческая аналитика, которая измеряет, кто ваши пользователи, что они делают и так далее. Кроме того, существует атрибутивная аналитика, которую вы можете использовать для измерения эффективности вашей рекламы в других кампаниях роста, не говоря уже о push-уведомлениях и отчетах о сбоях. Можно также проверять, какими функциями наиболее часто используют пользователи мобильного приложения. То есть логика состоит в том, что, если в приложении имеются не используемые функции, то их можно удалить из приложения с целью снижения нагрузки. А имеющиеся приоритетные функции модернизировать.

Функциональная структура самой системы представлена на рисунках 2.2-2.3.

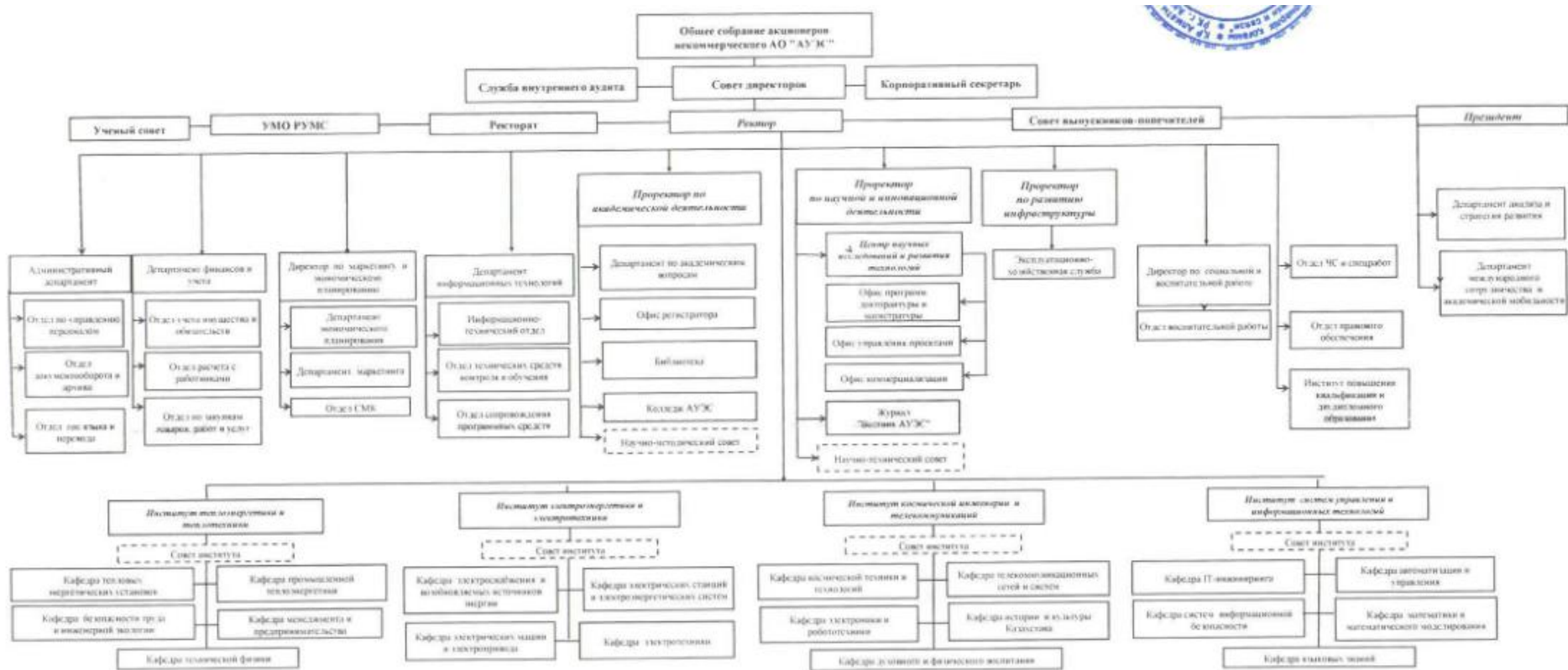


Рисунок 2.1 - Организационная структура университета

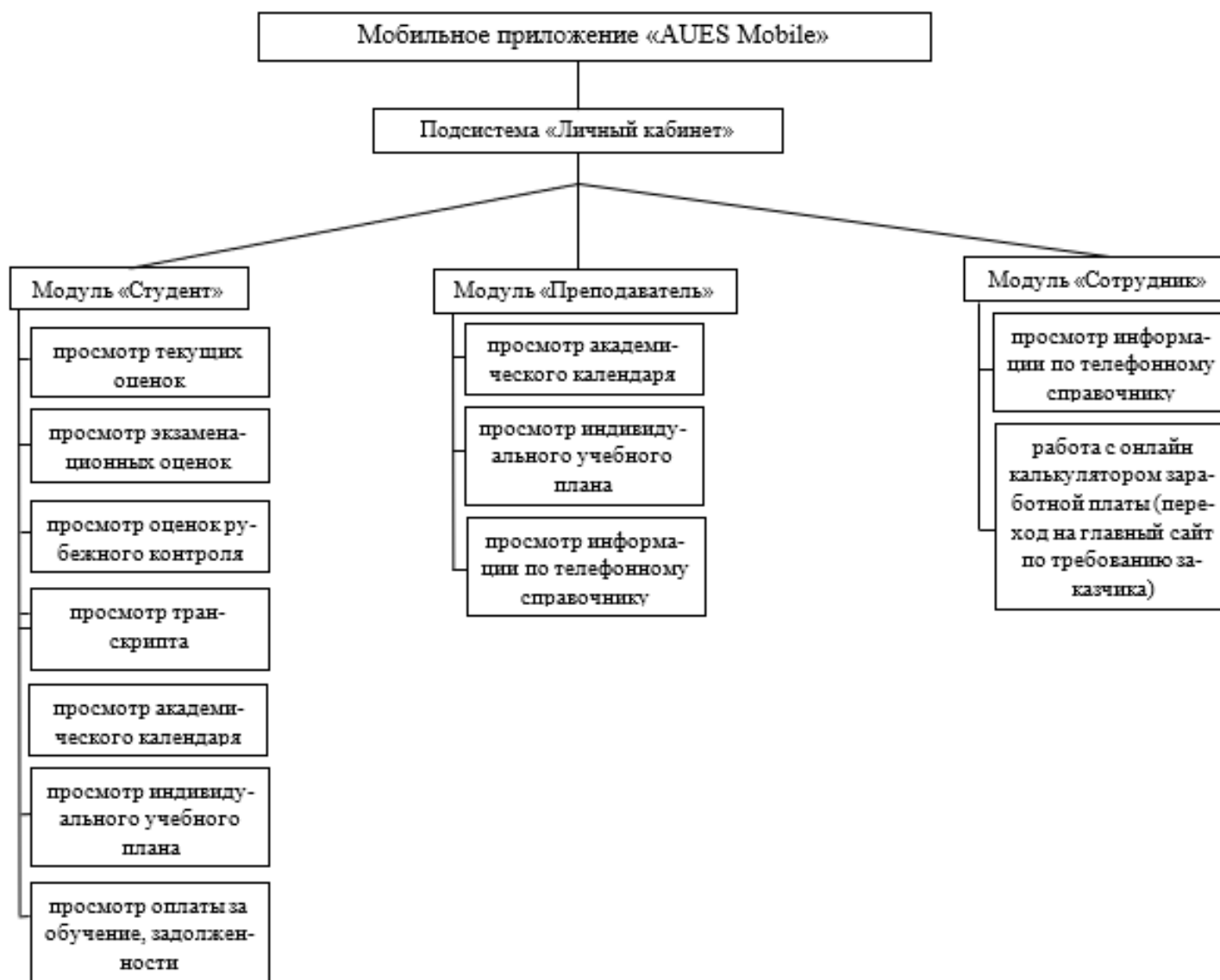


Рисунок 2.2 - Функциональная структура подсистемы «Личный кабинет»

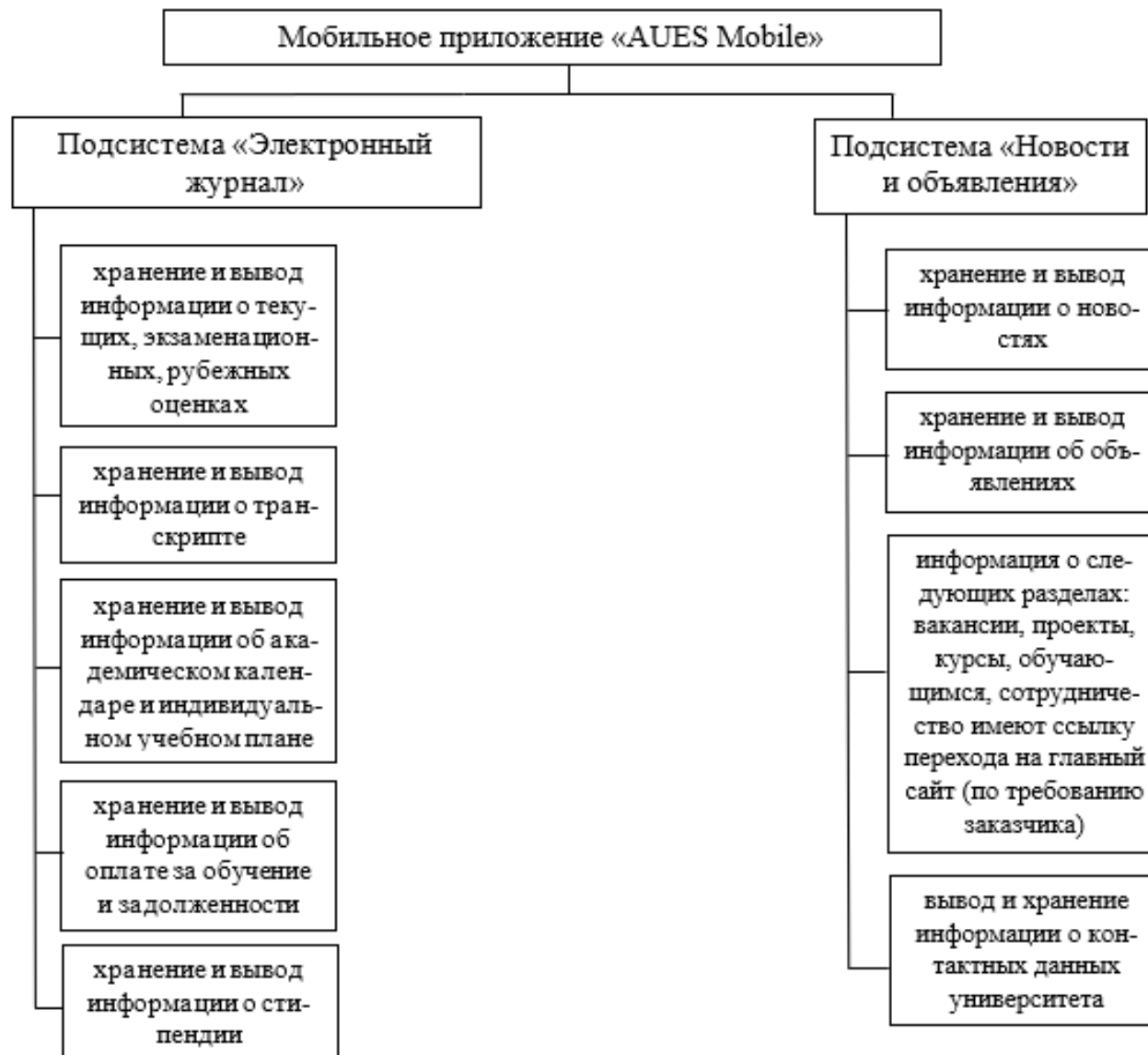


Рисунок 2.3 - Функциональная структура подсистем «Электронный журнал» и «Новости и объявления»

2.2 Разработка бизнес модели ИС

2.2.1 Предметная область

Расписание представляет собой таблицу, в которой хранятся такие данные, как номер занятия, время занятия, наименование группы, наименование дисциплины, номер аудитории, Ф.И.О преподавателя, вид занятия.

Электронный журнал представляет собой таблицу, в которой хранятся данные об оценках студента, Ф.И.О преподавателе, выставившего оценку, наименовании дисциплины и дате выставления оценки. Кроме того, данная подсистема хранит также информацию об академическом календаре, индивидуальном плане, задолженности, оплате и стипендии.

2.2.2 Предпроектное обследование

2.2.2.1 Функциональная и информационная модели

2.2.2.1.1 Функциональная модель

Главное назначение информационной системы университета является обеспечение надлежащих людей надлежащей информацией. Пользователем инициируется событие, далее мобильное приложение отправляет тот или иной запрос серверу базы данных. Сервер обрабатывает запрос и передает приложение, приложение выводит данные в удобном для пользователя виде. Кроме того, события регистрируются в Firebase Analytics и Google Analytics и хранятся там. Функциональная модель ИС представлена на рисунке 2.4.

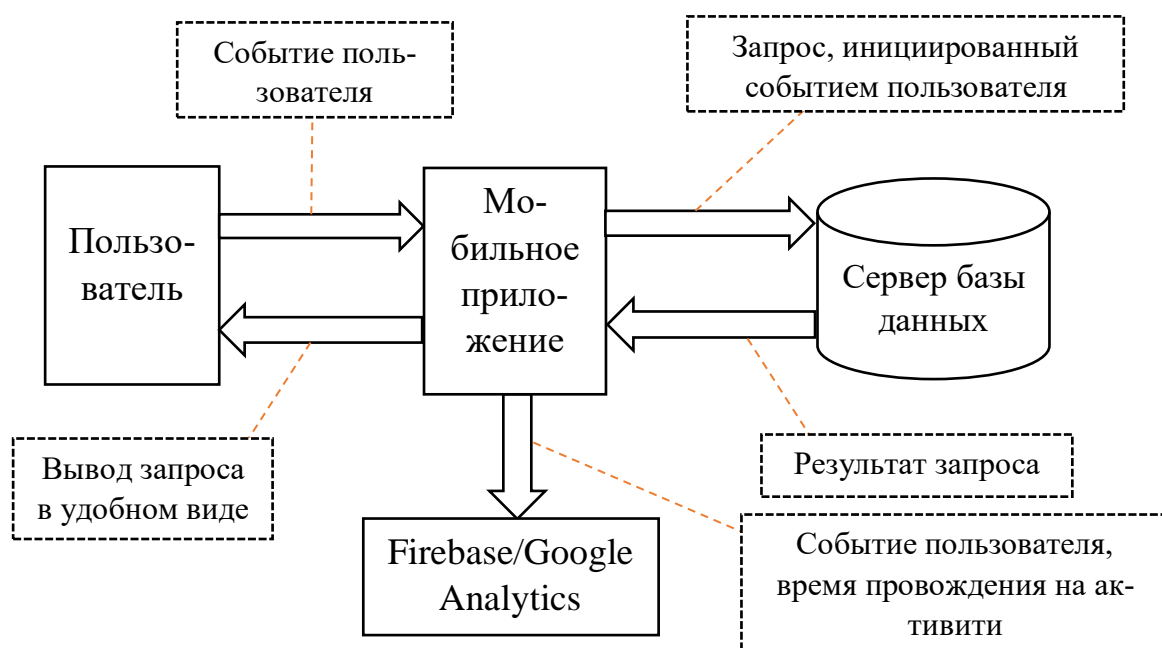


Рисунок 2.4 - Функциональная модель

2.2.2.1.2 Информационная модель

Информационная модель подсистемы «Электронный журнал» для студента показана на рисунке 2.5.



Рисунок 2.5 - Информационная модель подсистемы «Электронный журнал» для студента

2.2.3 Выделение бизнес-процессов

Необходимо выделить бизнес-процессы, отобразить их в виде схем.

В работе выделяются следующие бизнес-процессы: «Отправка данных в Firebase/Google Analytics» «Просмотр оценок студента». Просмотр оценок студента представлен на рисунке 2.6.

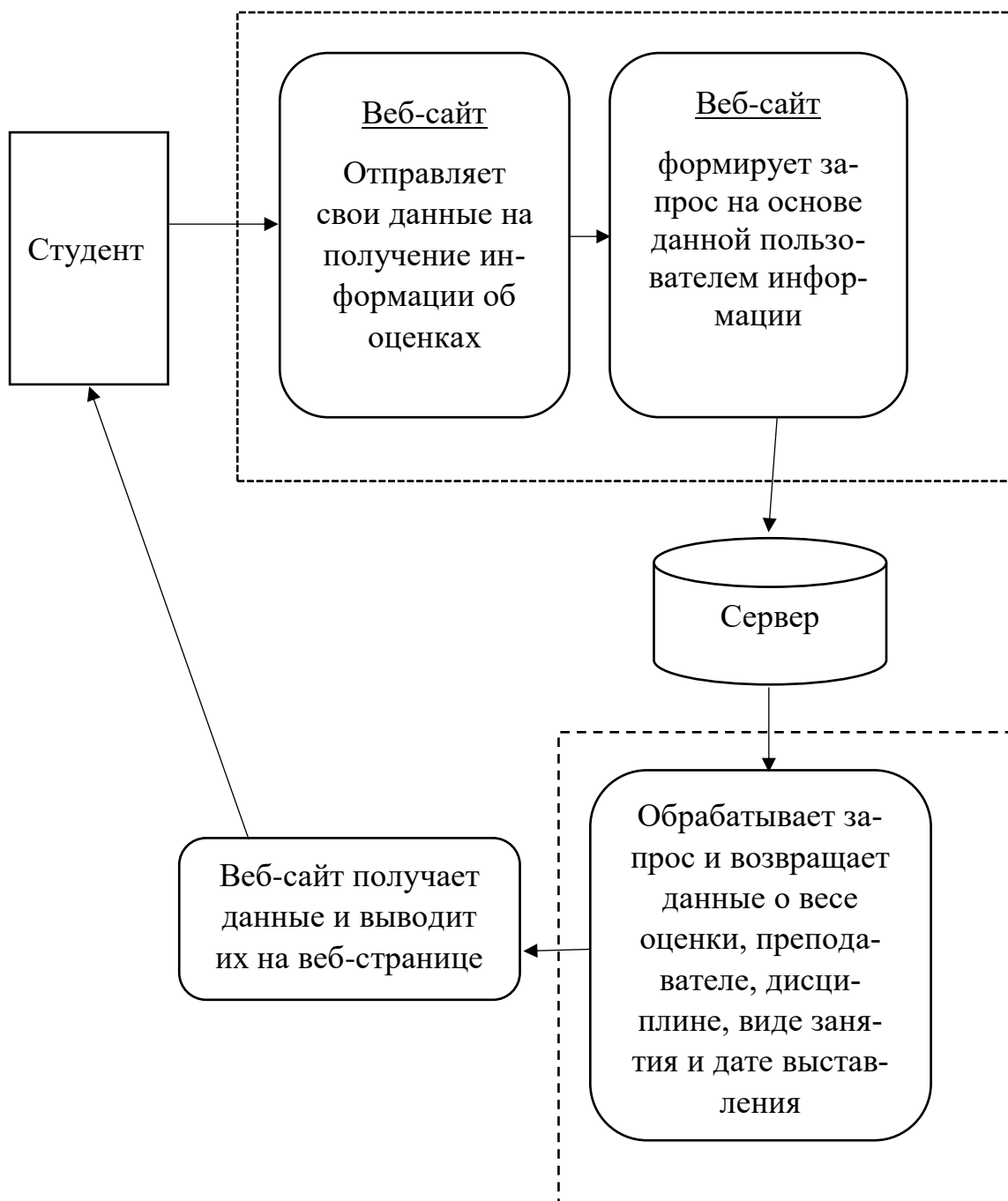


Рисунок 2.6 - Бизнес-процесс «Просмотр оценок студента»

2.2.4 Анализ и оптимизация бизнес-процессов

Необходимо провести анализ выявленных бизнес-процессов, провести оптимизацию бизнес-процессов и составить их схемы. В таблице 2.1 представлен анализ бизнес-процессов.

Таблица 2.1 - Анализ бизнес-процессов

Произведем оценку критических факторов успеха работы подсистем. Бизнес-процессы	Эффективное использовать всех функций смартфона	Автономный режим функционирования	Удобство отображения	Совместимость (поддержка ОС)
Отправка данных в Firebase/Google Analytics	0*	0	0	0
Просмотр оценок студента	5*	3*	4*	5

Как видно из таблицы аналитика сайта не настроена, автономный режим функционирования доступен только при кэшировании страниц (3), удобство отображения также имеет недостатки, поскольку адаптивный дизайн фактически урезает и уменьшает отображаемый контент (4), однако адаптивный веб-сайт имеет более широкий охват аудитории из-за своей совместимости со всеми мобильными ОС (5).

Итоговым решением становится разработка мобильного приложения, реализующее вышеперечисленные бизнес-процессы более оптимизированно. Мобильное приложение обеспечивает более быстрый доступ к контенту. Оптимизированные бизнес-процессы показаны ниже.

Отправка данных в Firebase/Google Analytics представлена на рисунке 2.7.

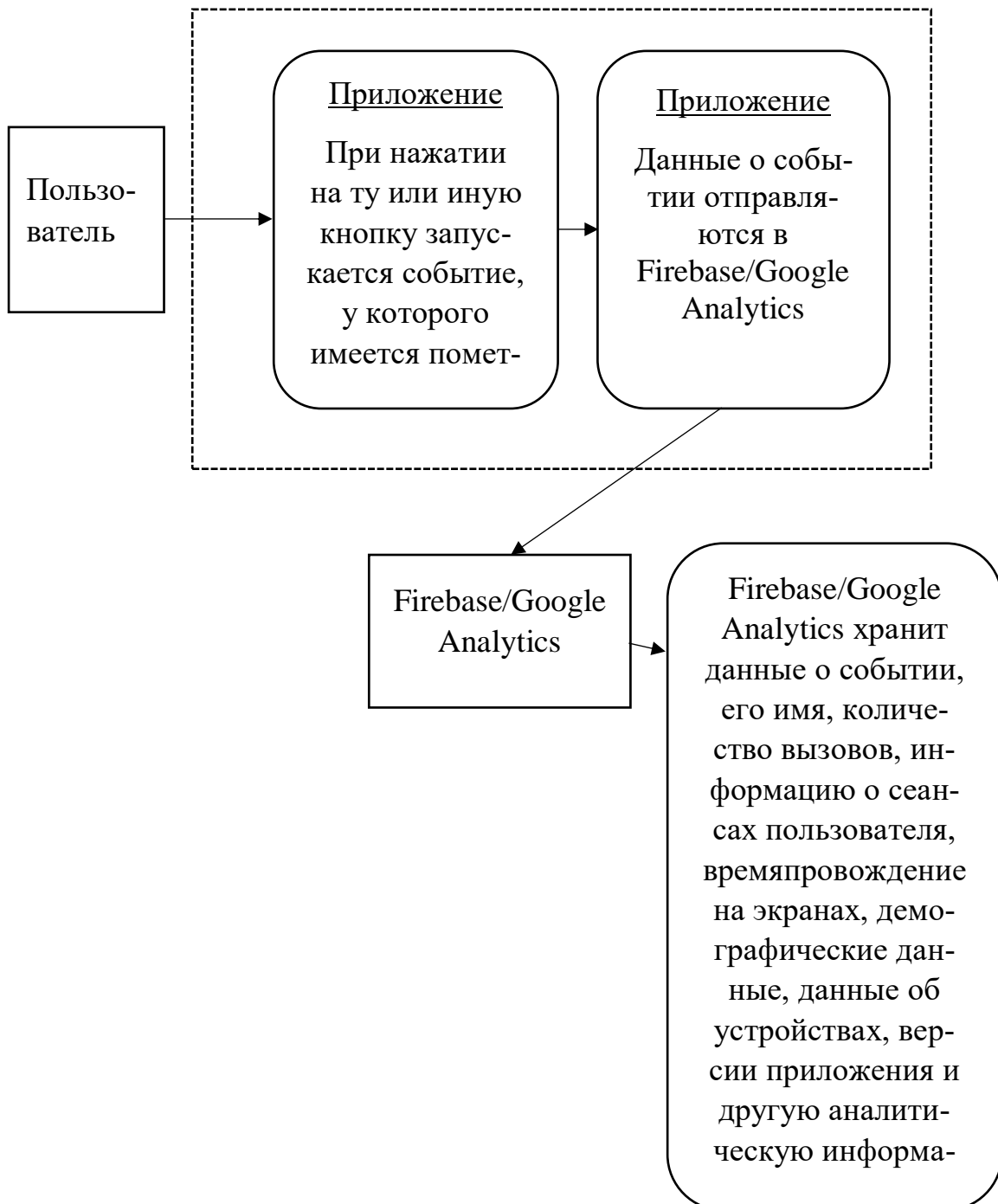


Рисунок 2.7 - Бизнес-процесс «Отправка данных в Firebase/Google Analytics»

Оптимизированный просмотр оценок студента представлен на рисунке 2.8.



Рисунок 2.8 - Бизнес-процесс «Просмотр оценок студента»

2.2.5 Разработка модели бизнес-объектов

Разработаем модель бизнес-объекта «Ответ на запрос предоставления информации об оценках», в которой будут внешний и внутренний исполнитель, а также бизнес-сущность.

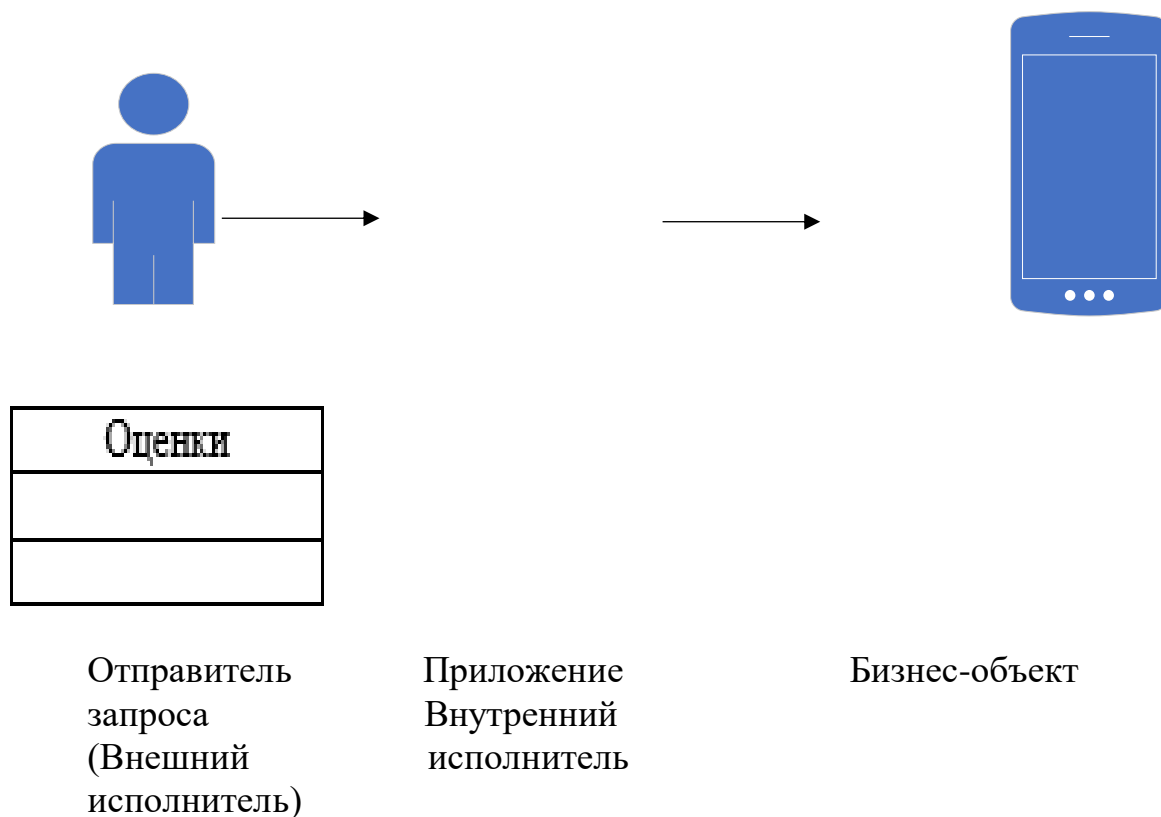


Рисунок 2.9 - Модель бизнес-объекта «Ответ на запрос предоставления информации об оценках»

Отправителем запроса могут выступать разные действующие лица. В следствие чего «Отправитель запроса» будет суперклассом по отношению к подклассам.

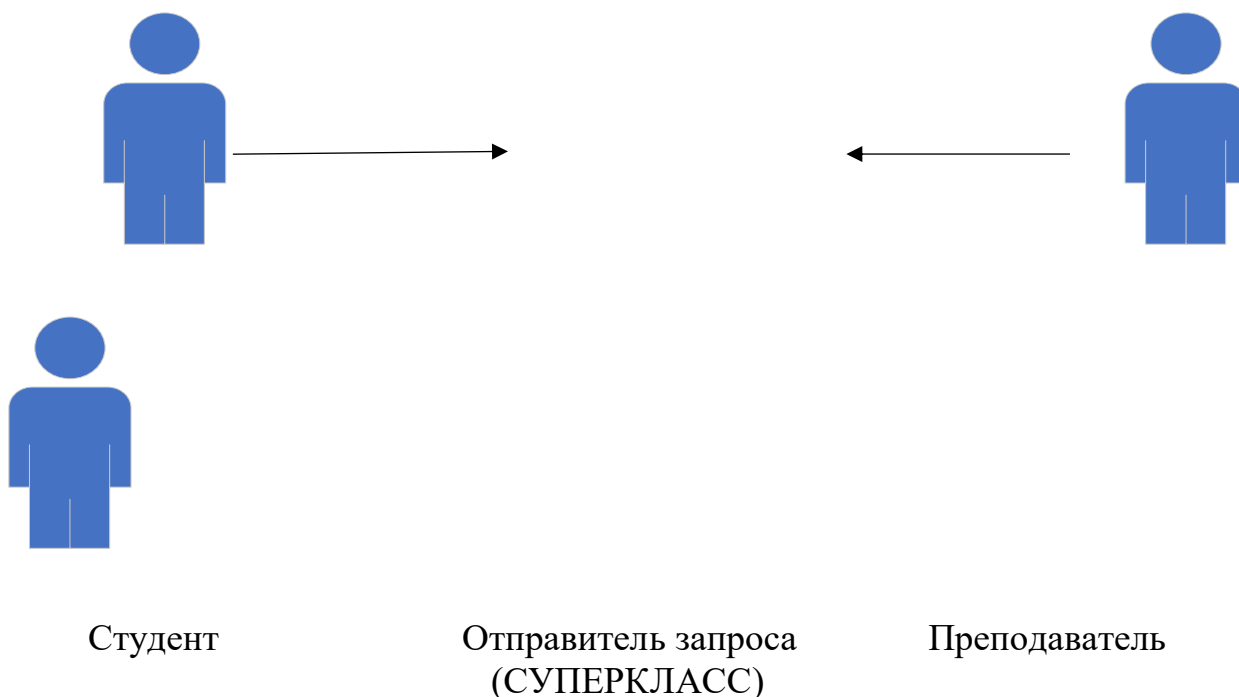


Рисунок 2.10 - Обобщение классов

Стоит отметить, что преподаватель получает информацию об оценках студента только по «своим» дисциплинам.

2.3 Описание информационного обеспечения

2.3.1 Диаграмма UML

Диаграмма UML - это диаграмма, основанная на UML (унифицированном языке моделирования), целью которой является визуальное представление системы, визуализация происходит вместе с основными участниками, действиями, ролями, классами и артефактами. Данные диаграммы помогают лучше понимать, поддерживать, изменять и документировать информацию о системе. Основными диаграммами, которые описывают систему, являются диаграммы вариантов использования, диаграммы последовательной и кооперативные диаграммы. Говоря элементарно, UML – это модернизированный подход к программному обеспечению для моделирования и документирования. Де-факто, это одна из самых популярных методик по моделированию бизнес-процессов [9].

Существуют две наиболее широкие категории – это диаграмма поведенческого UML и диаграмма структурного UML. Они охватывают все другие типы. Исходя из названия, некоторые диаграммы UML пытаются проанализировать и изобразить структуру системы или процесса, в то время как другие описывают поведение системы, ее участников (актеров) и компонентов ее построения. Различные типы разбиты следующим образом [9]:

- диаграмма классов;

- диаграмма компонентов;
- диаграмма композитной/составной структуры;
- диаграмма развёртывания;
- диаграмма объектов;
- диаграмма пакетов;
- диаграмма деятельности;
- диаграмма автомата;
- диаграмма вариантов использования;
- диаграммы коммуникации и последовательности;
- диаграмма обзора взаимодействия;
- диаграмма синхронизации.

2.3.2 Построение диаграммы вариантов использования

Краеугольным камнем системы являются функциональные требования, которые система выполняет. Диаграммы прецедентов используются для анализа требований к системе высокого уровня. Эти требования выражены в разных вариантах использования. Необходимо отметить три основные компоненты этой UML-диаграммы [9]:

- функциональные требования - представлены как варианты использования; глагол, описывающий действие;
- акторы, взаимодействующие с системой; актор может быть человеком, организацией или внутренним или внешним приложением;
- отношения между акторами и вариантами использования представлены прямыми стрелками.

В приведенных ниже рисунках показаны диаграммы UML вариантов использования для системы, описанной в техническом задании проекта. В данном случае у нас есть студент, преподаватель и сотрудник. На рисунке 2.11 изображена UML-диаграмма вариантов использования для актора «Студент», на рисунке 2.12 - для актора «Преподаватель», на рисунке 2.13 - для актора «Сотрудник».

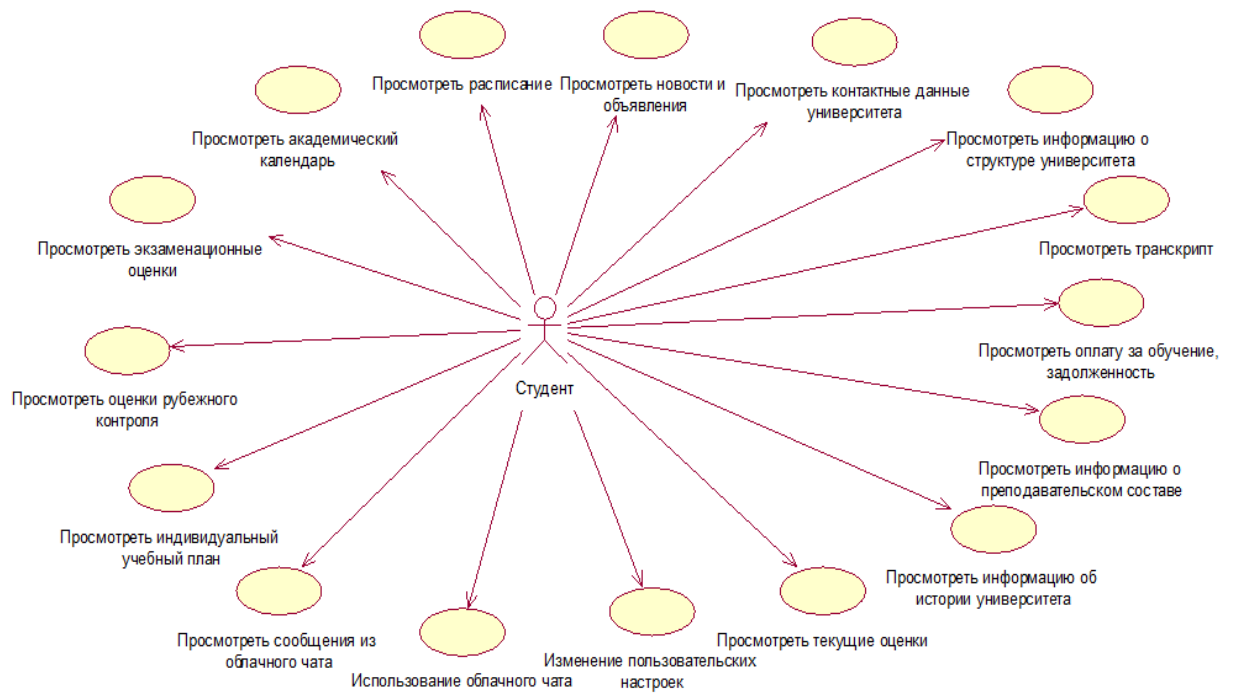


Рисунок 2.11 - UML-диаграмма вариантов использования для актора «Студент»

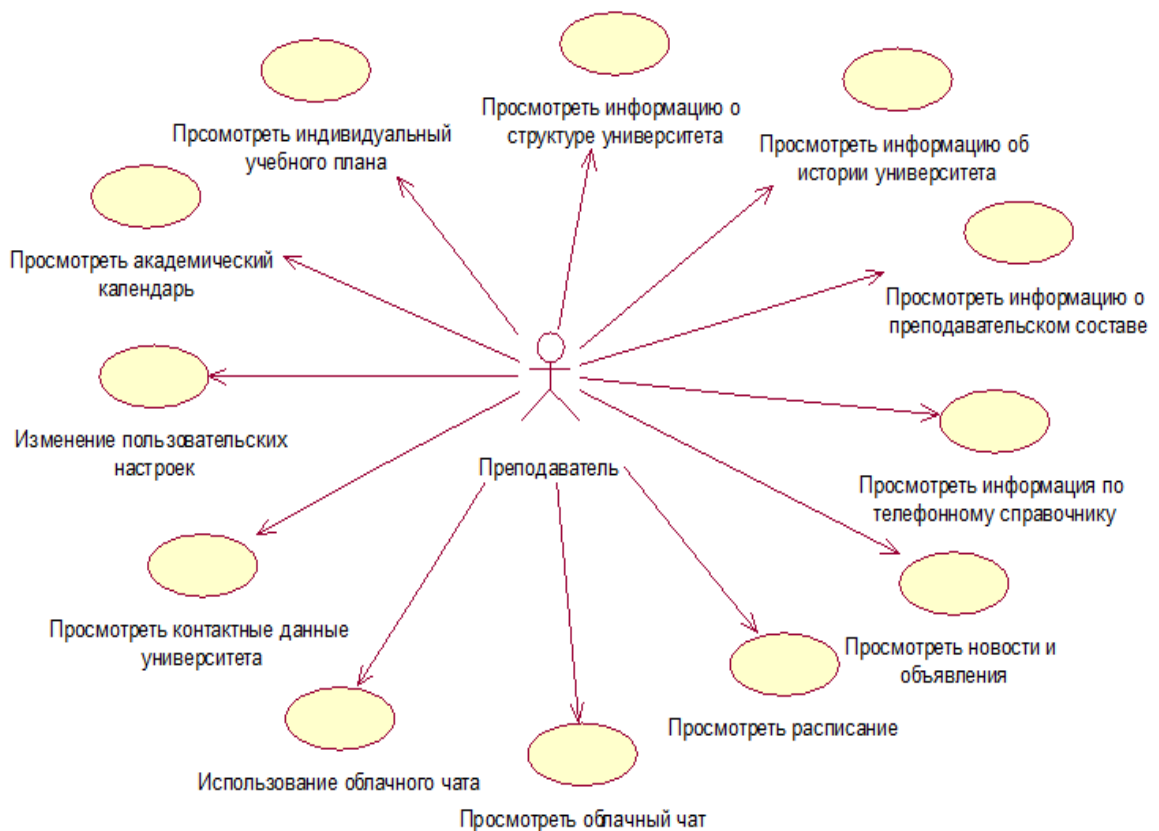


Рисунок 2.12 - UML-диаграмма вариантов использования для актора «Преподаватель»

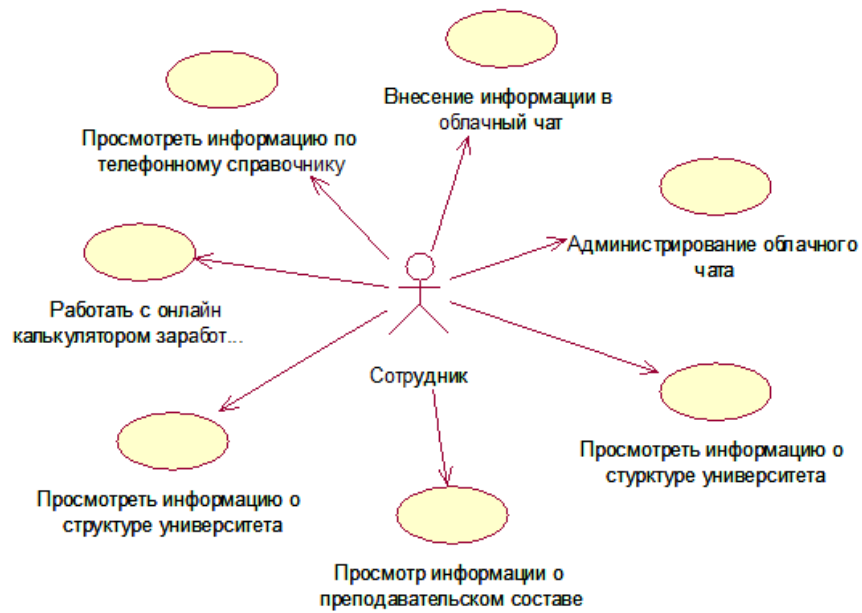


Рисунок 2.13 - UML-диаграмма вариантов использования для актора «Сотрудник»

Рассмотрим UML-диаграммы вариантов использования конкретно для подсистем основной системы с точки зрения отдельных акторов. На рисунке 2.14 изображена UML-диаграмма вариантов использования для подсистемы «Электронный журнал» актора «Студент», на рисунке 2.15 - для «Преподаватель».

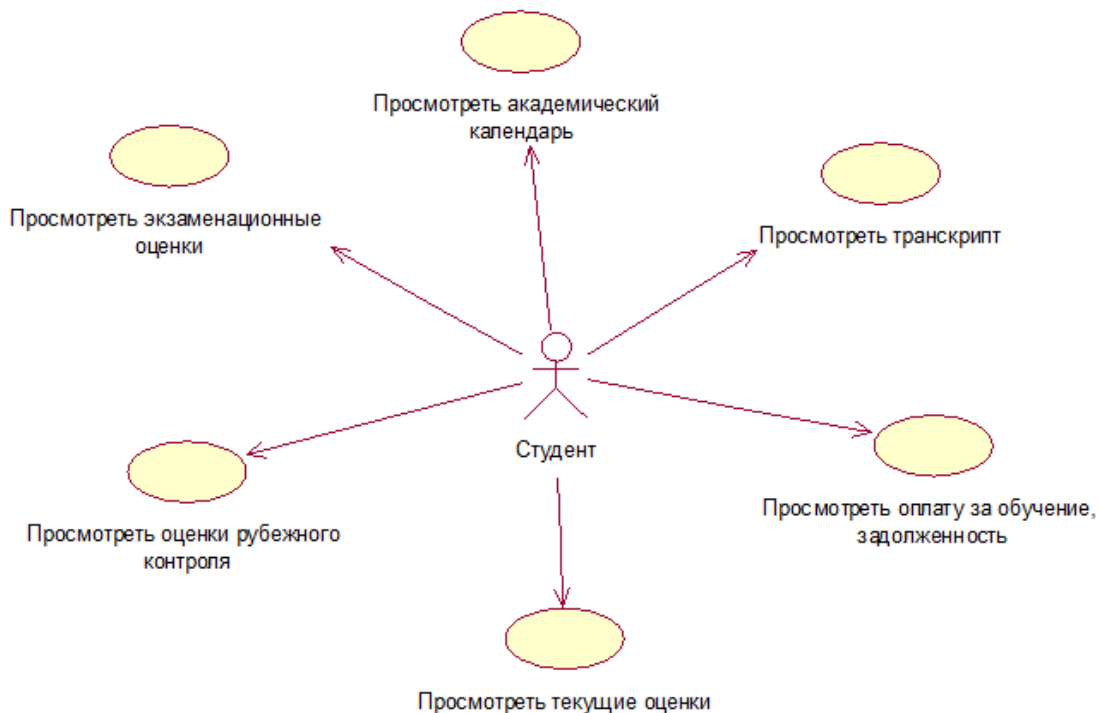


Рисунок 2.14 - UML-диаграмма вариантов использования для подсистемы «Электронный журнал» актора «Студент»

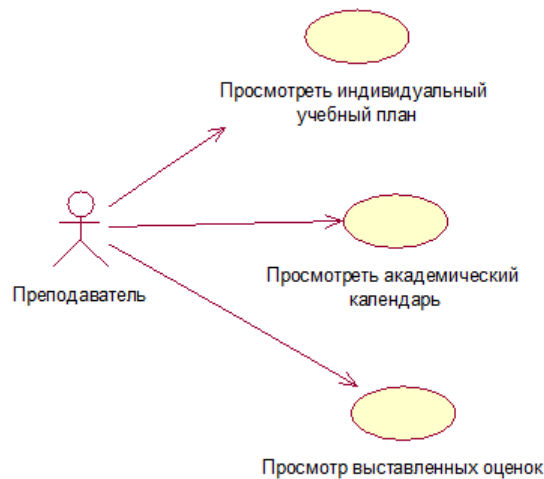


Рисунок 2.15 - UML-диаграмма вариантов использования для подсистемы «Электронный журнал» актора «Преподаватель»

В круглых контейнерах выражаются действия, которые выполняют актеры. Такими действиями являются: просмотреть академический календарь, просмотреть расписание, просмотреть новости и объявления, просмотреть контактные данные университета, просмотреть информацию о структуре университета, просмотреть транскрипт, просмотреть оплату за обучение, задолженность, просмотреть информацию о преподавательском составе, просмотреть информацию об истории университета, просмотреть текущие оценки, изменение пользовательских настроек, использование облачного чата, просмотреть сообщения из облачного чата, просмотреть индивидуальный учебный план, просмотреть оценки рубежного контроля, просмотреть экзаменационные оценки и т.д. Как можно заметить, UML-диаграммы сценариев использования хороши для демонстрации динамического поведения между участниками внутри системы, упрощая представление системы и не отражая детали реализации.

2.3.3 Построение диаграммы последовательностей и кооперативной диаграммы

Диаграммы последовательности, вероятно, являются наиболее важными диаграммами UML не только в компьютерном мире, но и в качестве моделей уровня разработки для разработки бизнес-приложений. В последнее время они стали популярными в описании бизнес-процессов из-за своей наглядности.

Исходя из названия, диаграммы последовательности описывают последовательность сообщений и взаимодействий, которые происходят между субъектами и объектами.

Поскольку основными компонентами являются сообщения, которыми обмениваются объекты, диаграммы связи строятся точно также, как и диаграммы последовательности. Единственное различие между двумя вышена-

званными диаграммами состоит в том, что объекты на диаграммах связи показаны с помощью ассоциативных соединений [9].

Основными и наиболее важными прецедентами для актора «Студент» являются «Просмотреть текущих/экзаменационных/рубежных оценок» и «Просмотреть транскрипт». Построим для них диаграммы последовательностей. На рисунке 2.16 и 2.17 продемонстрирована диаграмма последовательности и кооперативная диаграмма соответственно для прецедента «Просмотр текущих оценок».

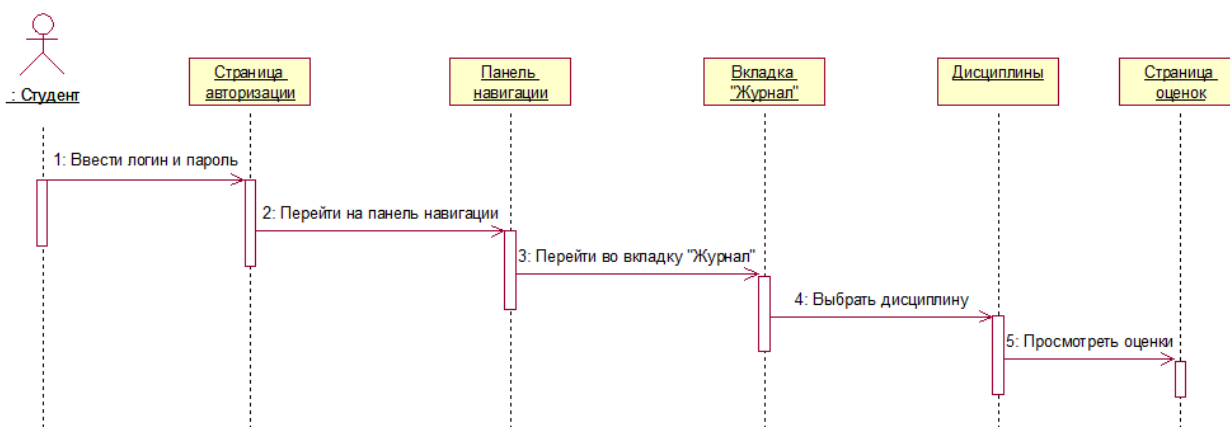


Рисунок 2.16 - UML-диаграмма последовательности для прецедента «Просмотр текущих оценок»

На рисунке 2.18 и 2.19 продемонстрирована диаграмма последовательности и кооперативная диаграмма соответственно для прецедента «Просмотр транскрипта».



Рисунок 2.17 - Кооперативная диаграмма для прецедента «Просмотр текущих оценок»



Рисунок 2.18 - UML-диаграмма последовательности для прецедента «Просмотр транскрипта»

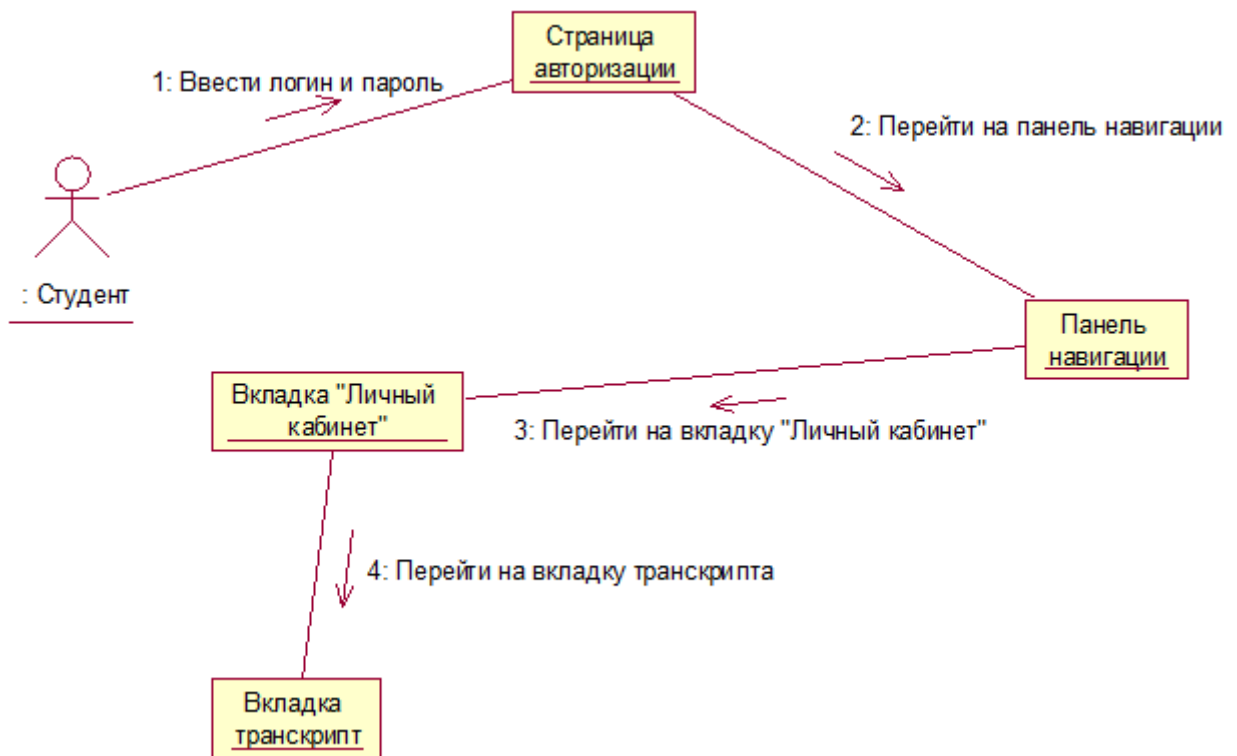


Рисунок 2.19 - Кооперативная диаграмма для прецедента «Просмотр транскрипта»

2.3.4 Обоснование выбора СУБД

Проанализируем следующие наиболее популярные СУБД на данный момент времени и на основе данного анализа выберем подходящую СУБД для нашей информационной системы мобильного приложения:

- Oracle;
- PostgreSQL;
- Microsoft SQL Server;

- MySQL.

Изучив литературу по СУБД Oracle [10], можно выделить следующие плюсы данной системы:

- инновации для повседневной работы. С Oracle 12c в качестве гибридного облачного программного обеспечения ежедневно появляются инновационные технологии облачных вычислений;

- сильная техническая поддержка и документация. Oracle обеспечивает достойную поддержку клиентов и предоставляет исчерпывающую техническую документацию по нескольким ресурсам. Таким образом, благодаря этому, можно найти решения любых возникающих проблем;

- большая емкость. Благодаря недавно выпущенной функции мультитенденности, архитектура базы данных теперь упрощает упаковку многих баз данных и плавное управление ими;

Однако, основополагаясь на вышеназванные достоинства СУБД Oracle, стоит отметить, что для информационной системы мобильного приложения нет необходимости использовать облачные вычисления. Большая емкость – это весомое достоинство, однако для приложения не нужно обрабатывать большие корпоративные данные. Кроме того, у данной СУБД имеются существенные недостатки [10]:

- высокая стоимость. Стандартная версия, которая не включает в себя все доступные функции, стоит 17 500 долларов за единицу. Enterprise Edition стоит более 47000 долларов за единицу;

- ресурсоемкая. Для установки требуется не только много места на диске, но также необходимо учитывать постоянные обновления оборудования, если происходит его развертывание на месте;

- кривая тяжелого обучения. База данных Oracle - это не система, которую можно сразу начать использовать. Для этого лучше иметь сертифицированных инженеров Oracle DB.

Рассмотрим следующую в списке СУБД PostgreSQL. Прочитав и изучив статьи, посвященные данной системе управления базами данных, можно выделить следующие плюсы [10]:

- масштабируемость. Вертикальная масштабируемость является отличительной чертой PostgreSQL, в отличие от СУБД MySQL;

- поддержка пользовательских типов данных. PostgreSQL по умолчанию поддерживает большое количество типов данных, таких как JSON, XML, H-Store и другие. Кроме того, он позволяет пользователям детерминировать свои собственные типы данных;

- открытый исходный код, управляемый сообществом. Postgres является полностью открытым и поддерживается сообществом, что укрепляет его как целостную экосистему.

Так как в проектируемой базе данных приложения будут использоваться только встроенные типы данных, нет необходимости в создании пользовательских типов данных. Также у данной СУБД есть существенный минус - наличие противоречивой документации. Поскольку сообщество PostgreSQL

довольно распространено, документация не соответствует одинаковым стандартам для всех функций Postgre.

Проанализировав источники по СУБД MSSQL, необходимо выделить следующие достоинства [10]:

- разнообразие версий;
- комплексное решение для бизнес-данных. MSSQL, ориентированный в основном на коммерческие решения, предоставляет множество дополнительных возможностей для бизнеса;
- богатая документация и помощь сообщества;
- поддержка облачных баз данных.

Мобильное приложение не является бизнес-ориентированным и не использует облачные базы данных, кроме Firebase Realtime Database, следовательно, данная СУБД не полностью подходит для управления БД. MSSQL Server - это разумный вариант для компаний с другими подписками на продукты Microsoft, не подходящий под мобильные приложения.

После изучения рекомендуемой литературы [10-11] по MySQL были отмечены следующие достоинства:

- бесплатная установка. Комьюнити версия MySQL бесплатна для скачивания. MySQL Community Edition - базовый набор инструментов для индивидуального использования;
- простой синтаксис и легкая сложность. Структура и стиль MySQL очень просты. Разработчики даже считают MySQL базой данных с человеческим языком;
- облако-совместимый.

Сравнение вышеперечисленных СУБД наглядно представлено в таблице 2. 2.

Таблица 2.2 - Сравнение СУБД

СУБД	Структура данных	Лицензия	Документация	Масштабируемость	Другие структуры данных	Кривая обучения
MySQL	SQL	GNU General-Public License	✓ ✓	Vertical, complex	*	mild
Oracle	Multi-model, SQL	Proprietary	✓ ✓ ✓	Vertical	SQL	hard

Продолжение таблицы 2.2

	Структура данных	Лицензия	Документация	Масштабируемость	Другие структуры данных	Кривая обучения
PostgreSQL	Object-relational, SQL	Open-source	✓ ✓	Vertical	NoSQL	hard
MSSQL Server	T-SQL	Proprietary	✓ ✓ ✓	Vertical, complex	SQL, NoSQL	hard

Изучив все аспекты вышеназванных СУБД и принимая во внимание все их достоинства и недостатки, в качестве системы управления базами данных для своей информационной системы я выбрал СУБД MySQL. Основной причиной выбора является тот факт, что данная СУБД бесплатная, имеет довольно несложную установку и легко настраиваемая. Так как выбор основывался в основном между двумя продуктами Oracle - MySQL и СУБД Oracle, первая отличительная особенность MySQL сыграла свою нишу. Для того, чтобы развернуть свою базу данных в Oracle необходимо заплатить немалые деньги. Кроме того, серверная часть приложения будет написана на языке программирования PHP. При помощи связки PHP + MySQL получаются быстрые приложения и сайты, работающие в режиме реального времени. Другим аспектом выбора СУБД MySQL является то, что ее синтаксис очень прост и похож на человеческий язык и для манипулирования данными используется язык SQL. Несмотря на то, что в MySQL нет многих функций, которые имеются в Oracle, разработчики постоянно занимаются усовершенствованием.

Таким образом, MySQL – это быстрая, надежная система баз данных с открытым исходным кодом, которая также предлагает большое количество функций и отлично подходит для средних баз данных.

Выбор, кроме вышеперечисленных аргументов, учитывался также из того факта, что заказчик использует СУБД MySQL.

2.3.5 Описание структур таблиц

База данных информационной системы для мобильного приложения состоит из 27 таблиц:

- ACADEMIC_CALENDAR (АКАДЕМИЧЕСКИЙ КАЛЕНДАРЬ);
- ACADEMIC_DEGREE (АКАДЕМИЧЕСКАЯ СТЕПЕНЬ);
- ADMINISTRATOR (АДМИНИСТРАТОР);
- ANNOUNCES (ОБЪЯВЛЕНИЯ);
- AUDIENCE (АУДИТОРИЯ);
- CLASS (ГРУППА);

- CURRENT_MARKS (ТЕКУЩИЕ ОЦЕНКИ);
- CYCLE (ЦИКЛ);
- DAY (ДЕНЬ);
- DEGREE (СТЕПЕНЬ);
- DEPARTAMENT (КАФЕДРА);
- DISCIPLINE (ДИСЦИПЛИНА);
- DISCIPLINE_CLASS (ДИСЦИПЛИНА_ГРУППА);
- EMPLOYEE (СОТРУДНИК);
- FINAL_MARKS (ИТОГОВЫЕ ОЦЕНКИ);
- INDIVIDUAL_PLAN (ИНДИВИДУАЛЬНЫЙ ПЛАН);
- NEWS (НОВОСТИ);
- ROLE (РОЛЬ);
- SCHEDULE (РАСПИСАНИЕ);
- SPECIALITY (СПЕЦИАЛЬНОСТЬ);
- STUDENT (СТУДЕНТ);
- TEACHER(ПРЕПОДАВАТЕЛЬ);
- TEACHER_TRAINING (ПРЕПОДВАТЕЛЬ_ЗАНЯТИЕ);
- TIME (ВРЕМЯ);
- TRAINING (ЗАНЯТИЕ);
- USERS (ПОЛЬЗОВАТЕЛИ).

Подсистема ИС «Электронный журнал» состоит из 9 таблиц. Рассмотрим более подробно каждую таблицу, значение и функции.

Таблица TEACHER (ПРЕПОДАВАТЕЛЬ) предназначена для хранения информации об идентификаторе преподавателя, его ФИО, степени, а также принадлежности к кафедре. Ее структура приведена в таблице 2.3.

Таблица 2.3 - Структура таблицы TEACHER (ПРЕПОДАВАТЕЛЬ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID_TEACHER	INTEGER		Является первичным ключом, а также хранит идентификатор преподавателя
2	SURNAME	VARCHAR2	60 BYTE	Хранит сведения о фамилии
3	NAME	VARCHAR2	60 BYTE	Хранит сведения об имени
4	PATRONYMIC	VARCHAR2	60 BYTE	Хранит сведения об отчестве
5	ID_DEGREE	INTEGER	50 BYTE	Хранит информацию об ID степени, которую имеет преподаватель
6	ID_DEPARTMENT	SMALLINT		Хранит информацию об ID кафедры, к которой относится преподаватель

7	LOGIN	VARCHAR	300BYTE	Хранится информация о логине преподавателя
---	-------	---------	---------	--

Таблица DISCIPLINE (ДИСЦИПЛИНА) предназначена для хранения сведений о дисциплине, коде дисциплины, количестве кредитов, академическом периоде и курсе обучения. Ее структура приведена в таблице 2.4.

Таблица 2.4 - Структура таблицы DISCIPLINE (ДИСЦИПЛИНА)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	CODE_DISCIPLINE	VARCHAR2	100 BYTE	Является первичным ключом, а также хранит код дисциплины
2	NAME_OF_DISCIPLINE	VARCHAR2	200 BYTE	Хранит сведения о наименовании дисциплины
3	ID_CYCLE	SMALLINT		Хранит информацию об ID цикла, к которой относится дисциплина (является внешним ключом из таблицы CYCLE)
4	SHORT_NAME	VARCHAR2	100 BYTE	Хранит короткое название дисциплины

Таблица CLASS (ГРУППА) предназначена для хранения информации о группе, годе поступления и окончания, количестве обучающихся в ней студентов, а также шифре специальности. Ее структура приведена в таблице 2.5.

Таблица 2.5 - Структура таблицы CLASS (ГРУППА)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID_GROUP	VARCHAR2	60 BYTE	Является первичным ключом и хранит идентификатор группы
2	YEAR_OF_RECEIVE	SMALLINT		Хранит информацию о годе поступления
3	CIPHER	VARCHAR2	60 BYTE	Хранит сведения о шифре специальности, к которой относится группа, является также внешним ключом из таблицы

				SPECIALITY
4	NAME_OF_GROUP	VARCHAR2	60 BYTE	Хранит сведения о наименовании группы
5	YEAR_OF_ENDING	SMALLINT		Хранит сведения о годе окончания
6	AMOUNT_OF_STUDENTS	SMALLINT		Хранит информацию о количестве студентов в группе

Таблица STUDENT (СТУДЕНТ) предназначена для хранения информации о студенте, его фамилии, имени и отчестве, группе. Ее структура приведена в таблице 2.6.

Таблица 2.6 - Структура таблицы STUDENT (СТУДЕНТ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID_STUDENT	INTEGER		Является первичным ключом и хранит идентификатор группы
2	SURNAME	VARCHAR2	60 BYTE	Хранит информацию о фамилии

Продолжение таблицы 2.6

3	NAME	VARCHAR2	60 BYTE	Хранит сведения об имени
4	PATRONYMIC	VARCHAR2	60 BYTE	Хранит сведения об отчестве
5	ID_GROUP	VARCHAR2	60 BYTE	Хранит сведения о ID группе, в которой обучается студент (внешний ключ из таблицы CLASS)
6	LOGIN	VARCHAR	300 BYTE	Хранит информацию о логине студента
7	SCIENTIFIC_DIRECTOR	INTEGER		Хранит идентификатор преподавателя, являющегося научным руководителем

				студента
8	DIPLOMA	VARCHAR	300 BYTE	Хранит информацию о наименовании дипломной работы
9	ACADEMIC_DEGREE	INTEGER		Хранит идентификатор академической степени (бакалавр, магистр)

Таблицы DISCIPLINE_CLASS (ДИСЦИПЛИНА_ГРУППА) хранит информацию, связывающую группу и дисциплины группы с ее основными характеристиками. Структура данной таблицы представлена в таблице 2.7.

Таблица 2.7 - Структура таблицы DISCIPLINE_CLASS (ДИСЦИПЛИНА_ГРУППА)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	CODE_DISCIPLINE	VARCHAR2	100 BYTE	Является первичным ключом, а также хранит код дисциплины
2	ID_GROUP	VARCHAR2	200 BYTE	Является первичным ключом и хранит идентификатор группы
3	CREDITS	INTEGER		Хранит сведения о количестве кредитов
4	COURSE	INTEGER		Хранит сведения о курсе, на котором должна проводиться дисциплина
5	ACADEMIC_PERIOD	INTEGER		Хранит сведения об академическом периоде
6	SEMESTER	INTEGER		Хранит информацию о семестре, на котором проходит дисциплина
7	ACADEMIC_YEAR	VARCHAR	20 BYTE	Хранит сведения об академическом годе, на котором должна проводиться дисциплина
8	LAB_PCT	INTEGER		Хранит информацию о весе лабораторных работ дисциплины в процентном эквива-

				ленте
--	--	--	--	-------

Продолжение таблицы 2.7

9	LECTION_PCT	INTEGER		Хранит информацию о весе лекционных занятий дисциплины в процентном эквиваленте
10	RGR_PCT	INTEGER		Хранит информацию о весе расчетно-графических работ дисциплины в процентном эквиваленте
11	PRACTICE_PCT	INTEGER		Хранит информацию о весе практических работ дисциплины в процентном эквиваленте
12	COURSE_WORK_PCT	INTEGER		Хранит информацию о весе курсовых работ дисциплины в процентном эквиваленте
13	RGR_AMOUNT	INTEGER		Хранит сведения о количестве РГР
14	COURSE_WORK	INTEGER		Хранит сведения о количестве курсовых работ

Таблица TRAINING (ЗАНЯТИЕ) предназначена для хранения информации о занятии, его принадлежности к дисциплине, длинное и короткое наименование. Ее структура приведена в таблице 2.8.

Таблица 2.8 - Структура таблицы TRAINING (ЗАНЯТИЕ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID_TRAINING	VARCHAR2	100 BYTE	Является первичным ключом и хранит идентификатор занятия
2	NAME_OF_TRAINING	VARCHAR2	200 BYTE	Хранит информацию о полном наименовании занятия
3	CODE_DISCIPLINE	VARCHAR2	100 BYTE	Хранит сведения о коде дисциплины (является внешним и первичным ключом одновременно)
4	SHORT_NAME	VARCHAR2	20 BYTE	Хранит сведения о коротком наименовании занятия

Таблица TEACHER_TRAINING (ПРЕПОДАВАТЕЛЬ_ЗАНЯТИЕ) предназначена для хранения ID преподавателя, ID группы, ID занятия и кода дисциплины. Ее структура приведена в таблице 2.9.

Таблица 2.9 - Структура таблицы TEACHER_TRAINING (ПРЕПОДАВАТЕЛЬ_ЗАНЯТИЕ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID_TEACHER	INTEGER		Является первичным ключом, а также хранит идентификатор преподавателя (одновременно является внешним ключом из таблицы TEACHER)

Продолжение таблицы 2.9

2	ID_GROUP	VARCHAR2	60 BYTE	Хранит сведения о ID группе, в которой обучается студент (внешний ключ из таблицы TRAINING)
3	ID_TRAINING	VARCHAR2	100 BYTE	Хранит идентификатор занятия (одновременно является внешним ключом из таблицы TRAINING)
4	CODE_DISCIPLINE	VARCHAR2	100 BYTE	Хранит сведения о коде дисциплины (является внешним из таблицы TRAINING и первичным ключом одновременно)

Таблица CURRENT_MARKS (ТЕКУЩИЕ ОЦЕНКИ) предназначена для хранения сведений об оценках студентов за конкретное занятие дисциплины и сведений о преподавателе, кто выставил оценку. Ее структура приведена в таблице 2.10.

Таблица 2.10 - Структура таблицы CURRENT_MARKS

№	Наименование поля	Тип поля	Размер	Назначение поля
1	DATE_OF	DATE		Хранит дату, за которую была выставлена оценка
2	ID_STUDENT	INTEGER		Является внешним ключом из таблицы STUDENT и хранит идентификатор группы

3	ID_GROUP	VARCHAR2	60 BYTE	Хранит сведения о ID группе, в которой обучается студент (внешний ключ из таблицы TEACHER_TRAINING)
4	MARK	INTEGER		Хранит числовую информацию об оценке
4	ID_TRAINING	VARCHAR2	100 BYTE	Хранит идентификатор занятия (является внешним ключом из таблицы TEACHER_TRAINING)
5	CODE_DISCIPLINE	VARCHAR2	100 BYTE	Хранит сведения о коде дисциплины (является внешним ключом из таблицы TEACHER_TRAINING)

Таблица CYCLE (ЦИКЛ) предназначена для хранения сведений о цикле дисциплины, названии цикла и короткого имени. Ее структура приведена в таблице 2.11.

Таблица 2.11 - Структура таблицы CYCLE (ЦИКЛ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID_CYCLE	INTEGER		Является первичным ключом, а также хранит идентификатор цикла
2	NAME_OF_CYCLE	VARCHAR2	50 BYTE	Хранит информацию о названии цикла
3	SHORT_NAME	CHAR	20 BYTE	Хранит сведения о коротком именовании цикла

Таблица SPECIALITY (СПЕЦИАЛЬНОСТЬ) предназначена для хранения сведений о специальности, шифре и кафедре, к которой относится специальность.

Таблица 2.12 - Структура таблицы SPECIALITY (СПЕЦИАЛЬНОСТЬ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	CIPHER	VARCHAR2	60 BYTE	Является первичным ключом, а также хранит шифр специальности.
2	NAME_OF_SPECIALITY	VARCHAR2	120	Хранит информацию

			BYTE	о наименовании специальности
3	ID_DEPARTMENT	SMALLINT		Хранит сведения о выпускающей кафедре (внешний ключ из таблицы «DEPARTMENT»)

Таблица ACADEMIC_CALENDAR (АКАДЕМИЧЕСКИЙ КАЛЕНДАРЬ) хранит ссылки на PDF-файлы, в которых отображаются академические календари для 1, 2, 3, 4 курсов. Данные файлы представляют собой списки с названием, при нажатии на которые будут отображаться PDF-файлы в специально предназначенном для этого приложении-посреднике. Структура данной таблицы представлена в таблице 2.13.

Таблица 2.13 - Структура таблицы ACADEMIC_CALENDAR (АКАДЕМИЧЕСКИЙ КАЛЕНДАРЬ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID	INTEGER		Является первичным ключом
2	DESCRIPTION	TEXT		Хранит описание (для какого курса предназначен академический календарь)
3	LINK	TEXT		Хранит ссылку на PDF-файл в теге <a> HTML

Таблица ACADEMIC_DEGREE (АКАДЕМИЧЕСКАЯ СТЕПЕНЬ) содержит информацию о академических степенях. Структура таблицы представлена в таблице 2.14.

Таблица 2.14 - Структура таблицы ACADEMIC_DEGREE (АКАДЕМИЧЕСКАЯ СТЕПЕНЬ)

№	Наименование поля	Тип поля	Размер	Назначение поля
1	ID_ACADEMIC_DEGREE	INTEGER		Является первичным ключом
2	NAME_OF_ACAD_DEGREE	VARCHAR	300 BYTE	Хранит наименование степени (бакалавриат, магистратура, докторантура)

Также имеются следующие таблицы ADMINISTRATOR (хранятся администраторы системы), EMPLOYEE (хранятся сотрудники системы), ROLE (хранятся роли системы), ANNOUNCES (хранятся объявления), NEWS (хранятся новости). Структура данных таблиц представлена на рисунках ниже.




#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	ID_ADMIN  	int(11)			Нет	Нет
2	FIRSTNAME	varchar(300)	utf8_unicode_ci		Да	NULL
3	NAME	varchar(300)	utf8_unicode_ci		Да	NULL
4	PATRONYMIC	varchar(300)	utf8_unicode_ci		Да	NULL
5	LOGIN 	varchar(300)	utf8_unicode_ci		Нет	Нет

Рисунок 2.20 - Структура таблицы ADMINISTRATOR



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	ID_ROLE  	int(11)			Нет	Нет
2	ROLE_NAME	varchar(300)	utf8_unicode_ci		Да	NULL

Рисунок 2.21 - Структура таблицы ROLE




#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	ID_EMPLOYEE  	int(11)			Нет	Нет
2	SURNAME	varchar(300)	utf8_unicode_ci		Нет	Нет
3	NAME	varchar(300)	utf8_unicode_ci		Нет	Нет
4	PATRONYMIC	varchar(300)	utf8_unicode_ci		Да	NULL
5	LOGIN 	varchar(300)	utf8_unicode_ci		Нет	Нет

Рисунок 2.22 - Структура таблицы EMPLOYEE



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	ID_ADMIN 	int(11)			Да	NULL
2	ID_ANNOUNCES 	int(11)			Нет	Нет
3	TITLE_ANNOUNCES_RU	text	utf8_unicode_ci		Нет	
4	BODY_ANNOUNCES_RU	text	utf8_unicode_ci		Нет	
5	TITLE_ANNOUNCES_KZ	text	utf8_unicode_ci		Да	
6	BODY_ANNOUNCES_KZ	text	utf8_unicode_ci		Да	
7	TITLE_ANNOUNCES_EN	text	utf8_unicode_ci		Да	
8	BODY_ANNOUNCES_EN	text	utf8_unicode_ci		Да	
9	DATE_ANNOUNCES	date			Нет	current_timestamp()

Рисунок 2.23 - Структура таблицы ANNOUNCES

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	ID_NEWS 🔑	int(11)			Нет	Нет
2	TITLE_NEWS_RU	text	utf8_unicode_ci		Нет	
3	BODY_NEWS_RU	text	utf8_unicode_ci		Нет	
4	DATE_NEWS	date			Нет	current_timestamp()
5	TITLE_NEWS_KZ	text	utf8_unicode_ci		Нет	
6	BODY_NEWS_KZ	text	utf8_unicode_ci		Нет	
7	TITLE_NEWS_EN	text	utf8_unicode_ci		Нет	
8	BODY_NEWS_EN	text	utf8_unicode_ci		Нет	
9	ID_ADMINISTRATOR 🔑	int(11)			Да	NULL
10	IMAGE_URL	text	utf8_unicode_ci		Да	

Рисунок 2.24 - Структура таблицы NEWS

2.3.6 Модели базы данных

На основе анализа предметной области необходимо создать логическую модель базы данных. Логическая модель базы данных была построена при помощи Data Modeler и показана на рисунке 2.25. На этапе проектирование логической модели БД были сразу удалены связи многие-ко-многим, хотя на этом уровне они имеют место быть. Таким образом таблицы соединены друг с другом связью один-ко-многим или многие-к-одному.

На основе логической модели была построена реляционная модель базы данных при помощи функции Engineer to Relational Model. Реляционная модель представлена на рисунке 2.27.

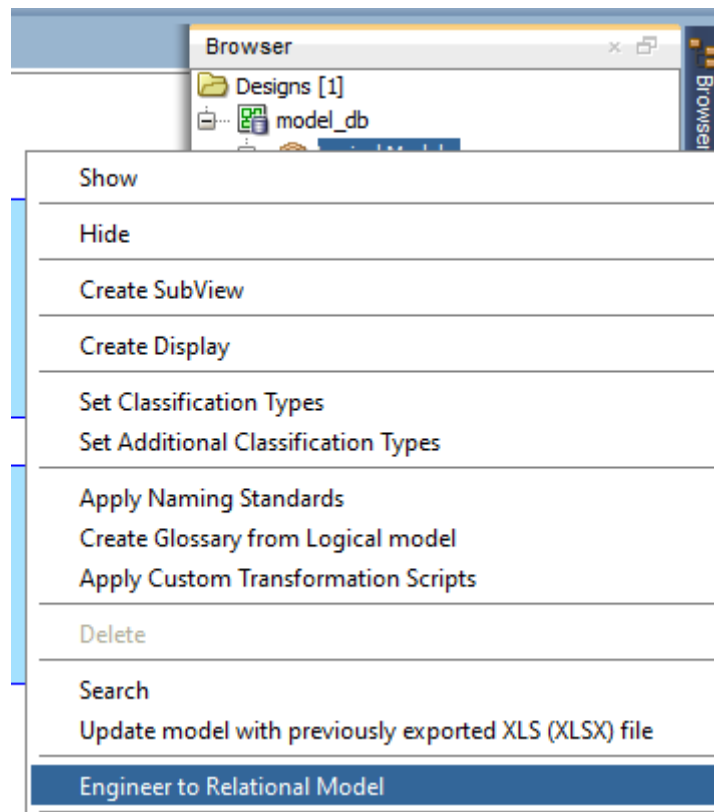


Рисунок 2.25 - Функция Engineer to Relational Model

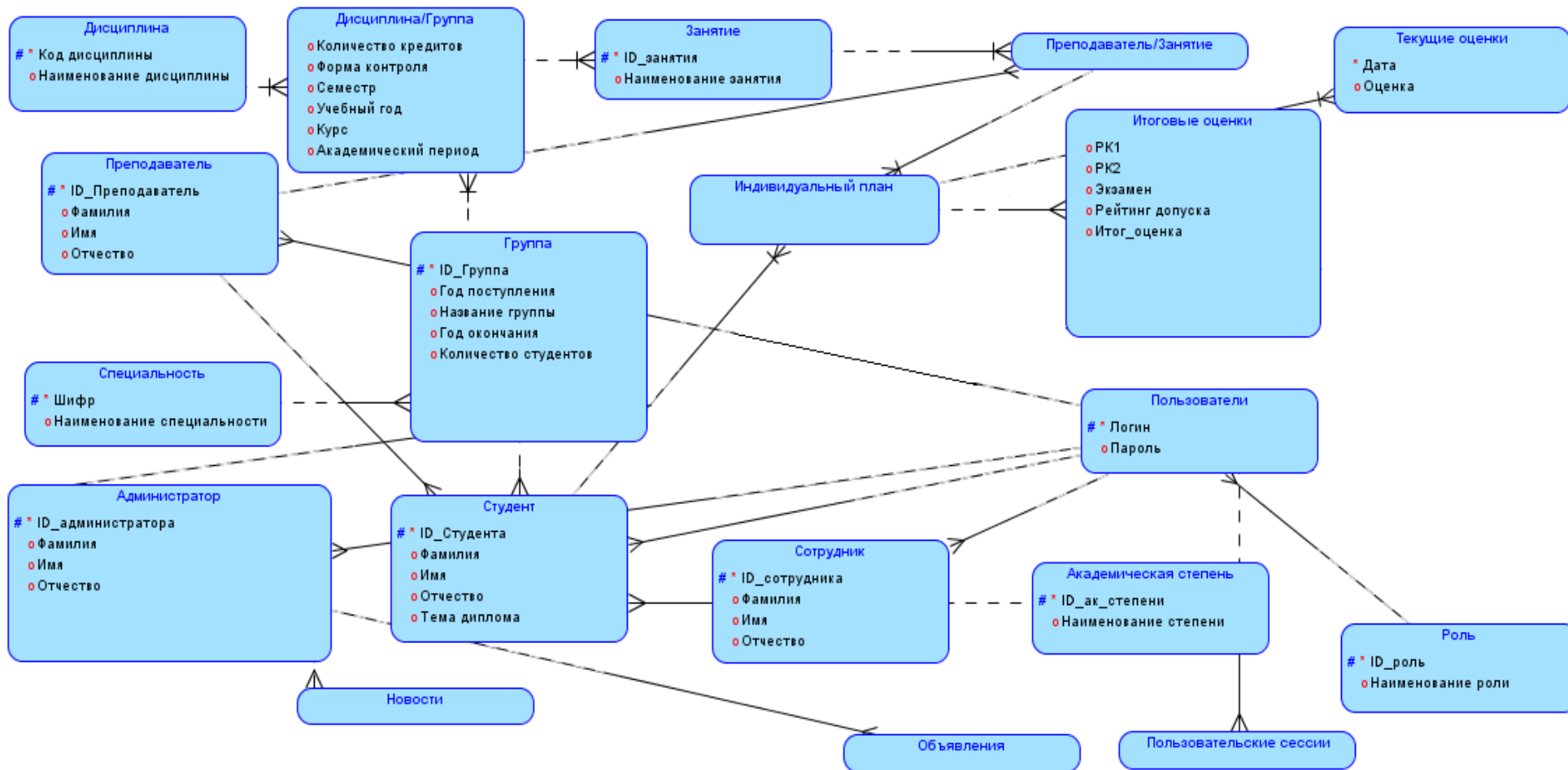


Рисунок 2.26 - Логическая модель базы данных

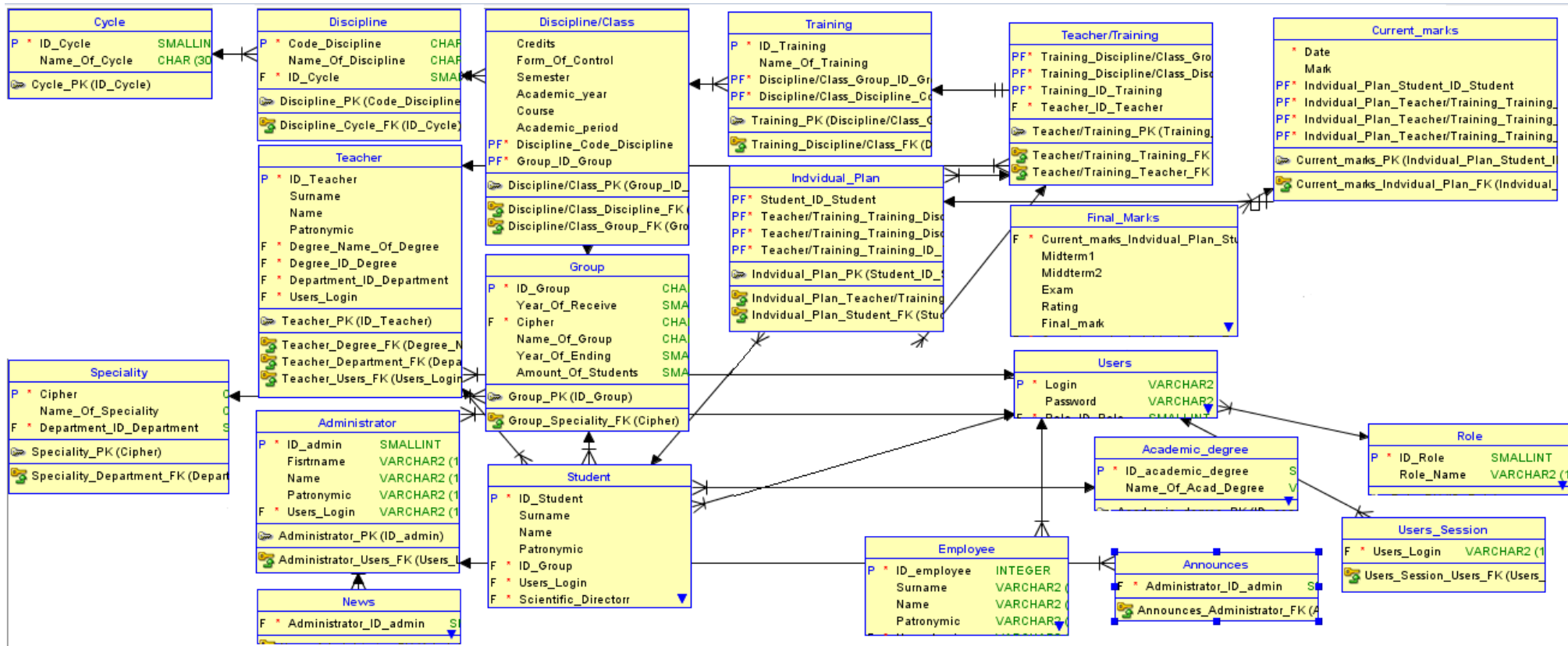


Рисунок 2.27 - Реляционная модель базы данных

2.3.7 Триггеры, хранимые процедуры, функции

Для нормальной работы базы данных для уменьшения времени добавления данных разрабатываются собственные хранимые процедуры, функции и триггеры. В данной работе разрабатывается хранимая процедура, которая вызывается в скрипте РНР, после каждого раза, когда студенту выставляется итоговая или текущая оценка. Хранимая процедура служит для подсчета средней текущей оценки, рейтинга допуска, итоговой оценки. Структура процедуры представлена на рисунке 2.28.

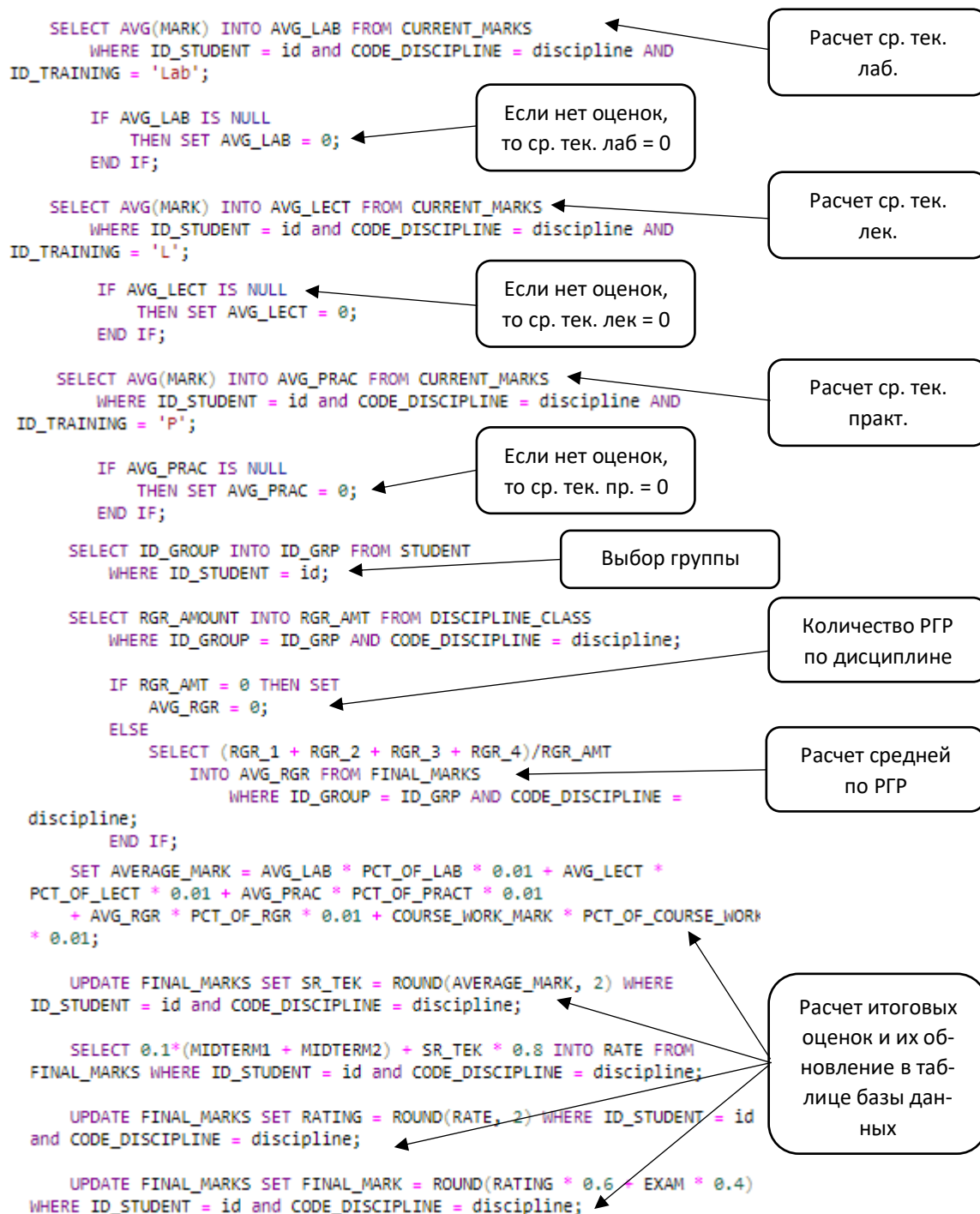


Рисунок 2.28 - Структура хранимой процедуры

Также имеется триггер на таблице DISCIPLINE_CLASS, структура которого представлена на рисунке 2.29.

Детали	
Название триггера	INSERT_INTO_TRAINING_AFTER_INSERT_INTO_DIS_CLASS
Таблица	DISCIPLINE_CLASS
Время	AFTER
Событие	INSERT
Определение	<pre> 1 BEGIN 2 IF (new.LAB_PCT > 0) THEN 3 INSERT INTO `TRAINING` (`ID_TRAINING`, `NAME_OF_TRAINING`, `ID_GROUP`, 4 `CODE_DISCIPLINE`, `SHORT_NAME`) VALUES ('Lab', 'Лабораторные работы', 5 new.ID_GROUP, new.CODE_DISCIPLINE, 'Лаб. '); 6 END IF; 7 IF (new.LECTION_PCT > 0) THEN 8 INSERT INTO `TRAINING` (`ID_TRAINING`, `NAME_OF_TRAINING`, `ID_GROUP`, 9 `CODE_DISCIPLINE`, `SHORT_NAME`) VALUES ('L', 'Лекции', new.ID_GROUP, 10 new.CODE_DISCIPLINE, 'Лек. '); 11 END IF; 12 IF (new.PRACTICE_PCT > 0) THEN 13 INSERT INTO `TRAINING` (`ID_TRAINING`, `NAME_OF_TRAINING`, `ID_GROUP`, 14 `CODE_DISCIPLINE`, `SHORT_NAME`) VALUES ('P', 'Практические работы', 15 new.ID_GROUP, new.CODE_DISCIPLINE, 'Прак. '); 16 END IF; 17 END </pre>
Определитель	id11311399_aues@%

Рисунок 2.29 - Структура триггера INSERT_INTO_TRAINING_AFTER_INSERT_INTO_DIS_CLASS

Смысл триггера состоит в том, что после того запись о дисциплине добавлена в таблицу DISCIPLINE_CLASS, создаются записи о занятиях в таблице TRAINING. Если то или иное занятие имеет процентный вес, то для него создается запись в таблице TRAINING.

2.4 Firebase Analytics

Firebase – это платформа для разработки мобильных и веб-приложений. С помощью этой платформы можно быстро разрабатывать качественные приложения, привлекать новых пользователей и повышать доходы. Платформа предлагает несколько интегрированных возможностей, таких как сервер для мобильных приложений, средства аналитики, инструменты для продвижения приложений и монетизации, которые можно использовать по отдельности или в любых сочетаниях. В данной работе этот инструмент от компании Google используется как средство аналитики. То есть каждое действие всех пользователей регистрируется и записывается в данной системе. Целью аналитики является выявление наиболее популярных функций у пользователей. Информация о действиях показаны на рисунке 2.30.

События		Параметры отчетов			
Существующие события					
Название события ↓	Количество	Изменение, %	Пользователи	Изменение, %	Отметить как конверсию ⓘ
Чат	2	-	1	-	<input type="checkbox"/>
Сотрудничество	1	-	1	-	<input type="checkbox"/>
Проекты	1	-	1	-	<input type="checkbox"/>
Обучающимся	1	-	1	-	<input type="checkbox"/>
Геопозиция	1	-	1	-	<input type="checkbox"/>
session_start	233	-	7	-	<input type="checkbox"/>
screen_view	3 499	-	7	-	<input type="checkbox"/>

Рисунок 2.30 - Информация о событиях

Также имеются функции просмотра активных пользователей, информация представлена на рисунке 2.31. Можно узнать с чем взаимодействовали пользователи системы - рисунок 2.32.

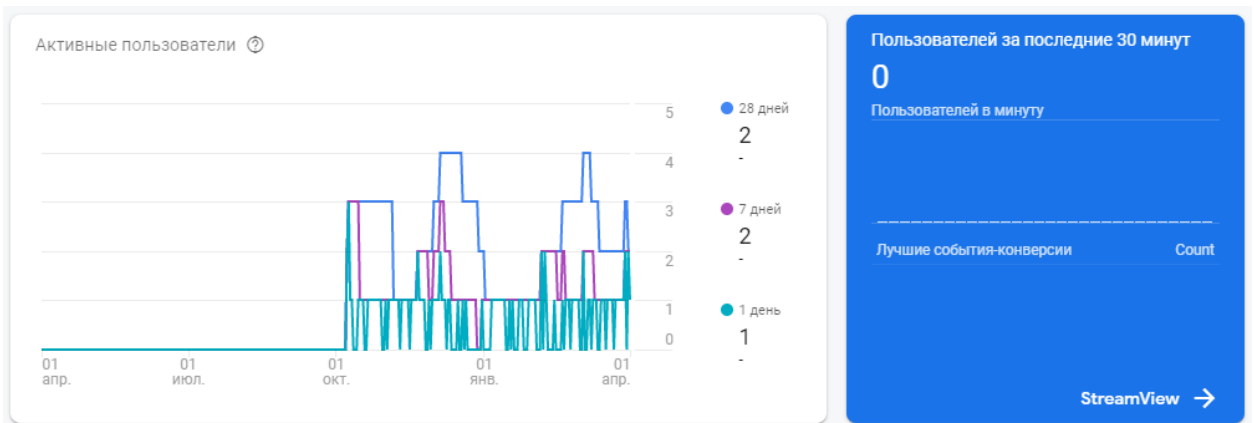


Рисунок 2.31 - Активные пользователи

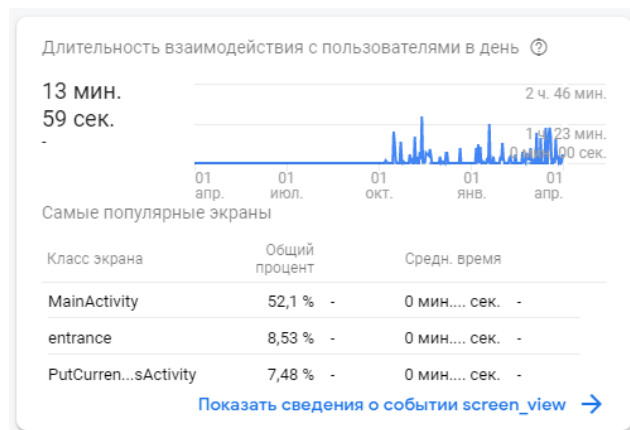


Рисунок 2.32 - Взаимодействие с экраном

3 Разработка программного обеспечения

3.1 Обоснование выбора среды программирования

Android Studio

Android Studio специально предназначена для создания приложений для Android, поэтому все инструменты используются для создания только Android, так что нет никакой путаницы, как Eclipse и Visual Studio. Каждое новое обновление дает лучшие функции и методы, которые могут сократить время написания кода [12].

Существуют также и следующие преимущества Android Studio:

- периодически обновляется;
- поддержка Gradle;
- выделенный эмулятор;
- не имеет отношения к Eclipse;
- чистый макет на основе IntelliJ IDEA;
- Android Studio поставляется с адаптивной поддержкой значков для Android 8.0+;
- новый APK Analyzer;
- выделенная командная строка;
- простое подключение и интеграция с выделенными помощниками с Firebase, Kotlin и Google Cloud;
- глубокий анализ кода;
- создание приложений одновременно для мобильных устройств, часов, телевидения и т. д.

Java

Java является предпочтительным по многим причинам для разработки приложений для Android [13]:

- Java безопасен (без угрозы для безопасности, потому что ничего не выполняется вне JVM)
- объектно-ориентированные парадигмы;
- богатый набор основных функций (основные функции Java полны и обширны. Кроме того, они регулярно обновляются и поддерживаются Oracle);
- JVM (оптимизировано для Android): виртуальная машина облегчает жизнь разработчику / программисту. Андроид использует dalvik, виртуальную машину, оптимизированную для удовлетворения потребностей Android;
- фреймворки и классы: за пределами основной библиотеки в Java есть много фреймворков и классов для таких функций, как работа в сети, многопоточность, операции ввода-вывода, и, таким образом, программисты могут использовать эти качества в своих приложениях.

Java является единственным языком, который может наилучшим образом удовлетворить потребности разработчика приложений. Ни один другой язык не может сравниться со скоростью и избыточностью Java, даже Python.

C / C ++ небезопасны из-за указателей, C # принадлежит Microsoft, Objective-C принадлежит Apple, PHP для веб. Java остается единственным вариантом.

СУБД и серверная часть

В качестве СУБД, как было указано в пункте описания информационного обеспечения, будет использована MySQL. MySQL – это одна из самых популярных и самых распространенных система управления базами данных (СУБД). MySQL отличается хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP. Приложение на PHP, использующее для хранения информации базу данных (в частности MySQL) всегда работает быстрее приложения, построенного на файлах.

В качестве сервера для созданного приложения будет использован хостинг 000webhost.com. В качестве веб-интерфейса для администрирования СУБД MySQL будет использоваться phpMyAdmin. Приложение пользуется большой популярностью, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс. Серверная часть приложения будет реализована на языке PHP. Главным фактором языка PHP является практичность. PHP должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер PHP обусловлен пятью важными характеристиками [4]:

- традиционностью;
- простотой;
- эффективностью;
- безопасностью;
- гибкостью.

Язык XML

Приложение «AUES Mobile» состоит из активностей(activity), фрагментов(fragment), отдельных классов(class) и макетов(layout). Макет состоит из отдельных объектов, которые позиционируются на экране и определяют пользовательский интерфейс. Макет создается из формализованного языка XML (eXtensible Markup Language) - расширяемый язык разметки.

Проанализировав статьи по XML, были выделены следующие достоинства [14-15]:

- он основан на тексте и, следовательно, его удобно читать;
- он поддерживает стандарт Unicode и, следовательно, может использоваться текст из любого набора символов;
- XML может хранить двоичную информацию. Двоичные данные должны быть преобразованы в текст с помощью «Base64»;
- XML хранит любой документ или данные в виде «древовидной структуры»;
- XML не зависит от платформы;

- XBRL (версия XML) рекомендована SEC (Комиссия по ценным бумагам и биржам США) США для отчетности о финансовых доходах федеральному правительству США;

- он может использоваться небольшими пользователями EDI (Electronic data interchange), у которых нет ресурсов для реализации полноценной установки EDI.

Также были выделены и недостатки [14-15]:

- XML относительно неэффективен по сравнению с другими алгоритмами хранения. Это связано с использованием тегов XML, которые требуют дополнительной памяти или пропускной способности;

- хранение двоичных данных, таких как данные изображений, также неэффективны. Хранение осуществляется с использованием кодировки Base64, которая занимает 4 байта для хранения 3 байтов двоичных данных. Следовательно, он потребляет больше памяти / пропускной способности;

- для отображения данных закодированного изображения, записанных в формате XML, необходима дополнительная программа;

- XML требует отдельную таблицу стилей XML, чтобы использовать ее для более простого отображения данных;

- XML - это базовая технология для хранения и передачи данных. В нем отсутствуют инструкции по визуализации;

- не все файлы хорошо представлены древовидными структурами. Примеры включают таблицы DB2 / 400, простые файлы, неопределенные файлы фиксированной длины записи, потоковые файлы и т. д.;

- размеры файлов очень большие.

Так как Android-приложения пишутся в интегрированной среде разработки Android Studio, весь интерфейс пишется на языке XML.

3.2 Построение макета пользовательского интерфейса

3.2.1 Макет главной активности (MainActivity)

По умолчанию при создании любой активности ее макет определяется двумя виджетами - LinearLayout и TextView. В интерфейсе MainActivity определено 5 виджетов:

- DrawerLayout;
- вертикальный LinearLayout;
- include;
- FrameLayout;
- NavigationView.

Тег `<include>` позволяет включать в один макет другой ранее заготовленный. В данном случае `<include>` включает в макет `app_bar_main.xml`, который в свою очередь состоит из двух виджетов:

- AppBarLayout;
- Toolbar.

NavigationView представляет стандартное меню навигации для приложения. NavigationView обычно размещается внутри DrawerLayout [16]. Содержимое меню заполнено файлом ресурсов меню. Меню находится в файле activity_main_drawer.xml.

DrawerLayout действует как контейнер верхнего уровня для содержимого окна, что позволяет извлекать интерактивные виды «drawer» из одного или обоих вертикальных краев окна [17].

LinearLayout представляет из себя view группу, которая выравнивает дочерние элементы либо по вертикали, либо по горизонтали.

На рисунке 3.1 показано, как из этих виджетов образуется интерфейс MainActivity.

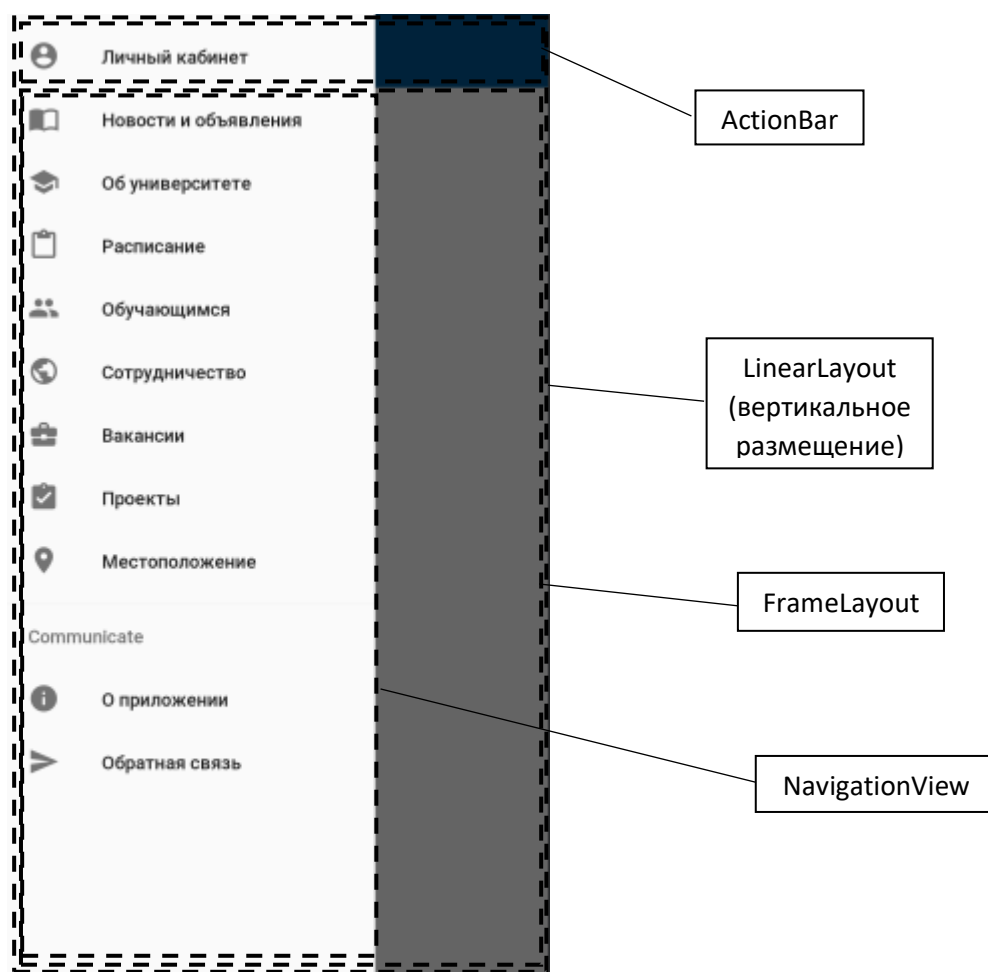


Рисунок 3.1 - Запланированное расположение виджетов MainActivity на экране

Далее определяются вышеназванные виджеты в файле activity_main.xml. Для определения виджетов в XML (activity_main.xml) был написан модуль, на котором позиционируются вышеназванные вью-элементы: LinearLayout, FrameLayout, NavigationView, include, DrawerLayout.

Данный модуль выполняет функцию позиционирования виджет-объектов на экране. Далее строится иерархия представлений для макета

MainActivity, которая показана на рисунке 3.2. Данный модуль выполняет функцию позиционирования виджет-объектов на экране.

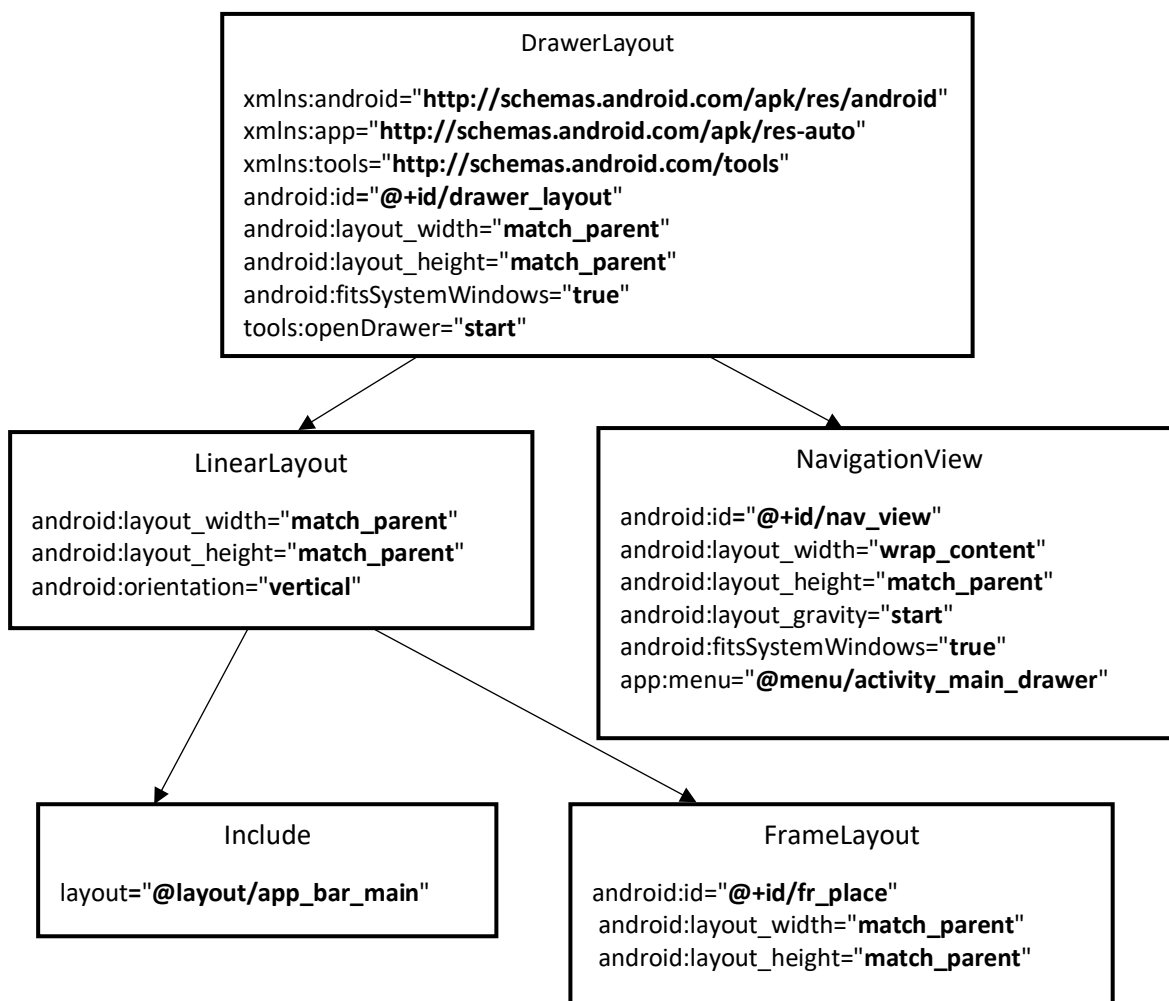


Рисунок 3.2 - Иерархия представлений макета MainActivity

Ниже показаны основные атрибуты, которые используются в вышеперечисленных виджетах [1]:

- `android:id = "@+id/name"`. Используется для идентификации виджета, может использоваться в Java-коде для доступа к нему. Кроме того, используется для позиционирования виджетов относительно других виджетов в `RelativeLayout`. При этом для использования идентифицированного виджета символ «+» не пишется.

- `android:layout_width` `android:layout_height`. Задают ширину и высоту виджета по размеру содержимого (значение задается как `wrap_content`) и по размеру родителя (`match_parent`). У самого корневого элемента родителем является сам экран устройства.

- `android:orientation`. Является атрибутом виджета `LinearLayout`, принимает значения `vertical` (все дочерние виджеты будут выстроены в вертикальном положении) и `horizontal` (все дочерние виджеты будут выстроены в горизонтальном положении). В вертикальном положении выше

тот виджет, что определён первым внутри корневого элемента и аналогично, в горизонтальном положении первый объявленный виджет будет крайним слева.

На рисунке 3.3 показано расположение макета хедера, который был включен в макет MainActivity посредством тэга `<include>`.

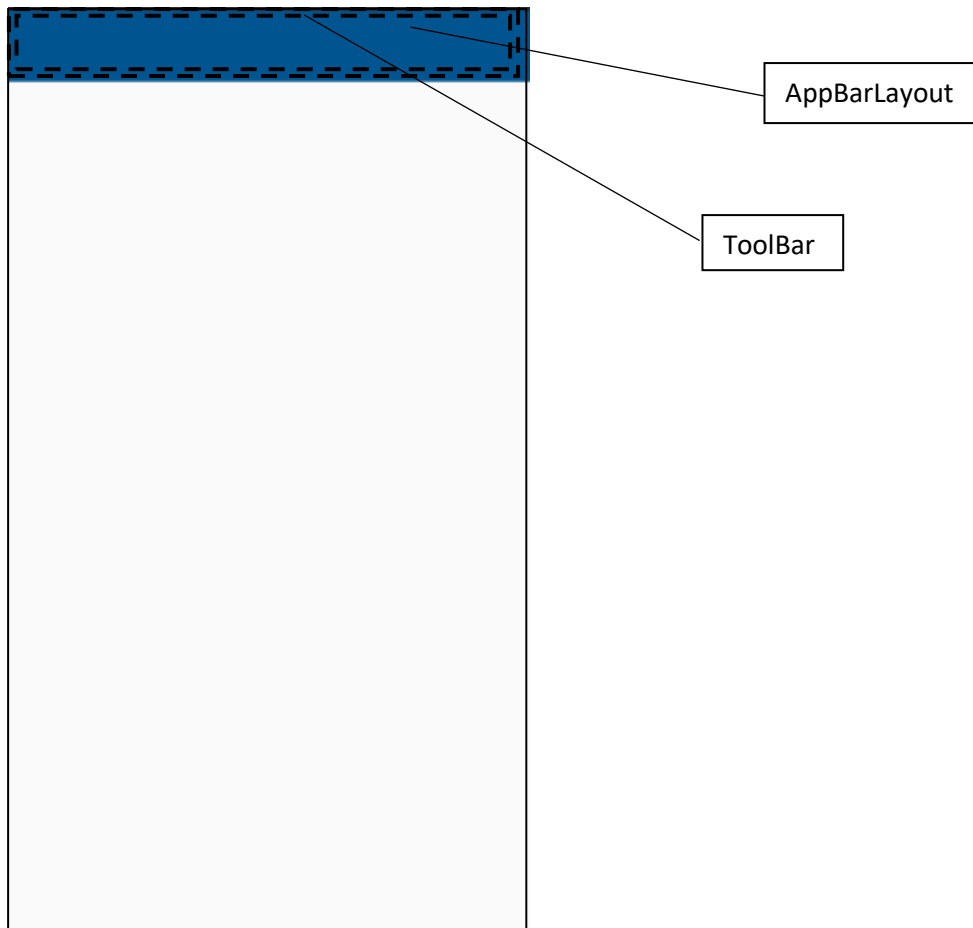


Рисунок 3.3 - Запланированное расположение виджетов AppBarLayout на экране

Далее виджеты, представленные на рисунке 3.3, определяются в файле `app_bar_main.xml`. Для определения виджетов в XML (`app_bar_main.xml`) был написан следующий модуль:

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.material.appbar.AppBarLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <androidx.appcompat.widget.Toolbar
```

```

android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="#005590"
app:popupTheme="@style/AppTheme.PopupOverlay" />

```

```
</com.google.android.material.appbar.AppBarLayout>
```

Как и все модули, написанные на языке XML, выполняют одну и ту же функцию - позиционирование виджетов на экране устройств. Иерархия представлений app_bar_main.xml показана на рисунке 3.4.

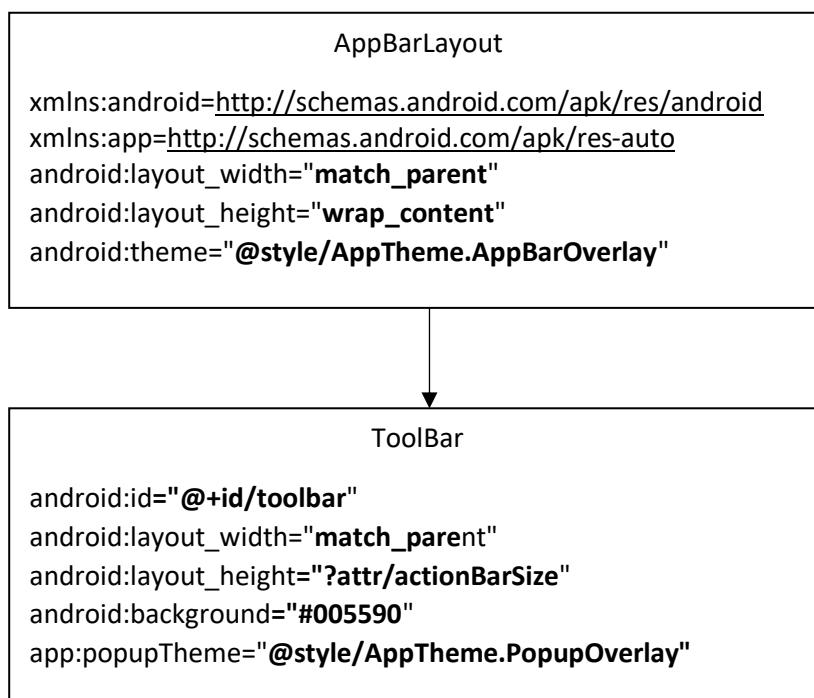


Рисунок 3.4 - Иерархия представлений макета ToolBar

3.2.2 Макеты подсистемы «Новости и объявления»

3.2.2.1 Макет newsss

В интерфейсе newsss определено 5 виджетов:

- RelativeLayout;
- RecyclerView;
- FloatingActionButton.

RelativeLayout - это группа представлений, которая отображает дочерние представления в относительных позициях. Положение каждого вида может быть задано относительно элементов одного уровня (например, слева или ниже другого вида) или в позициях относительно родительской области RelativeLayout (например, выровнено по низу, левому краю или центру). RelativeLayout - это очень мощная утилита для разработки пользовательского

интерфейса, поскольку она может исключить вложенные группы представлений и сохранить плоскость иерархии макета, что повышает производительность [18].

RecyclerView является представлением, которое повышает производительность работы со списками. Является намного оптимальнее, чем ListView.

FloatingActionButton - это представление модифицированной кнопки. Отличительной чертой является тот факт, что кнопка фиксируется в одной позиции экрана и не смещается при вертикальных скроллах.

На рисунке 3.5 показано, как из этих виджетов образуется интерфейс.

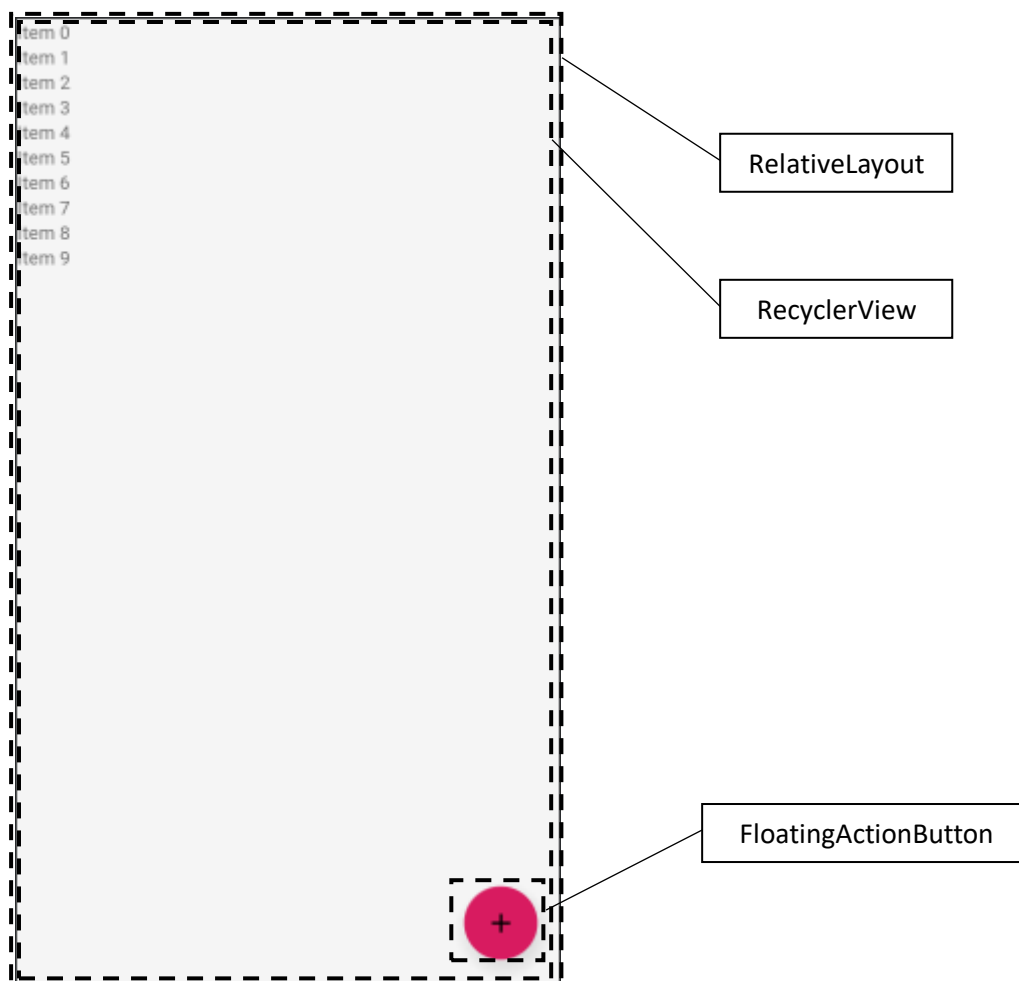


Рисунок 3.5 - Запланированное расположение виджетов newsss на экране

Для определения виджетов в XML (fragment_newsss.xml) был написан следующий модуль:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```

tools:context=".newsss">
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recycler_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F5F5F5"/>

    <com.google.android.material.floatingactionbutton.Float-
ingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="end|bottom"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:layout_margin="16dp"
        android:src="@drawable/ic_add_black_24dp" />
</RelativeLayout>

```

На основании данного модуля строится иерархия представления для данного макета, которая представлена на рисунке 3.6.

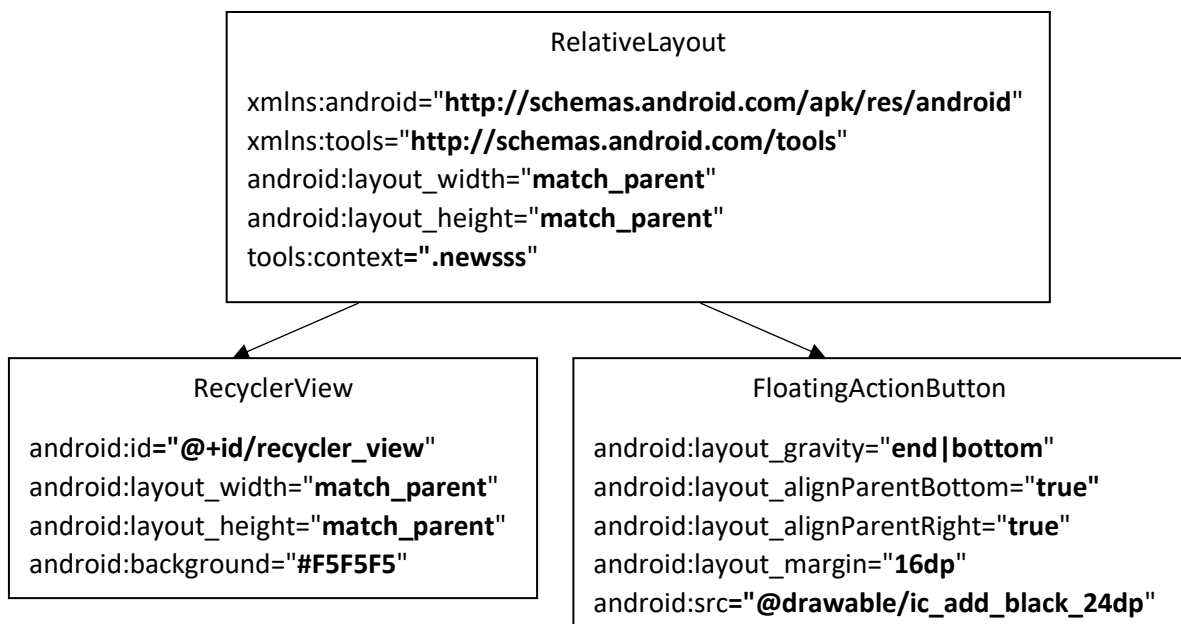


Рисунок 3.6 - Иерархия представлений макета newsss

3.2.2.2 Макет ExampleItem

В данном интерфейсе определено 7 виджетов:

- CardView;
- RelativeLayout;
- TextView (4);
- ImageView.

В Android 7.0 появился новый виджет `CardView`, который, по сути, можно представить как `FrameLayout` с закругленными углами и тенями в зависимости от его высоты. Стоит обратить внимание, что `CardView` оборачивает макет и часто является контейнером, используемым в макете для каждого элемента в `ListView` или `RecyclerView` [19]. `TextView` и `ImageView` исходя из названий служат для отображения текста и изображения соответственно.

Далее из этих виджетов образуется интерфейс `ExampleItem`, который представлен на рисунке 3.7.

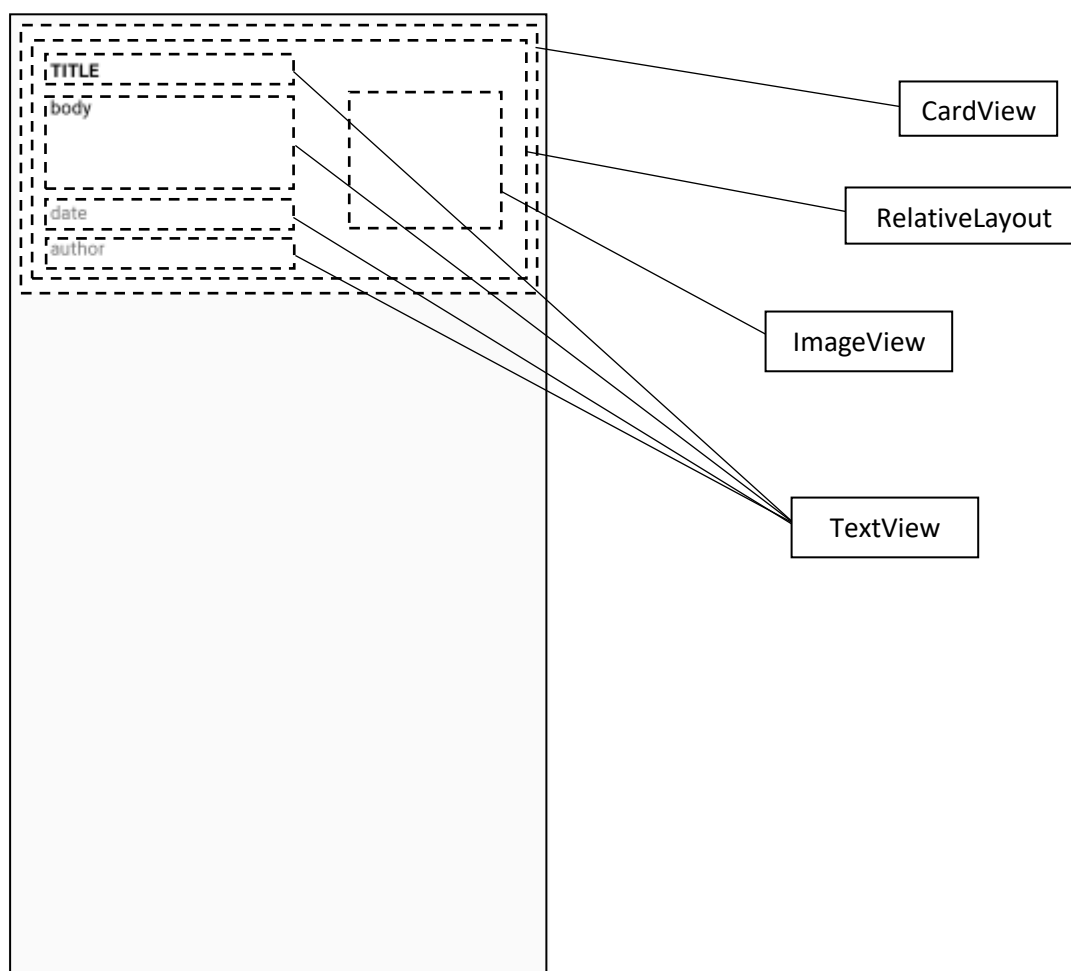


Рисунок 3.7 - Запланированное расположение виджетов `ExampleItem` на экране

Для определения виджетов в XML (`example_item.xml`) был написан следующий модуль:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    >
```

```

<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    android:background="#FFF"
    android:focusable="true"
    android:padding="16dp">

    <TextView
        android:id="@+id/title"
        android:layout_width="217dp"
        android:layout_height="wrap_content"
        android:clickable="false"
        android:text="TITLE"
        android:textColor="#000"
        android:textStyle="bold" />

    <ImageView
        android:id="@+id/imageUrl"
        android:layout_width="163dp"
        android:layout_height="123dp"
        android:layout_centerVertical="true"
        android:layout_marginLeft="2dp"
        android:layout_toRightOf="@id/title"
        android:clickable="false" />

    <TextView
        android:id="@+id/body"
        android:layout_width="218dp"
        android:layout_height="80dp"
        android:layout_below="@id/title"
        android:layout_marginTop="1dp"
        android:layout_marginRight="-218dp"
        android:layout_toLeftOf="@id/title"
        android:clickable="false"
        android:maxLines="4"
        android:paddingTop="8dp"
        android:text="body"
        android:textColor="#000"
    />

    <TextView
        android:id="@+id/date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/body"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@id/imageUrl"
        android:paddingTop="8dp"
        android:text="date" />

```

```

<TextView
    android:id="@+id/author"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/date"
    android:layout_alignParentLeft="true"
    android:paddingTop="8dp"
    android:text="author" />

</RelativeLayout>
</androidx.cardview.widget.CardView>

```

Исходя из вышеуказанного модуля была построена иерархия представлений, представленная на рисунке 3.8.

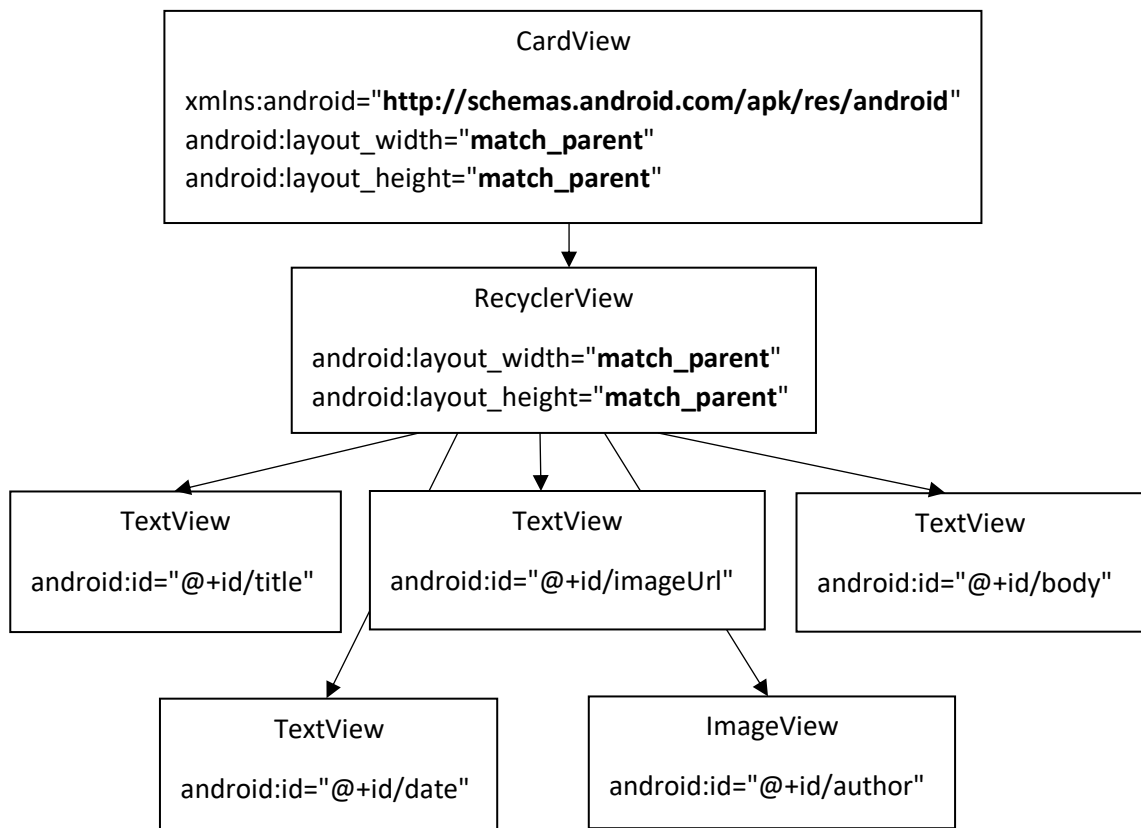


Рисунок 3.8 - Иерархия представлений макета ExampleItem

Макет для объявлений выглядит идентично макету новостей, единственное отличие в том, что в ExampleItem макет не содержит представления для изображения.

3.2.2.3 Макет InsertNews

Макет InsertViews состоит из 18 виджетов:

- ScrollView, который используется для прокрутки экрана по вертикали;
- вертикальный LinearLayout;
- TextView, используемые для отображения текстов;

- EditText служат для вставки текста пользователем;
- Button для выполнения события.

На рисунке 3.9 показано, как из этих виджетов образуется интерфейс InsertNews.

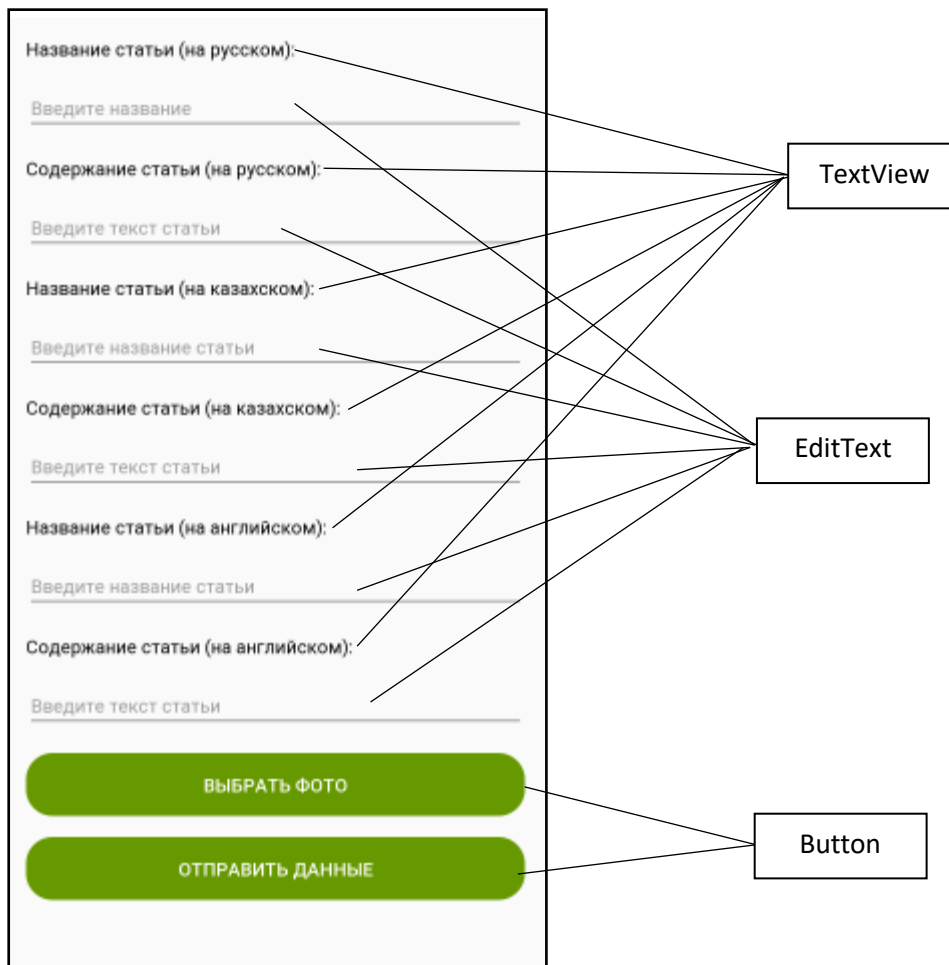


Рисунок 3.9 - Запланированное расположение виджетов ExampleItem на экране

3.2.2.4 Макет DetailActivityAnnounces

Макет DetailActivityAnnounces состоит из 6 виджетов:

- ScrollView, который используется для прокрутки экрана по вертикали;
- вертикальный LinearLayout;
- TextView, используемые для отображения текстов.

На рисунке 3.10 показано, как из этих виджетов образуется интерфейс DetailActivityAnnounces.

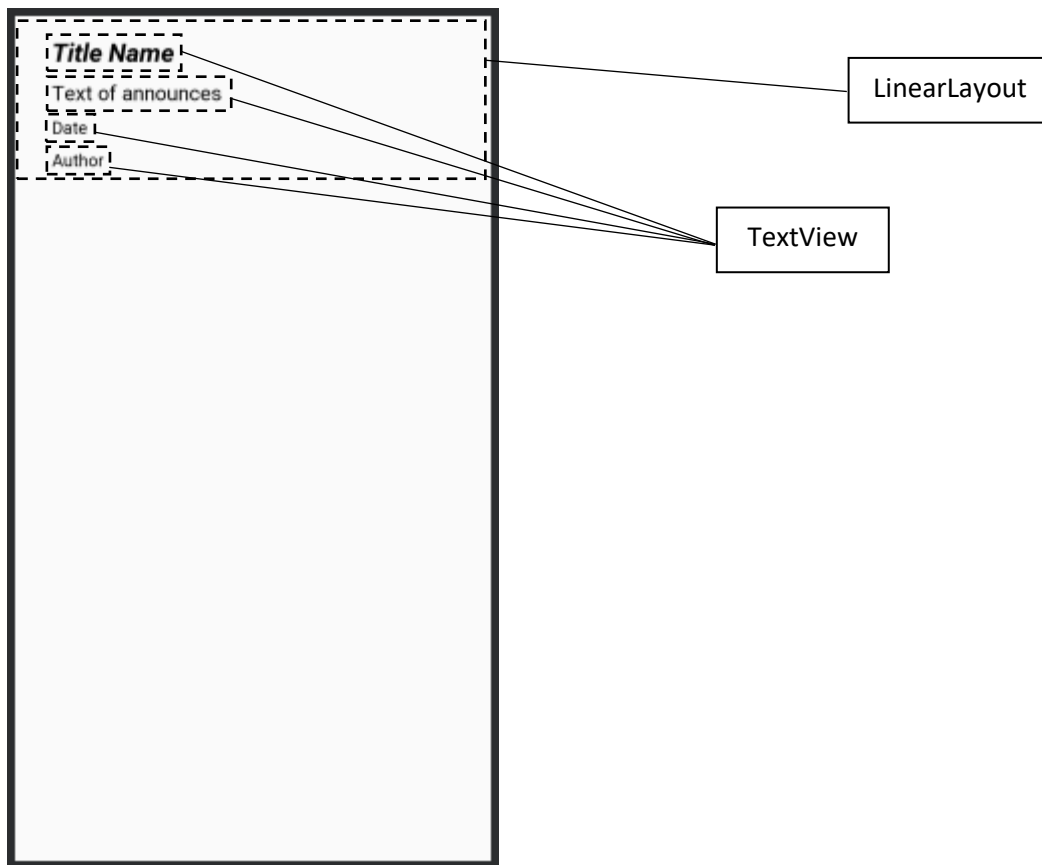


Рисунок 3.10 - Запланированное расположение виджетов `DetailActivityAnnounces` на экране

Для определения виджетов в XML (`activity_detail_announces.xml`) был написан следующий модуль:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scroll_two"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fillViewport="true">
    <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".DetailActivityAnnounces"
        android:orientation="vertical">

        <TextView
            android:textStyle="bold|italic"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/text_view_title_detail">
```

```
<TextView
    android:textStyle="bold|italic"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/text_view_title_detail"
    android:text = "Title Name"
    android:textSize="22sp"
    android:textColor="@android:color/black"
    android:layout_marginTop="16dp"
    android:layout_marginStart="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginEnd="32dp"
    android:layout_marginRight="32dp"
    android:layout_marginBottom="8dp"
/>
```

```
<TextView
    android:justificationMode="inter_word"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/text_view_body_detail"
    android:text = "Text of announces"
    android:textSize="18sp"
    android:layout_marginStart="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginEnd="32dp"
    android:layout_marginRight="32dp"
    android:layout_marginBottom="8dp"
    android:textColor="@android:color/black"
    android:autoLink="web"
/>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/text_view_date_detail"
    android:text = "Date"
    android:textSize="15sp"
    android:layout_marginStart="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginEnd="32dp"
    android:layout_marginRight="32dp"
    android:layout_marginBottom="8dp"
    android:textColor="@android:color/black"
/>
```

```

        android:layout_marginStart="32dp"
        android:layout_marginLeft="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginRight="32dp"
        android:layout_marginBottom="8dp"
        android:textColor="@android:color/black"
    />

</LinearLayout>
</ScrollView>

```

3.2.2.5 Макет DetailActivity

Макет DetailActivity состоит из 7 виджетов:

- ScrollView, который используется для прокрутки экрана по вертикали;
- вертикальный LinearLayout;
- TextView, используемые для отображения текстов;
- ImageView для отображения изображения новости.

На рисунке 3.11 показано, как из этих виджетов образуется интерфейс DetailActivity.

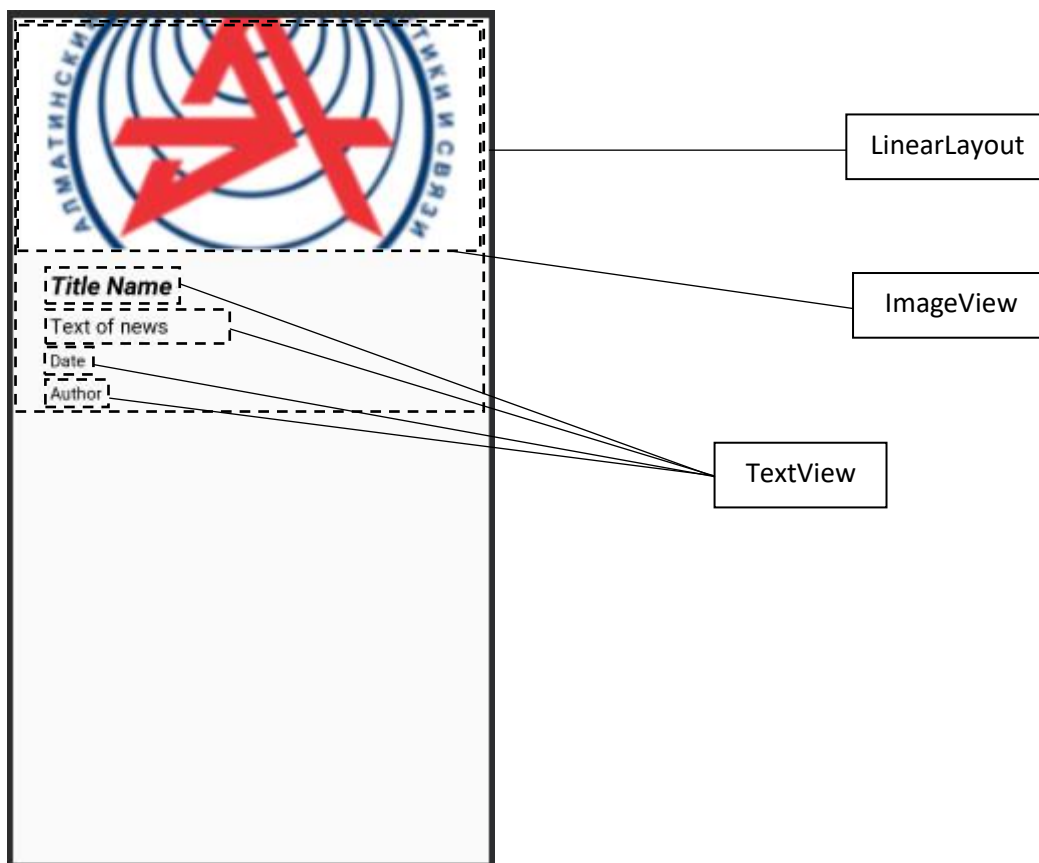


Рисунок 3.11 - Запланированное расположение виджетов DetailActivity на экране

Для определения виджетов в XML (activity_detail.xml) был написан следующий модуль:

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/scroll_two"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fillViewport="true">
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".DetailActivity"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/image_view_detail"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:src="@mipmap/ic_launcher"
            android:scaleType="centerCrop"
            />

        <TextView
            android:textStyle="bold|italic"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/text_view_title_detail"
            android:text = "Title Name"
            android:textSize="22sp"
            android:textColor="@android:color/black"
            android:layout_marginTop="16dp"
            android:layout_marginStart="32dp"
            android:layout_marginLeft="32dp"
            android:layout_marginEnd="32dp"
            android:layout_marginRight="32dp"
            android:layout_marginBottom="8dp"
            />

        <TextView
            android:justificationMode="inter_word"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/text_view_body_detail"
            android:text = "Text of news"
            android:textSize="18sp"
            android:layout_marginStart="32dp"
```

```

        android:layout_marginLeft="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginRight="32dp"
        android:layout_marginBottom="8dp"
        android:textColor="@android:color/black"
        android:autoLink="web"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/text_view_date_detail"
        android:text = "Date"
        android:textSize="15sp"
        android:layout_marginStart="32dp"
        android:layout_marginLeft="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginRight="32dp"
        android:layout_marginBottom="8dp"
        android:textColor="@android:color/black"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/text_view_author_detail"
        android:text = "Author"
        android:textSize="15sp"
        android:layout_marginStart="32dp"
        android:layout_marginLeft="32dp"
        android:layout_marginEnd="32dp"
        android:layout_marginRight="32dp"
        android:layout_marginBottom="8dp"
        android:textColor="@android:color/black"
    />

    </LinearLayout>
</ScrollView>

```

3.2.3 Модуль «Личный кабинет»

Данный модуль имеет 6 вкладок:

- индивидуальный учебный план, где описаны образовательные программы студента;
- академический календарь (календарь проведения учебных и контрольных мероприятий);
- транскрипт, где указаны названия пройденных курсов и полученные оценки;
- задолженности (информация о задолженности по оплате обучающихся);
- библиотека (электронная библиотека АУЭС);

- дистанционные курсы (дистанционные курсы АУЭС).

3.2.3.1 Макет ListOfCalendar

Макет ListOfCalendar состоит из привычных элементов-вью:

- TextView;
- LinearLayout;
- RecyclerView.

Основное назначение данного макета - отображение списком академические календари для 1, 2, 3, 4 курсов. В RecyclerView хранятся лишь ссылки на PDF-файлы, при нажатии на элемент списка открывается всплывающее окно, в котором предлагается открыть или загрузить файл. При нажатии одно из двух действий открывается приложение для отображения выбранного файла. Файлы хранятся в базе данных университета. На рисунке 3.12 показано, как из этих виджетов образуется интерфейс ListOfCalendar.

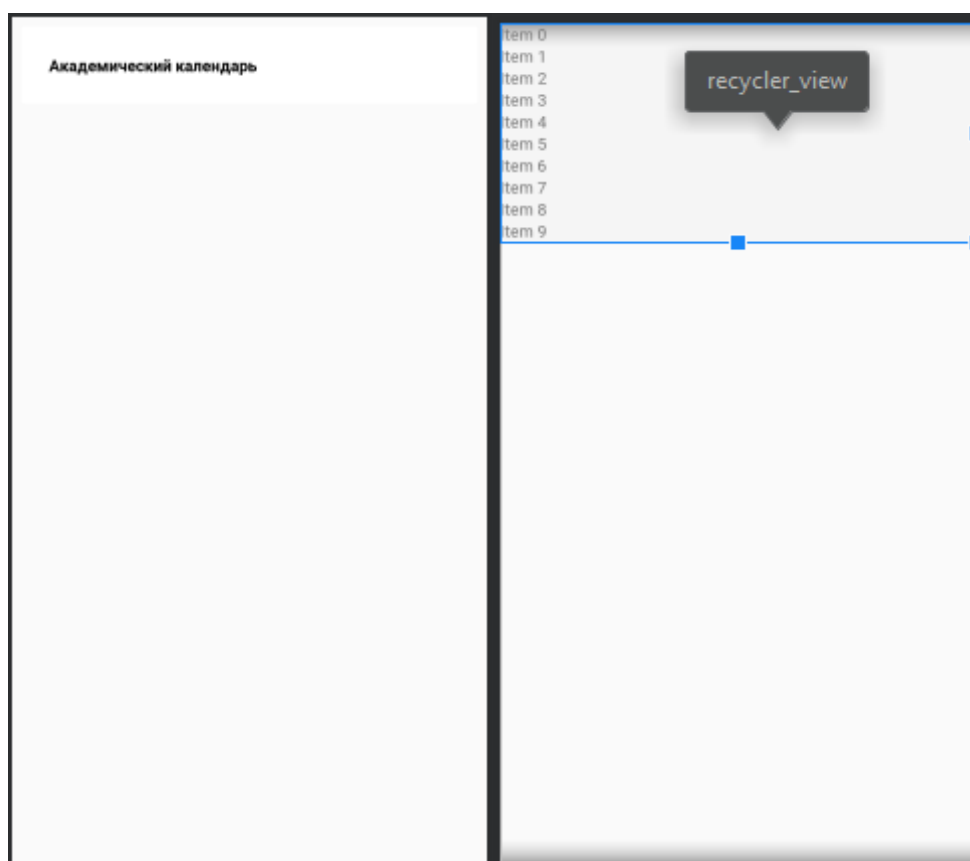


Рисунок 3.12 - Запланированное расположение виджетов ListOfCalendar на экране

3.2.4 Интерфейс подсистемы «Электронный журнал»

3.2.4.1 Макет ExampleItemCurrentMarks

Макет ExampleItemCurrentMarks состоит из 6 виджетов:

- CardView;
- LinearLayout;
- TextView (3).

В Android 7.0 появился новый виджет CardView, который, по сути, можно представить как FrameLayout с закругленными углами и тенями в зависимости от его высоты. Стоит обратить внимание, что CardView оборачивает макет и часто является контейнером, используемым в макете для каждого элемента в ListView или RecyclerView [19].

В TextView отображается дата выставления оценки, баллы и занятие. В TabView отображается наименование дисциплины, за которую выставлены оценки.

Далее из этих виджетов образуется интерфейс ExampleItemCurrentMarks, который представлен на рисунке 3.13.

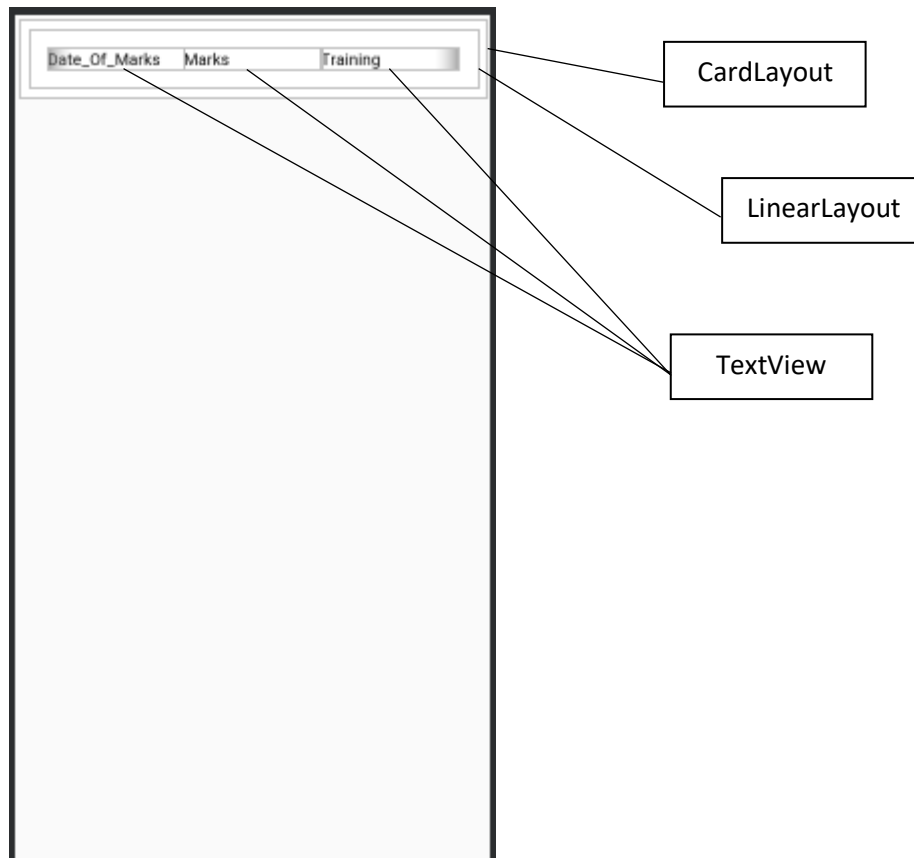


Рисунок 3.13 - Запланированное расположение виджетов ExampleItemCurrentMarks на экране

3.2.4.2 Макет ExampleItemFinalMarks

Макет ExampleItemFinalMarks состоит из 14 виджетов:

- CardView;
- RelativeLayout;

- LinearLayout;
- TextView (10).

В данном макете отображаются название дисциплины в верхней части. Снизу следуют два горизонтальных LinearLayout, в которых отображаются оценки:

- средняя текущая оценка;
- рубежный контроль 1;
- рубежный контроль 2;
- РГР1;
- РГР2;
- РГР3;
- РГР4;
- курсовая работа;
- рейтинг допуска;
- экзамен.

Далее из этих виджетов образуется интерфейс ExampleItemFinalMarks, который представлен на рисунке 3.14.

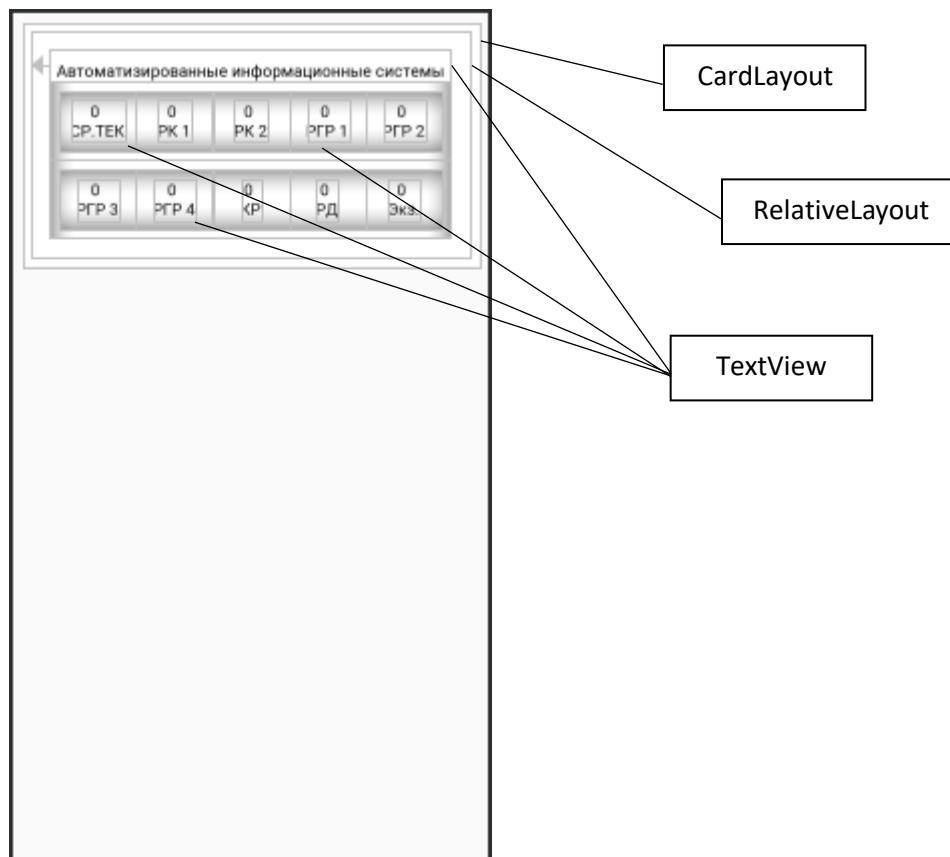


Рисунок 3.14 - Запланированное расположение виджетов ExampleItemFinalMarks на экране

3.2.4.3 Макет ExampleItemTeachersDiscipline

Макет ExampleItemTeachersDiscipline состоит из 9 виджетов:

- CardView;
- RelativeLayout;
- LinearLayout;
- View;
- TextView (5).

В данном макете отображаются название дисциплин, которую ведет преподаватель и группа, которая закреплена за этой дисциплиной и преподавателем.

Далее из этих виджетов образуется интерфейс ExampleItemTeachersDiscipline, который представлен на рисунке 3.15.

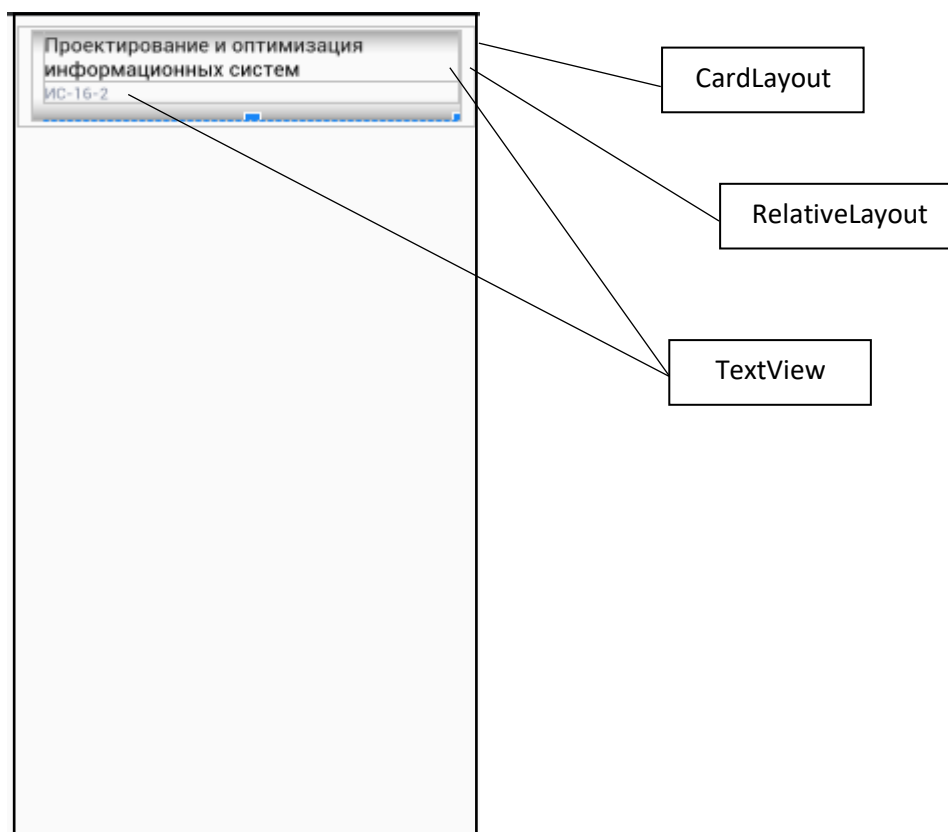


Рисунок 3.15 - Запланированное расположение виджетов ExampleItemTeachersDiscipline на экране

3.2.4.4 Макет ActivityPutCurrentMarks

Макет ActivityPutCurrentMarks состоит из 6 виджетов:

- LinearLayout (3);
- Button;
- RecyclerView (2).

В данном макете отображаются данные студентов и поля для выставления оценок, а также кнопка для отправки данных.

Далее из этих виджетов образуется интерфейс `ActivityPutCurrentMarks`, который представлен на рисунке 3.16.

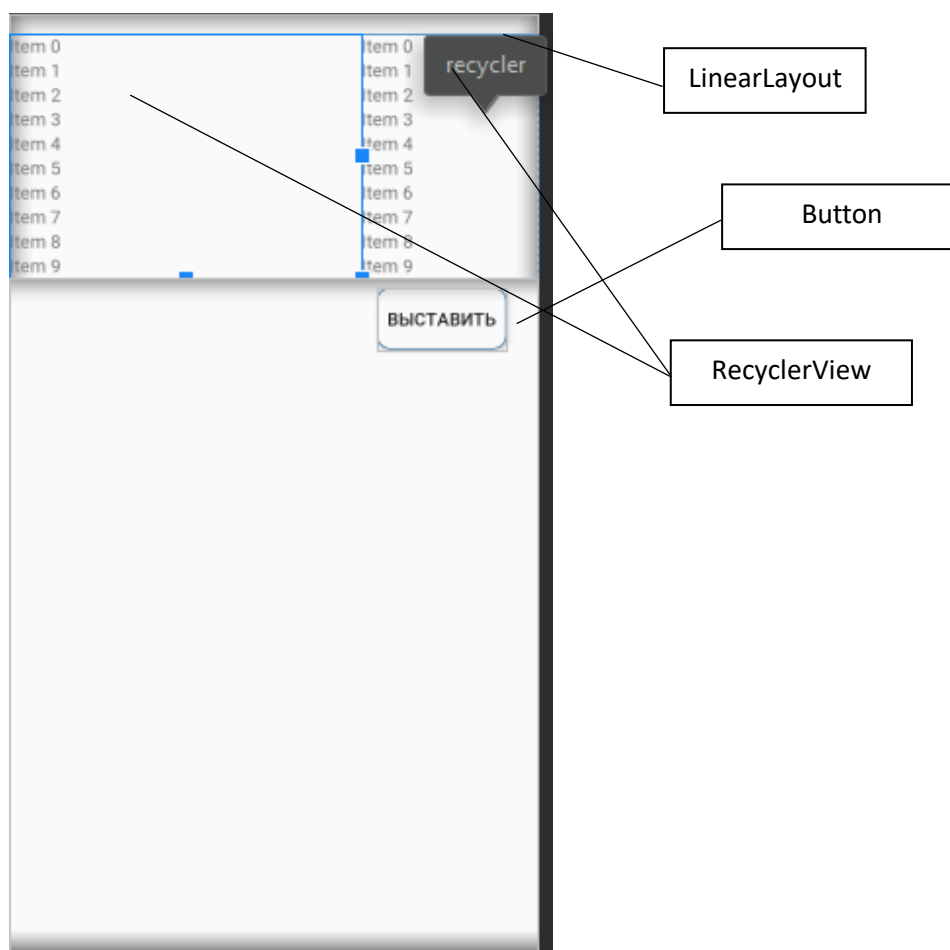


Рисунок 3.16 - Запланированное расположение виджетов `ActivityPutCurrentMarks` на экране

3.3 Структура программного обеспечения

Структура программного обеспечения ИС состоит из системного ПО, инструментального ПО и прикладного ПО. Для запуска мобильного приложения необходимо системное программное обеспечение, включающее в себя операционную систему Android 4.2 Jelly Bean и выше. Инструментальное программное обеспечение включает в себя IDE Android Studio 3.5, языка программирования Java SE 11, СУБД MySQL 8.0.17, хостинга 000webhost.com и серверного языка программирования PHP 7.3. Для написания мобильного приложения в среде IDE Android Studio использовалась ОС Windows 10. Запуск приложения производился на смартфонах Samsung Galaxy A50, Xiaomi Redmi 5 Plus на платформе Android 9 Pie, а также на Samsung Galaxy A51 на платформе Android Q. Схематично структура представлена на рисунке 3.17.

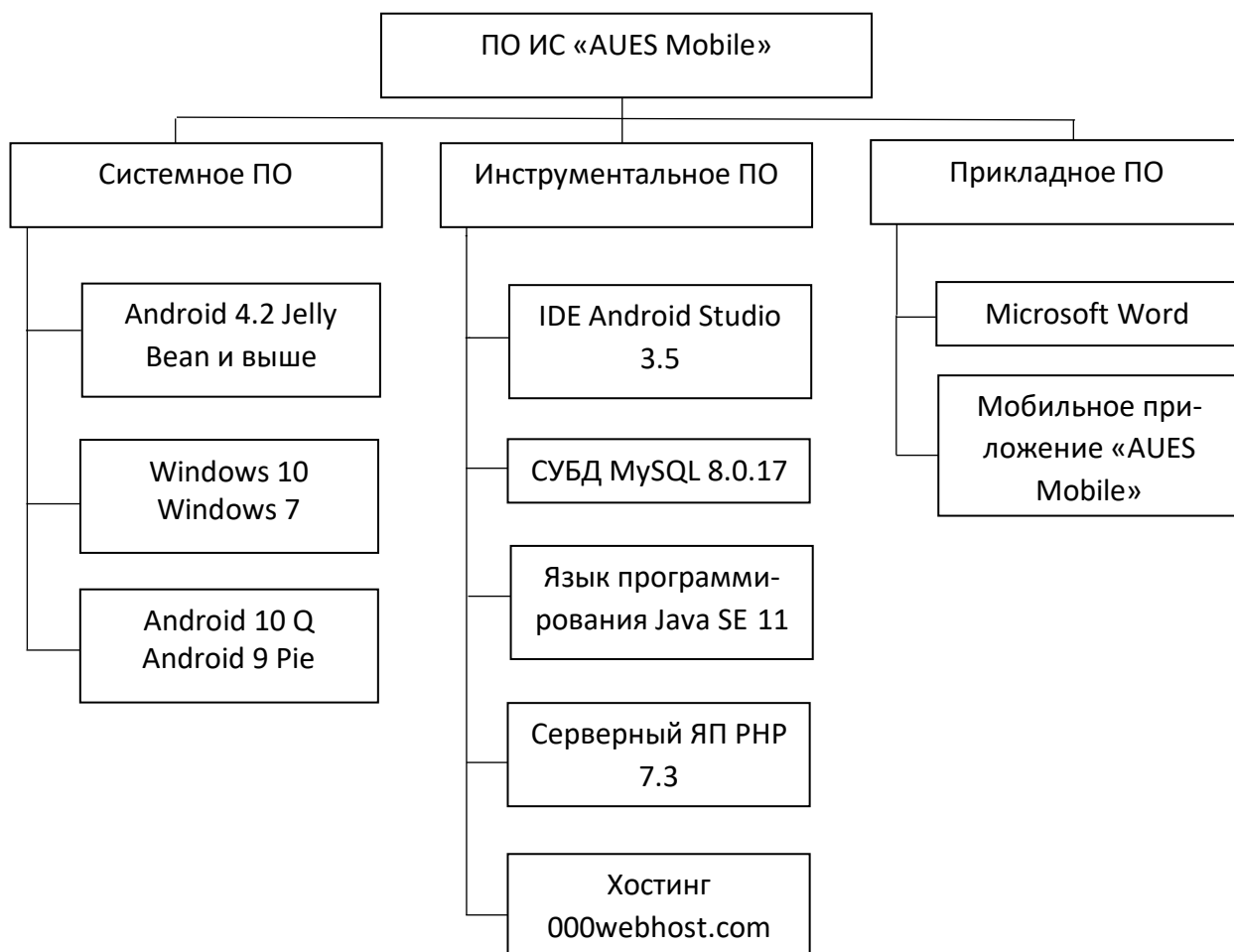


Рисунок 3.17 - Структура ПО

3.4 Клиентская часть

На этапе прототипирования приложения были созданы пользовательские UI интерфейсы с расположением View-объектов, теперь необходимо написать код для вывода необходимой информации. Код, представленный в листинге необходим для функционирования авторизации. Данный код использует основные конструкции, представленные в таблице 3.1.

Таблица 3.1 - Основные конструкции класса *entrance*

№	Конструкция	Пояснения
1	String login_url = "https://auesmobiledatabase.000webhostapp.com/login.php";	Переменная, хранящая URL-адрес скрипта PHP
2	StringRequest stringRequest = new StringRequest(Request.Method.POST, login_url, new Response.Listener<String>() {})	Объект, возвращающий от-

	вет сервера
--	-------------

Продолжение таблицы 3.1

3	<code>public void onResponse(String response) {}</code>	В переменной <code>response</code> содержится строковая переменная, содержащая в себе ответ сервера
4	<code>SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());</code>	Объект данного класса служит для хранения пользовательских данных в приложении
5	<code>protected Map<String, String> getParams() throws AuthFailureError{}</code>	Метод передает данные, введенные пользователем на странице авторизации, серверу

Код, представленный в листинге необходим для извлечения объявлений с сервера. Данный код использует основные конструкции, представленные в таблице 3.2.

Таблица 3.2 - Основные конструкции класса `announces`

№	Конструкция	Пояснения
1	<code>String url = "https://auesmobiledatabase.000webhostapp.co</code>	Переменная, хранящая URL

	<code>m/announces.php";</code>	нящая URL-адрес скрипта PHP
2	<pre>JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, url, (String) null, new Response.Listener<JSONObject>() {}</pre>	Объект, возвращающий ответ сервера в виде JSON-объекта
3	<pre>public void onResponse(JSONObject response) {}</pre>	В переменной response содержится JSON-объект, содержащий в себе ответ сервера
4	<pre>mSharedPreferences = PreferenceManager.getDefaultSharedPreferences(getActivity().getBaseContext());</pre>	При помощи данного кода извлекаются ранее сохраненные пользовательские данные
5	<pre>ExampleAdapterAnnounces.OnItemClickListener exampleAdapter= new ExampleAdapterAnnounces.OnItemClickListener() {}</pre>	Метод, позволяющий открывать подробное описание объявления

Код в листинге позволяет пользователю с ролью «Администратор» вносить данные об объявлении. Данный класс содержит следующие основные конструкции, представленные в таблице 3.3.

Таблица 3.3 - Основные конструкции класса announces

№	Конструкция	Пояснения
1	<code>String URL = "https://auesmobiledatabase.000webhostapp.com/addannounces.php";</code>	Переменная, хранящая URL-адрес скрипта PHP
2	<code>JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, url, (String) null, new Response.Listener<JSONObject>() {})</code>	Объект, возвращающий ответ сервера в виде JSON-объекта
3	<code>add_announces.setOnClickListener(new View.OnClickListener() {})</code>	Инициализируется обработчик события нажатия на кнопку
4	<code>StringRequest stringRequest = new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {})</code>	Строковый ответ сервера
5	<code>protected Map<String, String> getParams() throws AuthFailureError { Map<String, String> values = new HashMap<String, String>();</code>	Метод, передающий входные данные серверу

3.5 Серверная часть

Для того, чтобы извлечь данные, хранящиеся на сервере, необходимо написать скрипты. Сначала проверяется соединение к базе данных, далее в скрипте методом POST берутся данные о логине и пароле, которые передал клиент. Пишется соответствующий SQL-запрос, который проверяет имеется ли пользователь с отправленными параметрами. Далее проверяется, сколько строк возвратил запрос, если больше 0, значит пользователь существует и выполняются другие запросы, упаковывая данные в JSON-массив и отправ-

ляя его в качестве ответа клиенту. Скрипт ниже осуществляет функционал авторизации.

```

<?php
require "int.php";
mysqli_set_charset($conn, 'utf8');
$user_name = $_POST["user_name"];
$password = $_POST["password"];

$sql = "SELECT LOGIN, PASSWORD, ROLE_ID_ROLE, IMAGE_URL from USERS where LOGIN = '". $user_name."' and PASSWORD = '"
.$password."'";

$result = mysqli_query($conn, $sql);
$response = array();
if (mysqli_num_rows($result) > 0) {
    $row = mysqli_fetch_row($result);
    $login = $row[0];
    $code = "login success";
    $role_id = $row[2];
    $image_url = $row[3];
    $sql2 = "";
    if ($role_id == 2) {
        $sql2 = "SELECT CONCAT(NAME, ' ', SURNAME) AS ALIAS, NAME_OF_GROUP from STUDENT JOIN USERS ON USERS.LOGIN =
        STUDENT.LOGIN
        JOIN CLASS ON CLASS.ID_GROUP = STUDENT.ID_GROUP where USERS.LOGIN = '". $login."'";
        $result2 = mysqli_query($conn, $sql2);
        if (mysqli_num_rows($result2) > 0) {
            $row2 = mysqli_fetch_row($result2);
            $alias = $row2[0];

            $group_name = $row2[1];
            array_push($response,array("code"=>$code, "login"=>$login, "alias" => $alias, "image_url"=>$image_url,
            "role_id" =>$role_id, "group_name"=>$group_name));
        }
    }
    else if ($role_id == 3) {
        $sql3 = "SELECT CONCAT(NAME, ' ', SURNAME) AS ALIAS, ID_TEACHER from TEACHER JOIN USERS ON USERS.LOGIN =
        TEACHER.LOGIN where USERS.LOGIN = '". $login."'";
        $result3 = mysqli_query($conn, $sql3);
        if (mysqli_num_rows($result3) > 0) {
            $row3 = mysqli_fetch_row($result3);
            $alias = $row3[0];
            $id_teacher = $row3[1];
            array_push($response,array("code"=>$code, "login"=>$login, "alias" => $alias, "image_url"=>$image_url,
            "role_id" =>$role_id, "id_teacher"=>$id_teacher));
        }
    }
    else if ($role_id == 1) {
        $sql4 = "SELECT CONCAT(FIRSTNAME, ' ', NAME) AS ALIAS from ADMINISTRATOR JOIN USERS ON USERS.LOGIN =
        ADMINISTRATOR.LOGIN where USERS.LOGIN = '". $login."'";
        $result4 = mysqli_query($conn, $sql4);
        if (mysqli_num_rows($result4) > 0) {
            $row4 = mysqli_fetch_row($result4);
            $alias = $row4[0];
            array_push($response,array("code"=>$code, "login"=>$login, "alias" => $alias, "image_url"=>$image_url,
            "role_id" =>$role_id));
        }
    }
}

echo json_encode($response, JSON_UNESCAPED_UNICODE, JSON_FORCE_OBJECT);
}
else
{
    $code = "login_failed";
    $message = "User not found... Please try again";
    array_push($response,array("code"=>$code, "message"=>$message));
    echo json_encode($response);
}
mysqli_close($conn);
?>

```

Следующий скрипт также сначала проверяет подключение к базе данных. Пишется SQL-запрос, который возвращает все объявления из БД в порядке DESC. Все возвращаемые строки помещаются в JSON-массив и отправляются клиенту. Таким образом скрипт ниже извлекает все объявления из БД.

```

<?php
$host = "localhost";
$db_user = "id11311399_aues";
$password = "auesmobile";
$db_name = "id11311399_aues";

$conn = mysqli_connect($host, $db_user, $password, $db_name);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

```

```

<?php

require "int.php";
$lang = "ru"; //post must be
if ($lang == "ru") {
    $sql = "SELECT NAME, PATRONYMIC, TITLE_ANNOUNCES_RU, BODY_ANNOUNCES_RU, TITLE_ANNOUNCES_KZ, BODY_ANNOUNCES_KZ,
            TITLE_ANNOUNCES_EN, BODY_ANNOUNCES_EN, DATE_ANNOUNCES FROM ANNOUNCES JOIN ADMINISTRATOR ON ANNOUNCES.ID_ADMIN
            = ADMINISTRATOR.ID_ADMIN ORDER BY ID_ANNOUNCES DESC;";
    $result = mysqli_query($conn, $sql);
    $json_array = array();
    while ($row = mysqli_fetch_assoc($result))
    {
        $json_array[] = $row;
    }
    echo json_encode(array('announces' => $json_array), JSON_UNESCAPED_UNICODE, JSON_FORCE_OBJECT);
}

?>

```

Аналогично вышеописанному скрипту, следующий скрипт извлекает данные о новостях. Скрипт подобен скрипту с объявлениями, только данные берутся из другой таблицы БД.

```

<?php

require "int.php";
$lang = "ru"; //post must be
if ($lang == "ru") {
    $sql = "SELECT * FROM news ORDER BY ID_NEWS DESC;";
    $result = mysqli_query($conn, $sql);

    $json_array = array();
    while ($row = mysqli_fetch_assoc($result))
    {
        $json_array[] = $row;
    }
    echo json_encode(array('news' => $json_array), JSON_UNESCAPED_UNICODE, JSON_FORCE_OBJECT);
}

?>

```

Для пользователя администратор определены дополнительные скрипты, которые осуществляют добавления новостей и объявлений. Следующий скрипт принимает данные, отправленные клиентом, методом POST. Данные проверяются на пустые значения, если хотя бы одна переменная будет не заполнена, скрипт отправит клиенту сообщение об ошибке. В противном случае данные будут занесены в таблицу объявлений при помощи DDL-операции INSERT.


```

<?php
require "int.php";
$title_ru = $_POST["title_ru"];
$title_kz = $_POST["title_kz"];
$title_eng = $_POST["title_eng"];
$body_ru = $_POST["body_ru"];
$body_kz = $_POST["body_kz"];
$body_eng = $_POST["body_eng"];

function IsNullOrEmptyString($question)
{
    return (!isset($question) || trim($question)=='');
}

if (IsNullOrEmptyString($title_ru) || IsNullOrEmptyString($title_kz) || IsNullOrEmptyString($title_eng) ||
    IsNullOrEmptyString($body_eng) || IsNullOrEmptyString($body_ru) || IsNullOrEmptyString($body_kz))
{
    echo "Ошибка! Некорректные данные";
}
else
{
    $sql = "Insert into ANNOUNCES (TITLE_ANNOUNCES_RU, TITLE_ANNOUNCES_KZ, TITLE_ANNOUNCES_EN, BODY_ANNOUNCES_RU,
        BODY_ANNOUNCES_KZ, BODY_ANNOUNCES_EN) values('".$title_ru."', '".$title_kz."', '".$title_eng."', '".$body_ru
        ."', '".$body_kz."', '".$body_eng."');";
    if (mysqli_query($conn, $sql))
    {
        echo "Данные были успешно внесены";
    }
    else
    {
        echo "Ошибка! Данные не были внесены, попробуйте позже";
    }
}
mysqli_close($conn);
?>

```

Следующий скрипт позволяет преподавателям выставлять текущие оценки. В данном случае от приложения отправляется массив данных идентификаторы студента, группа, дисциплина, занятие.

```

<?php
require "int.php";
mysqli_set_charset($conn, 'utf8');
$response = array();
$answer = "";
$code = $_POST["code"];
$group = $_POST["group"];
$training = $_POST["training"];
$sql = "SELECT CONCAT(STUDENT.SURNAME, ' ', STUDENT.NAME, ' ', STUDENT.PATRONYMIC) AS STUDENT_NAME, INDIVIDUAL_PLAN
.ID_STUDENT AS ID FROM
INDIVIDUAL_PLAN JOIN STUDENT ON INDIVIDUAL_PLAN.ID_STUDENT = STUDENT.ID_STUDENT WHERE
INDIVIDUAL_PLAN.ID_GROUP = '".$group.'" AND
INDIVIDUAL_PLAN.CODE_DISCIPLINE = '".$code.'" AND
INDIVIDUAL_PLAN.ID_TRAINING = '".$training.'"
ORDER BY STUDENT_NAME;";

$result = mysqli_query($conn, $sql);
$json_array = array();
while ($row = mysqli_fetch_assoc($result))
{
    $json_array[] = $row['ID'];
}

$size = (int)$_POST["size"];
$array = array();

```

```

$array[0] = (int)$_POST["params_" . 0];
if ($size == 1) {
    $sql = "INSERT INTO CURRENT_MARKS (MARK, ID_STUDENT, ID_GROUP, CODE_DISCIPLINE, ID_TRAINING) VALUES ('"
        . $array[0]."', '". $json_array[0]."', '". $group."', '". $code."', '". $training."')";
}
else {
    $sql = "INSERT INTO CURRENT_MARKS (MARK, ID_STUDENT, ID_GROUP, CODE_DISCIPLINE, ID_TRAINING) VALUES ('"
        . $array[0]."', '". $json_array[0]."', '". $group."', '". $code."', '". $training."')";
    for ($i = 1; $i < $size; $i++)
    {
        $array[$i] = (int) $_POST["params_" . $i];
        if ($i == $size - 1) {
            $sql .= "(" . $array[$i]."', '". $json_array[$i]."', '". $group."', '". $code."', '". $training."')";
        }
        else {
            $sql .= "(" . $array[$i]."', '". $json_array[$i]."', '". $group."', '". $code."', '". $training."')";
        }
    }
}

if (mysqli_query($conn, $sql)) {
    $answer = "New records created successfully";
} else {
    $answer = "Error: " . $sql . "<br>" . mysqli_error($conn);
}
array_push($response, array("answer"=>$answer));
echo json_encode($response, JSON_UNESCAPED_UNICODE, JSON_FORCE_OBJECT);

for ($i = 0; $i < count($json_array); $i++) {
    $sql = "CALL MARKS('". $json_array[$i]."', '". $code."')";
    mysqli_query($conn, $sql);
}

```

3.6 Тестирование и отладка программного обеспечения

После разработки прототипа и описания классов активностей и фрагментов запускается приложение на смартфоне Samsung Galaxy A50 (6.4 дюйм).

Сначала пользователь должен авторизоваться в системе. Процесс авторизации показан на рисунке 3.18.

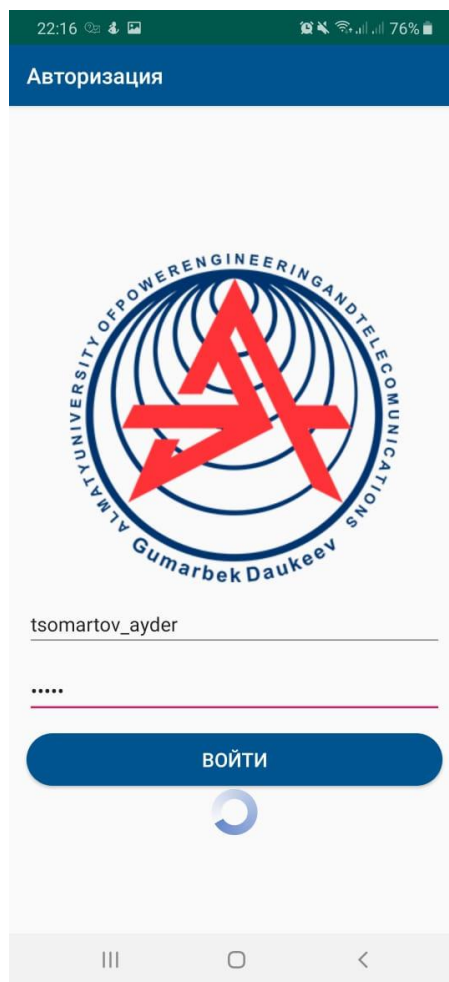


Рисунок 3.18 - Авторизация пользователя

После того как пользователь успешно авторизовался, ему открывается активити с фрагментом новостей. На рисунке 3.19 и 3.20 показаны фрагменты (принадлежащие MainActivity) объявлений и новостей соответственно.

После нажатия пользователем на одну из новостей или объявлений, открывается новая деятельность с детальным описанием выбранной новости и объявления. На рисунке 3.21 и 3.22 показана подробная новость и объявление соответственно. Кнопка FloatingActionButton на рисунках 3.19 и 3.20 предоставлена для пользователя «Администратор». По нажатию на одну из кнопок открывается активити с формой заполнения новости или объявления. На рисунках 3.23 и 3.24 показаны формы для них. На рисунке 3.24 показана форма после выбора изображения из внутреннего хранилища устройства к новости.

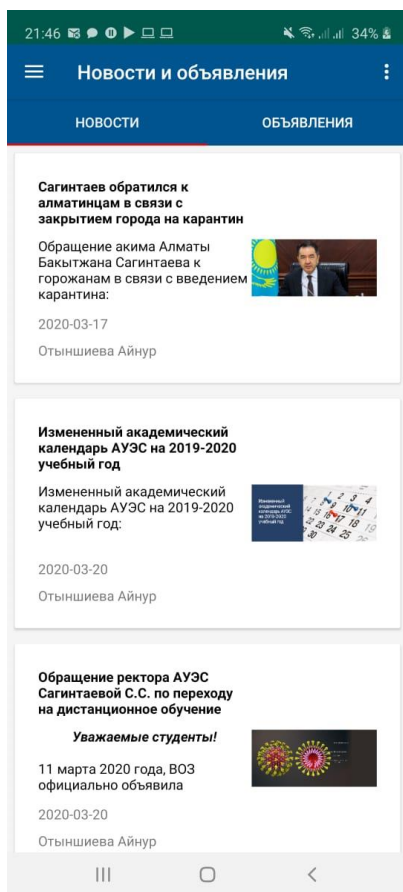


Рисунок 3.19 - Новости

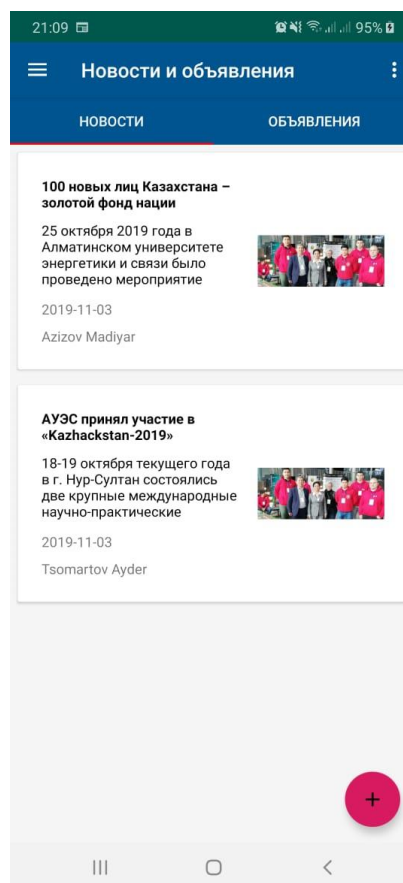


Рисунок 3.20 - Объявления

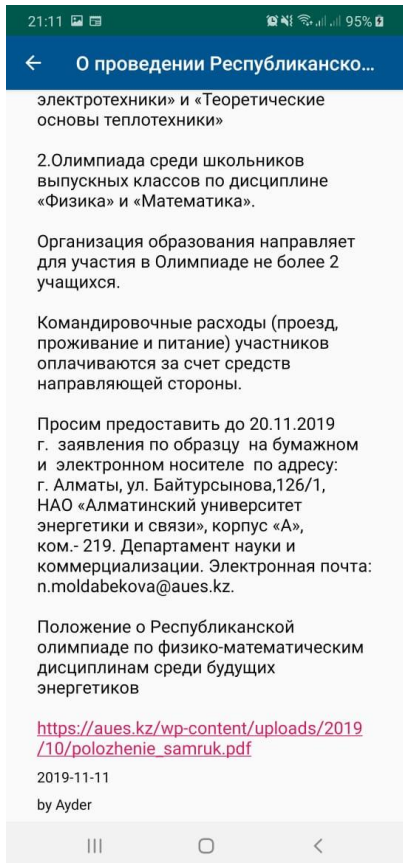


Рисунок 3.21 - Подробная новость
ление

21:10 95%

← AUES

Название статьи (на русском):
Введите название

Содержание статьи (на русском):
Введите текст статьи

Название статьи (на казахском):
Введите название статьи

Содержание статьи (на казахском):
Введите текст статьи

Название статьи (на английском):
Введите название статьи

Содержание статьи (на английском):
Введите текст статьи

ВЫБРАТЬ ФОТО

ОТПРАВИТЬ ДАННЫЕ

Рисунок 3.23 - Добавление новости
ния

Рисунок 3.22 - Подробное объяв-

21:10 95%

← AUES


Введите текст статьи

Название статьи (на казахском):
Введите название статьи

Содержание статьи (на казахском):
Введите текст статьи

Название статьи (на английском):
Введите название статьи

Содержание статьи (на английском):
Введите текст статьи



Введите название

ВЫБРАТЬ ФОТО

ОТПРАВИТЬ ДАННЫЕ

Рисунок 3.24 - Выбор изображе-

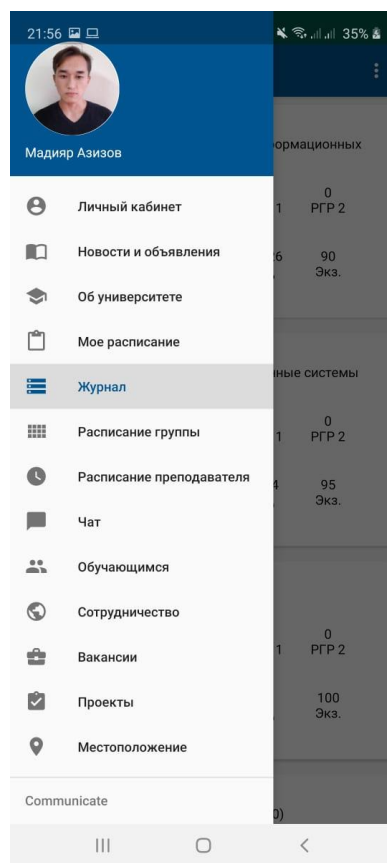
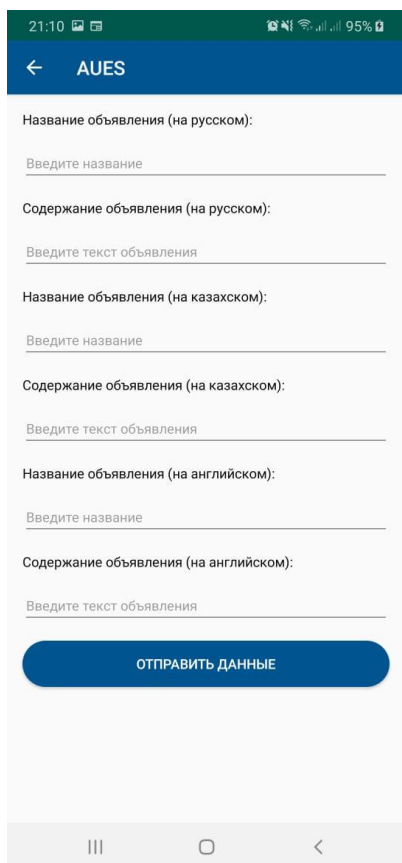


Рисунок 3.25 - Добавление объявления Рисунок 3.26 - Панель навигации

Теперь проверим работоспособность опубликования новостей. На рисунке 3.27 показан пример заполнения полей новостей с выбором изображения и сведения о том, что новость успешно добавлена. Для того, чтобы проверить это, необходимо вернуться на страницу новостей и объявлений. Видно, что на рисунке 3.28, новость добавлена и «висит» сверху.

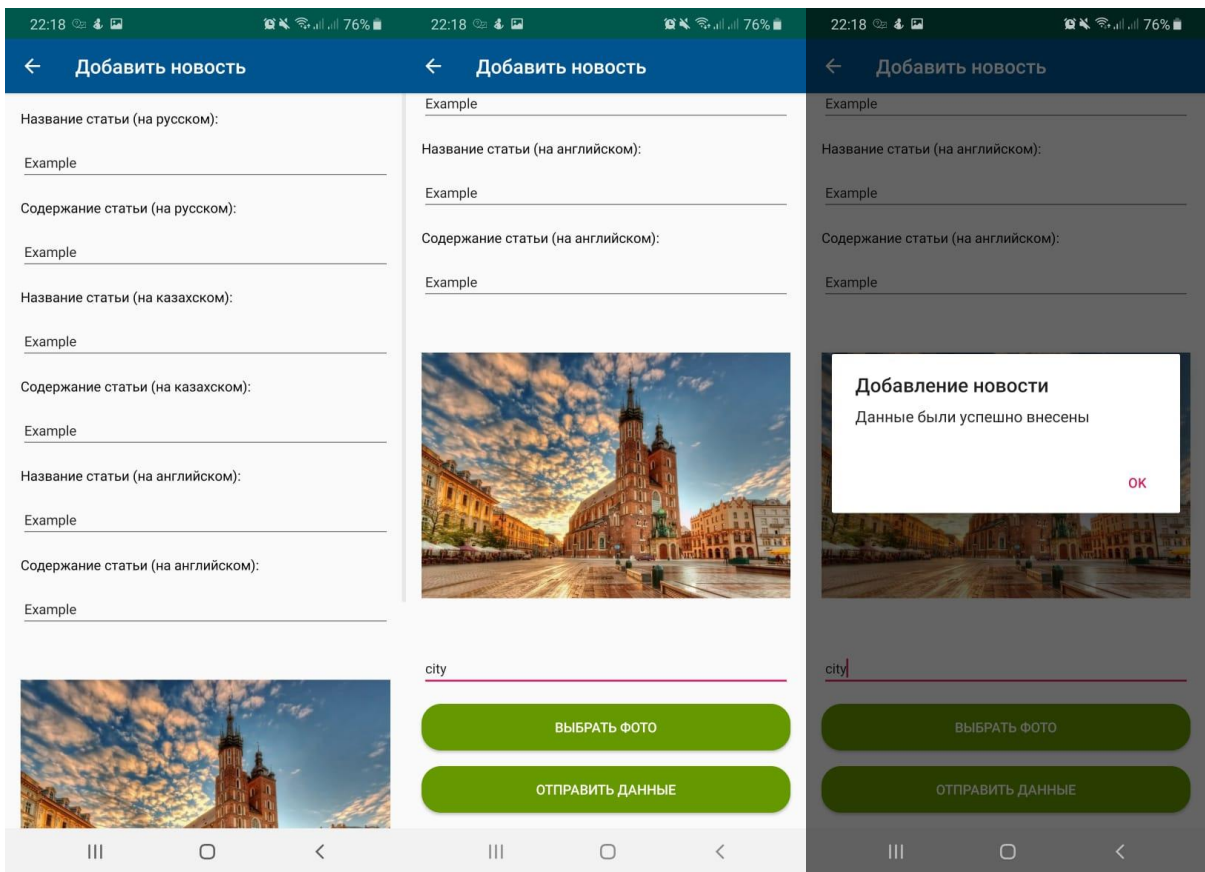


Рисунок 3.27 - Добавление новости

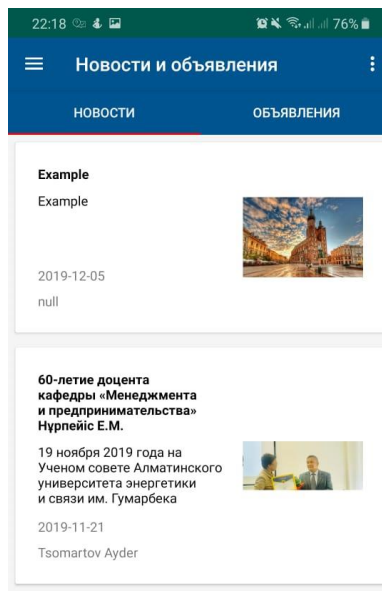


Рисунок 3.28 - Опубликованная новость

Проектирование и оптимизация информационных систем (87)									
82.2	90	95	0	0					
СР.ТЕК	ПК 1	ПК 2	РГР 1	РГР 2					
0	0	90	84.26	90					
РГР 3	РГР 4	КР	РД	Экз.					

Автоматизированные информационные системы (96)									
98	95	95	0	0					
СР.ТЕК	ПК 1	ПК 2	РГР 1	РГР 2					
0	0	0	97.4	95					
РГР 3	РГР 4	КР	РД	Экз.					

Математика III (98)									
100	70	100	0	0					
СР.ТЕК	ПК 1	ПК 2	РГР 1	РГР 2					
0	0	0	97	100					
РГР 3	РГР 4	КР	РД	Экз.					

Физическая культура (90)									

Рисунок 3.29 - Итоговые оценки

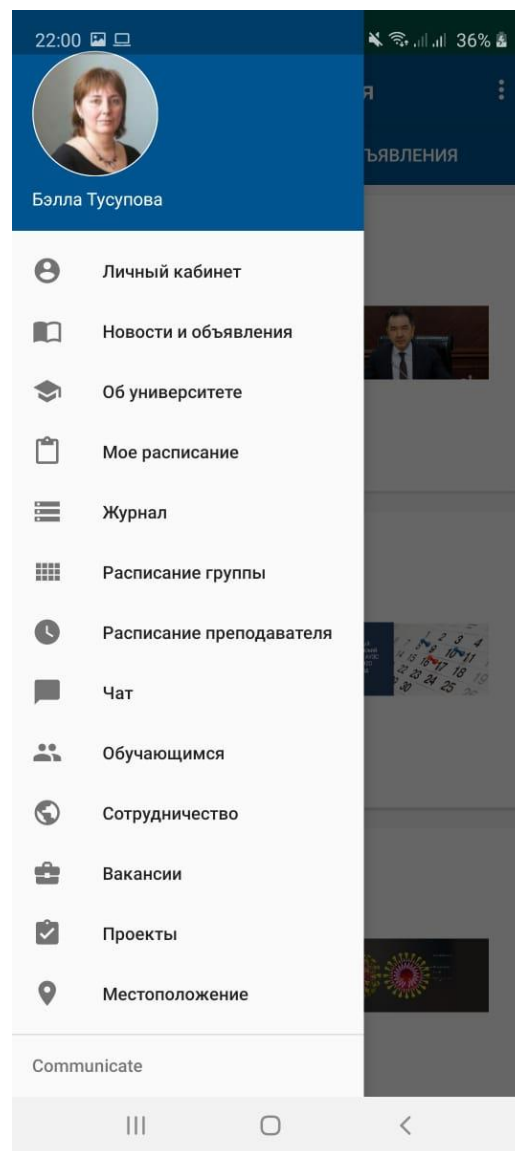


Рисунок 3.30 - Преподаватель

На рисунке 3.29 представлен список дисциплин с итоговыми оценками, при нажатии на которые студент может просмотреть информацию о текущих оценках, где представлены такие данные как дата выставления оценки, сама оценка и занятие. На рисунке 3.30 представлена панель навигации пользователя под ролью преподавателя. При нажатии на вкладку журнал открываются читаемые дисциплины с информацией о группах и занятии - представлен на рисунке 3.31. При переходе на ту или иную дисциплину открывается список студентов с итоговыми оценками, при нажатии на плавающую кнопку открываются две опции, показанные на рисунке 3.33. При нажатии на того или иного студента открываются его текущие оценки, такая же опция есть и у пользователя студент - рисунок 3.34. Далее преподаватель может выбрать одну из опций выставления оценки. При нажатии на опцию текущие оценки преподавателю открывается список студентов и поля ввода для выставления оценок, данный экран предоставлен на рисунке 3.35.

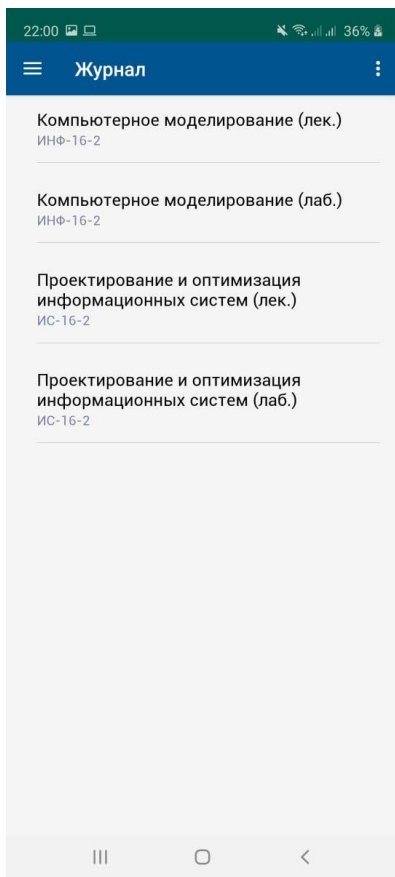


Рисунок 3.31 - Список дисциплин

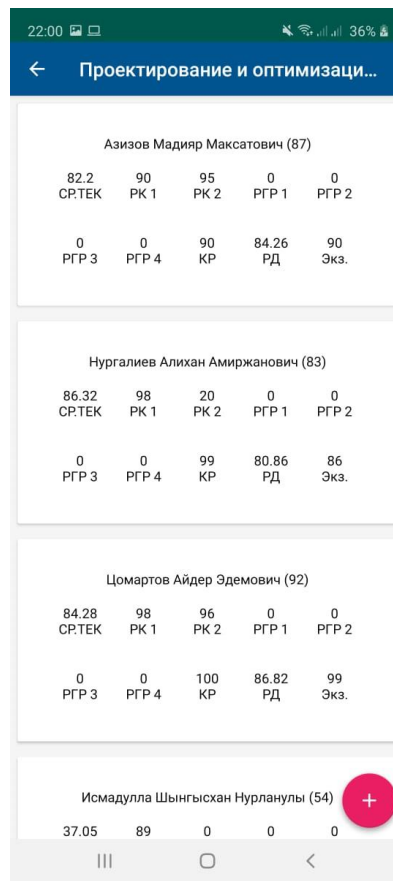


Рисунок 3.32 - Список студентов

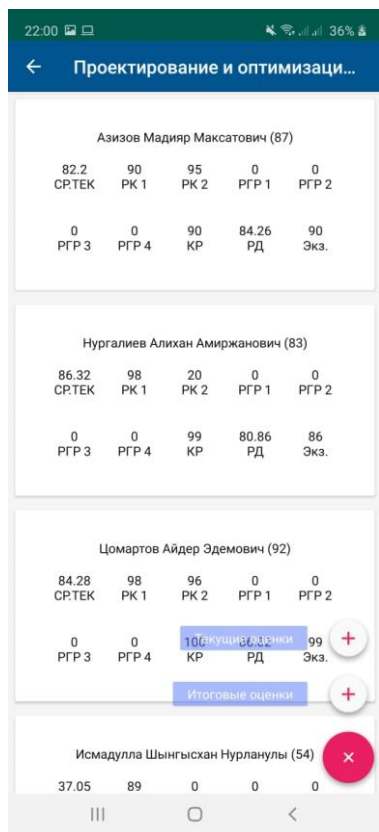


Рисунок 3.33 - Опции выставления

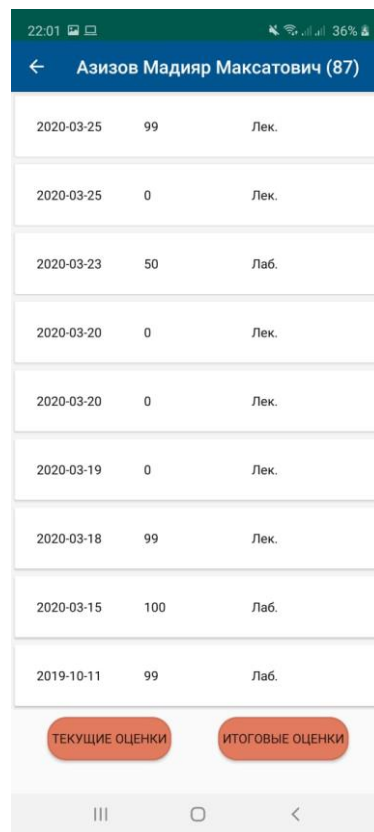


Рисунок 3.34 - Текущие оценки

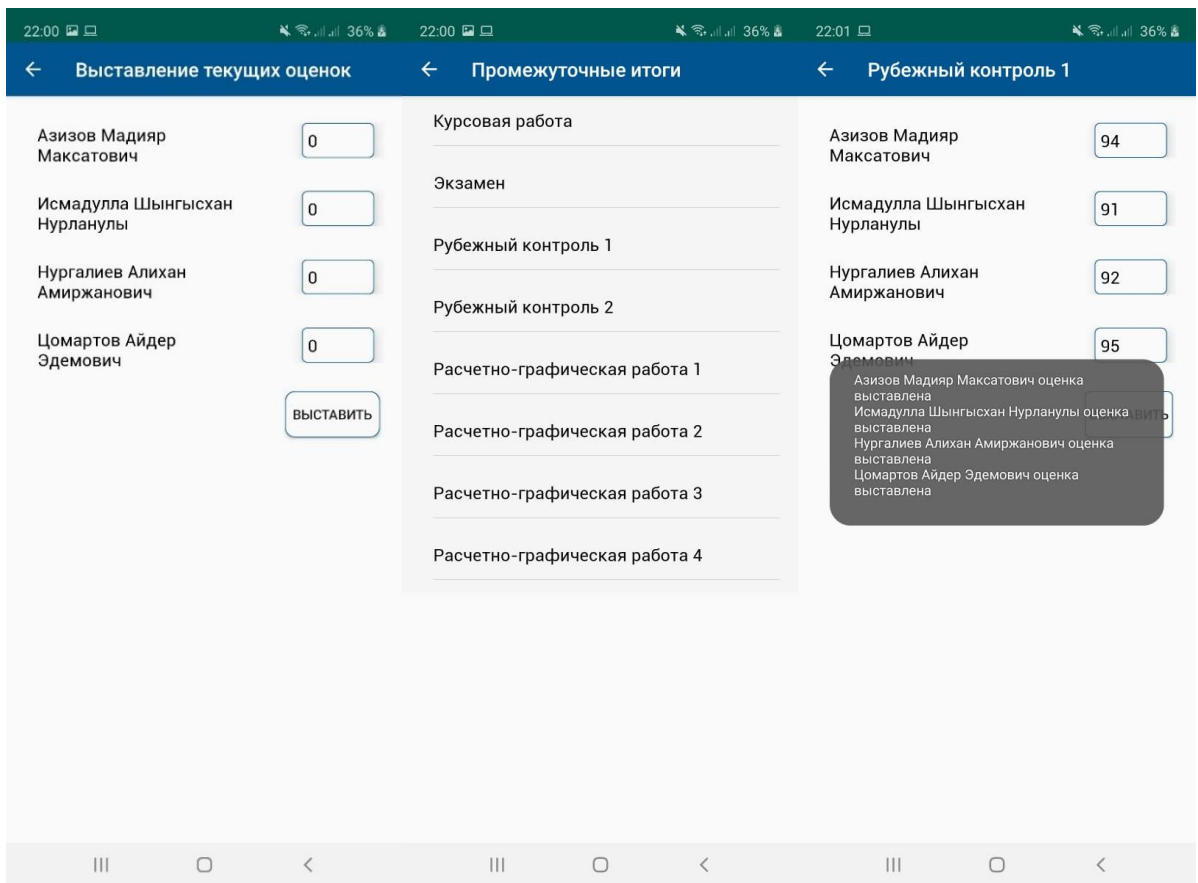


Рисунок 3.35 - Выставление оценок

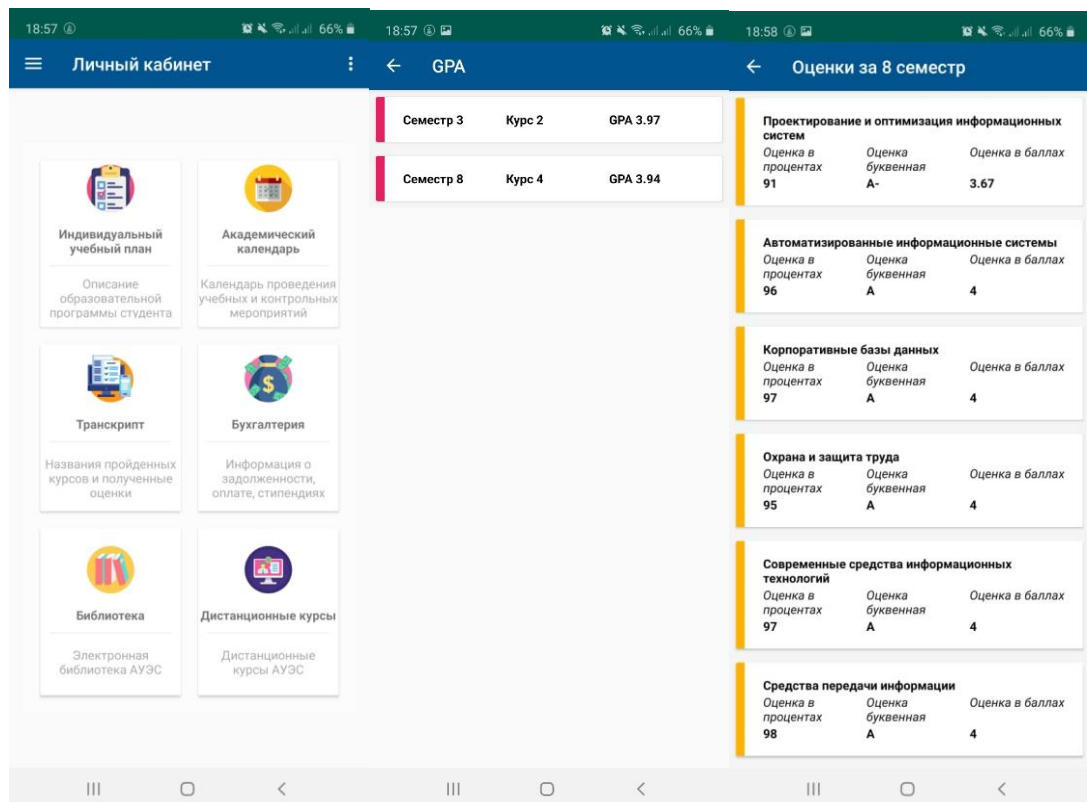


Рисунок 3.36 - Просмотр транскрипта

Также на данном рисунке представлен экран, который отображается при нажатии на опцию «Итоговые оценки». На экране отображаются промежуточные итоги при нажатии на которые также открывается список студентов и поля ввода. На рисунке отображено успешное добавление оценок.

4 Экономическое обоснование

В данном дипломном проектировании разрабатывается программный продукт «AUES Mobile», который является мобильным приложением. Конкретно разрабатываются подсистемы «Электронный журнал» и «Новости и объявления», основными функциями которых является предоставление информации об оценках студентов и выставление оценок преподавателями. Студентам доступна функция просмотра транскрипта, календарного плана, а также новостей и объявлений. Основная эффективность мобильного приложения - сокращение времени к доступу информации, обновления информации, а также работа в режиме реального времени. Основной задачей экономического раздела является расчет трудоемкости, себестоимости проекта (конкретно разрабатываемых подсистем) и определением экономической эффективности.

4.1 Определение объема и трудоемкости разработки программного обеспечения

Общий объем (V_0) программного продукта определяется исходя из количества и объема функций, реализуемых программой [23]. За уточненный объем ПО (V_y) берется количество строк исходного кода (Lines Of Code, LOC), таким образом,

$$V_y = 5000 \text{ строк}$$

Так как проект является крупным, трудоемкость рассчитывается по стадиям разработки и затем складывается.

Трудоемкость стадий определяется с учетом сложности, новизны, степени использования в разработке стандартных модулей ПО и удельного веса трудоемкости каждой стадии в общей трудоёмкости ПО [23]:

$$T_{yi} = T_n * d_{cmi} * K_c * K_m * K_n, \quad (4.1)$$

где T_{yi} - уточненная трудоемкость разработки ПО на i – й стадии;

T_n - нормативная трудоёмкость;

d_{cmi} - удельный вес трудоемкости i – й стадии разработки ПО в общей трудоемкости разработки ПО;

K_c - коэффициент, учитывающий сложность ПО, вводится на всех стадиях;

K_m - коэффициент, учитывающий степень использования стандартных модулей ПО, вводится только на стадии рабочего проекта;

K_n - коэффициент, учитывающий степень новизны ПО, вводится на всех стадиях.

Нормативная трудоёмкость T_H определяем по таблице Б.1 приложения Б [23]:

$$T_H = 170 \text{ часов.}$$

Коэффициент сложности рассчитывается по формуле [23]

$$K_c = 1 + \sum_{i=0}^n K_i, \quad (4.2)$$

где K_i - коэффициент, соответствующий степени повышения сложности ПО за счет конкретной характеристики;

n - количество характеристик.

Согласно таблице 1 и Г.1 приложения Г [23] программный продукт имеет следующие характеристики:

- интерактивный доступ ($K_c = 0,06$);
- режим работы в реальном времени и обеспечение настройки ПО на изменения структур входных и выходных данных ($K_c = 0,12$).

Следовательно,

$$K_c = 1 + \frac{0,06 + 0,12}{2} = 1,09.$$

Поправочный коэффициент, учитывающий степень использования готовых модулей, берется из таблицы 2 [23] и составляет $K_m = 0,8$ за счет использования стандартных библиотек Volley и Picasso.

Коэффициент, учитывающий степень новизны, определяется по таблице 3 [23] и равен $K_H = 1,6$, так как программный продукт не имеет аналогов и создается для новой ОС - Android. Значения коэффициентов удельных весов трудоёмкости стадий берутся из таблицы 4 [23], согласно степени новизны.

Занесем данные по этапам разработки и коэффициентам в таблицу 4.1.

Таблица 4.1 - Стадии разработки программного продукта

№	Название	d_{cmi} - удельный вес трудоёмкости	K_c - коэффициент сложности	T_H - нормативная трудоёмкость	K_H - коэффициент новизны
1	ТЗ (разработка технического задания)	0,11	1,09	170	1,6
2	ЭП (анализ, проектирование и разработка бизнес моделей)	0,09	1,09	170	1,6

Продолжение таблицы 4.1

3	ТП (проектирование интерфейса и информационного обеспечения)	0,11	1,09	170	1,6
4	РП (разработка программного обеспечения, тестирование и отладка)	0,55	1,09	170	1,6
5	ВН (ввод в действие)	0,14	1,09	170	1,6

Трудоемкость стадий ПО рассчитывается по следующим формулам [23]

$$\text{трудоемкость стадии ТЗ} \quad T_{уз} = T_{н} * d_{з} * K_{с} * K_{н} \quad (4.3)$$

$$\text{трудоемкость стадии ЭП} \quad T_{уэ} = T_{н} * d_{э} * K_{с} * K_{н} \quad (4.4)$$

$$\text{трудоемкость стадии ТП} \quad T_{ут} = T_{н} * d_{т} * K_{с} * K_{н} \quad (4.5)$$

$$\text{трудоемкость стадии РП} \quad T_{ур} = T_{н} * d_{р} * K_{с} * K_{н} * K_{т} \quad (4.6)$$

$$\text{трудоемкость стадии ВН} \quad T_{ув} = T_{н} * d_{в} * K_{с} * K_{н} \quad (4.7)$$

Тогда,

$$T_{уз} = 170 * 1,09 * 0,11 * 1,6 = 32,61 \text{ часов;}$$

$$T_{уэ} = 170 * 1,09 * 0,09 * 1,6 = 26,68 \text{ часов;}$$

$$T_{ут} = 170 * 1,09 * 0,11 * 1,6 = 32,61 \text{ часов;}$$

$$T_{ур} = 170 * 1,09 * 0,55 * 1,6 * 0,8 = 130,45 \text{ часов;}$$

$$T_{ув} = 170 * 1,09 * 0,14 * 1,6 = 41,51 \text{ часов.}$$

Общая трудоемкость будет равна

$$T_{у} = 32,61 + 26,68 + 32,61 + 130,45 + 41,51 = 263,86 \text{ часов} \approx 11 \text{ дней.}$$

4.2 Расчет затрат на разработку информационных технологий

Расчет полных затрат на разработку проектного решения осуществляется по формуле [23]:

$$C_{ni} = Z_{фот} + Z_{сзи} + Z_{сои} + M_i + P_{ci} + P_{ми} + P_{нки} + П_{зи} + P_{ни}, \quad (4.8)$$

где $Z_{фот}$ - общий фонд оплаты труда разработчиков, тенге;

$Z_{сзи}$ - отчисления по социальному налогу, тенге;

Z_{coi} - социальное отчисление, тенге;
 M_i - затраты на материалы, тенге;
 P_{ci} - затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;
 P_{mi} - затраты, связанные с эксплуатацией техники, тенге;
 P_{nki} - затраты на научные командировки, тенге;
 P_{zi} - прочие затраты, тенге;
 P_{ni} - накладные расходы, тенге.

Размер фонда оплаты труда разработчиков рассчитывается по формуле [23]:

$$Z_{\text{ФОТ}} = Z_{oi} + Z_{\varphi i}, \quad (4.9)$$

где Z_{oi} - основная заработная плата, тенге;
 $Z_{\varphi i}$ - дополнительная заработная плата, тенге.

Основная заработная плата рассчитывается по следующей формуле [23]

$$Z_{oi} = T_y * СП / (t_{\text{ср}} * 8), \quad (4.10)$$

где T_y - суммарные затраты труда, вычисляемые по формуле;
 $t_{\text{ср}}$ - среднее число дней в месяце, равно 22 дню, умножается на количество часов в рабочем дне - 8;
 СП - средняя заработная плата.

Средняя месячная зарплата в РК на 01.01.2020 г. составляет 203900 тенге, тогда основная заработная плата составит:

$$Z_{oi} = 263,86 * \frac{203\,900}{22 * 8} = 306\,000 \text{ тенге.}$$

Дополнительная заработная плата составляет 10% от основной и рассчитывается по формуле [23]:

$$Z_{\varphi i} = Z_{oi} * N_d / 100, \quad (4.11)$$

где N_d - коэффициент дополнительной заработной платы разработчиков.

Тогда,

$$Z_{\varphi i} = 306\,000 * \frac{10}{100} = 30\,600 \text{ тенге.}$$

Социальный налог составляет (9,5%) (ст. 358 п.1 НК РК) от дохода работника, и рассчитывается по формуле [23]:

$$Z_{czi} = (\text{ФОТ} - \text{ПО}) * 9,5\%, \quad (4.12)$$

где ПО - пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются:

$$\text{ПО} = \text{ФОТ} * 10\%. \quad (4.13)$$

Тогда,

$$\begin{aligned} \text{ПО} &= (306\,000 + 30\,600) * 10\% = 33\,660 \text{ тенге;} \\ \text{З}_{\text{сзи}} &= (336\,600 - 33\,660) * 9,5\% = 28\,780 \text{ тенге.} \end{aligned}$$

Социальное отчисление составляет (3,5%) (ст. 358 п.1 НК РК) от дохода работника, и рассчитывается по формуле:

$$\text{З}_{\text{сои}} = (\text{ФОТ} - \text{ПО}) * 3,5\%, \quad (4.14)$$

Тогда,

$$\text{З}_{\text{сои}} = (336\,600 - 33\,660) * 3,5\% = 10\,603 \text{ тенге.}$$

Величина затрат на материалы определяется по формуле [23]:

$$M_i = \frac{\text{З}_{\text{осн}} * \text{Н}_{\text{мз}}}{100} \%, \quad (4.15)$$

где $\text{Н}_{\text{мз}}$ - норма расхода материалов от основной заработной платы (3-5%).

Следовательно,

$$M_i = 306\,000 * \frac{4}{100} = 12\,240 \text{ тенге.}$$

Расходы по статье «Спецоборудование» (P_{ci}) включает в себя только оплату за покупку Google-аккаунта в Google Play, стоимость которого составляет 25\$, поскольку при разработке ПП были использованы стандартные бесплатные версии вспомогательных программных средств:

$$P_{ci} = 25\$ = 10\,900 \text{ тенге.}$$

Расходы по статье «Машинное время» (P_{mi}) включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машинно-часах) на 100 строк исходного кода ($\text{Н}_{\text{мв}}$) машинного времени в зависимости от характера решаемых задач и типа ПК, данное значение берется из приложения Д [23]:

$$P_{mi} = \text{Ц}_{mi} * \left(\frac{V_{oi}}{100}\right) * \text{Н}_{\text{мв}}, \quad (4.16)$$

где C_{mi} - цена одного машинно-часа (приблизительно в перерасчет на энергию равна 781,7 тенге);

V_{oi} - общий объем ПО (количество строк).

Тогда,

$$P_{mi} = 781,7 * \left(\frac{5\ 000}{100}\right) * 12 = 469\ 020 \text{ тенге.}$$

Расходы по статье «Научные командировки» (P_{nki}) на конкретное ПО определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате [23]:

$$P_{nki} = Z_{oi} * H_{рнк}/100 \quad (4.17)$$

где $H_{рнк}$ - норматив расходов на командировки в целом по организации в (%), равен 30%.

Тогда,

$$P_{nki} = 306\ 000 * \frac{30}{100} = 91\ 800 \text{ тенге.}$$

Расходы по статье «Прочие затраты» (P_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате [23]:

$$P_{zi} = Z_{oi} * H_{пз}/100 \quad (4.18)$$

где $H_{пз}$ - норматив прочих затрат в целом по организации в (%), равен 20%.

Тогда,

$$P_{zi} = 306\ 000 * \frac{20}{100} = 61\ 200 \text{ тенге.}$$

Затраты по статье «Накладные расходы» (P_{ni}), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды (P_{ni}), относятся на конкретное ПО по нормативу ($H_{нр}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации [23]:

$$P_{ni} = Z_{oi} * H_{nr} / 100 \quad (4.19)$$

где P_{ni} - накладные расходы на конкретную ПО (тыс. тенге);
 H_{nr} - норматив накладных расходов в целом по организации в (%), равен 70%.

Следовательно,

$$P_{ni} = 306\,000 * \frac{70}{100} = 204\,200 \text{ тенге.}$$

Результаты выполненных расчетов запишем в таблицу 4.2. На рисунке 4.1 предоставлена диаграмма затрат.

Таблица 4.2 - Затраты на разработку информационных технологий

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{фот}$	336 600	27,47
Социальный налог	$Z_{сзи}$	28 780	2,35
Социальное отчисление	$Z_{сои}$	10 603	0,86
Материалы	M_i	12 240	0,99
Спецоборудование	P_{ci}	10 900	0,89
Машинное время	P_{mi}	469 020	38,28
Научные командировки	$P_{нки}$	91 800	7,49
Прочие затраты	$P_{зи}$	61 200	4,99
Накладные расходы	P_{ni}	204 200	16,68
Итого:	C_0	1 225 343	100

Цена разрабатываемого программного обеспечения рассчитывается по формуле:

$$C_p = C_0 + \text{НДС}, \quad (4.20)$$

где C_0 - первоначальная цена.

Первоначальная цена рассчитывается через рентабельность разрабатываемого проекта. Рентабельность равна 50%, воспользуемся следующей формулой:

$$C_{п} = C \times \left(1 + \frac{P}{100}\right), \quad (4.21)$$

где P - рентабельность (50%).



Рисунок 4.1 - Диаграмма затрат на информационные технологии

Подставив полученные значения, первоначальная цена согласно формуле составит:

$$C_0 = 1\,225\,343 * \left(1 + \frac{50}{100}\right) = 1\,838\,015 \text{ тенге.}$$

Цена готовой продукции рассчитывается по формуле:

$$C_p = C_0 + \text{НДС}, \quad (4.22)$$

где НДС - 12% от начальной цены готового продукта. Рассчитаем НДС используя формулу (4.22)

$$\text{НДС} = 1\,838\,015 * 0,12 = 220\,562 \text{ тенге.}$$

Следовательно, конечная итоговая цена программного продукта составит:

$$C_p = 1\,838\,015 + 220\,562 = 2\,058\,577 \text{ тенге.}$$

4.3 Расчет динамических показателей эффективности капитальных вложений

То, какую прибыль принесет мобильное приложение департаменту информационных технологий прямо зависит от количества поступающих в университет, так как определенная часть от оплаты идет на обновление материально-технической базы (в том числе поддержка и эксплуатация мобильного приложения). Допустим, за содержание мобильного приложения в договоре указана сумма, равная 0,1% от их оплаты за год. Средняя оплата за год студента составляет 700 000 тенге. Итого, косвенная оплата за содержание

мобильного приложения будет составлять 700 тенге. Примерное количество обучающихся в университете каждый год примерно равна 6000 - 7000 студентов. За последние годы имеет место тенденция увеличения количества поступающих. Из них около 40% обучаются на платной основе. Тогда построим таблицу исходных данных.

Таблица 4.3 - Исходные данные для оценки проекта

Инвестиционные затраты, тг	Значения P_t , тг.		
	1 год	2 год	3 год
$T = 0$			
2 058 577	1 680 000	1 736 000	1 820 000

Чистый дисконтированный доход (NPV)

Коэффициент дисконтирования рассчитывается по формуле

$$\alpha_t = \frac{1}{(1+r)^t} \quad (4.23)$$

где α_t - коэффициент дисконтирования;

r - норма дисконта;

t - номер шага расчета.

Рассчитывается коэффициент дисконтирования для каждого года.

$$\text{за 1 год: } \alpha_1 = \frac{1}{(1+0,25)^1} = 0,8;$$

$$\text{за 2 год: } \alpha_2 = \frac{1}{(1+0,25)^2} = 0,64;$$

$$\text{за 3 год: } \alpha_3 = \frac{1}{(1+0,25)^3} = 0,512.$$

Чистая приведенная стоимость (Net Present Value - NPV) относится к группе методов дисконтирования денежных потоков или DCF-методов.

Пусть I_0 - сумма первоначальных затрат, т.е. сумма инвестиций на начало проекта;

PV - современная стоимость денежного потока на протяжении экономической жизни проекта.

Общая накопленная величина дисконтированных доходов рассчитывается по формуле

$$PV = \sum_{t=1}^n \frac{P_t}{(1+r)^t} = \sum_{t=1}^n P_t * \alpha_t \quad (4.24)$$

где r - норма дисконта;

n - число периодов реализации проекта;

P_t - чистый поток платежей в периоде t .

Рассчитывается величину дисконтированных доходов по каждому году.

Тогда,

$$PV_1 = 1\,680\,000 * 0,8 = 1\,344\,000 \text{ тенге};$$

$$PV_2 = 1\,736\,000 * 0,64 = 1\,111\,040 \text{ тенге};$$

$$PV_3 = 1\,820\,000 * 0,512 = 931\,840 \text{ тенге}.$$

Общая накопленная величина дисконтированных доходов будет равна

$$PV = 1\,344\,000 + 1\,111\,040 + 931\,840 = 3\,386\,880 \text{ тенге}.$$

Текущая стоимость затрат (I_0) сравнивается с текущей стоимостью доходов (PV). Разность между ними составляет чистую текущую стоимость проекта (NPV):

$$NPV = \sum_{t=1}^n \frac{CF_t}{(+r)^t} - I_0$$

(4.25)

$$NPV = 3\,386\,880 - 2\,058\,577 = 1\,328\,303 > 0$$

Индекс прибыльности проекта (PI)

Индекс рентабельности (доходности) инвестиций (PI). Индекс рентабельности представляет собой отношение суммы приведенных эффектов к величине инвестиционных затрат. Индекс рентабельности (PI) рассчитывается по формуле

$$PI = \sum_k \frac{P_k}{(1+r)^k} / I \quad (4.26)$$

Тогда,

$$PI = \frac{3\,386\,880}{2\,058\,577} = 1,64 > 1.$$

Дисконтированный срок окупаемости (DPP)

Срок окупаемости инвестиций - один из самых простых и широко распространен в мировой практике, не предполагает временной упорядоченности денежных поступлений [24].

Общая формула расчета показателя DPP имеет вид

$$DPP = t, \text{ при котором } P_t > I$$
$$DPP = t + \frac{\text{ост. долга инвестору}}{\text{денежный поток}(t+1)} \quad (4.27)$$

где P_t - чистый денежный поток доходов

$$DPP = 1 + \frac{714\,577}{1\,111\,040} = 1,64 \text{ года.}$$

График окупаемости проекта

Заполняется таблицу окупаемости проекта

Таблица 4.4 - Окупаемость проекта

Год	Доход, тенге	Расход, тенге	Прибыль/Чистый доход, тенге	Чистый доход нарастающим итогом, тенге
1	1 344 000	2 058 577	-714 577	-714 577
2	1 111 040	0	1 111 040	396 463
3	931 840	0	931 840	1 328 303

Строится график зависимости чистого дохода нарастающим итогом от года, данный график представлен на рисунке 4.2.



Рисунок 4.2 - График окупаемости проекта

Срок окупаемости инвестиционного проекта - период времени, за который чистый доход нарастающим доходом станет равным нулю. По графику визуально, достаточно точно можно определить данный период - 1,6 года.

Заключение о принятии решения по проекту

Рассчитанная чистая современная стоимость потока платежей имеет положительный знак ($NPV > 0$), это означает, что в течение своей экономической жизни проект возместит первоначальные затраты 2 058 577 тенге обеспечит получение прибыли, а также ее резерв, равный 1 328 303 тенге. Также проект полностью окупится за 1,64 года. Исходя из анализа данный проект является прибыльным.

5 Охрана труда и БЖД

5.1 Анализ потенциально опасных и вредных факторов, воздействующих на обслуживающий персонал в процессе труда

Чтобы осуществить первый шаг в любой оценке риска на рабочем месте, необходимо определить опасности на рабочем месте. На каждом рабочем месте имеются опасности, и, хотя для проведения официальной оценки рисков назначается специалист, все равно каждый должен помнить об опасностях на рабочем месте и минимизировать возникновение риска причинения вреда.

Не все опасности очевидны, и они будут уникальными для конкретного рабочего места. Это может затруднить немедленное выявление и защиту сотрудников от них. Поэтому необходимо понять различные категории опасностей и места их возникновения.

Для работающих в офисе характерны разнообразные опасности, которые можно классифицировать так [25]:

- биологический. Биологические опасности включают вирусы, бактерии, насекомых, животных и т. д., которые могут оказывать вредное воздействие на здоровье. В офисе присутствуют следующие опасности: плесень, вредные растения, пыль и паразиты. Для предотвращения данных опасностей в офисе необходимо составить график влажных уборок помимо ежедневного мытья полов, так как пыль является не только внешним раздражителем, а также может оседать внутри организма, установить специальные защитные механизмы на окна, чтобы при проветривании предотвратить попадания различных паразитов и пыли. Так как в офисе имеется кухня, то имеется риск возникновения таких паразитов как тараканы, в следствие этого необходимо несколько раз в месяц вызывать специалистов, которые занимаются дезинсекцией и дезинфекцией;

- химические вещества. Химическая опасность - это опасные вещества, которые могут причинить вред. Эти опасности могут привести как к здоровью, так и к физическим воздействиям, таким как раздражение кожи, раздражение дыхательной системы, слепота, коррозия и взрывы. В офисе основными химическими веществами являются моющие средства, которые хранятся в отдельном помещении - подсобке, доступ к которой имеет только клининг-работник, все средства находятся в специальных емкостях с надписями для предотвращения перепутывания средств;

- физическая. Физические опасности - это факторы окружающей среды, которые могут нанести вред работнику без необходимости прикасаться к ним, включая высоту, шум, радиацию и давление. Для нормальной работы в офисе был проведен расчет аспирационных систем для установки кондиционеров;

- угрозы безопасности. Это опасности, которые создают небезопасные условия труда. В офисе имеются такие опасности, как оголенные провода и

поврежденный ковер, которые могут привести к опасности споткнуться. Они иногда включаются в категорию физических опасностей;

- эргономичный. Эргономические опасности являются результатом физических факторов, которые могут привести к травмам опорно-двигательного аппарата. В расчетной части приведен анализ рабочего места, который выявил плохую настройку рабочей станции в офисе, в связи с чем были даны рекомендации и поручения по реконструкции рабочей области;

- психосоциальная. К психосоциальным опасностям относятся те, которые могут оказать неблагоприятное воздействие на психическое здоровье или благополучие работника. В офисе не присутствуют такие опасности, как виктимизация и насилие на рабочем месте, однако в связи с тем, что многие работники откладывают данные им задачи к концу дедлайна (в частности программисты), то имеется стресс, который оказывает негативное влияние на организм сотрудника. В результате этого необходимо мотивировать и проводить еженедельные проверки на рабочих местах и предотвращать застои на рабочем месте.

Также присутствуют такие угрозы безопасности как:

- рыхлые и потертые ковры и коврики, разливы, потертый линолеум. Все это может привести к скольжению или падению на рабочем месте, а также к физическим и психическим последствиям для пострадавшего работника. Такие угрозы появляются периодически со временем изношенности напольных материалов. Для минимизации возникновения несчастных случаев ковры реконструируются либо заменяются на новые, так же происходит и с линолеумом;

- изношенные и неисправные шнуры, проводка или кабели. Это может привести к поражению электрическим током, ожогам и пожару. Воздействие живого электричества также может привести к падению с высоты.

Физические опасности - это факторы окружающей среды, которые могут нанести вред работнику, не касаясь его.

В офисе имеется риск следующих физических опасностей:

- электричество. Воздействие электрических частей под напряжением может привести к серьезным травмам и смертельным исходам, включая поражение электрическим током, ожоги, взрывы и падения с высоты. Данная ситуация имеет место быть, когда работник офиса совершает обеденный перерыв на рабочем месте, он может разлить свой напиток на штекеры, что приведет к неисправности техники либо поражению электрическим током;

- пожары. Каждое рабочее место находится под угрозой пожара. В данном случае все сотрудники проходят учения по пожарной безопасности и обучаются владению огнетушителем.

Некоторые могут думать, что работа в удобном, контролируемом климате офиса безопасна и безвредна, но вокруг имеются много рисков для безопасности и здоровья. Бюро статистики труда сообщает о десятках тысяч травм или проблем со здоровьем, связанных с работой, от которых офисные работники страдают каждый год.

Поскальзывания являются наиболее распространенными несчастными случаями в офисе, на которые приходится наибольшее количество травм. Другие офисные опасности включают растяжение, напряжение, плохую эргономику рабочей станции, проблемы с качеством воздуха в помещении, недостаточное или чрезмерное освещение, шум, опасность поражения электрическим током и случайные акты насилия. Знание этих опасностей - это первый шаг к их устранению и снижению вероятности получения травм. Сотрудники отдела кадров внедряют процессы для выявления опасностей и устранения проблем, в том числе вводят обходные пути безопасности, создают официальную систему отчетности для небезопасных условий и проводят учебные занятия по устранению угроз безопасности.

Наиболее распространенные опасности, которые присутствуют в офисе изложены ниже [26].

Шликеры и падения

Некоторые сотрудники - виновники поскальзывания, спотыкания и падения включают оставленные без присмотра мокрые полы, открытые шнуры, неустойчивые рабочие поверхности, неровные полы, рыхлые коврики и загроможденные участки.

Неблагоприятные погодные условия, такие как дождь, снег и лед, создают опасность скольжения на открытом воздухе на наружных ступенях, пандусах, пешеходных дорожках, участках въезда и выезда, а также на парковках, а также в помещениях. Таяние льда и скользкие полы могут значительно снизить риск скольжения, спотыкания и падения в зимние месяцы.

Для того чтобы снизить риски опасности размещаются знаки, указывающие на опасности в местах, которые очищаются или недавно были очищены, а также в местах, подверженных скоплению воды и влажным поверхностям.

Электрические и телефонные шнуры надежно закреплены и не натягиваются на проход.

Эргономические травмы

Офисные работники проводят много часов в день, сидя за столом, работая за компьютером, что приводит к эргономическим нагрузкам и другим травмам, связанным с осанкой и повторяющимися движениями.

В офисе имеются разнообразные регулируемые стулья, столы, клавиатуры для удовлетворения самых разных стилей работы. Сотрудники информируются, как настраивать и использовать оборудование для наилучшего соответствия рабочей станции.

Кроме того, сотрудники офиса придерживаются следующих рекомендаций eTool, которое предлагается управлением по безопасности и гигиене труда (OSHA):

- расположить кресло, клавиатуру и монитор на одной линии с телом;
- сохранять расслабленную нейтральную позу;
- сидеть прямо, регулируя стул, чтобы обеспечить твердую поддержку

спины;

- пусть ваши руки свободно висят на плечах;
- во время набора текста держать локти под углом 90 градусов;
- использовать регулируемый лоток для клавиатуры, чтобы расположить клавиатуру и мышь на удобной высоте (обычно ниже поверхности стола). Поместить мышь рядом с клавиатурой и держать ее как можно ближе к телу, чтобы избежать ее попадания;
- регулировать высоту стула так, чтобы ноги прочно стояли на земле.

OSHA советует обращать внимание на любую боль, усталость, онемение или слабость, поскольку это может быть признаком проблемы с эргономикой и началом более серьезной проблемы.

Напряжение глаз

Поскольку сотрудники в офисе значительную часть рабочего дня проводят за компьютером имеется риск вызывания напряжения глаз. Глаза могут стать сухими и раздраженными, и у рабочих могут возникнуть проблемы с фокусировкой. В качестве превентивных мер руководством офиса периодически проверяется уровень освещенности, который должен соответствовать рабочей задаче, при недостаточности света на рабочих местах имеется настольные лампы.

Для уменьшения чрезмерного яркого света имеются жалюзи на окнах, которые затемняют верхний свет. Правильное расположение мониторов чуть ниже уровня глаз, минимизация бликов на экране и увеличение размера шрифта компьютера - всему этому сотрудники стараются придерживаться для уменьшения напряжения глаз.

Чтобы уменьшить утомляемость глаз и усталость сотрудники делают 10-минутный перерыв на каждый час, который проводится, глядя на экран компьютера, отдыхая и фокусируясь на вещах на разных расстояниях.

Пожарная безопасность

В офисном помещении шнуры питания регулярно проверяются на предмет износа и заменяются, если они изношены или имеют оголенный провод.

Шнуры не перегружают розетки, так как наиболее распространенными причинами пожаров, вызванных удлинителями, являются неправильное использование и перегрузка. Кроме того, на рабочих местах используются только удлинители, одобренные сертифицирующей лабораторией.

Также используются обогреватели, одобренные для коммерческого использования, в которых имеется переключатель, автоматически отключающих их. Обогреватели размещены на удаленном расстоянии от горючих материалов и бумаг.

Объекты не размещаются ближе, чем на 46 см ниже головок пожаротушения, чтобы обеспечить полный охват. Маршруты аварийного выхода никогда не должны быть заблокированы.

Каждый новый сотрудник обучается тому, что делать в случае возникновения пожара. Поскольку в соответствии с Законом о безопасности и гиги-

ене труда, когда работодатель предоставляет переносные огнетушители для использования работником, работодатель должен также обучать работников общим принципам использования огнетушителя. Работодатели имеют возможность требовать от всех сотрудников немедленной эвакуации помещений.

Качество воздуха в помещении

Циркулирование плохого качества воздуха в помещениях способствует росту профессиональной астмы и других респираторных расстройств, химической чувствительности и аллергии. В офисе причинами плохого воздуха являются неадекватные системы вентиляции; переполненность офиса; наличие чистящих химикатов и пестицидов; повреждение водой и рост плесени; конструкция кабины, которая блокирует поток воздуха в рабочие зоны; слишком большая и слишком низкая влажность; плохое ведение домашнего хозяйства, что приводит к грязной рабочей среде.

Качество воздуха в офисе периодически улучшается путем надлежащего технического обслуживания, очистки и фильтрации системы вентиляции, отопления и кондиционирования воздуха. Это помогает уменьшить респираторные раздражители, инфекции и болезни. В связи с увеличением работающих сотрудников в офисе, был произведен перерасчет аспирационных систем.

Предотвращение накопления пыли, пыльцы, грязи и других отложений на всех поверхностях, особенно на ковровых покрытиях, также уменьшает раздражение дыхательных путей, инфекции и болезни. Превентивные меры были описаны выше.

Каждый сотрудник содержит чистоту и порядок на рабочем месте, что также предотвращает распространение болезней и заболеваний. Кроме того, туалеты, комнаты отдыха, обеденные зоны и холодильники регулярно дезинфицируются, работники выбрасывают пищу до ее порчи.

5.2 Расчетная часть

5.2.1 Расчет аспирационных систем

Исходные данные

Необходимо произвести расчет аспирационных систем в ТОО «GLAS-MAN Trade», офис которого находится в г. Алматы и выбрать кондиционер(-ы). Помещение имеет следующие параметры 20 x 4 x 3,2. Данные по оборудованию: 12 шт., мощность каждого из которых равна 0,5 кВт/ч; КПД равно 0,75. Мощность люминесцентных ламп 60 Вт/м². Число сотрудников, из них: мужчины 6, женщины 6. Окна: количество 3/3. Площадь 1 окна равна 4 м², расположение окон С/Ю. Вид окон - жалюзи, деревянный переплет, загрязнение незначительное. Работы осуществляется в положении сидя.

Расчет тепловых нагрузок в помещении

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).

Наружные тепловые нагрузки

Данные нагрузки представлены следующими составляющими:

- теплопоступления или теплопотери в результате разности температур снаружи и внутри здания через стены, потолки, полы, окна и двери.

- разность температур снаружи здания и внутри него летом является положительной, в результате чего имеет место приток тепла снаружи во внутрь помещения; и наоборот - зимой эта разность отрицательна и направление потока тепла меняется;

- теплопоступления от солнечного излучения через застекленные площади; данная нагрузка проявляется в форме ощущаемого тепла;

- теплопоступления от инфильтрации.

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными.

Теплопоступления и теплопотери в результате разности температур определяются по формуле:

$$Q_{огр} = V_{пом} * X_0 * (t_{Нрасч} - t_{Врасч}), Дж$$

(5.1)

где $V_{пом}$ - объем помещения, м³:

$$V_{пом} = 20 * 4 * 3,2 = 256 \text{ м}^3;$$

X_0 - удельная тепловая характеристика, Вт/м³°С:

$$X_0 = 0,39 \text{ Вт/м}^3 \text{°С}$$

$t_{Нрасч}$ - наружная температура. Для холодного периода - средняя температура самого холодного месяца в 13 часов, для теплого периода - средней температуре самого жаркого месяца в 13 часов. В Алматы температура самого холодного месяца (января) равна $-4,7$ °С, самого тёплого месяца (июля) составляет $+23,8$ °С [27].

$t_{Врасч}$ - внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года

$$t_{Нрасч} = 23,8 \text{ °С}$$

$$t_{Врасч} = 22 \text{ °С}$$

$$Q_{\text{огр}} = 256 * 0,39 * (23,8 - 22) = 179,7 \text{ Дж}$$

Для холодного времени года

$$t_{\text{Нрасч}} = -4,7 \text{ }^{\circ}\text{C}$$

$$t_{\text{Врасч}} = 24 \text{ }^{\circ}\text{C}$$

$$Q_{\text{огр}} = 256 * 0,39 * |-4,7 - 24| = 2865,4 \text{ Дж}$$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле:

$$Q_p = (q^I F_0^I + q^{II} F_0^{II}) * \beta_{\text{с.з.}}, \text{ Дж}$$

(5.2)

где q^I, q^{II} - тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м²;

F_0^I, F_0^{II} - площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м²;

$\beta_{\text{с.з.}}$ - коэффициент теплопропускания. По приложению 8 [28]:

$$\beta_{\text{с.з.}} = 0,15.$$

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F_0^I = F_0; F_0^{II} = 0$:

$$Q_p = q^I F_0 * \beta_{\text{с.з.}} = (q_{\text{вп}} + q_{\text{вр}}) * K_1^c * K_2 * \beta_{\text{с.з.}} * n * S_0, \text{ Дж}$$

(5.3)

$q_{\text{вп}}; q_{\text{вр}}$ - тепловые потоки от прямой и рассеянной радиации, Вт/м². По таблице 8 [29] для широты в 46⁰с. ш. до полудня в 11-12 ч. при расположении Ю:

$$q_{\text{вп}} = 314 \text{ Вт/м}^2; q_{\text{вр}} = 114 \text{ Вт/м}^2;$$

$F_0 = nS_0 = 3 * 4 = 12 \text{ м}^2$ - площадь светового проема (n - число окон; S_0 - площадь 1 окна);

K_1 - коэффициент затемнения остекления переплетами (K_1^c - для облученных проемов). По таблице 3 [28]:

$$K_1^c = 0,75;$$

K_2 - коэффициент загрязнения остекления. По таблице 4 [28]:

$$K_2 = 0,95.$$

Тогда

$$Q_p = (314 + 114) * 0,75 * 0,95 * 0,15 * 12 = 548,9 \text{ Дж}$$

По таблице 8 [29] для широты в 46° с.ш. до полудня в 11-12 ч. при расположении С:

$$q_{вп} = 0 \text{ Вт/м}^2; q_{вр} = 80 \text{ Вт/м}^2;$$

$F_0 = nS_0 = 3 * 4 = 12 \text{ м}^2$ - площадь светового проема (n - число окон;
 S_0 - площадь 1 окна);

Тогда:

$$Q_p = (0 + 80) * 0,75 * 0,95 * 0,15 * 12 = 102,6 \text{ Дж}$$

Тогда общее теплопоступление солнечного излучения со всех окон равно:

$$Q_p = 548,9 + 102,6 = 651,5 \text{ Дж}$$

Внутренние тепловые нагрузки

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами фотокопировальными машинами пр.

В производственных и технологических помещениях различного назначения дополнительными источниками тепловыделений могут быть: нагретое производственное оборудование, горячие материалы, в том числе жидкости и различного рода полуфабрикаты, продукты сгорания и химических реакций.

Теплопоступления от людей зависит от интенсивности выполняемой работы и параметров окружающего воздуха. Тепло, выделяемое человеком, складывается из ощутимого (явного), то есть передаваемого в воздух помещения путем конвекции и лучеиспусканий, и скрытого тепла, затрачиваемого на испарение влаги с поверхности кожи и из легких.

Летом при 22°C один мужчина выделяет явного тепла 61 Вт, а общего - 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого

мужчины. Тогда выделение явного тепла в помещении составит:

$$Q_{\text{л}}^{\text{я}} = 61 * 6 + 61 * 6 * 0,85 = 677,1 \text{ Дж}$$

А выделение общего тепла:

$$Q = 102 * 6 + 102 * 6 * 0,85 = 1132,2 \text{ Дж}$$

Зимой при 24°C один мужчина выделяет явного тепла 85 Вт, а общего - 105 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины.

Тогда выделение явного тепла в помещении составит:

$$Q_{\text{л}}^{\text{я}} = 85 * 6 + 85 * 6 * 0,85 = 943,5 \text{ Дж}$$

А выделение общего тепла:

$$Q = 105 * 6 + 105 * 6 * 0,85 = 1165,5 \text{ Дж}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле:

$$Q_{\text{осв}} = \eta * N_{\text{осв}} * F_{\text{пол}}, \text{ Дж}$$

(5.4)

где η - коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп $\eta = 0.5 - 0.6$);

$N_{\text{осв}}$ - установленная мощность ламп ($N = 60 \text{ Вт/м}^2$);

$F_{\text{пол}}$ - площадь пола: $F_{\text{пол}} = 20 * 4 = 80 \text{ м}^2$

Тогда:

$$Q_{\text{осв}} = 0,5 * 60 * 80 = 2400 \text{ Дж}$$

Тепло, выделяемое производственным оборудованием, определяется по формуле:

$$Q_{\text{об}} = N_{\text{уст}} * K, \text{ Дж}$$

(5.5)

$$Q_{\text{об}} = 0,5 * 0,75 * 12 = 4,5 \text{ кДж}$$

Теплопритоки, возникающие за счет находящейся оргтехники, - это 30% мощности оборудования:

$$Q_{\text{орг}} = 0,3 * 4,5 = 1350 \text{ Дж.}$$

Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс теплоступлений в помещении:

$$Q_{\text{изб}} = Q_{\text{огр}} + Q_{\text{р}} + Q_{\text{л}}^{\text{я}} + Q_{\text{осв}} + Q_{\text{об}} + Q_{\text{орг}}, \text{ Дж} \quad (5.6)$$

Лето:

$$Q_{\text{изб}} = 179,7 + 651,5 + 677,1 + 2400 + 4500 + 1350 = 9758,3 \text{ Дж}$$

Зима:

$$Q_{\text{изб}} = 2865,4 + 651,5 + 943,5 + 2400 + 4500 + 1350 = 12709,4 \text{ Дж}$$

Так как тепловой баланс для зимы больше летнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле:

$$Q_{\text{н}} = Q_{\text{изб.зима}} * 860 / V_{\text{пом}} = 12709,4 * 860 / 256 = 42,7 \text{ ккал/м}^3.$$

При $Q_{\text{н}} > 20 \text{ ккал/м}^3$, $\Delta t = 8 \text{ }^\circ\text{C}$,

Определение количества воздуха, необходимое для поступления в помещение:

$$L = Q_{\text{изб}} * 860 / (C * \Delta t * \gamma) = 12709,4 * 860 / (0,24 * 1000 * 8 * 1,206) = 4720,4 \text{ м}^3/\text{час}$$

где $C = 0,24 \text{ ккал/(кг}^\circ\text{C)}$ - теплоемкость воздуха,

$\gamma = 1,206 \text{ кг/м}^3$ - удельная масса приточного воздуха.

Определение кратности воздухообмена:

$$n = L / V_{\text{пом}} = 4720,4 / 256 = 18,44 \text{ час}^{-1}.$$

Для создания комфортных условий труда необходимо установить в офисе кондиционеры. Исходя из полученных данных, выберем 2 кондиционера сплит-системы настенного типа. Основные характеристики установленных кондиционеров приведены в таблице 5.1.



Рисунок 5.1 - Кондиционер LG UB60/UU60

Таблица 5.1 - Основные характеристики

Разновидность	комплект
Площадь помещения	143 м ²
Напряжение	220 В
Мощность охлаждения	14.3 кВт

Продолжение таблицы 5.1

Мощность обогрева	17.5 кВт
Потребляемая мощность при обогреве	5 кВт
Режим вентиляции	
Расход воздуха	3000 м ³ /ч
Макс. уровень шума (внутр. блок)	46 дБ
Макс. уровень шума (внеш. блок)	55 дБ
Хладагент	R410a
Ширина (внутр. блок)	1230 мм
Глубина (внутр. блок)	590 мм
Высота (внутр. блок)	380 мм
Вес (внутр. блок)	62 кг
Ширина (внеш. блок)	950 мм
Глубина (внеш. блок)	330 мм
Высота (внеш. блок)	1380 мм
Вес (внеш. блок)	105 кг
Страна-производитель	Ю. Корея

Во внешних блоках находятся компрессор, конденсатор и вентилятор. Внешние блоки можно установить на стене здания, на крыше или на чердаке, в подсобном помещении или на балконе, то есть в таком месте, где горячий конденсатор может продуваться атмосферным воздухом более низкой температуры.

Внутренние блоки устанавливаются непосредственно в кондиционируемом помещении и предназначен для охлаждения или нагревания воздуха, фильтрации его и создания необходимой подвижности воздуха в помещении. Внутренние блоки поддерживают заданную температуру, обеспечивают равномерное распределение воздуха в помещении и работают практически бесшумно.

Управление работой настенных кондиционеров производится с дистанционного пульта, который позволяет задать режим работы кондиционера: обогрев, охлаждение, осушку, вентиляцию, ночной режим; задать требуемую температуру, которую должен поддерживать автоматически; выбрать режим работы вентилятора: настроить таймер, который включит или выключит кондиционер в заданное время;

автоматически регулировать положение направляющих шторок и изменить таким образом направление воздушного потока.

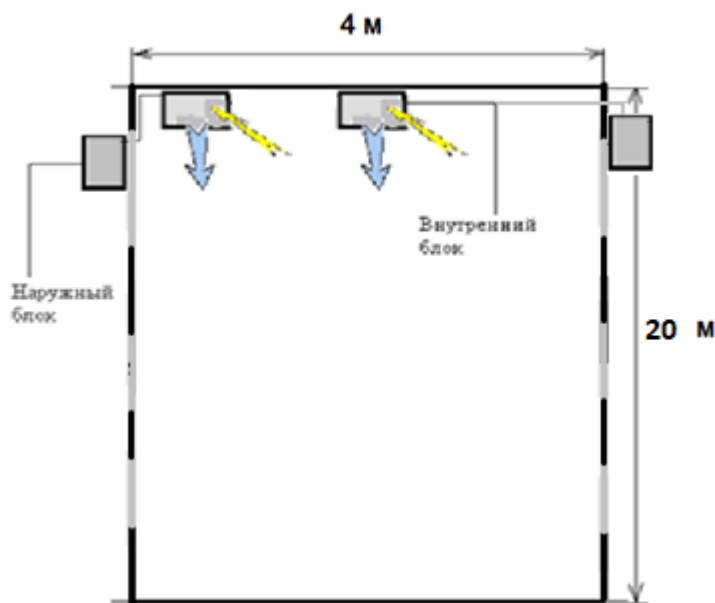


Рисунок 5.2 - Схема расположения кондиционеров

5.2.2 Анализ рабочего места и его корректировка

В современной жизни компьютер широко применяется в жизни человека: и дома, и в офисе, и в магазине, и в производстве, и даже в бытовой технике. Компьютеры прочно вошли в повседневную жизнь людей и их использование постоянно увеличивается.

В офисах компьютеры в основном используются как вспомогательные средства обработки информации, и такое введение компьютерных технологий принципиально изменило характер труда офисных работников и требования к организации и охране труда.

Несоблюдение требований безопасности приводит к тому, что при работе за компьютером сотрудник может ощущать дискомфорт: возникают головные боли и резь в глазах, появляются усталость и раздражительность. У некоторых людей нарушается сон, аппетит, ухудшается зрение, начинают болеть руки, шея, поясница и тому подобное. При ненормированной работе возможно нервное истощение.

Рассмотрим одно из рабочих мест, изображенное на рисунке 5.3.

Рабочее место имеет следующие параметры:

- ширина рабочего стола равна 990 мм;
- глубина рабочего стола равна 520 мм;
- ширина пространства для ног составляет 530 мм;
- высота пространства для ног составляет 580 мм;
- глубина на уровне колен 270 мм;
- глубина на уровне вытянутых ног 520 мм;

- высота рабочей поверхности стола равна 740 мм.

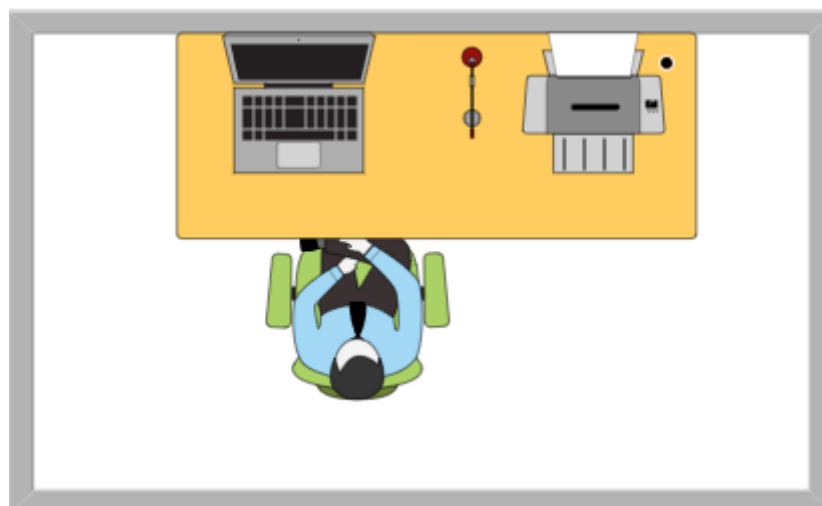


Рисунок 5.3 - Имеющееся рабочее место программиста

Клавиатура расположена от края стола на расстоянии 180 мм, экран ноутбука расположен на расстоянии 420 мм от глаз пользователя. На рабочем столе помимо ноутбука находится МФУ (многофункциональное устройство), настольная лампа и компьютерная мышь. Рабочий стол имеет выдвижную конструкцию снизу, а также шкаф с двумя полками под МФУ. Рабочее кресло подъемно-поворотное и легко фиксирующееся. Приблизительная площадь рабочего места составляет 4 квадратных метра. Материал рабочего стола - красное дерево.

Рассмотрим допустимые параметры рабочего места. Площадь рабочего места пользователя ПК с ЭЛТ-дисплеем должна составлять не менее 6 м², для ПК с плоским дисплеем - 4,5 м² (объемные нормы на одного человека - не менее 20 м³). В помещениях должна проводиться ежедневная влажная уборка и систематическое проветривание после каждого часа работы. Шумящее оборудование (печатающие устройства, сканеры, серверы и тому подобные), уровни шума которого превышают нормативные, должно размещаться вне рабочих мест сотрудников [30].

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования. Высота рабочей поверхности стола должна составлять 725 мм, рабочая поверхность стола должна иметь ширину 800...1400 мм и глубину 800...1000 мм. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм. Клавиатуру следует располагать на поверхности стола на расстоянии 100...300 мм от края, обращенного к пользователю, или на специальной поверхности, отделенной от основной столешницы. Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600...700 мм, но не ближе 500 [30].

Из этого наглядно видно, что рабочее место не удовлетворяет некоторым нормируемым значениям. Во-первых, необходимо увеличить площадь рабочего места до 4,5 - 5 м². Высота рабочей поверхности стола, а также его ширина соответствуют стандартам для нормальной рабочей среды, однако его глубина почти на 300 мм ниже минимального допустимого порога, в связи с этим глубина на уровне колен и на уровне вытянутых ног также не удовлетворяет нормам, высота пространства для ног также имеет незначительно отклонение от нормы. Поскольку на рабочей поверхности стола расположен ноутбук, то расстояние от экрана до глаз можно с легкостью регулировать, однако по нормам данное значение варьируется от 600 до 700 мм, а глубина стола составляет 520 мм, таким образом данная регулировка невозможно.

Можно сделать вывод о том, что площадь поверхности стола мала для нормальной работы сотрудника, решением будет являться его замена. Кроме того, на столе имеется шумящее оборудование, которое занимает почти половину рабочей поверхности - по рекомендациям данное устройство не советуют размещать непосредственно в пределах рабочего места сотрудника. Кроме того, необходимая документация находится шкафах - низко, так как нет места на столе, в связи с этим сотрудник будет постоянно напрягаться и нагибаться за ними. Рабочее кресло удовлетворяет требованиям, также на столе в случае недостаточности освещения размещен настольный светильник со светодиодными лампами. Также из плюсов можно сказать, что рабочее место расположено непосредственно слева от световых проемов, что сказывается положительно на зрительной системе.

Также необходимо придерживаться следующих рекомендаций [30]:

- не рекомендуется работать за компьютером более 6 часов за смену;
- рекомендуется делать перерывы в работе за ПК продолжительностью 10 минут через каждые 50 минут работы;
- продолжительность непрерывной работы за компьютером без регламентированного перерыва не должна превышать 2 часов;
- во время регламентированных перерывов целесообразно выполнять комплексы упражнений.

Заключение

В результате выполнения дипломного проекта были выполнены поставленные цели и задачи. Было разработано техническое задание проекта и написана постановка задачи. В ходе работы было спроектировано информационное обеспечение программного обеспечения: выбрана СУБД для хранения данных, построена физическая и логическая модели базы данных, на основе которого были созданы и описаны структуры таблиц БД подсистемы «Электронный журнал» и «Новости и объявления», описаны связи между ними.

Кроме того, была спроектированы диаграммы UML: диаграммы вариантов использования для различных акторов, а также диаграммы последовательности; разработаны основные бизнес-модели и показаны бизнес-объекты приложения.

В ходе проектирования программного обеспечения была описана структура ПО, обоснован выбор инструментально ПО (IDE Android Studio, Java SE 11, MySQL, серверный язык PHP), разработаны прототипы интерфейса и написаны основные классы и методы для нормального функционирования приложения.

В итоге дипломного проектирования было создано мобильное приложение «AUES Mobile» с подсистемами «Электронный журнал» и «Новости и объявления». Данное мобильное приложение относится к клиент-серверным ИС, а также относится к информационно-поисковым и информационно-справочным системам. Посредством первой подсистемы пользователь-студент может просматривать свои текущие и итоговые оценки в режиме реального времени по изучаемым дисциплинам, а функция транскрипта предоставляем ему доступ к просмотру итоговой оценки по всем дисциплинам в течение всего его обучения. Преподаватель имеет возможность выставления оценок по дисциплинам как текущих, так и итоговых. Кроме того, у обоих пользователей есть функция просмотра академического календаря.

Пользователи обеспечены актуальными новостями и объявлениями, касающиеся университета, а администраторы наделены функцией добавления новостей и объявлений.

Кроме того, мобильное приложение работает в интеграции с аналитической системой Firebase Analytics для выявления потребностей пользователей.

В конце работы были произведены тестирование и отладка, в ходе которых не были выявлены никакие ошибки.

Список использованной литературы

- 1 Харди Б., Филлипс Б., Стюарт К., Марсикано К. Android. Программирование для профессионалов. 2-е изд. — СПб.: Питер, 2016. — 640 с.: ил. — (Серия «Для профессионалов»). ISBN 978-5-496-02051-0
- 2 Шилдт, Герберт. Java 8. Полное руководство; 9-е изд.: Пер. с англ. - М.: ООО "И.Д. Вильямс", 2015. - 1376 с.: ил. - Парал. тит. англ. ISBN 978-5-8459-1918-2 (рус.)
- 3 Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие - М.: Postgress Professional, 2017. - 256 с.
- 4 Поляков Е. В. PHP на примерах. 2-е изд. - «Наука и техника», 2017. - 256с.
- 5 Гвоздева Т.В. Проектирование информационных систем. - Ростов-на-Дону: Феникс, 2009.
- 6 Приложение «Univer» на платформе Google Play <https://play.google.com/store/apps/details?id=kz.kaznu.univer.android&hl=ru>
- 7 Приложение «ИТМО» на платформе Google Play <https://play.google.com/store/apps/details?id=ru.ifmo.itmostudents&hl=ru>
- 8 Приложение «Narhoz Mobile» на платформе Google Play https://play.google.com/store/apps/details?id=kz.narhoz.narhoz_mobile&hl=ru
- 9 Статья «All You Need to Know About UML Diagrams: Types and 5+ Examples» на сайте <https://tallyfy.com/uml-diagram/>
- 10 Статья «Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch and others» на сайте <https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>
- 11 Статья «The advantages and disadvantages of MySQL» на сайте <http://makble.com/the-advantages-and-disadvantages-of-mysql>
- 12 Статья «What are the benefits of creating an app in Android Studio?» на сайте <https://www.quora.com/What-are-the-benefits-of-creating-an-app-in-Android-Studio>
- 13 Статья «Why is Java preferred for developing an Android app?» на сайте <https://www.quora.com/Why-is-Java-preferred-for-developing-an-Android-app>
- 14 Статья «Advantages of XML | disadvantages of XML» на сайте <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-XML.html>
- 15 Статья «Advantages and Disadvantages of XML» на сайте <https://beginnersbook.com/2018/10/advantages-and-disadvantages-of-xml/>
- 16 Официальная документация по NavigationView на сайте <https://developer.android.com/reference/android/support/design/widget/NavigationView>

17 Официальная документация по DrawerLayout на сайте <https://developer.android.com/reference/android/support/v4/widget/DrawerLayout.html>

18 Официальная документация по RelativeLayout на сайте <https://developer.android.com/guide/topics/ui/layout/relative>

19 Статья «Using the CardView» на сайте <https://guides.codepath.com/android/using-the-cardview>

20 ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы

21 ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы (пример) на сайте https://www.prj-exp.ru/patterns/pattern_tech_task.php

22 Грекул В.И. и др. Проектирование информационных систем. - М.: Интернет Университет Информационных Технологий, 2010.

23 Боканова Г.Ш. Методические указания по выполнению экономической части дипломных работ для студентов специальностей 5В070400 - «Вычислительная техника и программное обеспечение», 5В070300 - «Информационные системы», 5В060200 - «Информатика». Алматы, АУЭС - 2020, 35 с.

24 Шевчук Д. Организация и финансирование инвестиций, Litres, 2017, 877 с.

25 Статья Кэти Мартинели «A Guide to the Most Common Workplace Hazards» на сайте <https://www.highspeedtraining.co.uk/hub/hazards-in-the-workplace/>

26 Статья Роя Маурэра «5 Common Office Hazards to Prevent» на сайте <https://www.shrm.org/resourcesandtools/hr-topics/risk-management/pages/common-office-hazards.aspx>

27 Статья «Климат Алматы» на сайте https://ru.wikipedia.org/wiki/Климат_Алма-Аты

28 Приложение 8 «Коэффициенты тепло пропускания солнцезащитных устройств» на сайте <http://base.garant.ru/2306019/38d0e20d10a9099ed1e190ab-f152a12a/>

29 И.Б.Львовский, Б.В.Баркалов Пособие 2.91 к СНиП 2.04.05-91 расчет поступления теплоты солнечной радиации в помещения

30 Статья «Охрана труда за компьютером» на сайте https://ru.m.wikipedia.org/wiki/Охрана_труда_за_компьютером

ПРИЛОЖЕНИЕ А

Техническое задание

1 Общие сведения

1.2 Наименование системы

1.2.1 Полное наименование системы

Полное наименование:
Мобильное приложение для АУЭС

1.2.2 Краткое наименование системы

Краткое наименование:
AUES Mobile

1.2.3 Основания для проведения работ

Работа выполняется на основании договора от 4.05.2019 между руководством АУЭС им. Г. Даукеева и Азизовым М. и Цомартовым А.

1.2.4 Наименование организаций Заказчика и Разработчика

1.2.4.1 Заказчик

Заказчик: АО «Алматинский университет энергетики и связи» им. Г.Даукеева
Адрес фактический: г. Алматы, Байтурсынулы, 126/1
Телефон / Факс: +7 (727) 292 03 03, 323 11 75

1.2.4.2 Разработчик

Разработчик: Цомартов Айдер, Азизов Мадияр
Адрес фактический: г. Алматы
Телефон / Факс: +7 (778) 3816911, +7 (705) 7713016

1.2.5 Источники и порядок финансирования

Финансирование для данной системы необходимо для единственного пункта - добавление приложения в Play Маркет. Для того, чтобы опубликовать приложение для платформы Android, необходимо иметь аккаунт разработчика. Необходимая часть данного процесса - внесение единовременного платежа в размере \$25.

Продолжение приложения А

Финансирование необходимо после последнего этапа проекта - тестирование и отладка программного обеспечения ИС. Источником финансирования является АУЭС им. Г. Даукеева.

1.2.6 Плановые сроки начала и окончания работы

Работа проводится в соответствии графику, представленному в таблице 1.

Таблица 1 - Плановые сроки работы

№	Наименование работы	Начало работы	Конец работы
1	Анализ существующих мобильных приложений для ВУЗов	13.01.2020	13.02.2020
2	Проектирование функциональной структуры ИС	13.01.2020	10.03.2020
3	Разработка бизнес-модели	13.01.2020	10.03.2020
4	Проектирование информационного обеспечения ИС	13.01.2020	10.03.2020
5	Разработка прототипа ИС	13.01.2020	10.03.2020
6	Разработка программного обеспечения ИС	13.01.2020	30.04.2020
7	Тестирование и отладка программного обеспечения ИС	15.04.2020	30.04.2020

1.2.7 Порядок оформления и предъявления заказчику результатов работ

Работы по созданию данной системы предоставляются Разработчиком согласно календарному плану проекта поэтапно.

Результаты работ предъявляются заказчику в виде документов - отчетов, которые содержат скриншоты части работ, далее загружается арк-файл на смартфон с целью его дальнейшей установки и демонстрации функционала части системы.

1.3 Назначение и цели создания (развития) системы

1.3.1 Назначение системы

Поскольку в современном информационном обществе из года в год растет количество мобильных приложений, людям нет необходимости запоминать URL различных систем для удовлетворения своих информационных потребностей.

Также данный факт связан с увеличением количества пользователей мобильных смартфонов. Кроме того, мобильные приложения в большинстве имеют более компактный и удобный интерфейс по сравнению с мобильными адаптациями сайтов, а десктопные устройства не всегда могут оказаться под рукой. Возможен и тот факт, что тот или иной сайт не будет адаптирован под смартфоны и планшеты.

Мобильное приложение данной системы предназначено для оперативного и своевременного доступа и обеспечения его пользователям (в данном случае студентам, преподавателям и сотрудникам вуза) надлежащей информацией, удобства и персонализации.

1.3.2 Цели создания системы

Разработать мобильное приложение согласно современным требованиям интерфейсов «Material Design» и с помощью современных средств информационных технологий.

1.4 Характеристика объектов автоматизации

Объектом автоматизации является Алматинский Университет Энергетики и Связи, а точнее процессы, происходящие в нем. Мобильное приложение создается для трех видов пользователей: студент бакалавр, преподаватель и сотрудник. В первую очередь пользователи должны пройти аутентификацию и авторизацию в приложении. Для них необходимо автоматизировать процесс просмотра журнала (кроме пользователя «сотрудник»), транскрипта и новостей. Также приложение будет включать общую информацию о структуре университета. Администрирование данной подсистемы будут осуществлять один и или два пользователя «сотрудник», которые будут иметь необходимый функционал, также они будут ответственны за размещение новостей и объявлений.

1.5 Требования к системе

1.5.1 Требования к системе в целом

Взаимодействия производятся по протоколу HTTP(S).

В Системе предлагается выделить следующие функциональные подсистемы:

- подсистема хранения и загрузки данных, которая предназначена для реализации процессов просмотров новостей и объявлений;
- подсистема хранения данных, которая предназначена для электронного журнала.

Продолжение приложения А

Источниками данных для Системы должны быть:

- подсистема «Электронный журнал» (СУБД MySQL);
- подсистема «Новости и объявления» (СУБД MySQL);
- Google Analytics.

В режиме функционирования система должна обеспечивать:

- работу пользователей в режиме - 24 часа в день, 7 дней в неделю (24x7);
- выполнение своих функций: сбор, обработка и загрузка данных; хранение данных.

Необходимо обеспечить защиту пользовательских данных.

1.5.2 Требования к функциям, выполняемым системой

С точки зрения пользователя «Студент»:

1) просмотр новостей - отражается информация о новостях, включающая в себя дату опубликования, заголовок, и текст (ограничение 125 символов), при этом при нажатии на новость, должна открываться страница с полным текстом;

2) просмотр электронного журнала успеваемости - отражается информация о дисциплинах, текущих оценках с датами проставления, транскрипте, экзамене и рубежном контроле.

Требования к интерфейсу приложения:

На странице ВУЗа организовать три вкладки с правой стороны экрана:

1) вкладка «Главная»:

- новости (для всех);
- объявление (для всех).

2) вкладка зависит от авторизации пользователя:

- академический календарь;
- транскрипт;
- дистанционные курсы;
- телефонный справочник;
- библиотека.

С точки зрения пользователя «Преподаватель»:

1) просмотр новостей - отражается информация о новостях, включающая в себя дату опубликования, заголовок, и текст (ограничение 125 символов), при этом при нажатии на новость, должна открываться страница с полным текстом;

2) просмотр электронного журнала и выставление оценок;

3) информация о группах и преподаваемых дисциплинах, все виды оценок студентов.

С точки зрения пользователя «Сотрудник»:

Продолжение приложения А

- 1) просмотр новостей - отражается информация о новостях, включающая в себя дату опубликования, заголовок, и текст (ограничение 125 символов), при этом при нажатии на новость, должна открываться страница с полным текстом.
- 2) просмотр текущих и итоговых оценок.

1.5.3 Требования к ПО

Мобильное приложение для платформы Android необходимо разработать в среде интегрированной разработки «Android Studio» на языке программирования Java версии 11. Минимальная версия API 14, версия Android 4.4 Billy Jean.

Серверная часть пишется на языке программирования PHP версии 7.3.9. В качестве системы управления базами данных выступает СУБД MySQL версии 8.0.7.

1.6 Состав и содержание работ по созданию системы

Работы по созданию системы выполняются в три этапа: проектирование, разработка эскизного проекта, разработка технического проекта.

Подробные сведения о этапах разработки приведены в пункте 1.2.6. Плановые сроки начала и окончания работы.

1.7 Источник разработки

Настоящее Техническое Задание разработано на основе следующих документов и информационных материалов:

- договор от 4.05.2019 между руководством АУЭС им. Г. Даукеева и Азизовым М. и Цомартовым А.

ПРИЛОЖЕНИЕ Б

Листинг программы

Листинг 1. Adapter PDF

```
package com.example.aues;

import android.content.Context;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class AdapterPdf extends RecyclerView.Adapter<AdapterPdf.ExampleViewHolder> {
    private Context mContext;
    private ArrayList<ExampleAdapterPDF> mExampleList;
    private OnItemClickListener mListener;

    public interface OnItemClickListener {
        void onItemClick(int position);
    }

    public void setOnItemClickListener(OnItemClickListener listener) {
        mListener = listener;
    }

    public AdapterPdf(Context context, ArrayList<ExampleAdapterPDF> exampleList) {
        mContext = context;
        mExampleList = exampleList;
    }

    @Override
    public ExampleViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v = LayoutInflat-
```

```

er.from(mContext).inflate(R.layout.example_item_calenda
r, parent, false);
    return new ExampleViewHolder(v);
}

@Override
public void onBindViewHolder(ExampleViewHolder
holder, int position) {
    ExampleAdapterPDF currentItem = mExam-
pleList.get(position);

    String link = currentItem.getLink();
    hold-
er.mTextViewLink.setText(Html.fromHtml(link));

```

Продолжение приложения Б

```

}

@Override
public int getItemCount() {
    return mExampleList.size();
}

public class ExampleViewHolder extends Recycler-
View.ViewHolder {
    public TextView mTextViewLink;

    public ExampleViewHolder(View itemView) {
        super(itemView);

        mTextViewLink =
itemView.findViewById(R.id.calendar_name);

mTextViewLink.setMovementMethod(LinkMovementMethod.getI
nstance());

        itemView.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (mListener != null) {
                    int position = getAdapterPosi-
tion();

```



```

        String id_student = arguments.get("id_student").toString();
        Fragment fragment = null;
        FragmentManager fm = getSupportFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();
        if (role_id == 2) {
            setTitle("Дисциплина " + discipline);
            fragment = new cur_marks(code);
        }
        else if (role_id == 3) {
            setTitle(discipline);
            fragment = new cur_marks(code, id_student);
        }
        ft.replace(R.id.frag, fragment);
        ft.commit();
    }
}

```

Листинг 3. Detail Activity

```

package com.example.aues;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.widget.ImageView;
import android.widget.TextView;

import com.squareup.picasso.Picasso;

import static com.example.aues.newsss.EXTRA_AUTHOR;
import static com.example.aues.newsss.EXTRA_BODY;
import static com.example.aues.newsss.EXTRA_DATE;
import static com.example.aues.newsss.EXTRA_TITLE;
import static com.example.aues.newsss.EXTRA_URL;

public class DetailActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {

```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        Intent intent = getIntent();
        String imageUrl = intent.getStringExtra(EXTRA_URL);
        String author = intent.getStringExtra(EXTRA_AUTHOR);
        String title = intent.getStringExtra(EXTRA_TITLE);
        String body = intent.getStringExtra(EXTRA_BODY);
        String date = intent.getStringExtra(EXTRA_DATE);

        setTitle(title);

        ImageView imageView = findViewById(R.id.image_view_detail);
        TextView textViewAuthor = findViewById(R.id.text_view_author_detail);
        TextView textViewTitle = findViewById(R.id.text_view_title_detail);
        TextView textViewBody = findViewById(R.id.text_view_body_detail);
        TextView textViewDate = findViewById(R.id.text_view_date_detail);
        Picasso.with(this).load(imageUrl).fit().centerInside().into(imageView);
        textViewAuthor.setText("by " + author);
        textViewTitle.setText(title);

        textViewBody.setMovementMethod(LinkMovementMethod.getInstance());
        textViewDate.setText(date);
    }
}

```

Продолжение приложения Б

Листинг 4. Final Works List

```

package com.example.aues;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.text.Editable;
import android.text.TextWatcher;
import android.widget.EditText;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

```

Продолжение приложения Б

```

public class Final_Works_List extends AppCompatActivity
{
    private String code, group_id, training;
    private RecyclerView mRecyclerView;
    private AdapterTypesOfFinalWorks mExampleAdapter;
    private ArrayList<SampleItemTypesOfFinalWorks> mExampleList;
    private RequestQueue mRequestQueue;

    String url =
    "https://auesmobiledatabase.000webhostapp.com/works.php";

```

```

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_final__works__list);
        setTitle("Промежуточные итоги");

        Bundle arguments = getIntent().getExtras();
        code = arguments.get("code_discipline").toString();
        group_id = arguments.get("id_group").toString();
        training = arguments.get("training").toString();

        mRecyclerView = findViewById(R.id.recycler_view_works);
        mRecyclerView.setHasFixedSize(true);
        mRecyclerView.setLayoutManager(new LinearLayoutManager(Final_Works_List.this));

        mExampleList = new ArrayList<>();

        mRequestQueue = Volley.newRequestQueue(Final_Works_List.this);

        parseJSON();
    }

    private void parseJSON() {
        StringRequest request = new StringRequest(Request.Method.POST, url,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    try {
                        JSONObject jsonObject = new
                        JSONObject(response);
                        JSONArray jsonArray =
                        jsonObject.getJSONArray("works");
                        for (int i = 0; i < jso-

```

```

nArray.length(); i++) {
    JSONObject works = jsonArray.getJSONObject(i);
    String ID = works.getString("ID");
    String name = works.getString("NAME");
}

```

Продолжение приложения Б

```

        mExampleList.add(new
SampleItemTypesOfFinalWorks(ID, name));
    }

    mExampleAdapter = new
AdapterTypesOfFinalWorks(Final_Works_List.this, mExam-
pleList);

    mRecyclerView-
View.setAdapter(mExampleAdapter);

    AdapterTypesOfFinal-
Works.OnItemClickListener exampleAdapter = new Adapter-
TypesOfFinalWorks.OnItemClickListener() {
        @Override
        public void onItem-
Click(int position) {
            Intent detailIntent
= new Intent(Final_Works_List.this, Put-
FinalMarks.class);

            SampleItemTypesOf-
FinalWorks clickedItem = mExampleList.get(position);
            detail-
Intent.putExtra("id_work", clickedItem.getID());
            detail-
Intent.putExtra("work_name", clickedItem.getName());
            detail-
Intent.putExtra("code", code);
            detail-
Intent.putExtra("group_id", group_id);

            startActivi-
ty(detailIntent);
        }
}

```

```

        };

        mExampleAdapter
er.setOnItemClickListener(exampleAdapter);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            error.printStackTrace();
        }
    });

    mRequestQueue.add(request);
}
}

```

Листинг 5. Insert Announces

```

package com.example.aues;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
    Продолжение приложения Б

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;

```

```

import java.util.Map;

public class InsertAnnounces extends AppCompatActivity
{
    Button add_announces;
    EditText Title_ru, Title_kz, Title_eng, Body_ru,
Body_kz, Body_eng;
    String URL =
"https://auesmobiledatabase.000webhostapp.com/addannoun-
ces.php";
    AlertDialog.Builder builder;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insert_announces);
        setTitle("Добавить объявление");
        add_announces = (Button) findViewById(R.id.add_announces);
        Title_ru = (EditText) findViewById(R.id.title_ru_edit);
        Title_kz = (EditText) findViewById(R.id.title_kz_edit);
        Title_eng = (EditText) findViewById(R.id.title_eng_edit);
        Body_ru = (EditText) findViewById(R.id.body_ru_edit);
        Body_kz = (EditText) findViewById(R.id.body_kz_edit);
        Body_eng = (EditText) findViewById(R.id.body_eng_edit);

        builder = new AlertDialog.Builder(InsertAnnounces.this);

        add_announces.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final String title_ru, title_kz, ti-
tle_eng, body_ru, body_kz, body_eng;
                title_ru = Ti-
tle_ru.getText().toString();

```

```

        title_kz = Title_kz.getText().toString();
        title_eng = Title_eng.getText().toString();
        body_ru = Body_ru.getText().toString();
        body_kz = Body_kz.getText().toString();
        body_eng = Body_eng.getText().toString();
        StringRequest stringRequest = new
StringRequest(Request.Method.POST, URL,
                new Response.Listener<String>()
{

```

Продолжение приложения Б

```

                @Override
                public void onResponse(String response) {
builder.setTitle("Добавление новости");
                builder.setMessage(response);
                final String resp = response;
                builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        Title_ru.setText("");
                        Title_kz.setText("");
                        Title_eng.setText("");
                        Body_ru.setText("");
                        Body_kz.setText("");
                        Body_eng.setText("");
                    }
                });
                if (resp.equals("Данные были успешно внесены")) {
                    Intent in-

```



```

tent = new Intent(InsertAnnounces.this, MainActivity.class);
                                                                    startActivity(intent);
                                                                    }
                                                                    }
                                                                    });
                                                                    AlertDialog alertDialog = builder.create();
                                                                    alertDialog.show();
                                                                    }
                                                                    }
                                                                    , new Response.ErrorListener()
{
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(InsertAnnounces.this, "Ошибка" + error.toString(), Toast.LENGTH_SHORT).show();
        error.printStackTrace();
    }
}) {
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> values = new HashMap<String, String>();
        values.put("title_ru", title_ru);
        values.put("title_kz", title_kz);
        values.put("title_eng", title_eng);
        values.put("body_ru", body_ru);
        values.put("body_kz", body_kz);
        values.put("body_eng", body_eng);
        return values;
    }
};

```

Продолжение приложения Б

```

        stringRequest.setRetryPolicy(new Retry-
Policy() {
            @Override
            public int getCurrentTimeout() {
                return 50000;
            }

            @Override
            public int getCurrentRetryCount() {
                return 50000;
            }

            @Override
            public void retry(VolleyError er-
ror) throws VolleyError {
                }
            });
        MySingle-
ton.getInstance(InsertAnnounces.this).addToRequestque(s
tringRequest);
    }
}
}
}

```

Листинг 6. Insert News

```

package com.example.aues;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.provider.MediaStore;
import android.util.Base64;
import android.view.View;
import android.widget.Button;

```

```

import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;

```

```

import java.io.ByteArrayOutputStream;

```

Продолжение приложения Б

```

import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

```

```

public class InsertNews extends AppCompatActivity implements View.OnClickListener {
    SharedPreferences mSharedPreferences;
    Button add_news, choose_photo;
    String id_admin;
    EditText Title_ru, Title_kz, Title_eng, Body_ru,
Body_kz, Body_eng, Name_Of_Photo;
    ImageView image;
    private Bitmap bitmap;
    private final int IMG_REQUEST = 1;
    String URL =
"https://auesmobiledatabase.000webhostapp.com/addnews.php";
    AlertDialog.Builder builder;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_insert_news);
        setTitle("ДОБАВИТЬ НОВОСТЬ");
        image = (ImageView) findViewById(R.id.image);
        Name_Of_Photo = (EditText) findViewById(R.id.name_of_photo);
        add_news = (Button) findViewById(R.id.add_news);
        Title_ru = (EditText) findViewById(R.id.title_ru_edit);

```

```

        Title_kz = (Edit-
Text) findViewById(R.id.title_kz_edit);
        Title_eng = (Edit-
Text) findViewById(R.id.title_eng_edit);
        Body_ru = (Edit-
Text) findViewById(R.id.body_ru_edit);
        Body_kz = (Edit-
Text) findViewById(R.id.body_kz_edit);
        Body_eng = (Edit-
Text) findViewById(R.id.body_eng_edit);
        choose_photo = (But-
ton) findViewById(R.id.choose_photo);
        choose_photo.setOnClickListener(this);
        builder = new AlertDia-
log.Builder(InsertNews.this);

        mSharedPreferences = PreferenceManag-
er.getDefaultSharedPreferences(getBaseContext());
        id_admin = mSharedPrefer-
ence.getString("id_admin", "");

        add_news.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final String title_ru, title_kz, ti-
tle_eng, body_ru, body_kz, body_eng, name_of_photo;
                name_of_photo =
Name_Of_Photo.getText().toString();
                title_ru = Ti-
tle_ru.getText().toString();
                title_kz = Ti-
tle_kz.getText().toString();
                title_eng = Ti-
tle_eng.getText().toString();
                body_ru = Body_ru.getText().toString();
                body_kz = Body_kz.getText().toString();
                body_eng =
Body_eng.getText().toString();
                StringRequest stringRequest = new
StringRequest(Request.Method.POST, URL,
                new Response.Listener<String>()
{

```

Продолжение приложения Б

```

        @Override
        public void onRe-
sponse(String response) {

builder.setTitle("Добавление новости");
er.setMessage(response);
        final String resp = re-
sponse;
er.setPositiveButton("OK", new DialogInterface.OnClickListener() {
        @Override
        public void on-
Click(DialogInterface dialog, int which) {
t1e_ru.setText("");
t1e_kz.setText("");
t1e_eng.setText("");
Body_ru.setText("");
Body_kz.setText("");
Body_eng.setText("");
age.setImageResource(0);
age.setVisibility(View.GONE);
Name_Of_Photo.setText("");
Name_Of_Photo.setVisibility(View.GONE);
        if
(resp.equals("Данные были успешно внесены")) {
Intent in-
tent = new Intent(InsertNews.this, MainActivity.class);
startActiv-
ity(intent);
        }
        }
});

```

```

AlertDialog alertDialog
= builder.create();
        alertDialog.show();
    }
}
, new Response.ErrorListener()
{
    @Override
    public void onErrorRe-
sponse(VolleyError error) {
        Toast.makeText(InsertNews.this,
"Ошибка" + error.toString(),
Toast.LENGTH_SHORT).show();
        error.printStackTrace();
    }
}){
    @Override
    protected Map<String, String> get-
Params() throws AuthFailureError {
        Map<String, String> values = new
HashMap<String, String>();
        values.put("image", imageTo-
Str(bitmap));
        values.put("name",
name_of_photo);
        values.put("title_ru", ti-
tle_ru);
        values.put("title_kz", ti-
tle_kz);
        values.put("title_eng", ti-
tle_eng);
        values.put("body_ru", body_ru);
        values.put("body_kz", body_kz);
        values.put("body_eng",
body_eng);
        values.put("id_admin",
id_admin);
        return values;
    }
};
stringRequest.setRetryPolicy(new Retry-
Policy() {

```

Продолжение приложения Б

```

        @Override
        public int getCurrentTimeout() {
            return 50000;
        }

        @Override
        public int getCurrentRetryCount() {
            return 50000;
        }

        @Override
        public void retry(VolleyError error) throws VolleyError {

        }

        });
        MySingleton
ton.getInstance(InsertNews.this).addToRequestque(string
Request);
    }
    });
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.choose_photo:
            ChooseImage();
            break;
    }
}

private void ChooseImage() {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(intent, IMG_REQUEST);
}

@Override
protected void onActivityResult(int requestCode,
int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode,
data);
    if (requestCode == IMG_REQUEST && resultCode ==

```

```

RESULT_OK && data != null) {
    Uri path = data.getData();
    try {
        bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(),
path);
        image.setImageBitmap(bitmap);
        image.setVisibility(View.VISIBLE);

Name_Of_Photo.setVisibility(View.VISIBLE);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

private String imageToStr(Bitmap bitmap) {
    ByteArrayOutputStream byteArrayOutputStream =
new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG,
100, byteArrayOutputStream);
    byte[] imageBytes = byteArrayOut-
putStream.toByteArray();
    return Base64.encodeToString(imageBytes,
Base64.DEFAULT);
}
}

```

Продолжение приложения Б

Листинг 7. List Of Calendar

```

package com.example.aues;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;

```



```

import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class ListOfCalendar extends AppCompatActivity {

    private RecyclerView mRecyclerView;
    private AdapterPdf mExampleAdapter;
    private ArrayList<ExampleAdapterPDF> mExampleList;
    private RequestQueue mRequestQueue;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_list_of_calendar);
        setTitle("Академический календарь");
        Продолжение приложения Б

        mRecyclerView = findViewById(
            R.id.recycler_view);
        mRecyclerView.setHasFixedSize(true);
        mRecyclerView.setLayoutManager(
            new LinearLayoutManager(
                ListOfCalendar.this));

        mExampleList = new ArrayList<>();

        mRequestQueue = Volley.newRequestQueue(
            ListOfCalendar.this);
        parseJSON();
    }

    private void parseJSON() {
        String url =
            "https://auesmobiledatabase.000webhostapp.com/calendar.
            php";
        StringRequest request = new StringRe-
            quest(Request.Method.POST, url,
                new Response.Listener<String>() {
                    @Override

```

```

        public void onResponse(String re-
response) {
            try {
                JSONObject jsonObject= new
JSONObject(response);
                JSONArray jsonArray =
jsonObject.getJSONArray("calendar");

                for (int i = 0; i < jso-
nArray.length(); i++) {
                    JSONObject lessons =
jsonArray.getJSONObject(i);

                    String link = les-
sons.getString("Link");

                    mExampleList.add(new
ExampleAdapterPDF(link));
                }

                mExampleAdapter = new
AdapterPdf(ListOfCalendar.this, mExampleList);
                mRecycler-
View.setAdapter(mExampleAdapter);

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse (VolleyError er-
ror) {
                error.printStackTrace();
            }
        });

        mRequestQueue.add(request);
    }
}

```

Продолжение приложения Б

Листинг 8. Put Current Marks

```

package com.example.aues;

```

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class PutCurrentMarksActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private CustomAdapter customAdapter;
    public ArrayList<EditModel> editModelArrayList;
    private String code;
    private String group_id;
    private String training;
    private int answer;
    SharedPreferences mSharedPreferences = null;

    private RecyclerView mRecyclerView; // recycler
with names

```

```

        private StudentsNameAdapter mExampleAdapter; //
adapter with names
        private ArrayList<Students_Name> mExampleList; //
list with names
        private RequestQueue mRequestQueue; // request

        @Override
        protected void onCreate(Bundle savedInstanceState)
        {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_put_current_marks);
            Bundle arguments = getIntent().getExtras();
            code = arguments.get("code_discipline").toString();
                Продолжение приложения Б

            group_id = arguments.get("id_group").toString();
            training = arguments.get("training").toString();
            recyclerView = (RecyclerView) findViewById(R.id.recycler);
            setTitle("Выставление текущих оценок");

            mRecyclerView = findViewById(R.id.recycler_for_marks);
            mRecyclerView.setHasFixedSize(true);
            mRecyclerView.setLayoutManager(new LinearLayoutManager(PutCurrentMarksActivity.this));

            mExampleList = new ArrayList<>();

            mRequestQueue = Volley.newRequestQueue(PutCurrentMarksActivity.this);
            parseJSON();
            editModelArrayList = populateList();
            customAdapter = new CustomAdapter(this, editModelArrayList);
            recyclerView.setAdapter(customAdapter);
            recyclerView.setLayoutManager(new LinearLayoutManager(getApplicationContext(), LinearLayoutManager.VERTICAL, false));

            Button btn = (Button) findViewById(R.id.put);

```

```

        btn.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            for (int i = 0; i < CustomAdapt-
er.editModelArrayList.size(); i++){
                int mark = Inte-
ger.parseInt(CustomAdapter.editModelArrayList.get(i).ge
tEditTextValue());
                if (mark > 100 || mark < 0) {
                    answer = 404;
                    break;
                }
            }
            if (answer == 404) {
                Toast toast =
Toast.makeText(getApplicationContext(),
                    "Некорректная оценка!",
                    Toast.LENGTH_SHORT);
                toast.setGravity(Gravity.CENTER, 0, 0);
                toast.show();
                answer = 0;
            }
            else {
                String url =
                "https://auesmobiledatabase.000webhostapp.com/put_curre
                nt_marks.php";
                StringRequest request = new
                StringRequest(Request.Method.POST, url,
                    new Re-
                sponse.Listener<String>() {
                    @Override
                    public void onRe-
                sponse(String response) {
                        try {
                            JSONArray
                jsonArray = new
                JSONArray(response);
                            JSONObject
                jsonObject = jsonArray.getJSONObject(0);
                            String

```

Продолжение приложения Б

```

feedback = jsonObject.getString("answer");

Log.d("ANSWER", feedback);

if(feedback.equals("New records created successfully"))
    {
        Toast
toast = Toast.makeText(getApplicationContext(), "Данные
успешно добавлены",
Toast.LENGTH_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();
    }
    else {
        Toast
toast = Toast.makeText(getApplicationContext(),
jsonObject.getString("answer"),
Toast.LENGTH_SHORT);

toast.setGravity(Gravity.CENTER, 0, 0);

toast.show();
    }
} catch (JSONException e) {
e.printStackTrace();
}
}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
error.printStackTrace();
}
})

```

```

        {
            @Override
            protected Map<String,
String> getParams() throws AuthFailureError {
                Map<String, String>
params = new HashMap<String, String>();
                params.put("code",
code);
                params.put("group",
group_id);
                params.put("training",
training);
                params.put("size",
String.valueOf(CustomAdapter.editModelArrayList.size())
);
                for(int i = 0; i < Cus-
tomAdapter.editModelArrayList.size(); i++)
                {
                    Продолжение приложения Б

                    params.put("params_" + i, CustomAdapt-
er.editModelArrayList.get(i).getEditTextValue());
                }

                return params;
            }
        };

        mRequestQueue.add(request);
    }
}
catch (Exception e) {
    Toast toast_string =
Toast.makeText(getApplicationContext(),
        "Некорректные данные!",
        Toast.LENGTH_SHORT);
    toast_string.setGravity(Gravity.CENTER,
0, 0);
    toast_string.show();
}
}
});
}
}

```

```

        private void parseJSON() {
            String url =
"https://auesmobiledatabase.000webhostapp.com/students_
name.php";
            StringRequest request = new StringRe-
quest(Request.Method.POST, url,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String re-
sponse) {
                        try {
                            JSONObject jsonObject = new
JSONObject(response);
                            JSONObject jsonObject1 =
jsonObject.getJSONObject("1");
                            JSONArray jsonArray =
jsonObject1.getJSONArray("students");

                            for (int i = 0; i < jso-
nArray.length(); i++) {

                                JSONObject students =
jsonArray.getJSONObject(i);

                                String name = stu-
dents.getString("STUDENT_NAME");

                                mExampleList.add(new
Students_Name(name));
                            }
                        }
                    }
                }
            );
        }
    }
}

```

Продолжение приложения Б

```

        mExampleAdapter = new Stu-
dentsNameAdapter(PutCurrentMarksActivity.this, mExam-
pleList);
        mRecycler-
View.setAdapter(mExampleAdapter);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```



```

        }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                error.printStackTrace();
            }
        })
        {
            @Override
            protected Map<String, String> getParams()
            throws AuthFailureError {
                Map<String, String> params = new
                HashMap<String, String>();
                params.put("code", code);
                params.put("group", group_id);
                params.put("training", training);

                return params;
            }
        };

        mRequestQueue.add(request);
    }

    private ArrayList<EditModel> populateList () {
        mSharedPreferences = PreferenceManag-
        er.getDefaultSharedPreferences(getBaseContext());
        int size = mSharedPreferences.getInt("size", 0);
        ArrayList<EditModel> list = new ArrayList<>();

        for (int i = 0; i < size; i++) {
            EditModel editModel = new EditModel();
            editModel.setEditTextValue("0");
            list.add(editModel);
        }

        return list;
    }
}

```

Листинг 9. Put Final Marks

```
package com.example.aues;
```

```
import androidx.appcompat.app.AppCompatActivity;
import an-
droidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
```

Продолжение приложения Б

```
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.text.Html;
import android.util.Log;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
```

```
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
```

```
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
```

```
public class PutFinalMarks extends AppCompatActivity {

    private RecyclerView recyclerView;
    private CustomAdapter customAdapter;
    public ArrayList<EditModel> editModelArrayList;
    private String code, group_id, id_work, work_name;

    private int answer;
    SharedPreferences mSharedPreferences = null;

    private RecyclerView mRecyclerView; // recycler
with names
```

```

        private StudentsNameAdapter mExampleAdapter; //
adapter with names
        private ArrayList<Students_Name> mExampleList; //
list with names
        private RequestQueue mRequestQueue; // request

        @Override
        protected void onCreate(Bundle savedInstanceState)
        {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_put_final_marks);
            Bundle arguments = getIntent().getExtras();
            code = arguments.get("code").toString();
            group_id = arguments.get("group_id").toString();
            id_work = arguments.get("id_work").toString();
            work_name = arguments.get("work_name").toString();
            recyclerView = (RecyclerView) findViewById(R.id.recycler);
            setTitle(work_name);

            mRecyclerView = findViewById(R.id.recycler_for_marks);
            mRecyclerView.setHasFixedSize(true);

            mRecyclerView.setLayoutManager(new LinearLayoutManager(PutFinalMarks.this));

            mExampleList = new ArrayList<>();

            mRequestQueue = Volley.newRequestQueue(PutFinalMarks.this);
            parseJSON();
            editModelArrayList = populateList();
            customAdapter = new CustomAdapter(this, editModelArrayList);
            recyclerView.setAdapter(customAdapter);
            recyclerView.setLayoutManager(new LinearLayoutManager(getApplicationContext(), LinearLayoutManager.VERTICAL, false));

            Button btn = (Button) findViewById(R.id.put);

```

Продолжение приложения Б

```

        btn.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            for (int i = 0; i < CustomAdapt-
er.editModelArrayList.size(); i++){
                int mark = Inte-
ger.parseInt(CustomAdapter.editModelArrayList.get(i).ge-
tEditTextValue());
                if (mark > 100 || mark < 0) {
                    answer = 404;
                    break;
                }
            }
            if (answer == 404) {
                Toast toast =
Toast.makeText(getApplicationContext(),
                    "Некорректная оценка!",
                    Toast.LENGTH_SHORT);
                toast.setGravity(Gravity.CENTER, 0, 0);
                toast.show();
                answer = 0;
            }
            else {
                String url =
                "https://auesmobiledatabase.000webhostapp.com/put_final-
                _marks.php";
                StringRequest request = new
                StringRequest(Request.Method.POST, url,
                    new Re-
                sponse.Listener<String>() {
                    @Override
                    public void onRe-
                    sponse(String response) {
                        try {
                            JSONArray
                            jsonArray = new JSONArray(response);
                            JSONObject
                            jsonObject = jsonArray.getJSONObject(0);
                            String
                            feedback = jsonObject.getString("answer");
                            Toast toast
                            = Toast.makeText(getApplicationContext(),

```

Продолжение приложения Б

```
Html.fromHtml(feedback),
Toast.LENGTH_LONG);
toast.setGravity(Gravity.CENTER, 0, 0);
toast.show();

} catch (JSONException e) {
e.printStackTrace();
}
}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
error.printStackTrace();
})
{
@Override
protected Map<String, String>
getParams() throws AuthFailureError {
Map<String, String>
params = new HashMap<String, String>();
params.put("code",
code);
params.put("group",
group_id);
params.put("id_work",
id_work);
params.put("size",
String.valueOf(CustomAdapter.editModelArrayList.size()));
Log.d("CODE", code +
group_id + id_work);
for(int i = 0; i < CustomAdapter.editModelArrayList.size(); i++)
```

```

        {
            Log.d("PARAMS",
CustomAdapt-
er.editModelArrayList.get(i).getEditTextValue());

params.put("params_" + i, CustomAdapt-
er.editModelArrayList.get(i).getEditTextValue());
        }

        return params;
    }
};

mRequestQueue.add(request);
}
}
catch (Exception e) {
    Toast toast_string =
Toast.makeText(getApplicationContext(),
        "Некорректные данные!",
        Toast.LENGTH_SHORT);

toast_string.setGravity(Gravity.CENTER, 0, 0);
    toast_string.show();
        Продолжение приложения Б
    }
}
});
}

private void parseJSON() {
    String url =
"https://auesmobiledatabase.000webhostapp.com/students_
name.php";
    StringRequest request = new StringRe-
quest(Request.Method.POST, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String re-
sponse) {
                try {
                    JSONObject jsonObject = new

```

```

JSONObject (response) ;
                                JSONObject jsonObject1 =
jsonObject.getJSONObject ("1");
                                JSONArray jsonArray =
jsonObject1.getJSONArray ("students");

                                for (int i = 0; i < json-
nArray.length(); i++) {

                                JSONObject students =
jsonArray.getJSONObject (i);

                                String name = stu-
dents.getString ("STUDENT_NAME");

                                mExampleList.add (new
Students_Name (name));
                                }

                                mExampleAdapter = new Stu-
dentsNameAdapter (PutFinalMarks.this, mExampleList);
                                mRecycler-
View.setAdapter (mExampleAdapter);

                                } catch (JSONException e) {
                                    e.printStackTrace();
                                }
                                }, new Response.ErrorListener() {
@Override
public void onErrorResponse (VolleyError er-
ror) {
    error.printStackTrace();
}
}))
{
@Override
protected Map<String, String> getParams ()
throws AuthFailureError {
    Map<String, String> params = new
        Продолжение приложения Б
HashMap<String, String> ();
        params.put ("code", code);

```

```

        params.put("group", group_id);
        params.put("training", "L");

        return params;
    }
};

mRequestQueue.add(request);

}

private ArrayList<EditModel> populateList () {
    mSharedPreferences = PreferenceManager.
er.getDefaultSharedPreferences(getApplicationContext());
    int size = mSharedPreferences.getInt("size", 0);
    ArrayList<EditModel> list = new ArrayList<>();

    for (int i = 0; i < size; i++) {
        EditModel editModel = new EditModel();
        editModel.setEditTextValue("0");
        list.add(editModel);
    }

    return list;
}
}
}

```


ПРИЛОЖЕНИЕ В Авторское свидетельство

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ
МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

ҚҮӨЛІК

2020 жылғы «27» ақпан № 8425

Автордың (ардың) аты, аты, әкесінің аты (егер-ол жеке бағын куәландыратын құжатта көрсетілсе):
ТУСУПОВА БАТІА Б.ТЯ.ЛЕВНА

Авторлық құқық объектісі: **ЗЕМ-ге арналған бағдарлама**

Объектінің атауы: **Мәліметтік приложении «AUES mobile»**

Объектінің алынған күні: **18.06.2019**



Құқық тағайындаушының жеке басын куәландіретін қолданылған
"Авторлық құқық" бағдарламасымен бірге берілген. Ақпарат: <http://www.kazpatent.kz>

Подлинность документа электронной подписью не заменяет и не отменяет
его действительности. Авторское право © 2019, Государственный патентный центр РК

Подписано ЭЦП

Абулкаиров Н.А.

Продолжение приложения В

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

СВИДЕТЕЛЬСТВО
О ВНЕСЕНИИ СВЕДЕНИЙ В ГОСУДАРСТВЕННЫЙ РЕЕСТР
ПРАВ НА ОБЪЕКТЫ, ОХРАНЯЕМЫЕ АВТОРСКИМ ПРАВОМ
№ 8425 от «27» февраля 2020 года

Фамилия, имя, отчество (если оно указано в документе, удостоверяющем личность) автора (ов):
ТУСУПОВА БЭЛЛА БИГІЗІВНА

Вид объекта авторского права: программа для ЭВМ

Название объекта: Мобильное приложение «AUES mobile»

Дата создания объекта: 18.06.2019



Адрес: <http://www.kazpatent.kz> или <http://www.kazpatent.kz/ru>
"Авторские права", Белаякан, Сибирский бульвар, 119а, 120000, Алматы, Республика Казахстан

Подлинность документа можно проверить на сайте [kazpatent.kz](http://www.kazpatent.kz)
в разделе «Авторское право» <http://www.kazpatent.kz/ru>

Подписано ЗЦП

Абупкаиров Н.А.