

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
«Расписание»

Разработка мобильного приложения «AUES Mobile». Подсистема «Расписа-
ние».

Специальность 5В070300 - Информационные системы

Выполнил Цомартов А.Э.

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: _____
(ученая степень, звание, Ф.И.О.)

_____ « ____ » _____ 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 Цомартову Айдеру Эдемовичу

Тема проекта «Разработка мобильного приложения «AUES Mobile».
Подсистема «Расписание»»

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» и информационно-аналитическая
система для расписания «Архит».

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, согласно подсистеме, «Расписание»;
- б) проведение анализа существующих систем - аналогичных мобиль-
ных приложений для различных университетов;
- б) разработка информационного обеспечения ИС подсистемы «Распи-
сание»;
- в) разработка дружественного интерфейса и программная реализация
системы;
- г) экономическое обоснование проекта;
- 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. - (Се-
рия «Для профессионалов»).

2 Медникс З., Дорнин Л., Мик Б., Накамура М. Программирование под
Android. 2-е изд. - СПб.: Питер, 2013. - 560 с.: ил. - (Серия «Бестселлеры
O'Reilly»).

3 Гриффитс Дон, Гриффитс Дэвид Head First. Программирование для
Android. - СПб.: Питер, 2016. - 704 с.: ил. - (Серия «Head First O'Reilly»).

4 Хорстманн К.С. Java. Библиотека Профессионала. 9-е изд. - СПб.: Питер, 2014. - 860 с.: ил. - (Серия «Библиотека Профессионала»).

5 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. 2-е изд. - СПб.: Питер, 2014. - 544 с.: ил. - (Серия «Бестселлеры O'Reilly»). 6 Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие - М.: Postgress Professional, 2017. - 256 с.

8 Блох Д. Java: эффективное программирование 3-е изд. - СПб.: Питер, 2014. - 310 с.: ил. - (Серия «Java «из первых рук»).

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 _____ Цомартов А.Э.
(подпись)

Аннотация

Объектами автоматизации в ходе проектирования являются процесс оперативного получения информации о расписании студентов и преподавателей, процесс мгновенного обмена и получения информации между пользователями системы. Кроме того, мобильное приложение «AUES Mobile» имеет функции, которые позволяют отслеживать студентам сведения об их задолженности, информацию об оплате и стипендиях. Перед пользователями открывается вся информации об университете. Интерфейс представлен согласно правилам Material Design. При проектировании использовались методы хеширования паролей, а также система токенов для безопасной защиты данных и не получения информации вторыми лицами. Мобильное приложение является программным обеспечением поискового и справочного типа, а также системой ввода и обработки информации. Данная информационная система служит для своевременного обеспечения надлежащей информацией надлежащих пользователей в лице преподавателей и студентов. В результате дипломного проектирования было создано мобильное приложение «AUES Mobile» для смартфонов на базе операционной системы Android для АУЭС имени Гумарбека Даукеева.

Андатпа

Жобалау барысында автоматтандыру объектілері бұл студенттер мен оқытушылардың сабақ кестесі, жедел алмасу және жүйенің пайдаланушылары арасында ақпараттар алу туралы жедел ақпарат алу процесі. Сонымен қатар, «AUES Mobile» мобильді қосымшасында студенттерге қарыздары, төлемдер мен шәкіртақылар туралы ақпараттарды бақылауға мүмкіндік беретін функциялар бар. Пайдаланушылар университет туралы барлық ақпаратты ашпас бұрын. Интерфейс Material Design ережелеріне сәйкес ұсынылған. Дизайн кезінде парольді хэштеу әдістері, сонымен қатар мәліметтерді қауіпсіз қорғаудың және екінші тараптардың ақпарат алмауының таңбалы жүйесі қолданылды. Мобильді қосымша - бұл іздеу және анықтамалық типтегі бағдарламалық қамтамасыз ету, сонымен қатар ақпаратты енгізу және өңдеу жүйесі. Бұл ақпараттық жүйе дұрыс пайдаланушыларға оқытушылар мен студенттердің тұлғаларында уақытылы ақпарат беруге қызмет етеді. Дипломдық жұмыстың нәтижесінде Ғумарбек Даукеев атындағы АЭЖ-БУ-ға арналған Android операциялық жүйесі негізінде смартфондар үшін «AUES Mobile» мобильді қосымшасы жасалған.

The Summary

The objects of automation during the design process are the process of promptly obtaining information about the timetable of students and teachers, the process of instant exchange and obtaining information between users of the system. In addition, the «AUES Mobile» mobile application has functions that allow students to track information about their debts, information about payments and scholarships. Before users opens all the information about the university. The interface is presented according to the rules of Material Design. During the design, password hashing methods were used, as well as a token system for secure data protection and not receiving information by second persons. The mobile application is search and reference type software, as well as an information input and processing system. This information system serves to timely provide appropriate information to the right users in the person of teachers and students. As a result of graduation design, the mobile application «AUES Mobile» was created for smartphones based on the Android operating system for the «AUPET» named after Gumarbek Daukeev.

Содержание

Введение	9
1 Обзор литературы и анализ современного состояния вопроса.....	10
1.1 Обзор литературы.....	10
1.2 Анализ существующих систем	11
1.3 Постановка задачи.....	16
2 Разработка информационного обеспечения системы	18
2.1 Проектирование функциональной структуры	18
2.2 Разработка бизнес модели ИС	22
2.3 Описание информационного обеспечения	31
3 Разработка программного обеспечения	52
3.1 Обоснование выбора среды программирования.....	52
3.2 Построение макета пользовательского интерфейса	54
3.3 Структура программного обеспечения	69
3.4 Клиентская часть	70
3.5 Серверная часть.....	74
3.6 Тестирование и отладка программного обеспечения.....	77
4 Экономическое обоснование проекта	85
4.1 Определение объема и трудоемкости разработки программного обеспечения	85
4.2 Расчет затрат на разработку информационных технологий	87
4.3 Расчет динамических показателей эффективности капитальных вложений	93
5 Охрана труда и БЖД	96
5.1 Теоретическая часть	96
5.2 Расчетная часть.....	98
Заключение	111
Список использованной литературы.....	112
ПРИЛОЖЕНИЕ А Техническое задание	115
ПРИЛОЖЕНИЕ Б Листинг программы	120
ПРИЛОЖЕНИЕ В Акт внедрения.....	147
ПРИЛОЖЕНИЕ Г Авторское свидетельство.....	148

Введение

Целью данной дипломной работы является разработка Android-приложения «AUES Mobile». 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. Целью подсистемы «Расписание» является предоставление пользователю о расписании конкретного студента или преподавателя в режиме реального времени. Система должна поддерживать основной режим функционирования, в котором подсистемы выполняют функции согласно индивидуальной модели доступа к ним, определяемой системным администратором ВУЗа. Основной режим требует авторизации и аутентификации от пользователя для системы.

Объектом автоматизации является НАО «Алматинский Университет Энергетики и Связи имени Гумарбека Даукеева», 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 расписания, журнала (кроме пользователя «сотрудник»), транскрипта и новостей. Также приложение будет включать общую информацию о структуре университета.

1 Обзор литературы и анализ современного состояния вопроса

1.1 Обзор литературы

В процессе дипломного проектирования было задействовано множество источников, в том числе научная литература, статьи из интернета, а также ГОСТы. При разработке мобильного приложения «AUES Mobile», в значительной степени, помогла книга «Программирование для Android» от автора Д.Гриффитс. В книге должным образом описано: как структурировать мобильное приложение, разрабатывать гибкие и интерактивные интерфейсы, запускать службы в фоновом режиме, заставить приложение работать на различных смартфонах и планшетах. Также хорошим помощником послужила официальная документация от Google для Android-разработчиков, размещенная на сайте [https://develo-per.android.com/](https://developer.android.com/). Кроме того, были получены глубокие знания по взаимодействию XML и Java в среде Android Studio, что сильно помогло в разработке мобильного приложения «AUES Mobile» [1].

Теоретический анализ литературы позволил выделить перспективное направление разработки мобильного приложения на базе языка программирования Java. Для изучения данного языка программирования было задействовано несколько источников, важнейшим из которых является книга «Java. Библиотека профессионала» автора Хортсманн К.С. Существует огромное количество литературы по Java, но данная книга выделяется тем, что охватывает основы языка и объектно-ориентированной модели и включает в себя обзоры среды и инструменты для работы с Java, посредством чего в качестве среды был выбран Android Studio. Немаловажной является глава об исключениях, которая содержит введение в то, когда и почему используются исключения, описание особенностей механизма исключений в Java, а также ряд полезных советов по обработке исключений [2].

Для подробного изучения языка SQL была выбрана книга «SQL. Библия пользователя» авторов Кригель А, Трухнова Б. Объединяя теорию с повседневной практичностью, этот труд содержит самую актуальную информацию, новые функции и пояснения, необходимые для максимально эффективного использования SQL и его последнего стандарта. Книга уникальна тем, что в каждой главе рассказывается о том, как новый стандарт SQL применяется к трем основным СУБД: Oracle 11g, Microsoft SQL Server 2008 и MySQL. Были изучены основы SQL и реляционных баз данных, получены обширные знания по функциям, процедурам и транзакциям, а также знания по обеспечению безопасности данных [3].

В качестве основной литературы для изучения языка PHP и его кооперацией с MySQL была выбрана книга «PHP и MYSQL. Исчерпывающее руководство» автора Маклафлин. Б. В книге детально описаны базовые принципы программирования на языке PHP, структуру управления PHP, а также основные функции и процедуры. Кроме того, описано, как совместно использовать PHP и MySQL, что сильно помогло при разработке мобильного при-

ложения «AUES Mobile», т. к. сервер приложения полностью разработан при помощи кооперации PHP и MySQL на базе PHP MyAdmin[4].

Также стоит отметить, что было задействовано множество интернет-ресурсов, которые помогли в выборе основного языка программирования для клиентской части и базы данных для серверной части.

1.2 Анализ существующих систем

В данном разделе рассматриваются приложения таких университетов, как 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».

НИЯУ МИФИ

Данное приложение расположено в магазине приложений Google Play[5].

Название приложения : «НИЯУ МИФИ»

Дата выхода: 5 сентября 2016 года

Основные возможности:

- просмотр расписаний;
- общение с Мифистами присутствующими с вами на одной паре;
- общение с одноклассниками;
- поиск свободных аудиторий;
- новости МИФИ;
- просмотр отзывов на преподавателей;
- просмотр перлов преподавателей;
- удобный доступ к энциклопедии НИЯУ МИФИ.

Исходя из отзывов пользователей, приложению не хватает:

- сортировка результатов поиска по библиотеке;
- просмотра расписания экзаменов и зачетов;
- постоянные вылеты приложения по непонятным причинам.

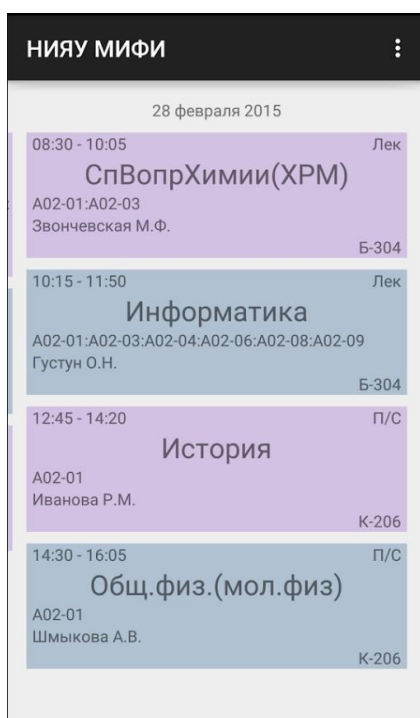


Рисунок 1.1 - Аудитории

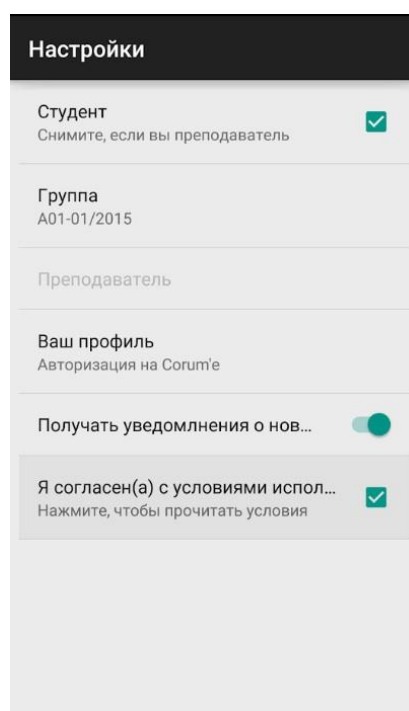


Рисунок 1.2 - Авторизация

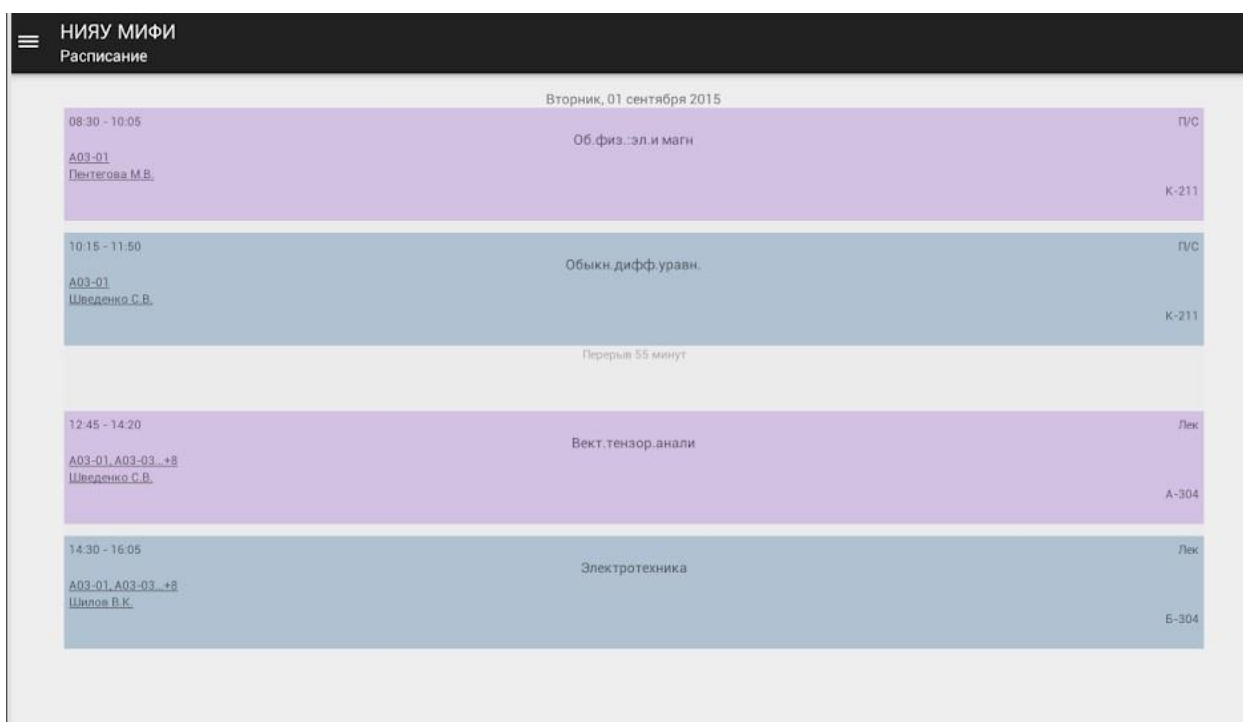


Рисунок 1.3 - Расписание

МГИМО

Данное приложение расположено в магазине приложений Google Play[6].

Название приложения : «МГИМО»

Официальное приложение «МГИМО» стало доступно 28 августа 2018 года.

Имеются сведения о расписании, информация об университете, новости, касающиеся университета, а также вкладка с предстоящими и прошедшими событиями. Во вкладке расписания есть возможность переключения по датам, при этом пишется полная информация о каждом занятии, включая факультет, группу, тип занятия (лекция, практика), предмет, преподавателя и аудиторию. Также есть возможность обратной связи с разработчиками во вкладке «Пожелания по приложению», к тому же студенты имеют возможность контактировать с администрацией университета во вкладке «Написать в администрацию», еще в приложении есть своя карта учебного заведения, а также контакты. Судя по отзывам пользователей, приложение достаточно неплохо справляется с функцией просмотра расписания, но также есть и недочеты: нет фотографий преподавательского состава, нет оффлайн расписания, а также пользователи отмечают постоянные вылеты [6].



Рисунок 1.4 - Главная страница

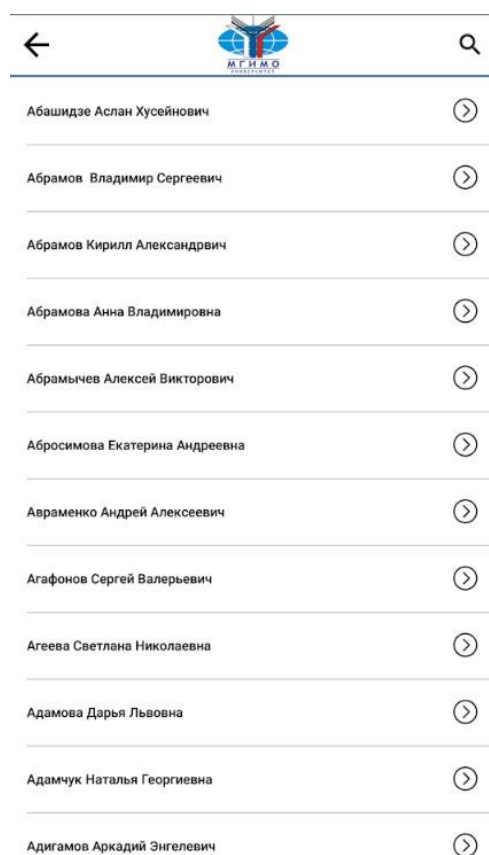


Рисунок 1.5 - Преподавательский состав

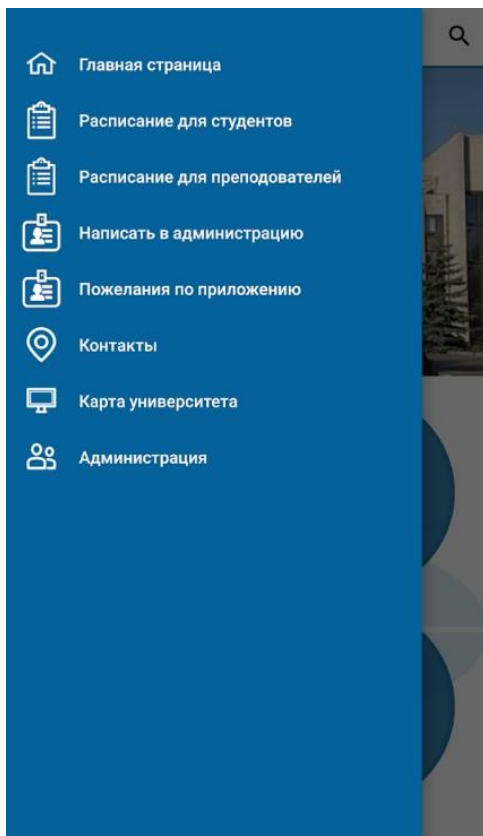


Рисунок 1.6 - Панель навигации



Рисунок 1.7 - Новости

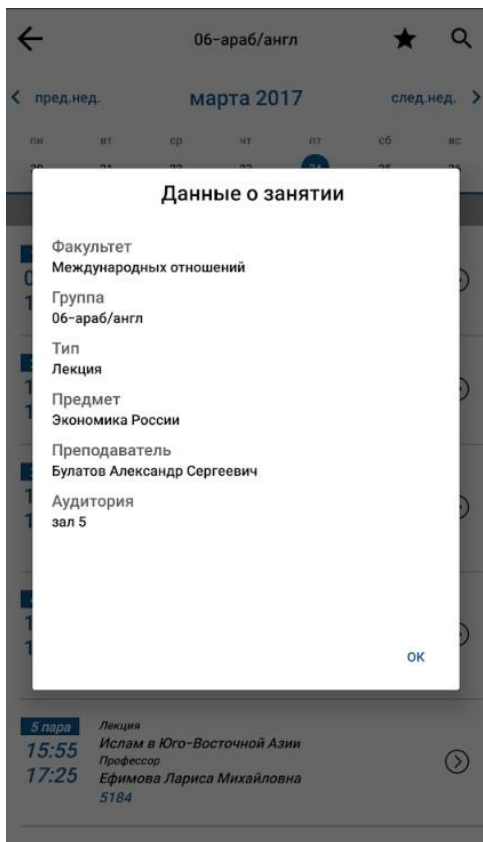


Рисунок 1.8 - Данные о занятии

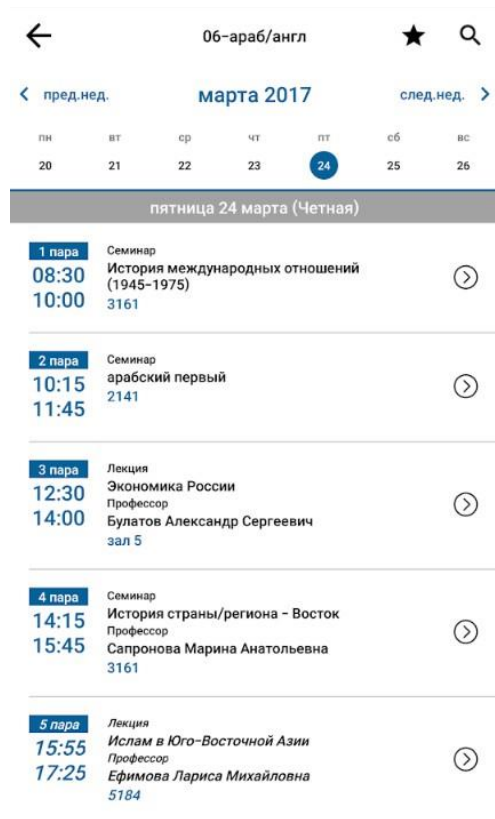


Рисунок 1.9 - Расписание

Сибирский Федеральный Университет

Данное приложение расположено в магазине приложений Google Play[7].

Название приложения: «Студент СФУ»

Дата выхода: 25 мая 2017 г.

Официальное приложение «Студент СФУ» предоставляет студентам [7]:

- актуальное расписание занятий абсолютно всех групп с прогрессирующей визуализацией времени;
- возможность найти попутчиков в университет и домой;
- бюро находок;
- карта кампуса СФУ;
- объявления кафедр и институтов СФУ;
- срочные уведомления от преподавателей;
- новости и др.

Имеется возможность авторизации пользователя, также для абитуриентов имеется «Подбор специальности», в котором высчитывается вероятность поступления на грант по разным специальностям, принимая во внимание ситуацию конкретного абитуриента.

Однако, пользователи жалуются на плохую оптимизацию, например, приложение не работает с версией Android 9.0, также расписание обновляется несвоевременно, т.е. нет связи с официальным сайтом университета, а также на постоянные вылеты.

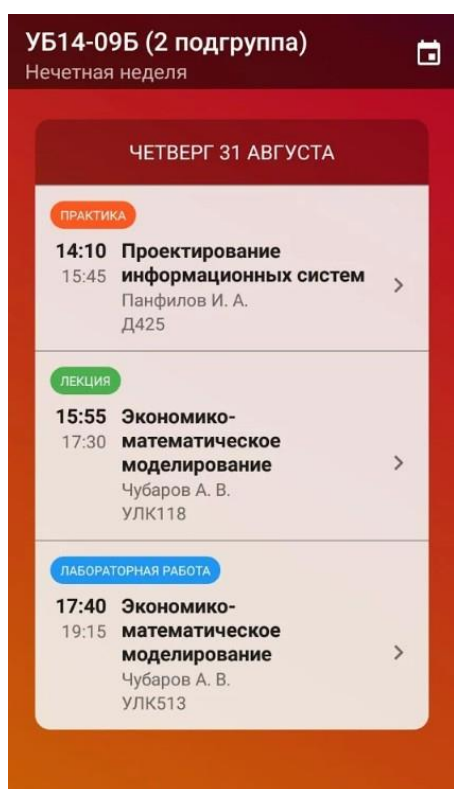


Рисунок 1.10 - Расписание группы

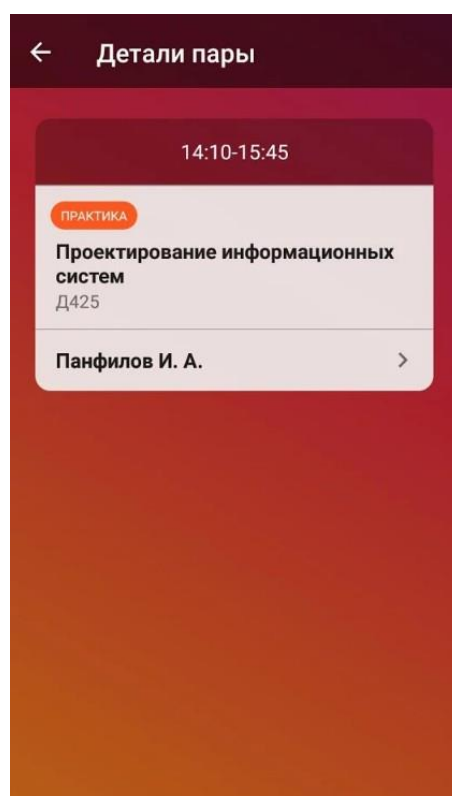


Рисунок 1.11 - Информация о занятии

Посредством проведения обзора аналогичных существующих приложений были выявлены их достоинства и недостатки, также получилось узнать мнение пользователей о функциональности приложений, таким образом, с помощью проведенного обзора и ТЗ от заказчика - университета АУЭС, будет проведена разработка мобильного приложения АУЭС, которое включает в себя такие функции как [7]:

- получение информации о расписании занятий группы и преподавателя для обоих типов пользователей: студент и преподаватель в режиме реального времени;
- информация об индивидуальном плане как студента, так и преподавателя;
- механизм обратной связи, посредством которого пользователи могут связываться с разработчиками и администрацией университета;
- информация о контактных данных университета, его структуре, истории;
- обеспечение пользователей информацией о преподавательском составе университета.

1.3 Постановка задачи

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

- обеспечить получение информации о расписании занятий группы и преподавателя для обоих типов пользователей: студент и преподаватель в режиме реального времени;
- предоставить информацию об индивидуальном плане как студента, так и преподавателя;
- обеспечить надежную защиту данных, используя современные средства информационных технологий;
- осуществить механизм получения экстренных сообщений от администрации университета;
- осуществить механизм обратной связи, посредством которого пользователи могут связываться с разработчиками и администрацией университета;
- предоставить информацию о контактных данных университета, его структуре, истории;
- отобразить Google-карту, на которой будут отображены географические координаты и само местоположение университета;
- обеспечить пользователей информацией о преподавательском составе университета;
- обеспечить механизм отправки и получения сообщений между пользователями приложения;

- предоставить информацию телефонного справочника преподавателям и сотрудникам университета;

- предоставить информацию о стипендиях, оплате, задолженностях.

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;

- должны быть созданы связи между сущностями: один-к-одному, многие-к-одному (или наоборот) и многие-ко-многим (обязательные или необязательные);

- в проектируемой модели необходимо исключить лишние связи без потери данных;

- 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 нижеуказанным 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 ИС можно выделить следующие подсистемы, касающиеся обучающихся и преподавателей:

1) подсистема «Расписание». С точки зрения пользователя «Преподаватель»: отражается информация о расписании занятий преподавателя по дням. Данная информация включает в себя время начала и конца занятия, номер занятия, дисциплину, вид занятия, группу, корпус и аудиторию, в которой проходит то или иное занятие. С точки зрения пользователя «Студент»: отражается информация о расписании занятий той группы, в которой обучается пользователь «Студент» по дням. Данная информация включает в себя время начала и конца занятия, номер занятия, дисциплину, вид занятия, преподавателя, его степень, корпус и аудиторию, в которой проходит то или иное занятие.

2) подсистема «Облачный чат». С точки зрения пользователя «Студент»: просмотр экстренных новостей. С точки зрения пользователя «Администратор»: просмотр и внесение экстренных новостей. То есть при помощи инструмента от Google при использовании Realtime Database будет осуществлена работа по обмену сообщениями.

3) подсистема, связанная с информацией об университете. В данную подсистему входят функции просмотра таких данных как история университета, информация о преподавателях университета, информация о корпусах и общежитиях университета, информация об институтах и специальностях и информация о руководстве университета. Имеется отдельная вкладка в панели навигации для обучающихся, информация о сотрудничестве и академической мобильности и информация о проектах.

4) подсистема, в которой студенты могут получать данные о стипендиях, задолженностях и оплате.

Функциональная структура самой системы представлена на рисунках 2.2-2.3.

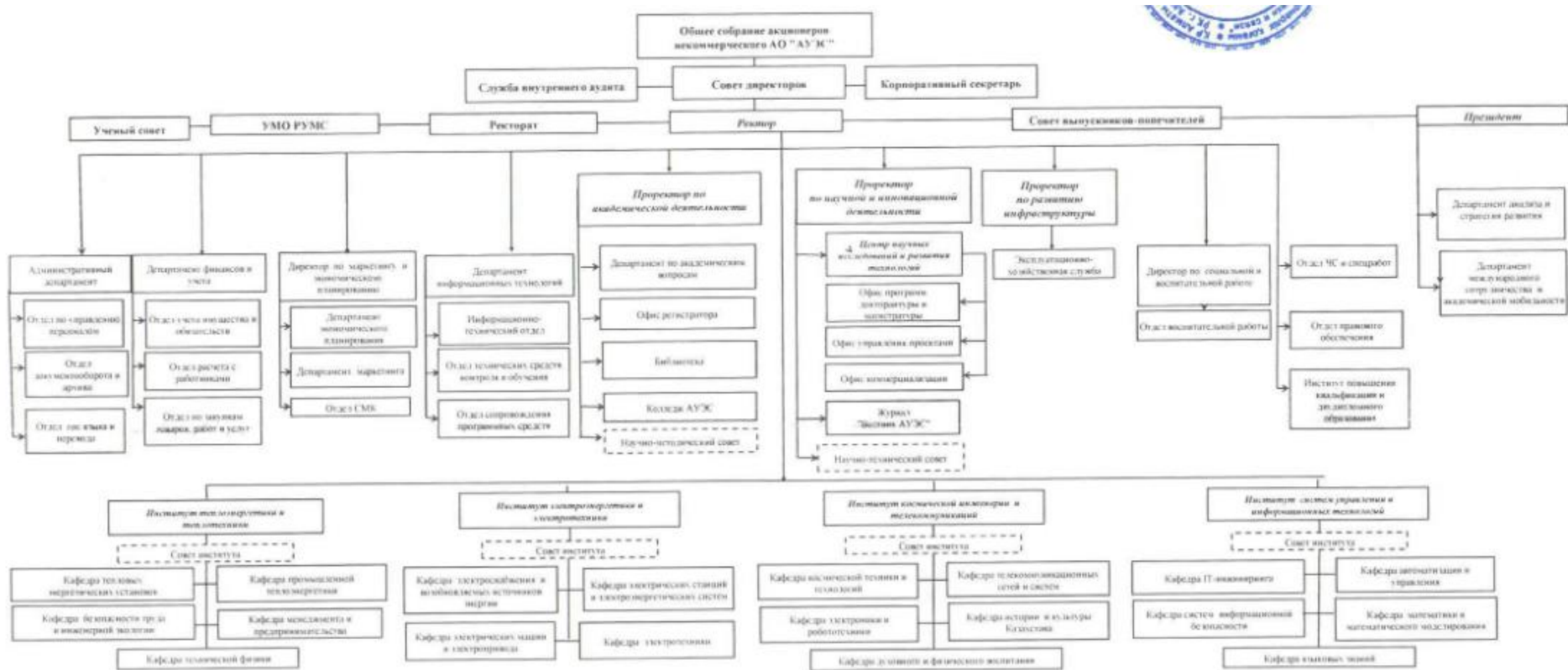


Рисунок 2.1 - Организационная структура университета



Рисунок 2.2 - Функциональная структура подсистем «Расписание» и «Личный кабинет»

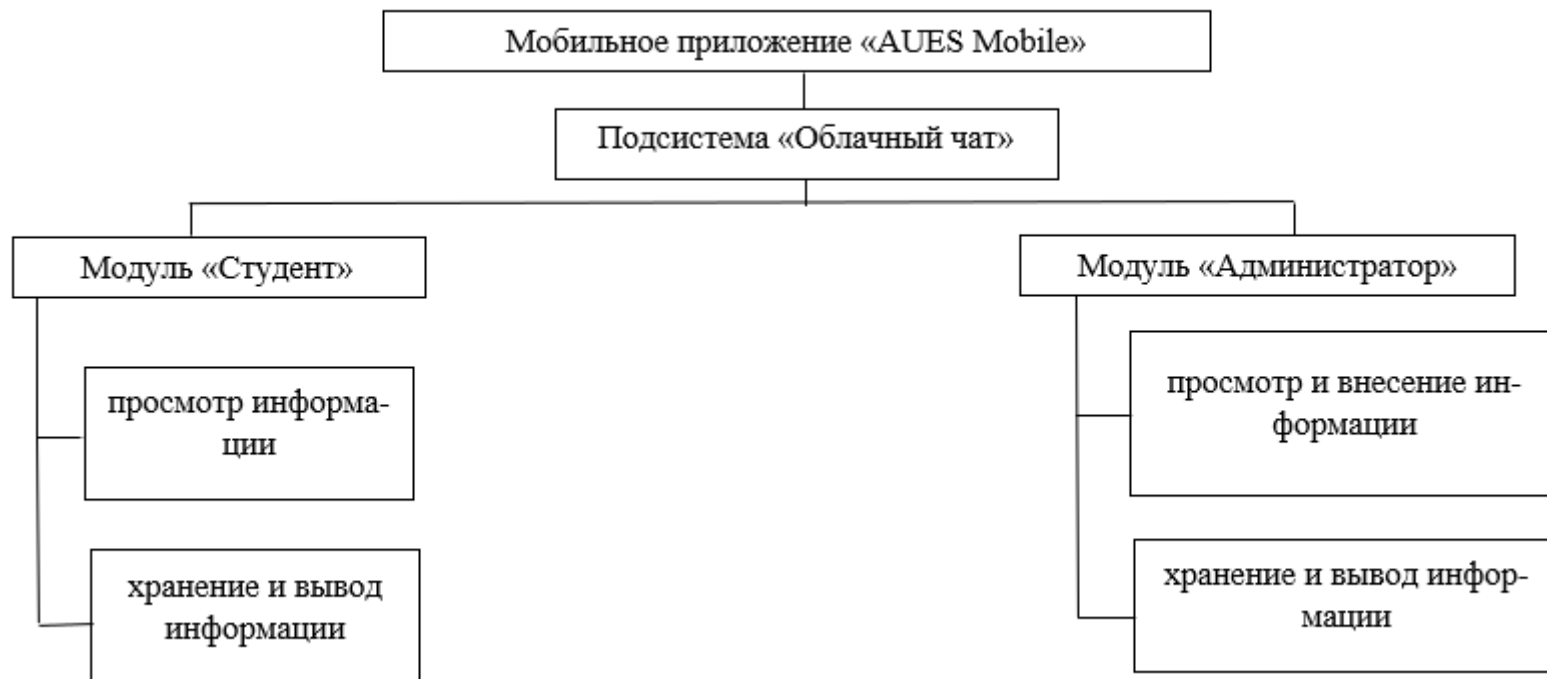


Рисунок 2.3 - Функциональная структура подсистемы «Облачный чат»

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.

2.2.2 Предпроектное обследование

Цель работы: необходимо создать организационную, функциональную и информационную модели для комплексного описания бизнеса и выделения проблемной области для построения информационной системы подсистемы «Расписание».

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.

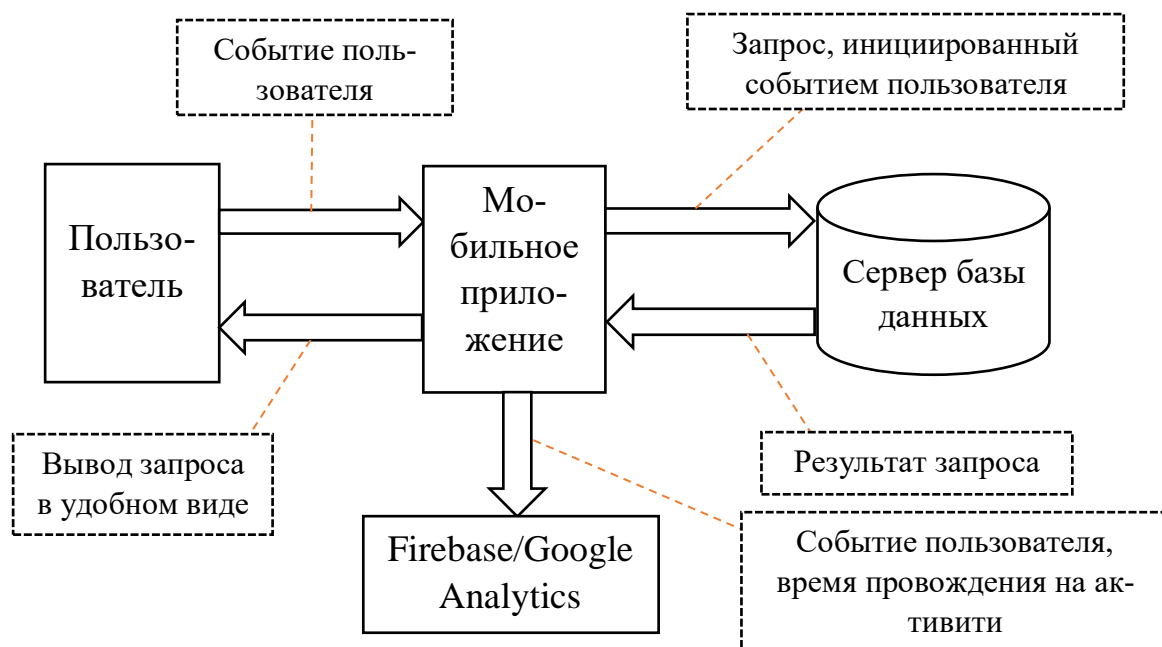


Рисунок 2.4 - Функциональная модель

2.2.2.1.2 Информационная модель

Информационная модель подсистемы «Расписание» для пользователя «Студент»

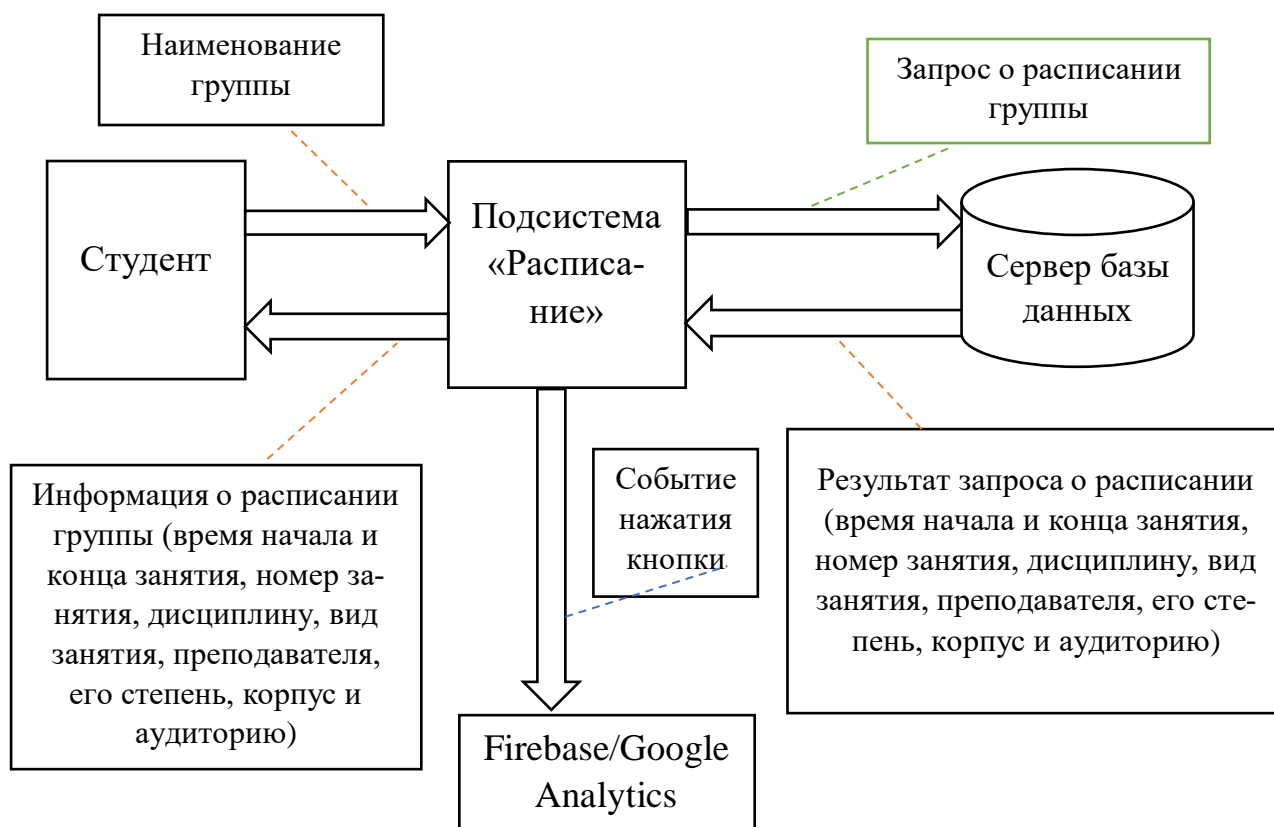


Рисунок 2.5 - Информационная модель подсистемы «Расписание» для пользователя «Студент»

Информационная модель подсистемы «Расписание» для пользователя «Преподаватель»

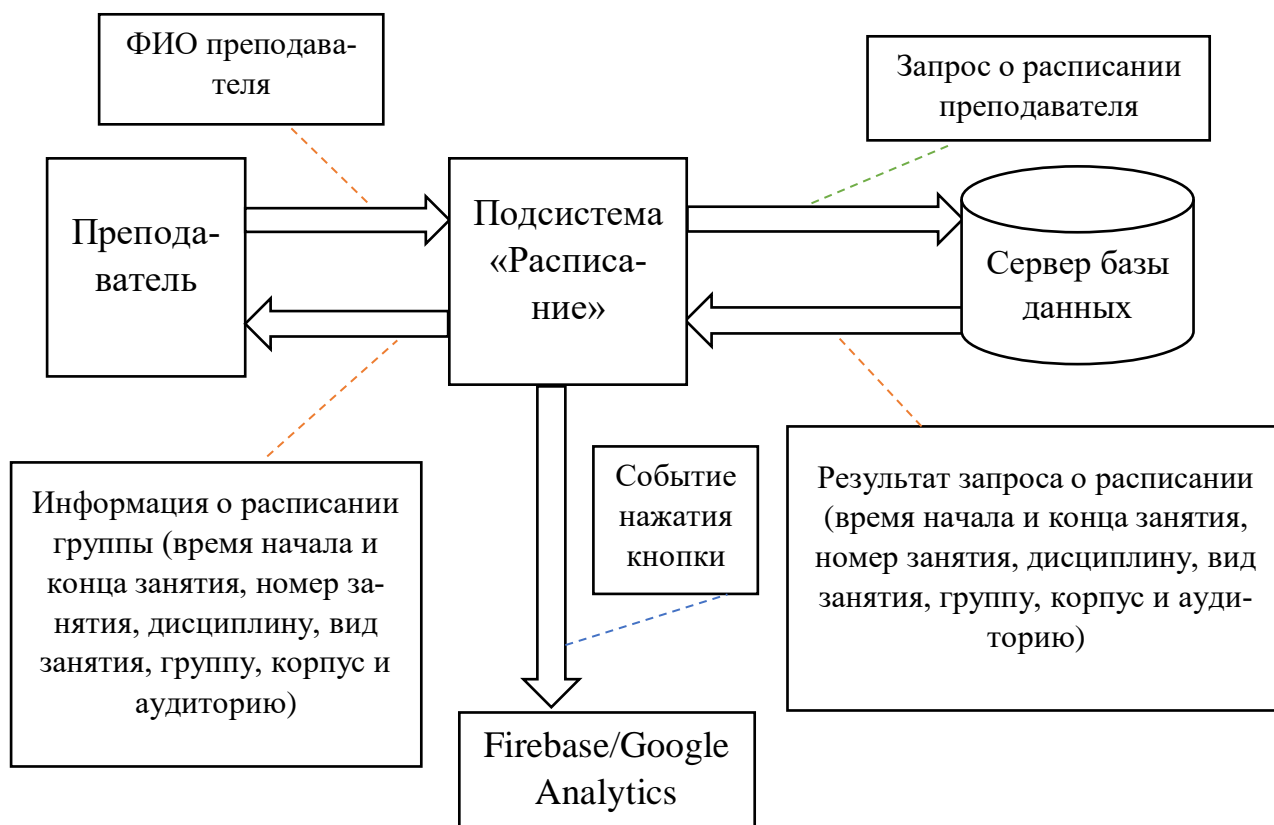


Рисунок 2.6 - Информационная модель подсистемы «Расписание» для пользователя «Преподаватель»

2.2.3 Выделение бизнес-процессов

В данном пункте будут рассмотрены все бизнес-процессы, которые есть в приложении «AUES Mobile». В бизнес-процессах представлено взаимодействие приложения и сервера посредством функций и библиотек Volley.

В работе выделяются следующие бизнес-процессы: «Предоставление расписания студенту», «Предоставление расписания преподавателю».

Ниже представлено взаимодействие студента с приложением и базой данных, при предоставлении расписания (рис. 2.7). Студент отправляет свои данные на получение информации о расписании на веб-сайт, который формирует запрос на сервер, на основе данных, полученных от пользователя. Сервер обрабатывает запрос и возвращает данные о времени начала и конца занятия, номере занятия, дисциплине, виде занятия, преподавателе, его степени, корпусе и аудитории, веб-сайт получает данные и выводит их на веб-странице, которую может просмотреть конечный адресат - студент.

Предоставление расписания студенту

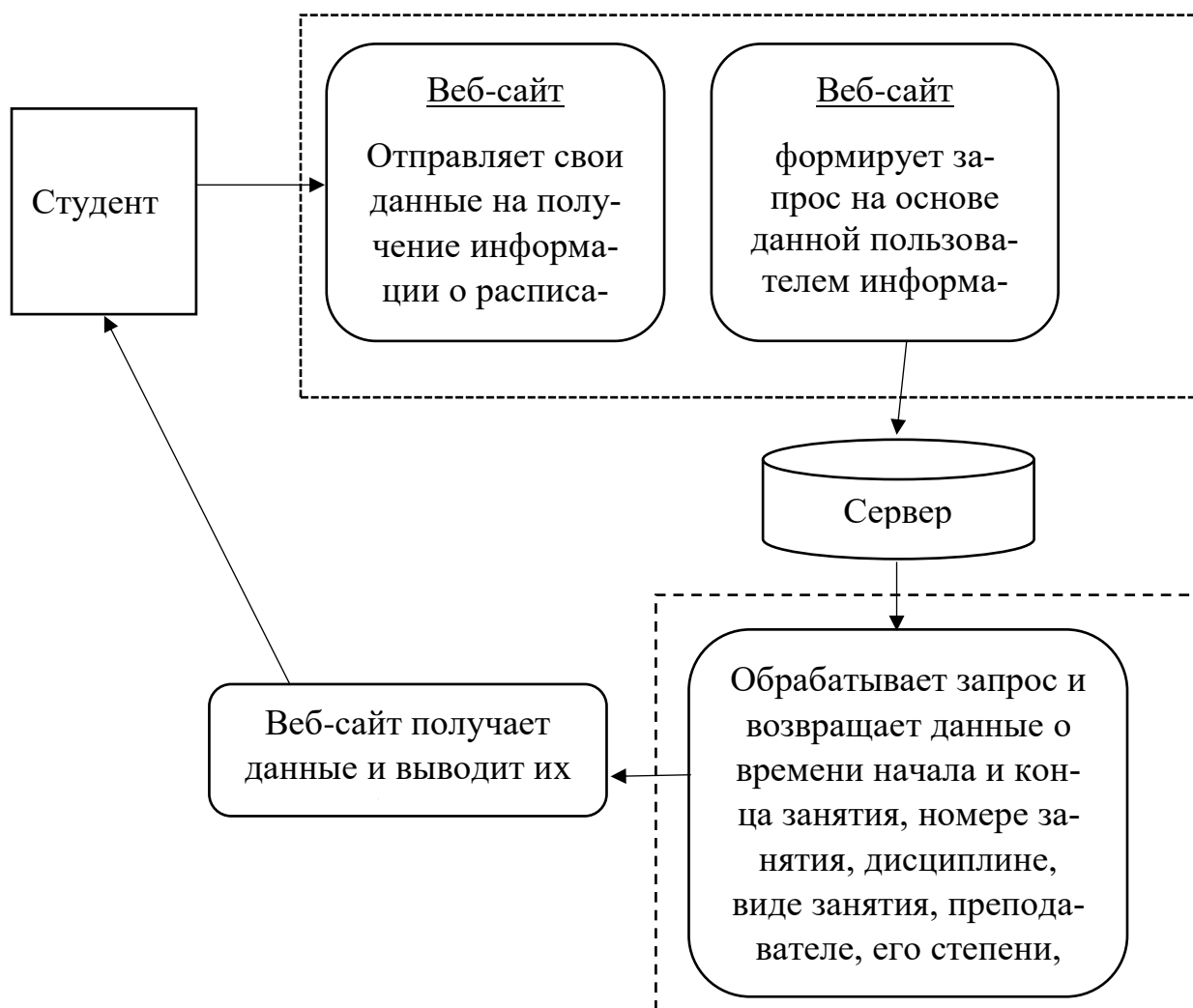


Рисунок 2.7 - Бизнес-процесс «Предоставление расписания студенту»

Предоставление расписания преподавателю

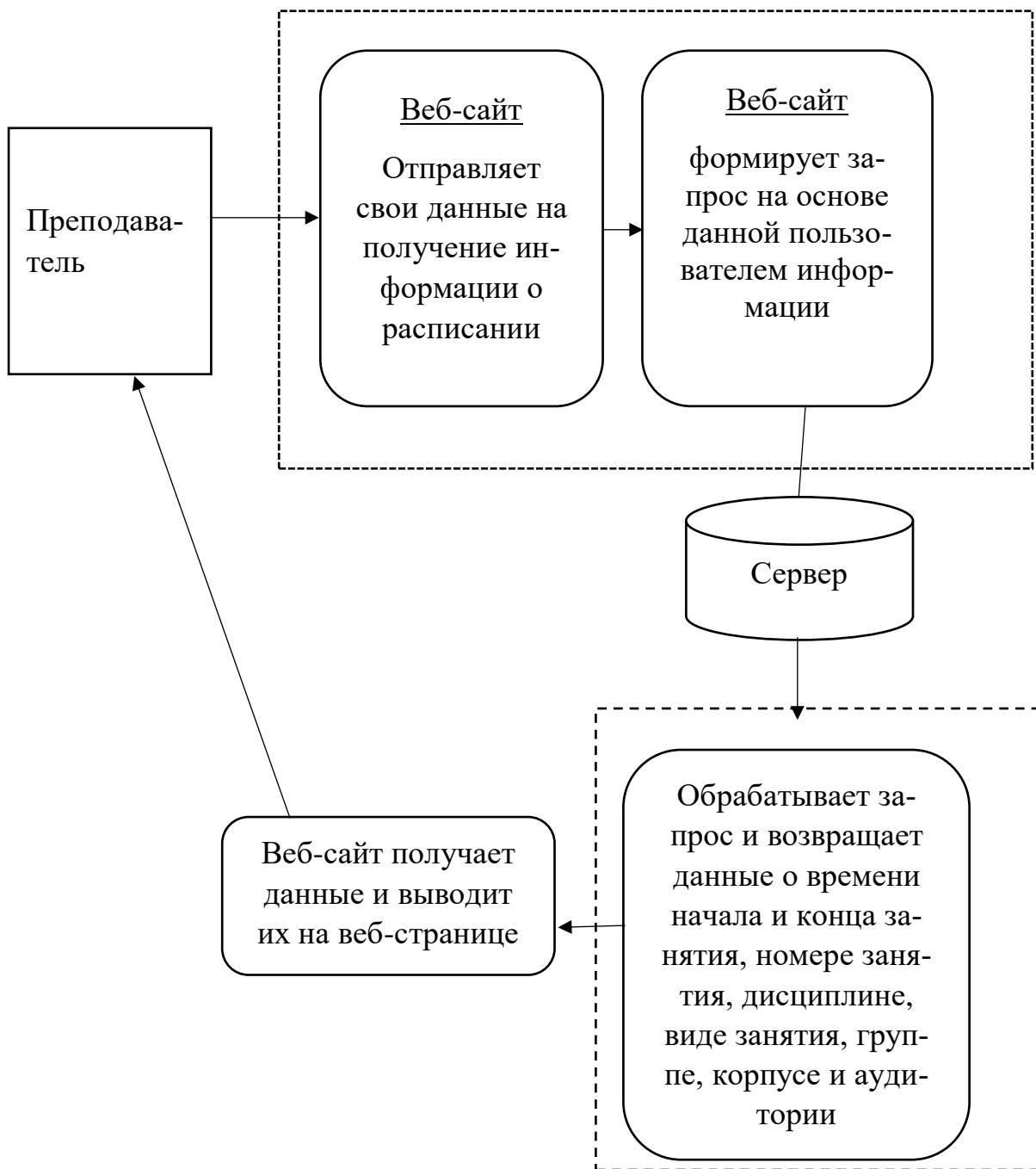


Рисунок 2.8 - Бизнес-процесс «Предоставление расписания преподавателю»

2.2.4 Анализ и оптимизация бизнес-процессов

Необходимо провести анализ выявленных бизнес-процессов, провести оптимизацию бизнес-процессов и составить их схемы.

Таблица 2.1 - Бизнес-процессы системы

Произведем оценку критических факторов успеха работы подсистем. Бизнес-процессы	Эффективное использовать всех функций смартфона	Автономный режим функционирования	Удобство отображения	Совместимость (поддержка ОС)
Предоставление расписания студенту	5	3	4	5
Предоставление расписания преподавателю	5	3	4	5
Обмен сообщениями между студентами	0	0	0	0

Как видно из таблицы автономный режим функционирования доступен только при кэшировании страниц, удобство отображения также имеет недостатки, поскольку адаптивный дизайн фактически урезает и уменьшает отображаемый контент, однако адаптивный веб-сайт имеет более широкий охват аудитории из-за своей совместимости со всеми мобильными ОС. Также имеется отсутствие обмена сообщениями между пользователями.

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

Предоставление расписания студенту



Рисунок 2.9 - Бизнес-процесс «Предоставление расписания студенту»

Предоставление расписания преподавателю

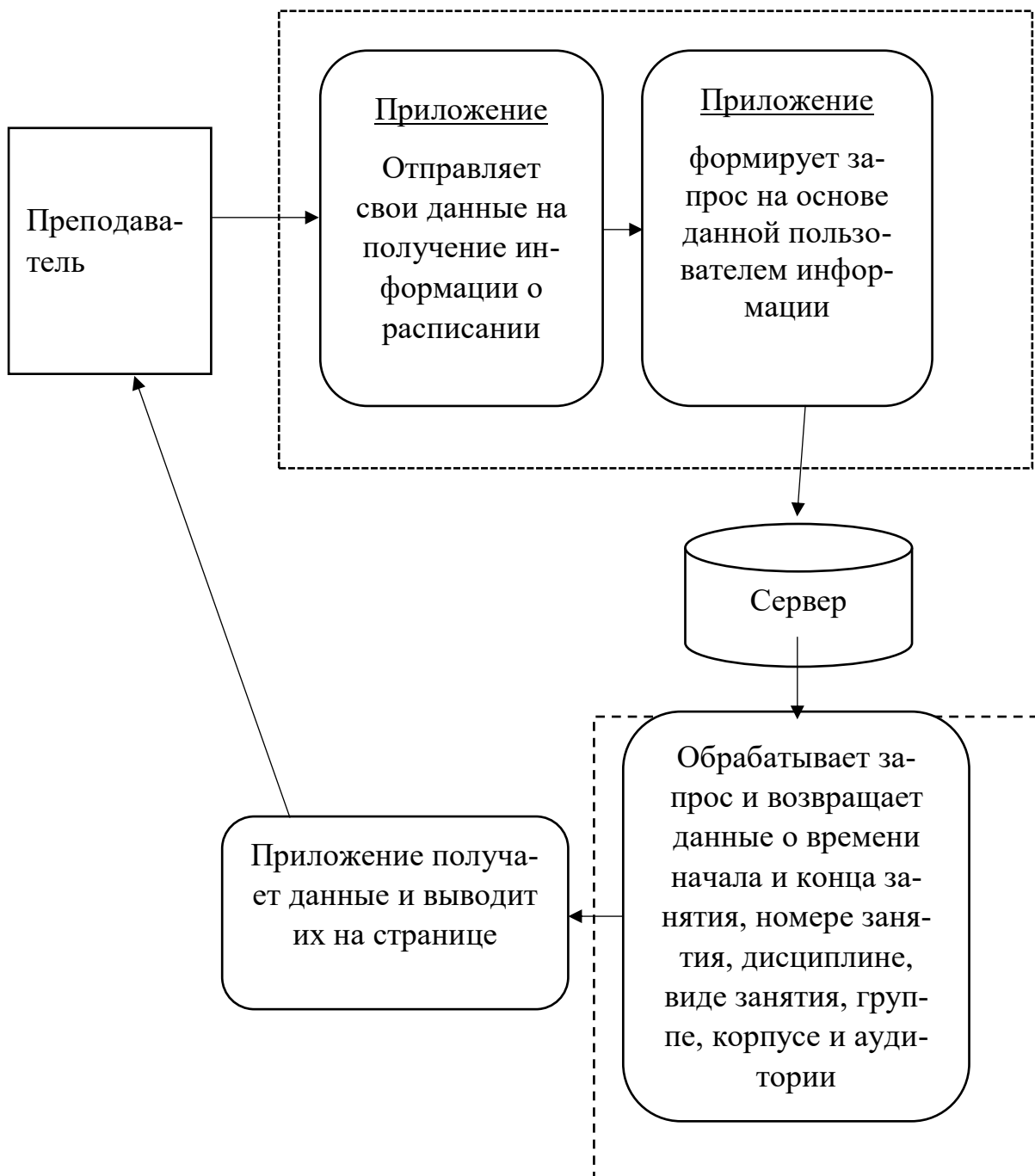


Рисунок 2.10 - Бизнес-процесс «Предоставление расписания преподавателю»

2.2.5 Разработка модели бизнес-объектов

Разработаем также модель бизнес-объекта «Ответ на запрос предоставления информации о расписании», в которой будут иметься внешний, внутренний исполнитель и бизнес-сущность.

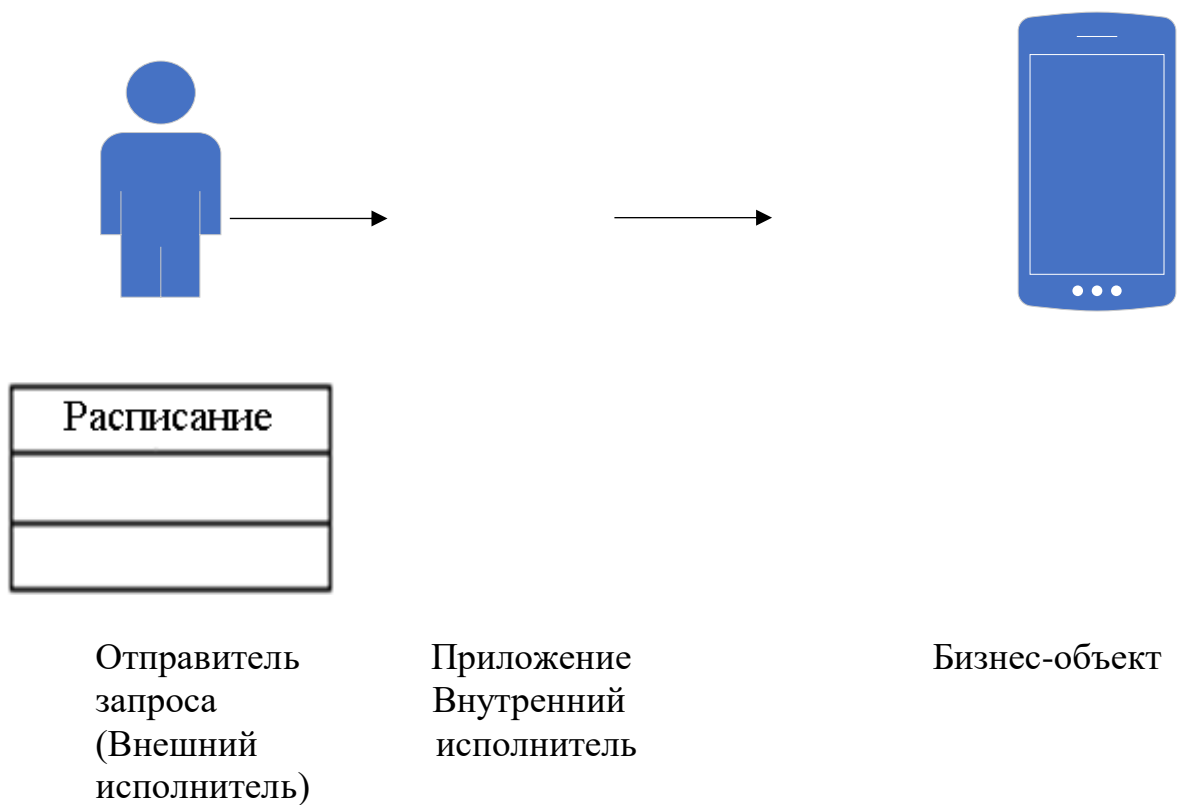
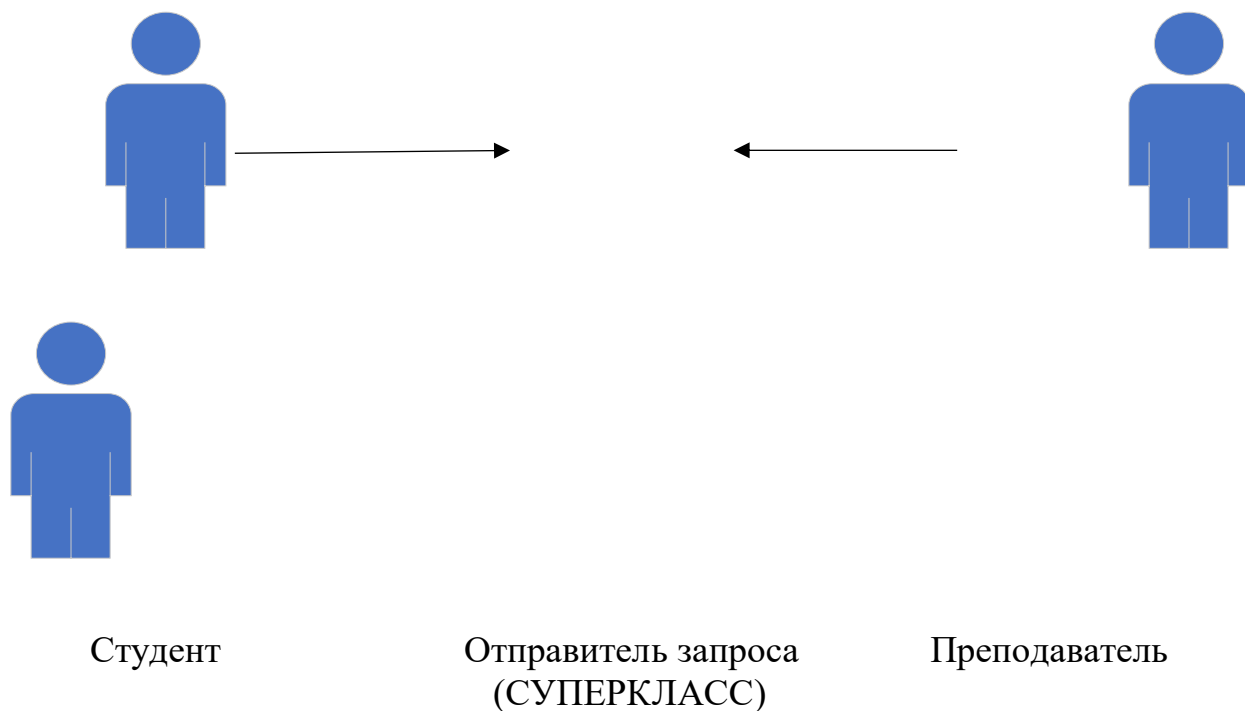


Рисунок 2.11 - Модель бизнес-объекта «Ответ на запрос предоставления информации о расписании»

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



2.3 Описание информационного обеспечения

2.3.1 Обоснование выбора СУБД

При разработке информационного обеспечения необходимо выбрать такую СУБД, которая отвечала бы всем требованиям данного приложения. Для этого проведем краткий анализ существующих СУБД [8].

Рассмотрим достоинства и недостатки СУБД, например, Oracle.

Достоинства [8]:

- удовлетворенность клиентов: одно из преимуществ использования баз данных Oracle связано с удовлетворенностью клиентов. Например, все базы данных Oracle обратно совместимы. Это позволяет предприятиям модернизировать свои системы без полного пересмотра своей системы баз данных. Это обеспечивает эффективные и недорогие обновления. Кроме того, новые версии баз данных Oracle предоставляют новые функции, сохраняя при этом популярные функции более старых версий. Это гарантирует, что их продукт основан на функции клиента, а не на том, что является экономически эффективным для Oracle;

- функциональность: базы данных Oracle используются практически для всех приложений корпоративного уровня. Таким образом, базы данных Oracle доминируют в банковской сфере. Десять ведущих банков мира используют базы данных Oracle для ведения своего бизнеса. Это связано прежде всего с функциональностью Oracle. Они обеспечивают сочетание технологий высокого уровня и интегрированных бизнес-решений. Это идеально подходит практически для всех корпораций, которые имеют огромные объемы данных для хранения и доступа;

- надежность: еще одним важным преимуществом, предлагаемым базами данных Oracle, является их надежность. Oracle - это база данных, которая обеспечивает отличную производительность при выполнении сложных задач. Тест ACID, который является важным инструментом, используемым для обеспечения целостности хранимых данных, был легко пройден базами данных Oracle. Этот тест важен, поскольку надежное хранение данных является основным назначением базы данных. Таким образом, доказано, что базы данных Oracle обеспечивают высокую целостность хранения данных;

- многосторонность: Oracle SQL дает вам гибкость выбора для запуска вашей базы данных в любой операционной системе.

Недостатки [8]:

- стоимость: затраты на использование Oracle SQL ставят его в невыгодное положение по сравнению с другими версиями SQL. Разница в стоимости начинается с лицензионных сборов, которые могут быть в десять раз выше для Oracle. Поскольку Oracle SQL является сложным и узкоспециализированным языком, может быть сложно найти квалифицированных админи-

страторов баз данных для запуска вашей системы. Когда вы найдете подходящего кандидата, ожидайте, что вы будете платить в два-три раза больше, чем вы платите сравнимому администратору Microsoft или PostgreSQL;

- сложность: Oracle SQL также сложнее в освоении и эксплуатации, чем его конкуренты. Удобная для пользователя версия, такая как SQL Server, проста в установке и настройке с минимальными настройками. Он включает в себя пять примеров баз данных, которые вы можете использовать в качестве основы для информации вашей собственной компании. Microsoft также предоставляет инструменты управления SQL Server на основе Visual Studio, которые помогут вам изменить структуру базы данных, изменить содержащиеся в ней данные и выполнить запросы.

Теперь нужно отметить достоинства и недостатки другой популярной СУБД - MySQL [9]:

Достоинства:

- простота использования: MySQL очень прост в установке, и благодаря множеству сторонних инструментов, которые могут быть добавлены в базу данных, настройка MySQL является относительно простой задачей. Кроме того, это также простая база данных для работы;

- поддержка: MySQL начиналась как платформа с открытым исходным кодом, это значит, что существует большое сообщество разработчиков и энтузиастов, к которому можно обратиться за помощью. Во многом это связано с популярностью решения, конечным результатом которого является нехватка специалистов;

- стоимость: в зависимости от того, для чего вы планируете его использовать, реализация MySQL может варьироваться в цене от бесплатной до 10000 долларов. В любом случае, это значительно дешевле, чем большинство других вариантов баз данных на рынке (за исключением конкурентов MySQL с открытым исходным кодом).

Недостатки [10]:

- стабильность: MySQL имеет тенденцию быть несколько менее надежным, чем его аналоги. Эти проблемы со стабильностью связаны с тем, как он обрабатывает определенные функции (например, ссылки, транзакции и аудит). Хотя база данных, безусловно, все еще пригодна для использования в свете этих проблем, она, как правило, делает MySQL плохим выбором для определенных случаев использования;

- производительность: хотя MySQL оснащен для обработки практически неограниченного объема данных, он имеет тревожную тенденцию останавливаться, если вынужден выполнять слишком много операций в данный момент времени;

Также стоит рассмотреть такую СУБД как Microsoft SQL Server, которая также является очень популярной в IT сфере:

Достоинства [10]:

- отличная поддержка восстановления данных: поврежденные данные всегда являются проблемой, когда происходит сбой питания или неправиль-

ное отключение. Microsoft SQL Server имеет ряд функций, которые способствуют восстановлению и восстановлению данных. Хотя отдельные таблицы не могут быть зарезервированы или восстановлены, доступны полные варианты восстановления базы данных.

Недостатки [10]:

- стоимость: одним из основных недостатков использования Microsoft SQL Server вместо альтернативной системы управления реляционными базами данных является то, что варианты лицензирования довольно дороги. Хотя использование программного обеспечения для разработки или в образовательных целях является бесплатным, любое деловое использование требует лицензионного сбора.

- ограниченная совместимость: Microsoft SQL Server предназначен только для работы на серверах под управлением Windows. По разным причинам, включая стоимость лицензий и проблемы безопасности, разработчики могут разместить свои сайты на компьютерах с Unix.

Учитывая все вышеперечисленные достоинства и недостатки рассмотренных СУБД, самым оптимальным вариантом будет выбор СУБД MySQL.

2.3.2 UML - диаграмма

UML, сокращенно Unified Modeling Language, является стандартизированным языком моделирования, состоящим из интегрированного набора диаграмм, разработанных, чтобы помочь разработчикам систем и программного обеспечения в определении, визуализации, конструировании и документировании артефактов программных систем, а также для бизнес-моделирования и другие не программные системы. UML представляет собой набор лучших инженерных практик, которые доказали свою эффективность в моделировании больших и сложных систем. UML является очень важной частью разработки объектно-ориентированного программного обеспечения и процесса разработки программного обеспечения. UML использует в основном графические обозначения для выражения дизайна программных проектов. Использование UML помогает проектным группам общаться, изучать потенциальные проекты и проверять архитектурный дизайн программного обеспечения [11].

Цель UML - предоставить стандартную нотацию, которая может использоваться всеми объектно-ориентированными методами, а также выбрать и интегрировать лучшие элементы нотаций-предшественников. UML был разработан для широкого спектра приложений. Следовательно, он предоставляет конструкции для широкого спектра систем и видов деятельности (например, распределенных систем, анализа, проектирования и развертывания систем) [11].

2.3.2.1 Диаграмма вариантов использования

Модель варианта использования описывает функциональные требования системы с точки зрения вариантов использования. Это модель предполагаемой функциональности системы (варианты использования) и ее среды (акторы). Варианты использования позволяют связать то, что вам нужно от системы, с тем, как система удовлетворяет эти потребности [11].

Представить модель вариантов использования можно как меню, очень похожее на меню, которое есть в ресторане. Просматривая меню, можно узнать, что в меню доступно, отдельные блюда, а также их цены. Также можно узнать, какую кухню предлагает ресторан: итальянская, мексиканская, китайская и так далее. Посмотрев на меню, можно также получить общее впечатление от ужина, который ждет вас в этом ресторане. По сути, меню «моделирует» поведение ресторана. [11]

Поскольку это очень мощный инструмент планирования, модель вариантов использования обычно используется на всех этапах цикла разработки всеми членами команды.

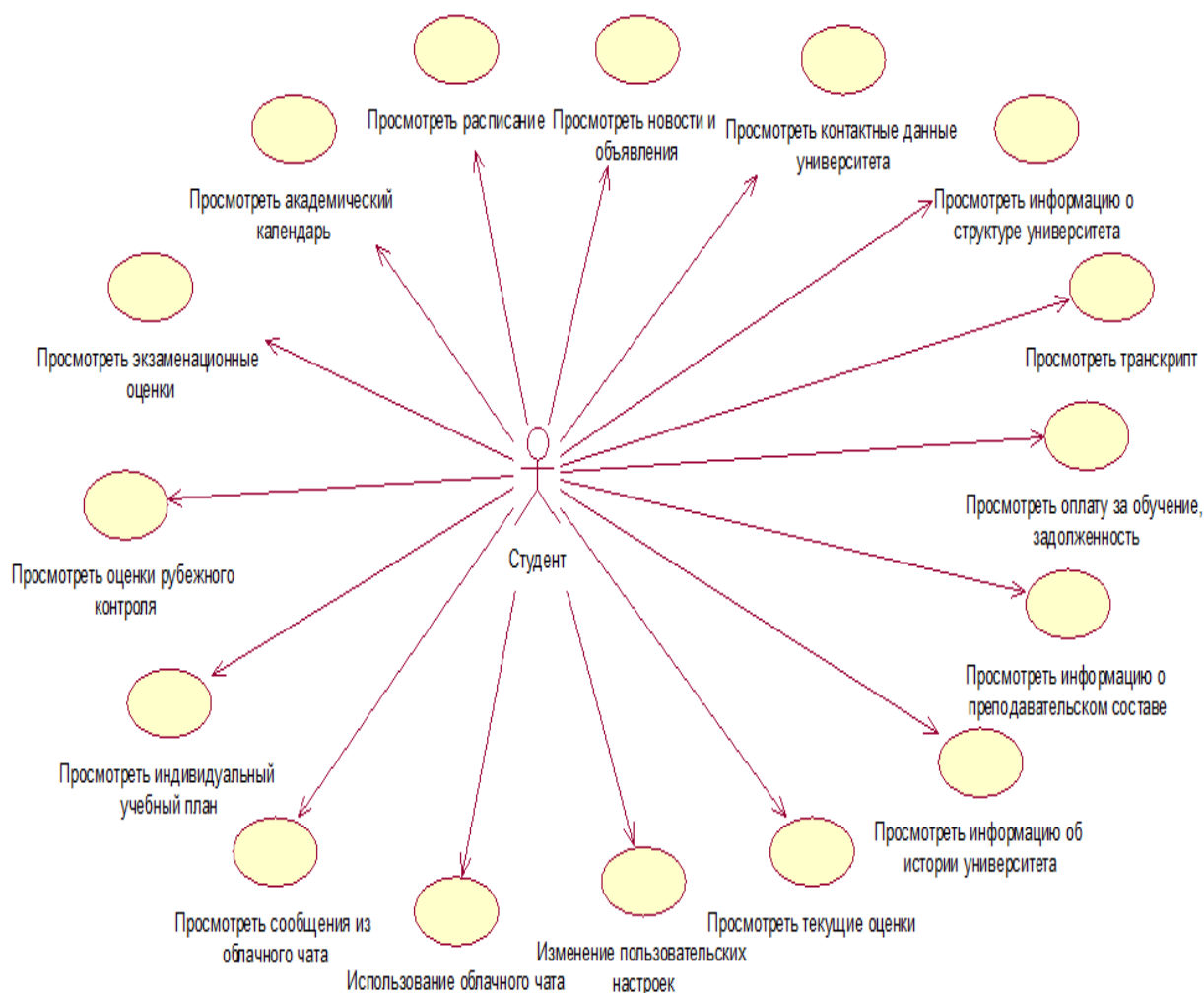


Рисунок 2.13 - Диаграмма вариантов использования приложения «AUES Mobile» для актора «Студент»

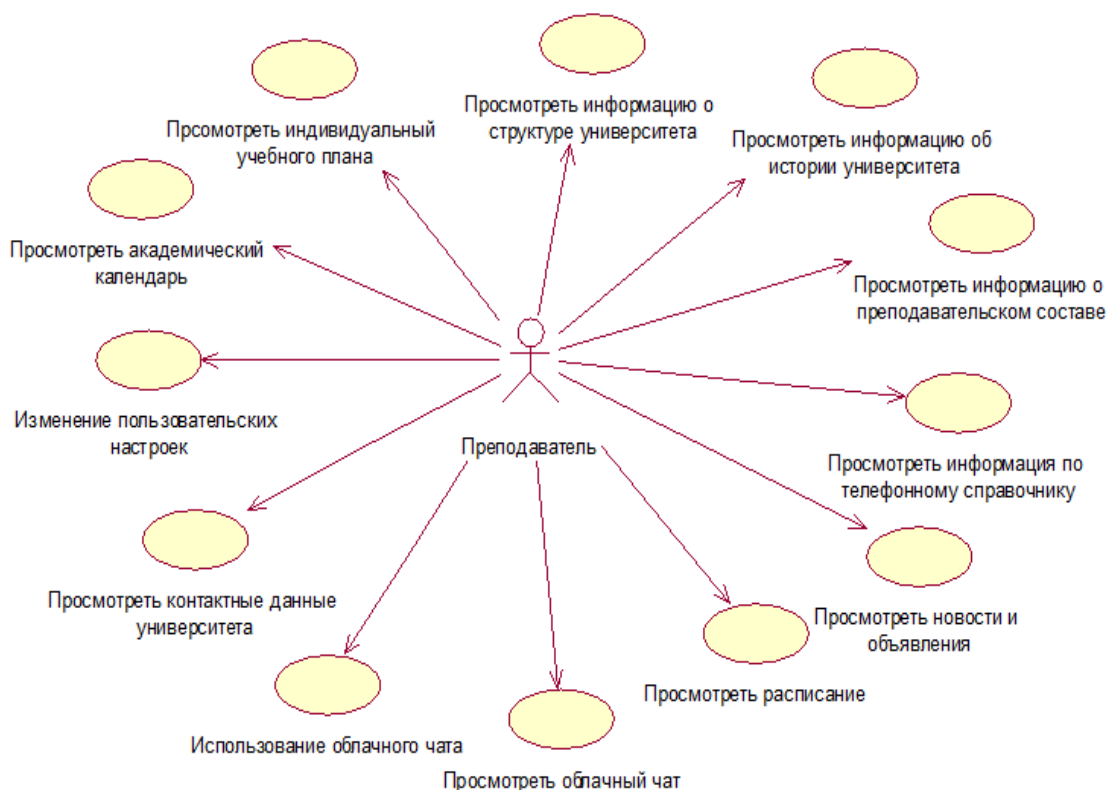


Рисунок 2.14 - Диаграмма вариантов использования приложения «AUES Mobile» для актора «Преподаватель»

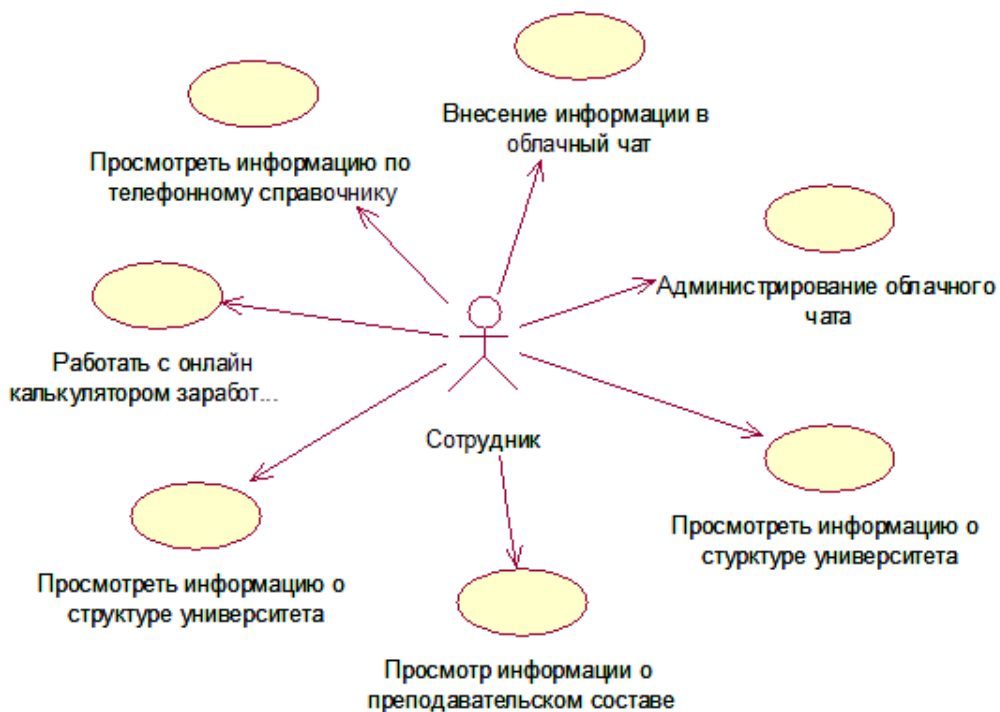


Рисунок 2.15 - Диаграмма вариантов использования приложения «AUES Mobile» для актора «Сотрудник»

2.3.2.2 Диаграмма последовательностей

Диаграмма последовательности просто отображает взаимодействие между объектами в последовательном порядке, то есть в порядке, в котором эти взаимодействия происходят. Мы также можем использовать термины диаграммы событий или сценарии событий для ссылки на диаграмму последовательности. Диаграммы последовательности описывают, как и в каком порядке объекты в системной функции. Эти диаграммы широко используются бизнесменами и разработчиками программного обеспечения для документирования и понимания требований к новым и существующим системам [11].

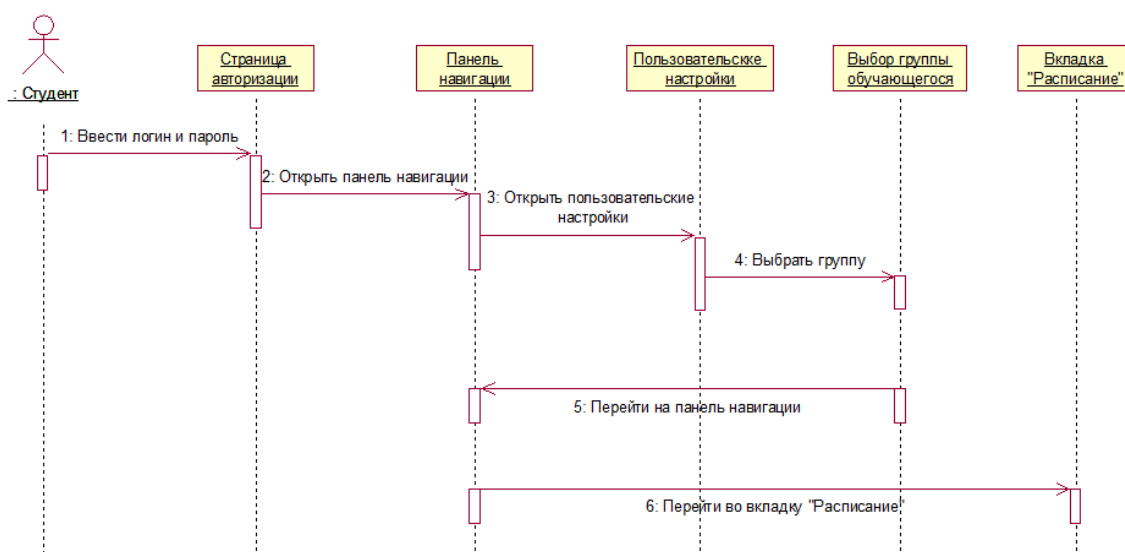


Рисунок 2.16 - Диаграмма последовательностей приложения «AUES Mobile» подсистемы «Расписание» для актора «Студент»

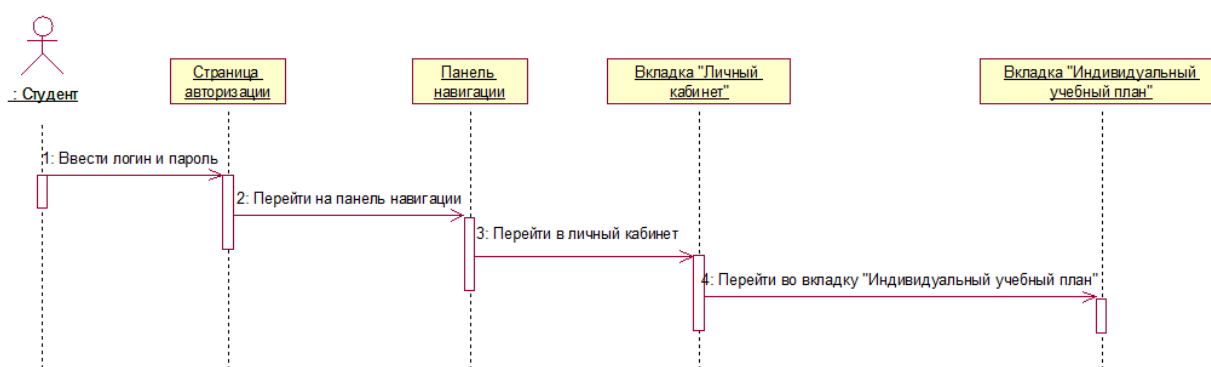


Рисунок 2.17 - Диаграмма последовательностей приложения «AUES Mobile» подсистемы «Индивидуальный учебный план» для актора «Студент»

2.3.2.3 Диаграмма коммуникации

Диаграммы коммуникации UML, как и диаграммы последовательности, показывают, как взаимодействуют объекты. Диаграмма связи - это расширение диаграммы объектов, которая показывает объекты вместе с сообщениями, которые передаются от одного к другому. В дополнение к ассоциациям между объектами, диаграмма связи показывает сообщения, которые объекты посылают друг другу [11]

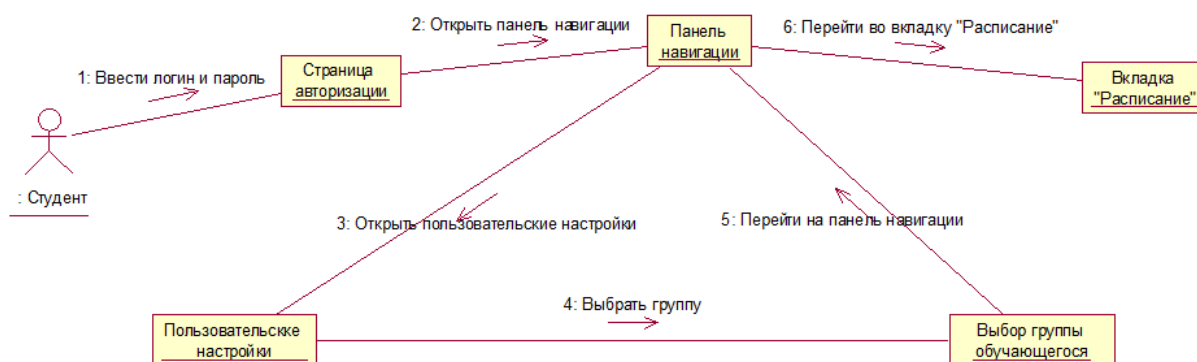


Рисунок 2.18 - Диаграмма коммуникации приложения «AUES Mobile» подсистемы «Расписание» для актора «Студент»

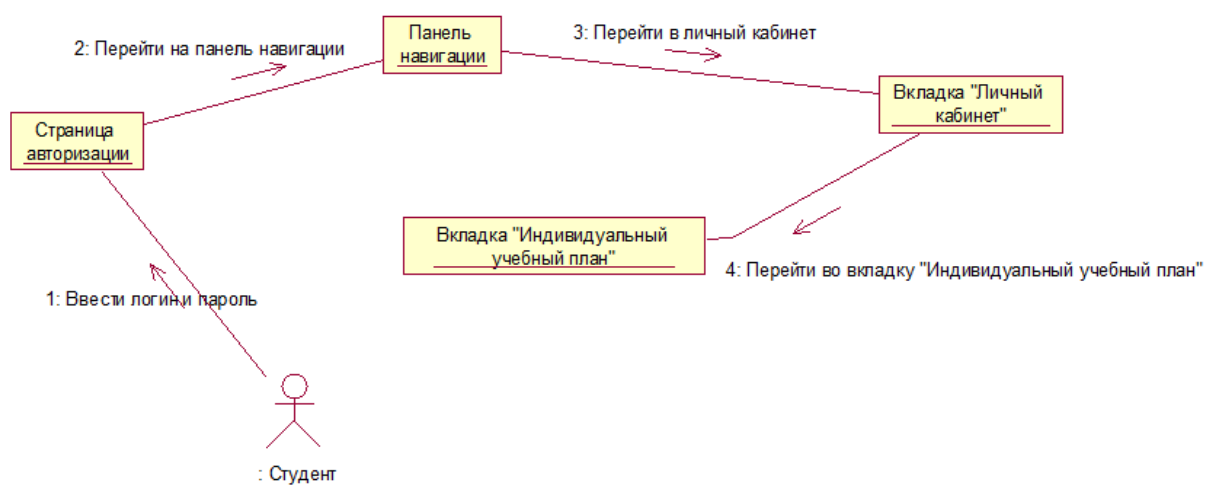


Рисунок 2.19 - Диаграмма коммуникации приложения «AUES Mobile» подсистемы «Индивидуальный учебный план» для актора «Студент»

2.3.3 Модели базы данных подсистем информационной системы приложения

На основе анализа предметной области были выделены сущности:

- преподаватель;
- дисциплина;
- студент;
- степень;
- группа;
- занятие;
- аудитория;
- день;
- время;
- преподаватель_занятие;
- расписание.

На основе этих данных при помощи программы Data Modeler была построена логическая модель, которая представлена на рисунке 2.21.

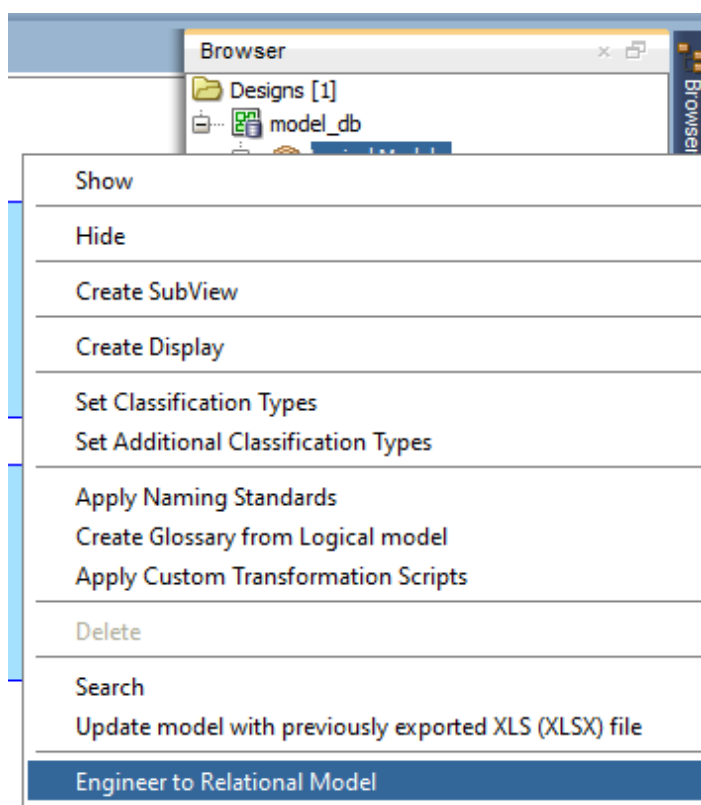


Рисунок 2.20 - Функция Engineer to Relational Model

Исходя из логической модели при помощи инструмента строится реляционная модель БД, которая показана на рисунке 2.22 ниже.

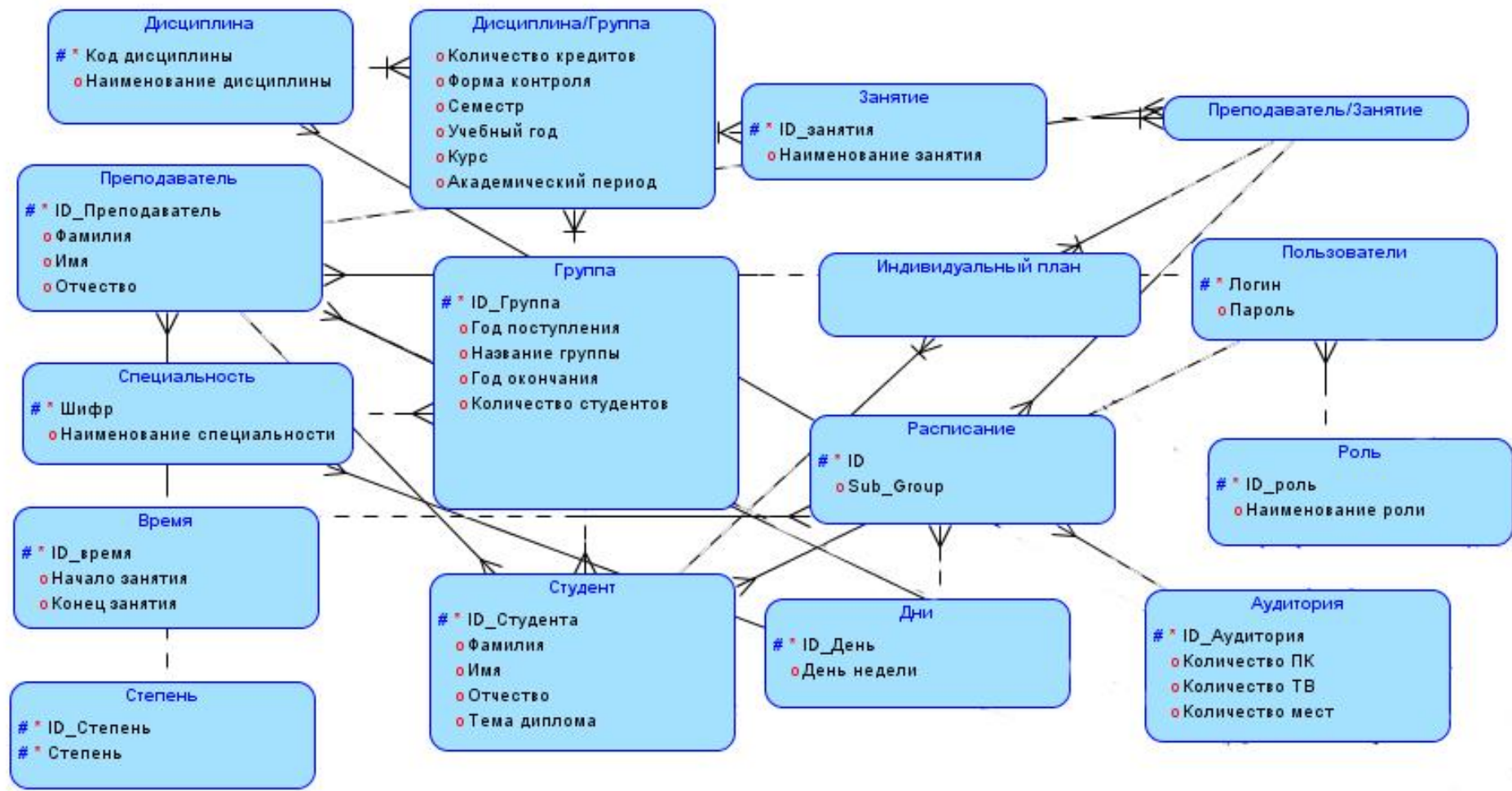


Рисунок 2.21 - Логическая модель базы данных подсистемы «Расписание»

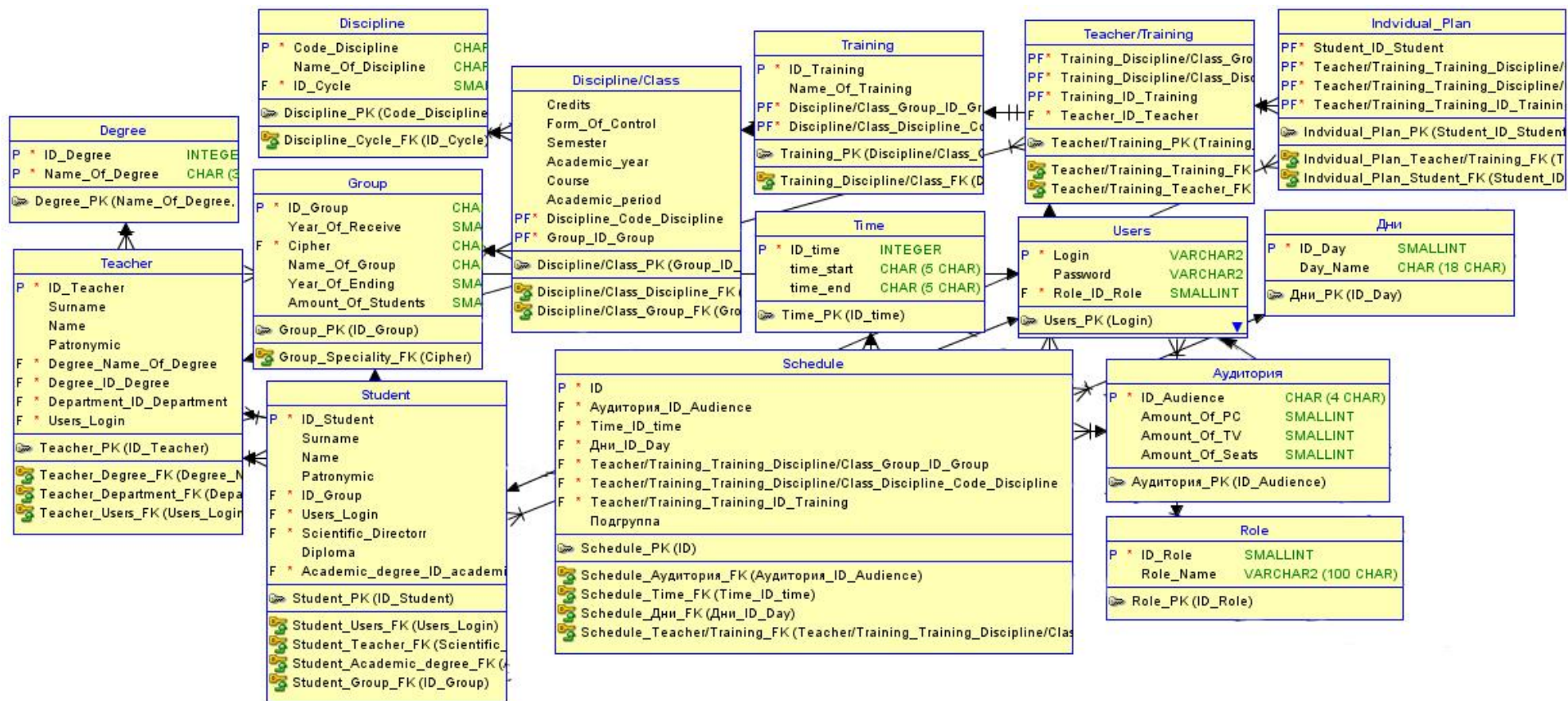


Рисунок 2.22 - Реляционная модель базы данных подсистемы «Расписание»

2.3.4 Структура таблиц БД

База данных подсистемы «Расписание» ИС состоит из 11 таблиц: TEACHER (ПРЕПОДАВАТЕЛЬ), DISCIPLINE (ДИСЦИПЛИНА), STUDENT (СТУДЕНТ), DEGREE (СТЕПЕНЬ), GROUP (ГРУППА), TRAINING (ЗАНЯТИЕ), AUDIENCE (АУДИТОРИЯ), DAY (ДЕНЬ), TIME (ВРЕМЯ), TEACHER_TRAINING (ПРЕПОДАВАТЕЛЬ_ЗАНЯТИЕ), SCHEDULE (РАСПИСАНИЕ).

Рассмотрим более подробно каждую таблицу, их значение и функции.

Таблица «TEACHER» («ПРЕПОДАВАТЕЛЬ») предназначена для хранения сведений о преподавателях, их ФИО, степени и кафедре, к которой они относятся. Ее структура приведена в таблице 2.2.

Таблица 2.2 - Структура таблицы «TEACHER» («ПРЕПОДАВАТЕЛЬ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Teacher	INTEGER		NOT NULL	Является первичным ключом, хранит идентификатор преподавателя
2	Surname	VARCHAR2	50	NOT NULL	Хранит фамилию преподавателя
3	Name	VARCHAR2	50	NOT NULL	Хранит имя преподавателя
4	Patronymic	VARCHAR2	50	NULL	Хранит отчество преподавателя
5	ID_Degree	INTEGER		NOT NULL	Является внешним ключом из таблицы «DEGREE», хранит идентификатор степени
6	ID_Department	SMALLINT		NOT NULL	Является внешним ключом из таблицы «DEPARTMENT», хранит идентификатор кафедры

Таблица «TRAINING» («ЗАНЯТИЕ») предназначена для хранения сведений о наименовании занятия и коде дисциплины, в качестве ключа служит идентификатор ID.

Таблица 2.3 - Структура таблицы «TRAINING» («ЗАНЯТИЕ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_training	SMALLINT		NOT NULL	Является первичным ключом, хранит идентификатор занятия
2	Name_Of_training	VARCHAR2	50	NOT NULL	Хранит наименование занятия
3	Code_discipline	VARCHAR2	50	NOT NULL	Является внешним ключом из таблицы «DISCIPLINE». Хранит информацию о коде дисциплины

Таблица «DISCIPLINE» («ДИСЦИПЛИНА») предназначена для хранения сведений о коде дисциплины, наименовании дисциплины, номере курса, об академическом периоде, количестве кредитов, а также сокращенное наименование дисциплины.

Таблица 2.4 - Структура таблицы «DISCIPLINE» («ДИСЦИПЛИНА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	Code_Discipline	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит код дисциплины
2	Name_Of_Discipline	VARCHAR2	200	NOT NULL	Хранит наименование дисциплины
3	Credit	NUMBER		NOT NULL	Хранит информацию о количестве кредитов
4	Course	NUMBER		NULL	Хранит информацию о номере курс
5	Academic_period	NUMBER		NOT NULL	Хранит информацию о номере семестра
6	ID_Cycle	NUMBER		NOT NULL	Является внешним ключом из таблицы «CYCLE», хранит идентификатор цикла дисциплины

Таблица «STUDENT» («СТУДЕНТ») предназначена для хранения сведений о номере студенческого билета, фамилии, имени, отчестве, а также наименовании группы конкретного студента.

Таблица 2.5 - Структура таблицы «STUDENT» («СТУДЕНТ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Student	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит номер студенческого билета
2	Surname	VARCHAR2	200	NOT NULL	Хранит информацию о фамилии студента
3	Name	NUMBER		NOT NULL	Хранит информацию об имени студента
4	Patronymic	NUMBER		NULL	Хранит информацию о отчестве студента
5	ID_Group	NUMBER		NOT NULL	Является внешним ключом из таблицы «GROUP», хранит информацию о наименовании группы

Таблица «TEACHER_TRAINING» («ПРЕПОДАВАТЕЛЬ_ЗАНЯТИЕ») предназначена для хранения сведений о преподавателе и группу, у которой он ведет определенную дисциплину, а также тип занятия. Структура таблицы приведена в таблице ниже.

Таблица 2.6 - Структура таблицы «TEACHER_TRAINING» («ПРЕПОДАВАТЕЛЬ_ЗАНЯТИЕ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Teacher	VARCHAR2	100	NOT NULL	Является внешним ключом из таблицы «TEACHER», хранит идентификатор преподавателя

Продолжение таблицы 2.6

2	ID_Group	VARCHAR2	200	NOT NULL	Является внешним ключом из таблицы «GROUP», хранит информацию о наименовании группы
3	ID_Training	NUMBER		NOT NULL	Является внешним ключом из таблицы «TRAINING», хранит идентификатор занятия
4	Code_Discipline	NUMBER		NULL	Является внешним ключом из таблицы «TRAINING», хранит код дисциплины

Таблица «AUDIENCE» («АУДИТОРИЯ») предназначена для хранения сведений о номере аудитории, количестве компьютеров, телевизоров, посадочных мест, а также типе аудитории.

Таблица 2.7 - Структура таблицы «AUDIENCE» («АУДИТОРИЯ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Audience	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит информацию о номере аудитории
2	Amount_Of_PC	VARCHAR2	200	NOT NULL	Хранит информацию о количестве персональных компьютеров
3	Amount_Of_TV	NUMBER		NOT NULL	Хранит информацию о количестве телевизоров

Продолжение таблицы 2.7

4	Amount_Of_Seats	NUMBER	100	NOT NULL	Хранит информацию о количестве посадочных мест
5	Audience_Type	VARCHAR2	50	NOT NULL	Хранит информацию о типе аудитории

Таблица «ДНИ» («DAY») предназначена для хранения сведений об идентификаторе дня недели, а также наименовании дня недели.

Таблица 2.8 - Структура таблицы «DAY» («ДНИ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Day	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит идентификатор дня недели
2	Day_Name	VARCHAR2	200	NOT NULL	Хранит информацию о наименовании дня недели

Таблица «ВРЕМЯ» («TIME») предназначена для хранения сведений о номере занятия, времени начала и конца занятия.

Таблица 2.9 - Структура таблицы «TIME» («ВРЕМЯ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Time	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит информацию о номере занятия
2	Time_Start	VARCHAR2	200	NOT NULL	Хранит информацию о времени начала занятия

Продолжение таблицы 2.9

3	Time_End	NUMBER	100	NOT NULL	Хранит информацию о времени конца занятия
---	----------	--------	-----	----------	---

Таблица «CLASS» («ГРУППА») предназначена для хранения сведений о наименовании группы, годе поступления, шифра специальности, годе выпуска, а также количестве учащихся.

Таблица 2.10 - Структура таблицы «CLASS» («ГРУППА»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Group	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит идентификатор группы
2	Year_Of_Receive	VARCHAR2	200	NOT NULL	Хранит год поступления
3	Cipher	NUMBER	200	NOT NULL	Является внешним ключом из таблицы «SPECIALITY» Хранит информацию о шифре специальности
4	Name_Of_Group	NUMBER	200	NULL	Хранит информацию о наименовании группы
5	Year_Of_Ending	NUMBER	200	NOT NULL	Хранит информацию о годе выпуска
6	Amount_Of_Students	NUMBER	200	NOT NULL	Хранит информацию о количестве обучающихся в группе

Таблица «DEGREE» («СТЕПЕНЬ») предназначена для хранения сведений о идентификаторе, а также наименовании степени преподавателя.

Таблица 2.11 - Структура таблицы «DEGREE» («СТЕПЕНЬ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID_Degree	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит идентификатор степени
2	Name_Of_Degree	VARCHAR2	200	NOT NULL	Хранит наименование степени
3	Short_Name	NUMBER	100	NOT NULL	Хранит информацию о сокращенном наименовании степени

Таблица «SCHEDULE» («РАСПИСАНИЕ») предназначена для хранения сведений о номере аудитории, наименовании группы, наименовании занятия, времени начала и конца занятия, наименовании дня недели, коде дисциплины, номере подгруппы, ФИО преподавателя, его степени, а также о типе занятия.

Таблица 2.12 - Структура таблицы «SCHEDULE» («РАСПИСАНИЕ»)

№	Наименование поля	Тип поля	Размер	NULL	Назначение поля
1	ID	VARCHAR2	100	NOT NULL	Является первичным ключом, хранит авто инкремент
2	ID_Audience	VARCHAR2	200	NOT NULL	Является внешним ключом из таблицы «AUDIENCE», хранит информацию о номере аудитории
3	ID_Group	NUMBER	100	NOT NULL	Является внешним ключом таблицы «TEACHER_TRAINING», хранит информацию о наименовании группы
4	ID_Training	NUMBER	м	NULL	Является внешним ключом таблицы «TEACHER_TRAINING», хранит информацию об идентификаторе занятия

Продолжение таблицы 2.12

5	ID_Time	NUMBER	100	NOT NULL	Является внешним ключом таблицы «TIME», хранит информацию о номере занятия
6	ID_Day	NUMBER	100	NOT NULL	Является внешним ключом таблицы «DAY», хранит информацию об идентификаторе дня недели
7	Code_Discipline	NUMBER	100	NOT NULL	Является внешним ключом из таблицы «TEACHER_TRAINING». Хранит информацию о коде дисциплины
8	Sub_Group	NUMBER	100	NOT NULL	Хранит информацию о номере подгруппы
9	ID_Teacher	NUMBER	100	NOT NULL	Является внешним ключом из таблицы «TEACHER_TRAINING». Хранит информацию об идентификаторе преподавателя

2.3.5 Используемые триггеры и функции СУБД

Для полноценного функционирования информационной системы необходимо производить автоматизацию БД. Это подразумевает написание триггеров и процедур, где они необходимы. В данной работе используется хранящая процедура, которая рассчитывает GPA студента по некоторой дисциплине, принимает один числовой аргумент в виде итоговой оценке и вычисляет GPA. Структура процедуры показана на рисунке 2.23. В зависимости от введенного параметра, процедура простыми операторами if-else высчитывает значение GPA.

```

BEGIN
    DECLARE GPA DOUBLE;
    SET GPA = 0;

    IF starting_value >= 95 AND starting_value <=100
    THEN SET GPA = 4.0;
    ELSEIF starting_value >= 90 and starting_value <=94
    THEN SET GPA = 3.67;
    ELSEIF starting_value >=85 and starting_value <=89
    THEN SET GPA = 3.33;
    ELSEIF starting_value >=80 and starting_value <=84
    THEN SET GPA = 3.0;
    ELSEIF starting_value >=75 and starting_value <=79
    THEN SET GPA = 2.67;
    ELSEIF starting_value >=70 and starting_value <=74
    THEN SET GPA = 2.33;
    ELSEIF starting_value >=65 and starting_value <=69
    THEN SET GPA = 2.0;
    ELSEIF starting_value >=60 and starting_value <=64
    THEN SET GPA = 1.67;
    ELSEIF starting_value >=55 and starting_value <=59
    THEN SET GPA = 1.33;
    ELSEIF starting_value >=50 and starting_value <=54
    THEN SET GPA = 1.0;
    ELSE SET GPA = 0;
    END IF;
    RETURN GPA;
END
    
```

Рисунок 2.23 - Структура процедуры GPACalc

На рисунке 2.24 предоставлена структура триггера, основная суть которого заключается в следующем: после того, как в индивидуальном плане появляется запись о дисциплине и закрепленном студенте с занятием - типа лекция, то в таблице итоговых оценок сразу создается запись с дисциплиной и студентов, где оценки выставлены автоматически - балл 0. Если же добавляется запись в индивидуальный план с типом семинар (практика) или лабораторные занятия, то программа смотрит, ведутся ли лекционные занятия у данной группы по данной дисциплине, если да, то триггер будет ожидать, когда появится запись с занятием типа лекция, если же не имеется, определённая запись появится в итоговых оценках.

Детали	
Название триггера	ADD_MARKS
Таблица	INDIVIDUAL_PLAN
Время	AFTER
Событие	INSERT
Определение	<pre> 1 BEGIN 2 IF (new.ID_TRAINING = 'L') THEN 3 INSERT INTO `FINAL_MARKS` (`MIDTERM1`, `MIDTERM2`, `EXAM`, `RATING`, `FINAL_MARK`, `ID_STUDENT`, `ID_GROUP`, `CODE_DISCIPLINE`, `ID_TRAINING`, `RGR_1`, `RGR_2`, `RGR_3`, `RGR_4`, `CR`, `SR_TEK`, `GPA`) VALUES ('0', '0', '0', '0', '0', new.ID_STUDENT, new.ID_GROUP, new.CODE_DISCIPLINE, NULL, '0', '0', '0', '0', '0', '0', '0'); 4 END IF; 5 SET @col = (SELECT COUNT(ID_TRAINING) FROM TEACHER_TRAINING WHERE ID_GROUP = new.ID_GROUP AND CODE_DISCIPLINE = new.CODE_DISCIPLINE AND ID_TRAINING = 'L'); 6 IF (@col = 0) THEN 7 INSERT INTO `FINAL_MARKS` (`MIDTERM1`, `MIDTERM2`, `EXAM`, `RATING`, `FINAL_MARK`, `ID_STUDENT`, `ID_GROUP`, `CODE_DISCIPLINE`, `ID_TRAINING`, `RGR_1`, `RGR_2`, `RGR_3`, `RGR_4`, `CR`, `SR_TEK`, `GPA`) VALUES ('0', '0', '0', '0', '0', new.ID_STUDENT, new.ID_GROUP, new.CODE_DISCIPLINE, NULL, '0', '0', '0', '0', '0', '0', '0'); 8 END IF; 9 END </pre>
Определитель	id11311399_aues@%

Рисунок 2.24 - Структура триггера ADD_MARKS

2.3.6 Realtime Database

При написании облачного чата были использованы средства Firebase Analytics Realtime Database. База данных Firebase Realtime - это база данных, размещенная в облаке, то есть она работает в облаке, и доступ к ней предоставляется как услуга. Она хранит данные в формате JSON (JavaScript Object Notation), формат для хранения или транспортировки данных. Все подключенные к нему пользователи могут получить доступ к данным в режиме реального времени. Особенности данного средства показаны на рисунке 2.25.



Рисунок 2.25 - Особенности Realtime Database

Как хранятся данные изображено на рисунке 2.26.

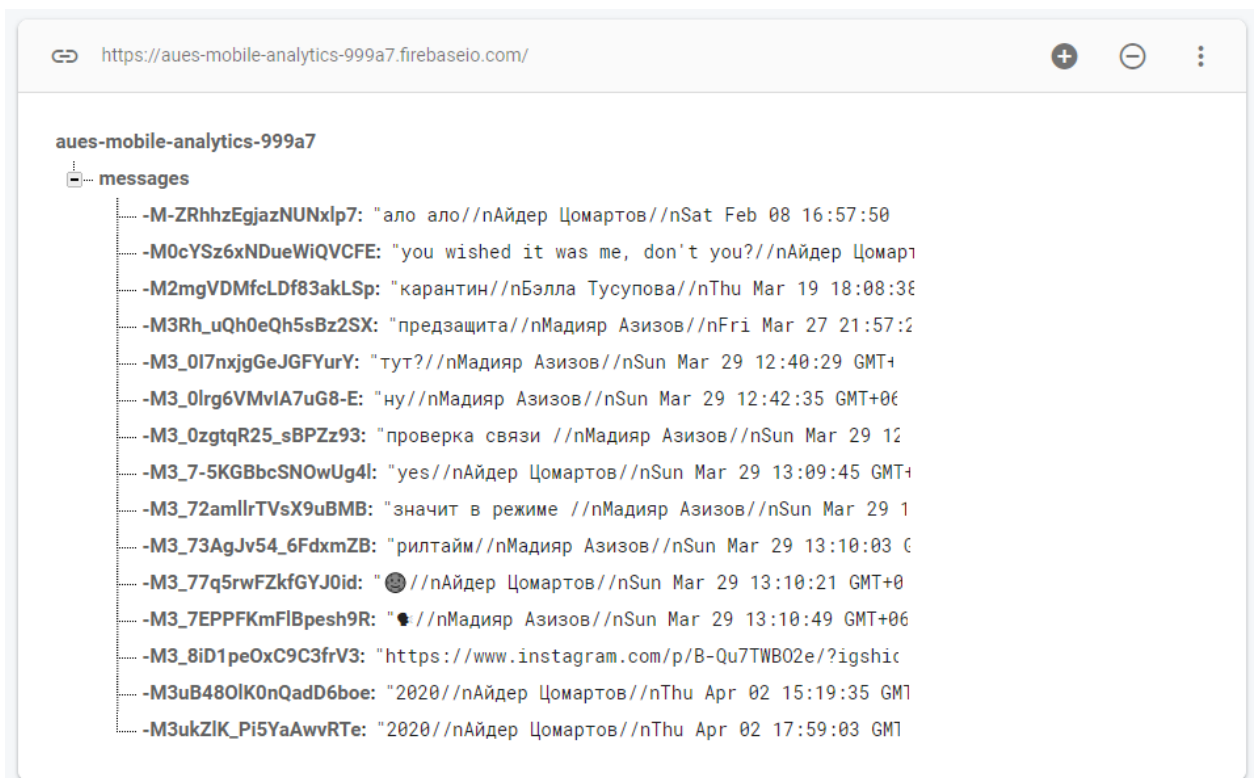


Рисунок 2.26 - Хранилище данных облачного чата

Также имеются аналитические данные по использованию облачного чата пользователями - рисунок 2.27.

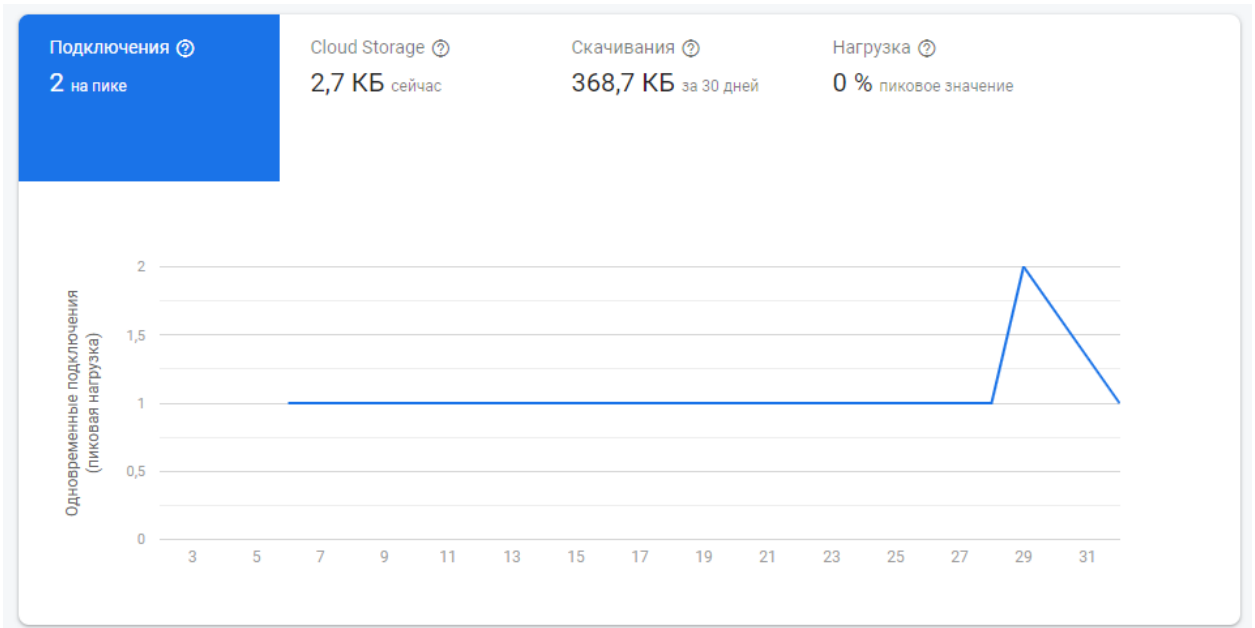


Рисунок 2.27 - Использование облачного чата

3 Разработка программного обеспечения

3.1 Обоснование выбора среды программирования

Язык XML

Графическая часть приложения написана на языке развертки XML, т.к он достаточно удобен в использовании и имеет возможность редактирования как в текстовом, так и в графическом виде. Основным преимуществом XML является то, что он предоставляет разработчикам инструмент, который кратко и однозначно определяет формат записей данных. Если компания решает, что в ее интересах, чтобы устройства, созданные другими компаниями, взаимодействовали со своими собственными устройствами, она может поддерживать эту возможность, публикуя свои схемы. Поскольку схемы интерпретируются программным обеспечением, человеческие ошибки в интерпретации устраняются [12].

XML не зависит от языка программирования и платформы, поэтому он поддерживает любые технологические изменения и используется в любом устройстве. Также он совместим с Java.

Основным недостатком XML является то, что его парсеры имеют тенденцию быть очень большими, хотя большой объем памяти парсеров XML может быть уменьшен до разумного размера за счет устранения ненужных функций. Так как Android-приложения пишутся в интегрированной среде разработки Android Studio, весь интерфейс пишется на языке XML [12].

Android Studio

Клиентская часть приложения «AUES Mobile» написана на языке Java при помощи среды разработки «Android Studio». Основанием для выбора языка программирования Java послужило то, что он является платформонезависимым, а также достаточно простым в использовании. Android Studio - это программное обеспечение, созданное самим Google для использования в Android разработке. Таким образом, несомненно, что его поддержка никогда не упадет, пока Android-приложения все еще разрабатываются [13].

Поскольку Google создан для использования в качестве средства разработки для Android, Google также очень легко добавляет отличную поддержку своих услуг в IDE и упрощает интеграцию служб Google в приложения для Android, созданные с помощью Studio.

Android Studio использует Gradle в качестве официального инструмента для сборки проектов, отходя от устаревшего Apache Ant.

Gradle - это мощный инструмент сборки, особенно для разработки под Android, с помощью которого очень легко делать вещи, которые в противном случае невозможно или очень трудно сделать в других системах сборки, например, обновить систему сборки, не нарушая сам проект, или позволяя вам отдельно определить версию разработки и производства [13].

Пожалуй, главным минусом является то, что Android Studio является официальной IDE для Android и прекрасно работает в качестве Android IDE.

К сожалению, это единственное, для чего он может использоваться. Если потребуется разрабатывать приложения для других платформ на Java, придется изучить другую среду разработки.

Одним из преимуществ Java в том, что он был спроектирован так, чтобы быть простым в использовании, и поэтому на нем легче писать, компилировать, отлаживать и изучать, чем в других языках программирования. Причина, по которой Java намного проще, чем C ++, заключается в том, что Java использует автоматическое выделение памяти и сборку мусора, в то время как C ++ требует от программиста выделять память и собирать мусор [13].

Java является объектно-ориентированным, потому что программирование на Java сосредоточено на создании объектов, управлении объектами и обеспечении совместной работы объектов. Это позволяет создавать модульные программы и повторно использовать код. Одним из наиболее значительных преимуществ Java является его способность легко переходить с одной компьютерной системы на другую [14].

Возможность запуска одной и той же программы на многих различных системах имеет решающее значение для программного обеспечения World Wide Web, и Java преуспевает в этом, будучи независимым от платформы как на исходном, так и на двоичном уровнях [14].

Главным недостатком является то, что Java работает значительно медленнее и потребляет больше памяти, чем нативно скомпилированные языки, такие как C или C ++.

База данных реализована при помощи СУБД MySQL на базе PHP MyAdmin, язык MySQL является бесплатной СУБД и достаточно простой в использовании, что и послужило основанием для выбора. Серверная часть написана на языке PHP, т. к. он совмещен с платформой PHP MyAdmin[15].

Теперь нужно отметить достоинства и недостатки другой популярной СУБД - MySQL [15]:

Достоинства [15]:

- простота использования: MySQL очень прост в установке, и благодаря множеству сторонних инструментов, которые могут быть добавлены в базу данных, настройка MySQL является относительно простой задачей. Кроме того, это также простая база данных для работы;

- поддержка: MySQL начиналась как платформа с открытым исходным кодом, это значит, что существует большое сообщество разработчиков и энтузиастов, к которому можно обратиться за помощью. Во многом это связано с популярностью решения, конечным результатом которого является нехватка специалистов [15];

- стоимость: в зависимости от того, для чего вы планируете его использовать, реализация MySQL может варьироваться в цене от бесплатной до 10000 долларов. В любом случае, это значительно дешевле, чем большинство других вариантов баз данных на рынке (за исключением конкурентов MySQL с открытым исходным кодом) [15].

- функциональность: базы данных MySQL используются практически для всех приложений корпоративного уровня. Таким образом, базы данных MySQL доминируют в банковской сфере. Десять ведущих банков мира используют базы данных MySQL для ведения своего бизнеса. Это связано прежде всего с функциональностью MySQL. Они обеспечивают сочетание технологий высокого уровня и интегрированных бизнес-решений. Это идеально подходит практически для всех корпораций, которые имеют огромные объемы данных для хранения и доступа [16];

Недостатки [16]:

- стабильность: MySQL имеет тенденцию быть несколько менее надежным, чем его аналоги. Эти проблемы со стабильностью связаны с тем, как он обрабатывает определенные функции (например, ссылки, транзакции и аудит). Хотя база данных, безусловно, все еще пригодна для использования в свете этих проблем, она, как правило, делает MySQL плохим выбором для определенных случаев использования [16];

- производительность: хотя MySQL оснащен для обработки практически неограниченного объема данных, он имеет тревожную тенденцию останавливаться, если вынужден выполнять слишком много операций в данный момент времени [16];

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` [17]. 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.

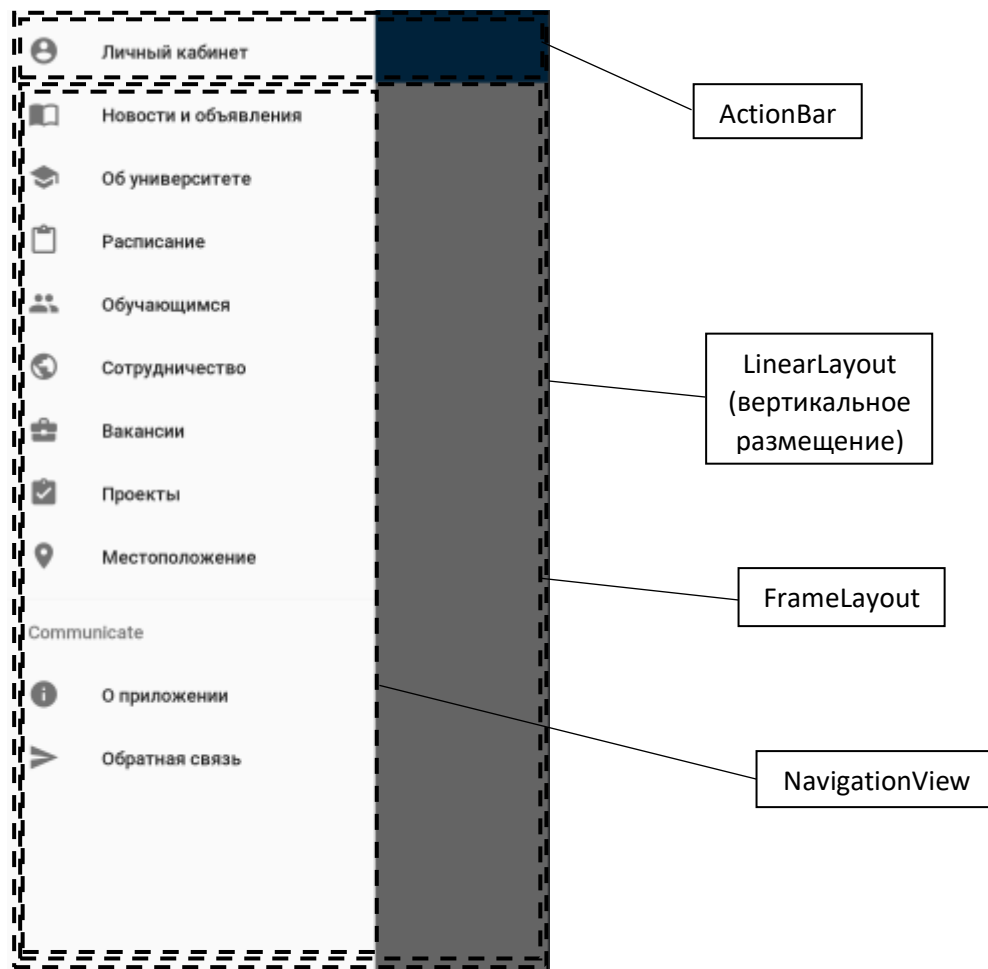


Рисунок 3.1 - Запланированное расположение виджетов MainActivity на экране

Далее определяются вышеназванные виджеты в файле activity_main.xml.

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

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">
```

```

<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical">

  <include
    layout="@layout/app_bar_main" />

  <FrameLayout
    android:id="@+id/fr_place"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
  </FrameLayout>

</LinearLayout>

<com.google.android.material.navigation.NavigationView
  android:id="@+id/nav_view"
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:layout_gravity="start"
  android:fitsSystemWindows="true"
  app:menu="@menu/activity_main_drawer" />

</androidx.drawerlayout.widget.DrawerLayout>

```

Данный модуль выполняет функцию позиционирования виджет-объектов на экране. Далее строится иерархия представлений для макета MainActivity, которая показана на рисунке 3.2.



Рисунок 3.2 - Иерархия представлений макета MainActivity

Ниже показаны основные атрибуты, которые используются в вышеперечисленных виджетах:

`-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 определен первым внутри корневого элемента и аналогично, в горизонтальном положении первый объявленный виджет будет крайним слева.

Как было сказано выше данный макет состоит из двух виджетов: `AppBarLayout` и `Toolbar`. `com.google.android.material.appbar.AppBarLayout` используется для обертывания `androidx.appcompat.widget.Toolbar`. Элемент `ToolBar` используется для быстрого и удобного доступа пользователя к часто используемым функциям.

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

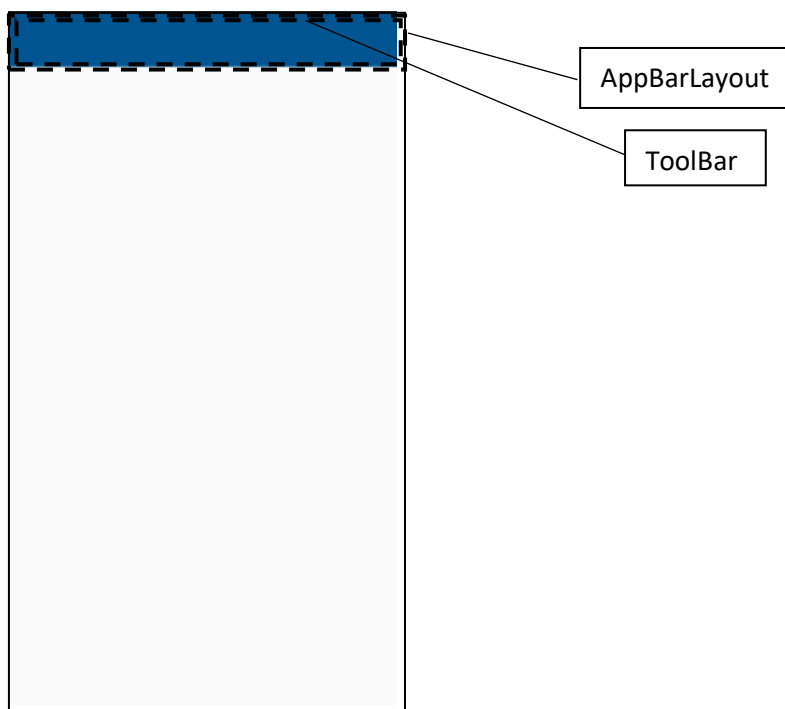


Рисунок 3.3 - Запланированное расположение виджетов AppBarLayout на экране

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.

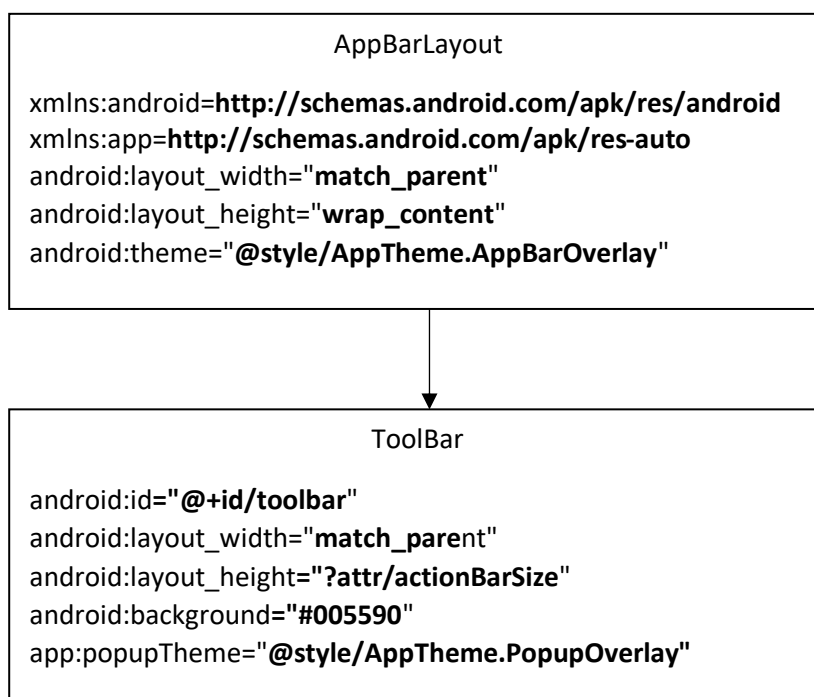


Рисунок 3.4 - Иерархия представлений макета ToolBar

3.2.2 Макеты подсистемы «Расписание»

3.2.2.1 Макет monday

В интерфейсе `fragment_monday` определено 5 виджетов:

- `RelativeLayout`;
- `RecyclerView`.

`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 [17].

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

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

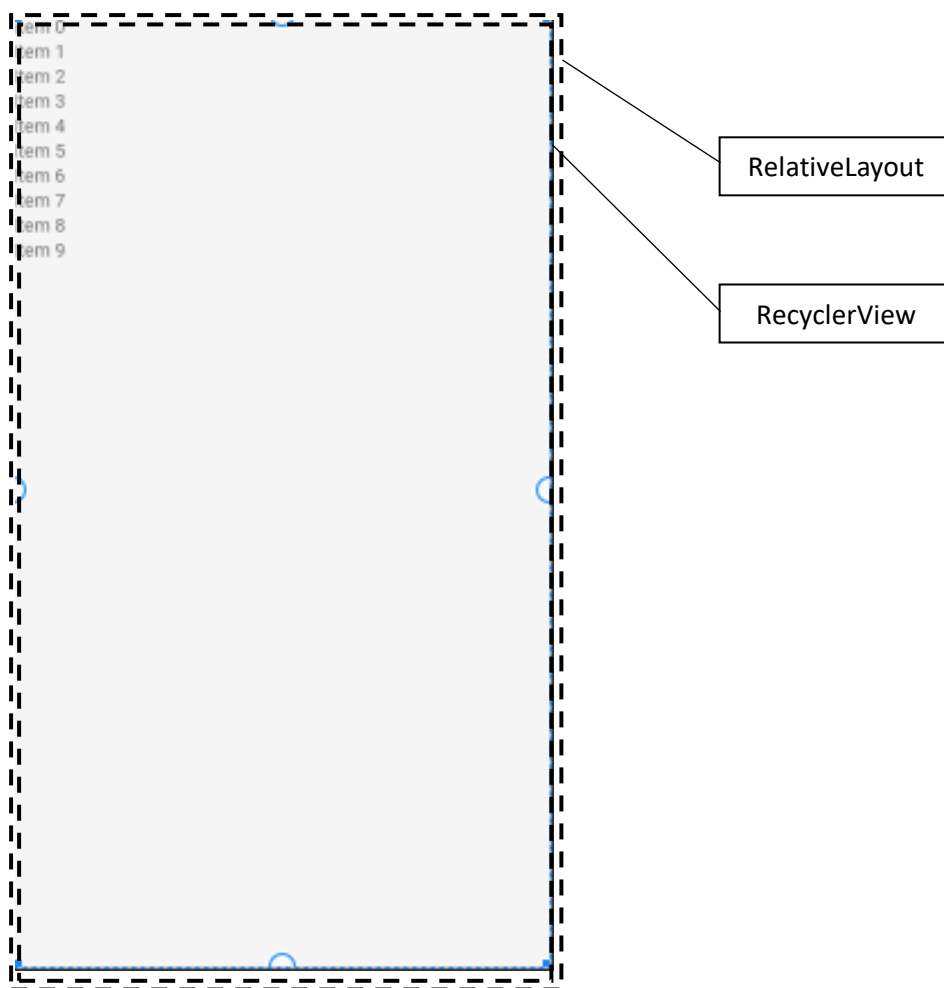


Рисунок 3.5 - Запланированное расположение виджетов `monday` на экране

Для определения виджетов в XML (fragment_monday.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=".monday">

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

    <TextView
        android:id="@+id/empty_day"
        android:textStyle="bold"
        android:textSize="24sp"
        android:textColor="#000"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="В этот день занятий нет"
        android:visibility="gone"
        android:gravity="center" >
    </TextView>

</RelativeLayout>
```

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

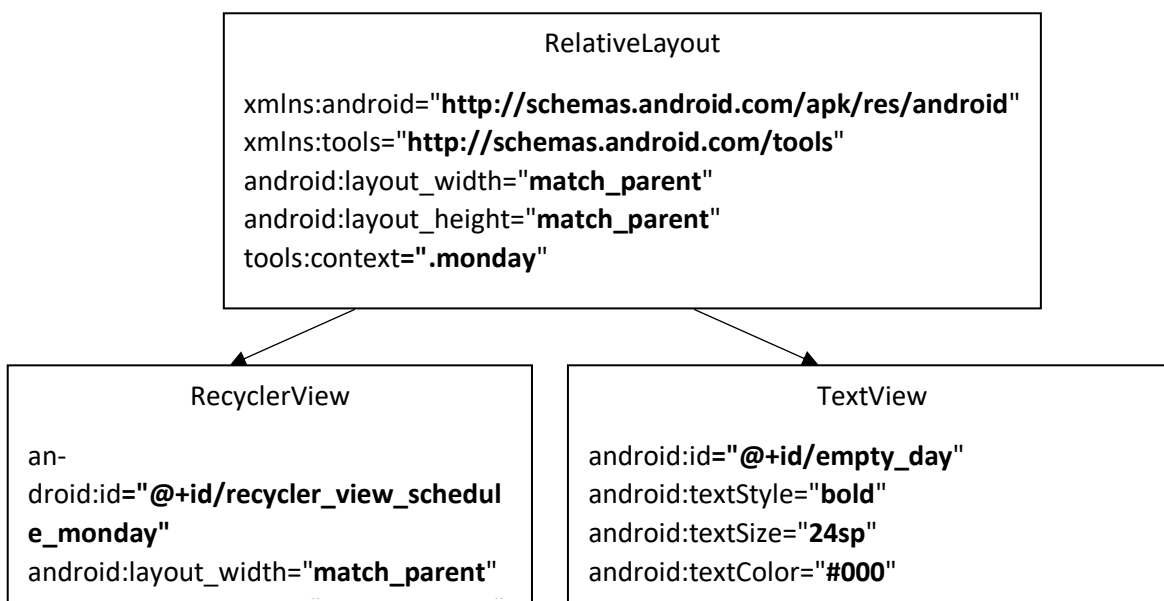


Рисунок 3.6 - Иерархия представлений макета monday

3.2.2.2 Макет Example_Item_Schedule

В данном интерфейсе определены следующие виджеты:

- CardView;
- LinearLayout;
- TextView;
- View.

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[18].

TextView исходя из названия служит для отображения текста.

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

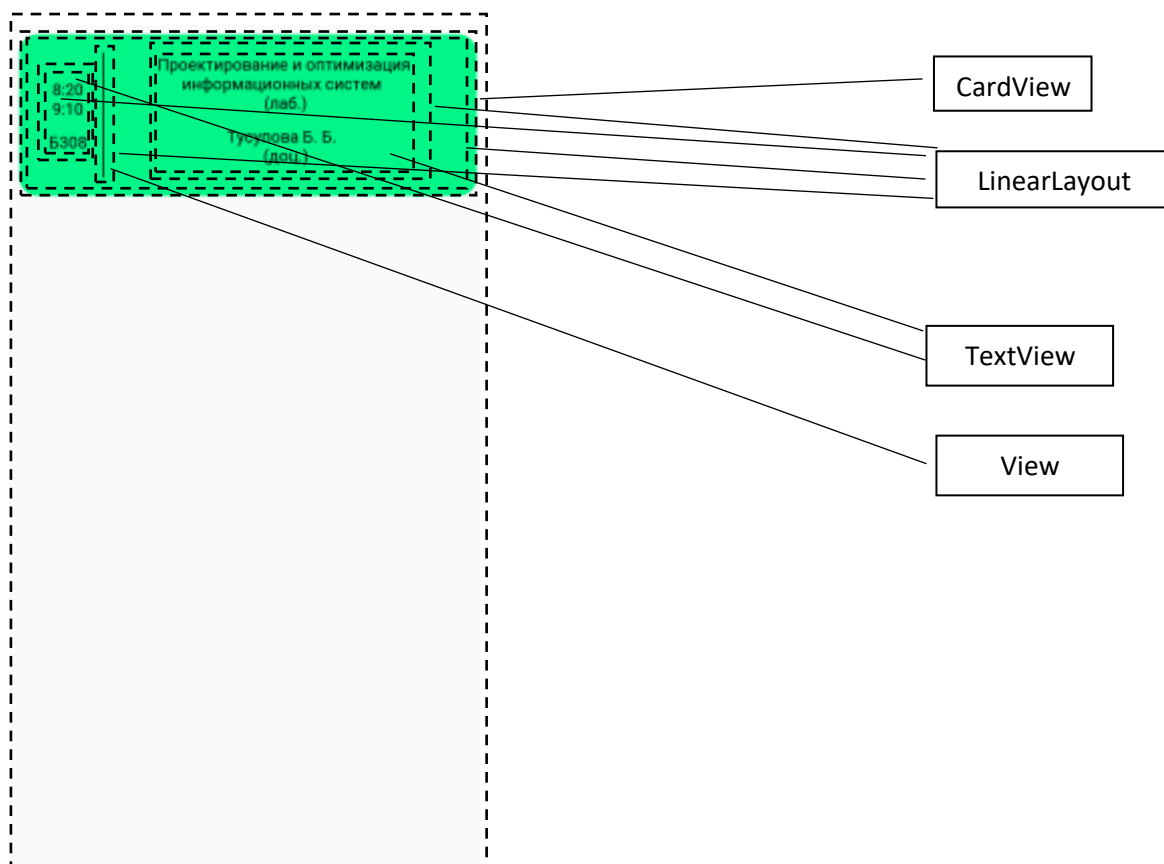


Рисунок 3.7 - Запланированное расположение виджетов ExampleItem на экране

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

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="8dp"
android:focusable="true"
android:orientation="horizontal"
android:padding="8dp"
android:weightSum="7">

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_weight="1"
    android:orientation="vertical">

    <TextView
        android:id="@+id/time"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginBottom="10dp"
        android:gravity="center"
        android:text="8:20\n9:10"
        android:textColor="#000"></TextView>

    <TextView
        android:id="@+id/audience"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:text="E308"
        android:textColor="#000"></TextView>
</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="0.2"
    android:gravity="center"
    android:orientation="horizontal">

    <View
        android:layout_width="1sp"
        android:layout_height="match_parent"
        android:background="#000"></View>

</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="5.8"
    android:orientation="vertical">

    <TextView
        android:id="@+id/disc"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:layout_marginBottom="10dp"
        android:gravity="center"
        android:text="Проектирование и оптимизация ин-
формационных систем \n (лаб.)"
        android:textColor="#000"></TextView>

<TextView
    android:id="@+id/teacher"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:gravity="center"
    android:text="Тусупова Б. Б. \n (доц.)"
    android:textColor="#000"></TextView>
</LinearLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>

```

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

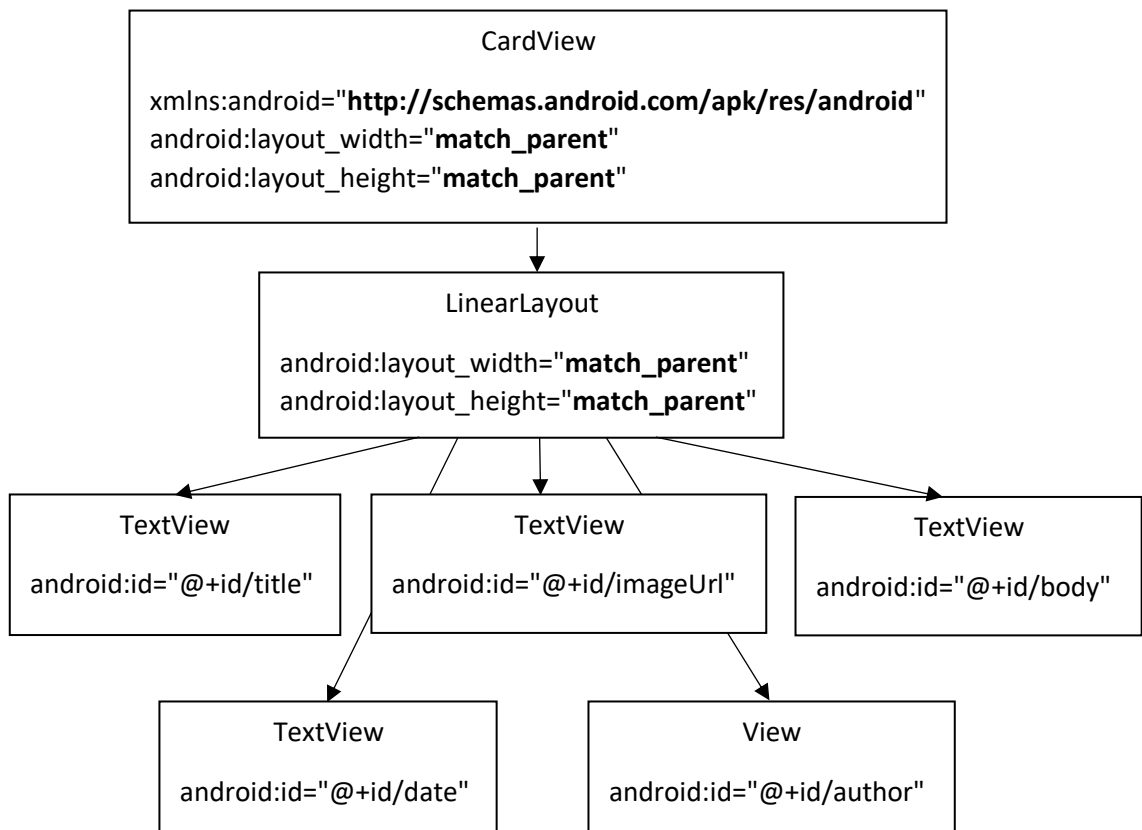


Рисунок 3.8 - Иерархия представлений макета ExampleItem

3.2.2.3 Макет Example_Item_For_Groups

Макет Example_Item_For_Groups состоит из 2 виджетов:

- LinearLayout, который используется как каркас для TextView
- TextView, используемые для отображения текстов;

- EditText служат для вставки текста пользователем;

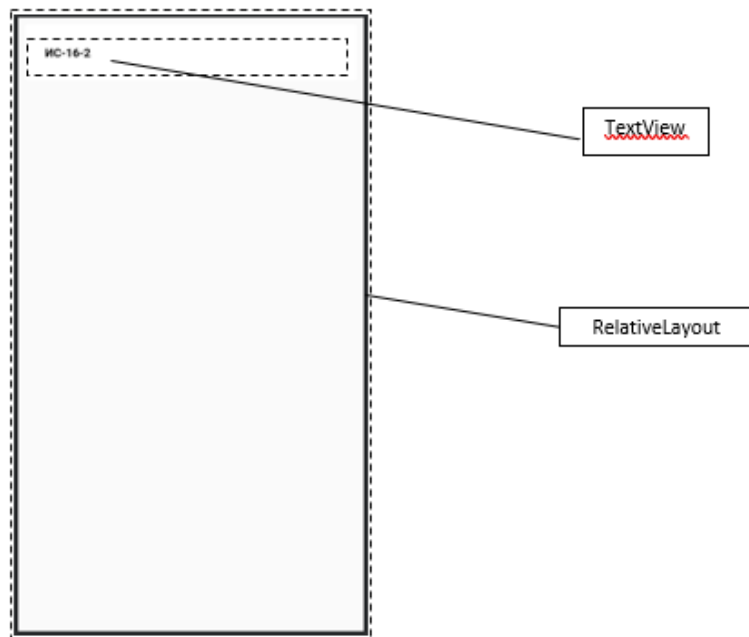


Рисунок 3.9 - Запланированное расположение виджетов ExampleItemForGroups на экране

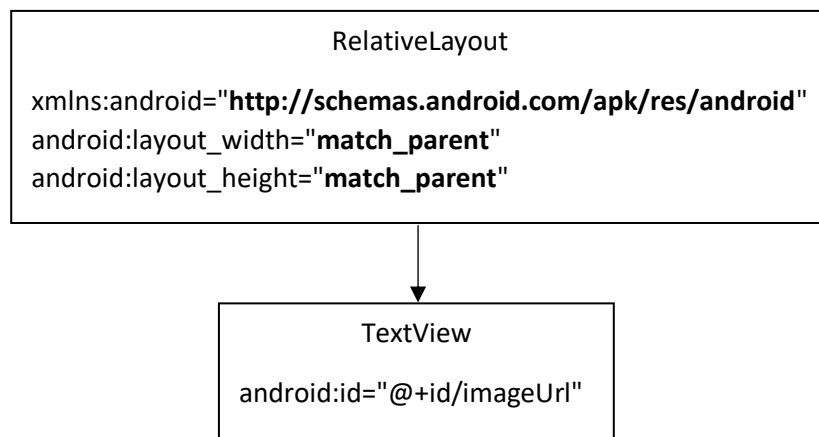


Рисунок 3.10 - Иерархия представлений макета ExampleItemForGroups

Для данной иерархии так же был написан 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"
        android:layout_marginBottom="0dp"
    >

    <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/group_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:clickable="false"
            android:text="ИС-16-2"
            android:textColor="#000"
            android:textStyle="bold" />

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

```

3.2.2.4 Макет Example_Item_For_Teacher

Макет Example_Item_For_Teacher состоит из 3 виджетов:

- LinearLayout, который используется как каркас для TextView
- TextView, используемые для отображения текстов;
- EditText служат для вставки текста пользователем.

Для данного интерфейса был написан модуль:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_marginBottom="0dp"
    android:layout_margin="8dp"
    card_view:cardBackgroundColor="@android:color/transparent"
    card_view:cardElevation="0dp"
    >

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="12dp"
        android:paddingRight="12dp"
        android:paddingBottom="5dp"
        android:paddingTop="5dp">

```

```

<TextView
    android:id="@+id/id_teacher"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:clickable="false"
    android:text="1"
    android:textColor="#000"
    android:textStyle="bold"
    android:visibility="gone"/>

<com.github.siyamed.shapeimageview.CircularImageView
    android:id="@+id/image_icon"
    android:layout_toStartOf="@id/id_teacher"
    android:layout_toLeftOf="@id/id_teacher"
    android:src="@drawable/icon"
    android:layout_width="38dp"
    android:layout_height="38dp"
    app:siBorderWidth="1dp"
    app:siBorderColor="#fff"/>

<LinearLayout
    android:id="@+id/linear"
    android:orientation="vertical"
    android:layout_toRightOf="@id/image_icon"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/teacher_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:clickable="false"
        android:text="Тусупова Бэлла Борисовна"
        android:textSize="18sp"
        android:textColor="#000"
        android:layout_marginLeft="12dp"
        android:fontFamily="@font/roboto"
        android:layout_centerVertical="true"
        />
    <View
        android:layout_marginTop="18dp"
        android:layout_width="match_parent"
        android:layout_height="0.5sp"
        android:layout_marginLeft="12dp"
        android:background="#A9A9A9"/>
</LinearLayout>

</RelativeLayout>

</androidx.cardview.widget.CardView>

```

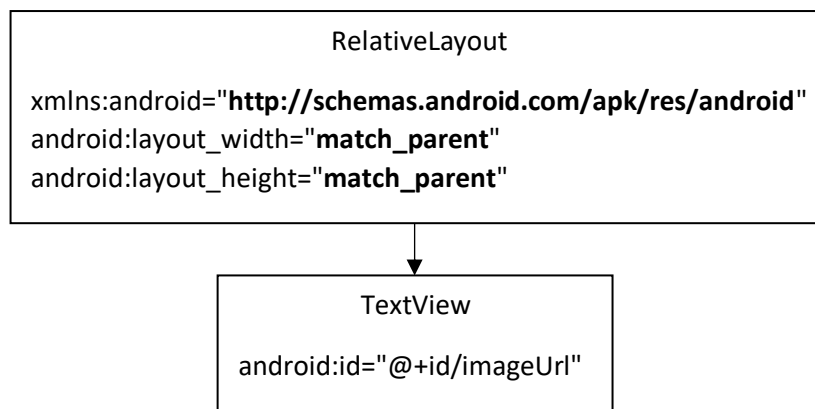


Рисунок 3.11 - Иерархия представлений макета ExampleItemForTeacher

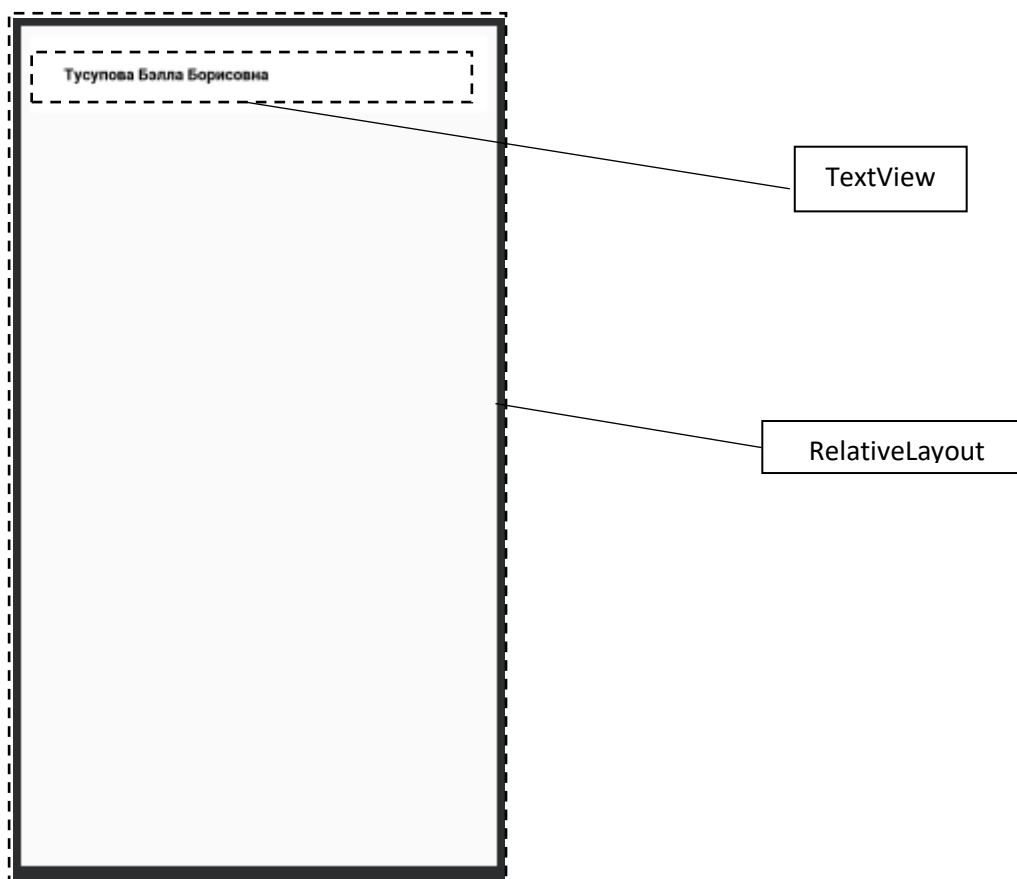


Рисунок 3.12 - Запланированное расположение виджетов ExampleItemForTeacher на экране

3.2.3 Макеты подсистемы «Местоположение»

В данном интерфейсе определены следующие виджеты:

- FrameLayout, который используется в качестве каркаса для фрагмента с местоположением;
- Fragment, главной функцией которого является отображение геолокации университета «АУЭС»;

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

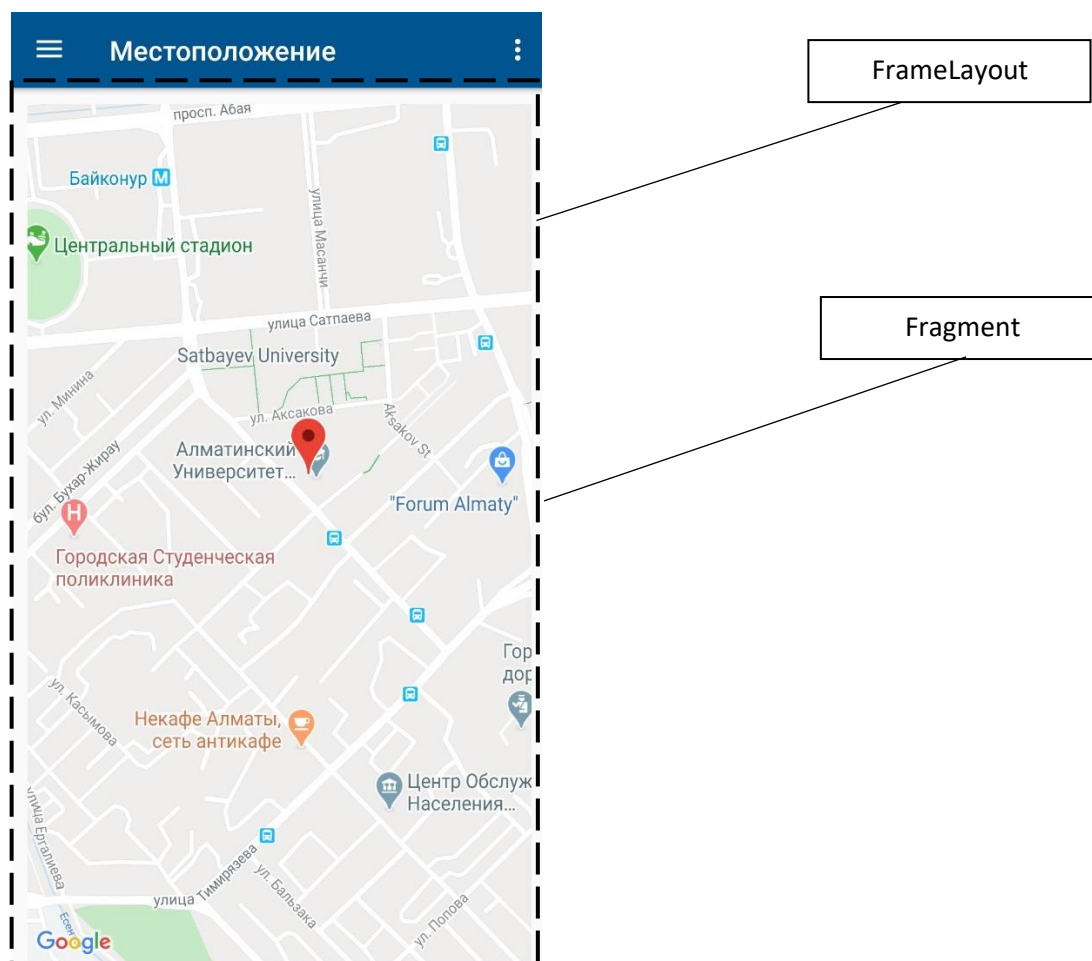


Рисунок 3.13 - Запланированное расположение виджетов **fragment_geo_position** на экране

3.2.4 Макет **fragment_info_university**

В данном интерфейсе определены следующие виджеты:

- **FrameLayout**, который используется в качестве каркаса для фрагмента с местоположением;
- **Fragment**, главной функцией которого является отображение всех остальных виджетов;
- **LinearLayout**, в котором отображается одна из функций;
- **TextView** для отображения текстовой информации;
- **ImageView** для отображения графической информации (изображение в формате **svg**).

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

Иерархическая структура представлена на рисунке 3.15.

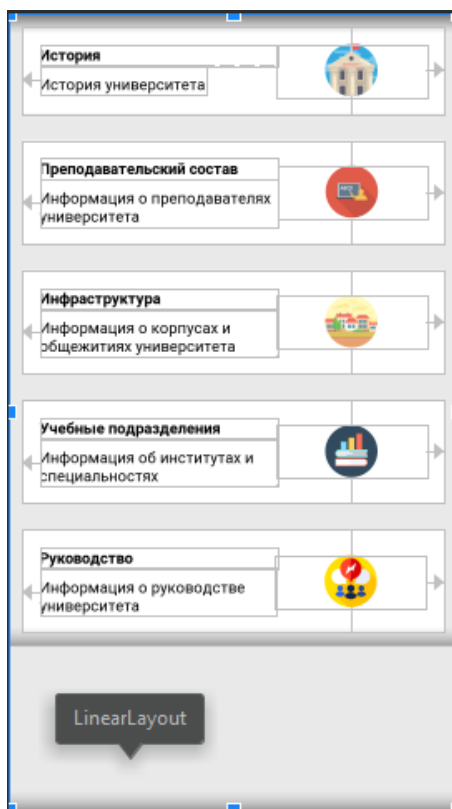


Рисунок 3.14 - Расположение виджетов fragment_info_university на экране

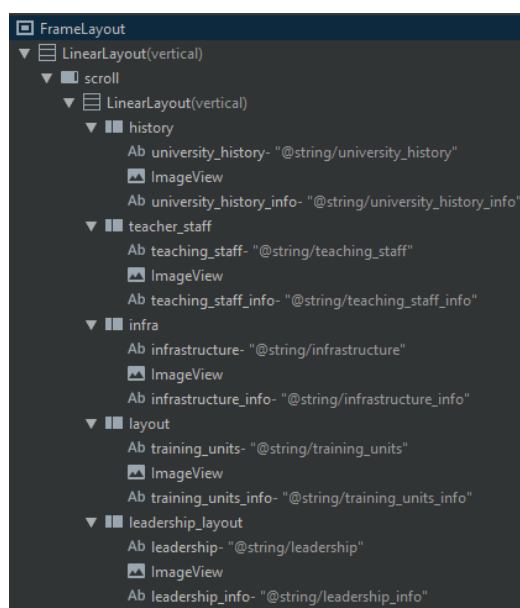


Рисунок 3.15 - Иерархическая структура

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. Схематично структура представлена на рисунке 3.16.

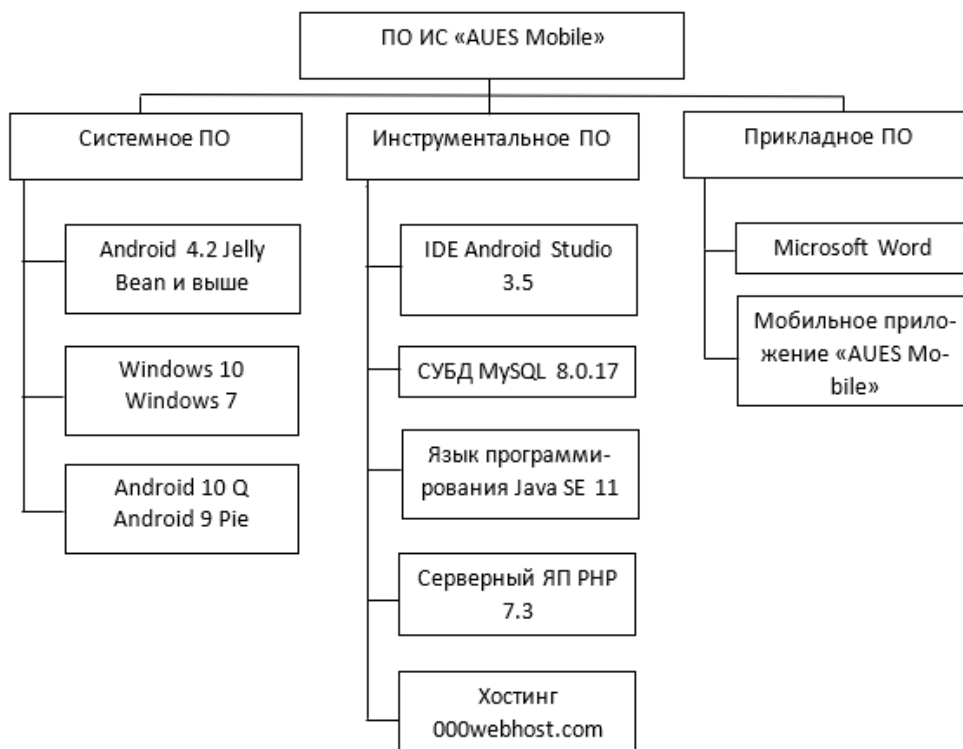


Рисунок 3.16 - Структура ПО

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

Продолжение таблицы 3.1

2	<pre>StringRequest stringRequest = new StringRe- quest(Request.Method.POST, login_url, new Response.Listener<String>() { }</pre>	Объект, возвращающий ответ сервера
3	<pre>public void onResponse(String response) {}</pre>	В переменной response содержится строковая переменная, содержащая в себе ответ сервера
4	<pre>SharedPreferences prefs = PreferenceManag- er.getDefaultSharedPreferences(getContext());</pre>	Объект данного класса служит для хранения пользовательских данных в приложении
5	<pre>protected Map<String, String> getParams() throws AuthFailureError{}</pre>	Метод передает данные, введенные пользователем на странице авторизации, серверу

Код, представленный в листинге, представляет собой класс `ExampleItemForTeacher`. Данный класс имеет конструктор, который принимает два параметра, в этом случае данные о преподавателе и его идентификатор, далее инициализирует переменные, объявленные выше конструктора. Так как переменные имеют модификатор доступа `private`, необходимо создать геттеры, при помощи которых можно обращаться к сокрытым данным. Основные конструкции представлены в таблице 3.2.

Таблица 3.2 - Основные конструкции класса `ExampleItemForTeacher`

№	Конструкция	Пояснения
1	<pre>public ExampleItemForTeacher(String title, String ID) { ...; }</pre>	Конструктор, инициализирующий поля объекта при его создании

2	<pre>public String getTitle() { return mTitle; }</pre>	Метод, возвращающий значение поля mTitle
3	<pre>public String getID() { return mID; }</pre>	Метод, возвращающий значение поля mID

Код, представленный в листинге, представляет собой класс `ExampleItemForGroup`. Данный класс имеет конструктор, который принимает один строковый параметр: наименование группы. Конструктор инициализирует переменную `mTitle`. Так как переменная `mID` имеет модификатор доступа `private`, необходимо создать геттер, при помощи которого можно обращаться к сокрытым данным. Основные конструкции представлены в таблице 3.3.

Таблица 3.3 - Основные конструкции класса `ExampleItemForGroup`

№	Конструкция	Пояснения
1	<pre>public ExampleItemForGroup(String title) { mTitle = title; }</pre>	Конструктор, инициализирующий поле объекта при его создании
2	<pre>public String getTitle() { return mTitle; }</pre>	Метод, возвращающий значение поля mTitle

Код, представленный в листинге, представляет собой класс `ExampleItemForSchedule`. По аналогии с вышеперечисленными классами, данный класс имеет конструктор, который принимает четыре параметра: время, занятие, преподавателя и аудиторию. Конструктор инициализирует переменные. Так как переменные имеют модификатор доступа `private`, необходимо создать геттеры, при помощи которых можно обращаться к сокрытым данным. В данном классе используются такие же конструкции, что использовались в двух предыдущих классах.

Также в данной подсистеме используются адаптеры, которые принимают список данных из базы данных и помещают в список. Один из адаптеров `ExampleAdapterSchedule` представлен в листинге. Основные методы, использующиеся в данном классе, показаны в таблице 3.4.

Таблица 3.4 - Основные методы класса `ExampleAdapterSchedule`

№	Конструкция	Пояснения
---	-------------	-----------

1	<pre>public ExampleViewHolder onCreate- ViewHolder(ViewGroup parent, int viewType) { ... }</pre>	Создает новый объект ViewHolder всякий раз, когда RecyclerView нуждается в этом
2	<pre>public void onBindViewHold- er(ExampleViewHolder holder, int po- sition) { ... }</pre>	Принимает объект ViewHolder и устанавливает необходимые данные для соответствующей строки во view-компоненте

Продолжение таблицы 3.4

3	<pre>public int getItemCount() { return mExampleList.size(); }</pre>	Возвращает количество записей в списке
---	--	--

Основной класс, реализующий функционал подсистемы - класс Monday, представлен в листинге. Класс Monday имеет 4 конструктора, которые принимают параметры группы обучающегося, параметры преподавателя, который вошел в систему, параметры любой группы (при нажатии на нее) и параметры любого преподавателя (при нажатии на него).

Таблица 3.5 - Основные конструкции класса Monday

№	Конструкция	Пояснения
1	<pre>String url = "https://auesmobiledatabase.000webhostapp.com/sch- edule.php";</pre>	Переменная, хранящая URL-адрес скрипта PHP
2	<pre>StringRequest stringRequest = new StringRe- quest(Request.Method.POST, login_url, new Response.Listener<String>() { }</pre>	Объект, возвращающий ответ сервера
3	<pre>public void onResponse(String response) {}</pre>	В переменной response содержится строковая переменная, содержащая в себе ответ сервера
4	<pre>SharedPreferences prefs = PreferenceManag-</pre>	Объект дан-

	<code>er.getDefaultSharedPreferences(getContext());</code>	ного класса служит для хранения пользовательских данных в приложении
5	<code>protected Map<String, String> getParams() throws AuthFailureError{}</code>	Метод передает данные, необходимые для вывода расписания, серверу

3.5 Серверная часть

Для того, чтобы извлечь данные, хранящиеся на сервере, необходимо написать скрипты. Сначала проверяется проверка соединения 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 отправляя 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);
?>

```

В следующем скрипте проверяется соединение к базе данных, далее методом POST берутся данные о группе или преподавателе, которые передал клиент. В зависимости от текущей даты, скрипт определяет академический год и академический период обучающегося/преподавателя. Пишется соответствующий SQL-запрос, который возвращает расписание за определенный день недели. Далее данные упаковываются в JSON-массив и отправляются в качестве ответа клиенту. Скрипт ниже осуществляет вышеописанный функционал.

```

<?php
require "int.php";
$id_day = (int)$_POST["id_day"];
$role_id = (int)$_POST["role_id"];
$any_group = $_POST["any_group"];
$any_teacher = $_POST["any_teacher"];
$year = date("Y");
$academic_year = "";
$academic_period = "";
$month = date("m");
$sql = "";
if ($month >=8) {
    $year = date("Y") + 1;
    $academic_year = date("Y") . "-" . $year;
    $academic_period = 1;
}
else if ($month < 8) {
    $year = date("Y") - 1;
    $academic_year = $year . "-" . date("Y");
    $academic_period = 2;
}
}

```

```

if ($any_group == "true") {
    $id_group = $_POST["id_group"];
    $sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>', TIME.TIME_END, '<br>',
        '(', COALESCE(SCHEDULE.SUB_GROUP,''), ')'), '()', '') AS TIME,
    CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(', LOWER(TRAINING.SHORT_NAME),
    ')') AS TRAINING,
    CONCAT(TEACHER.SURNAME, ' ', LEFT(TEACHER.NAME, 1), '. ', LEFT(TEACHER.PATRONYMIC, 1),
    '.', '<br>',
    DEGREE.SHORT_NAME) AS NAME,
    AUDIENCE.ID_AUDIENCE
FROM
DISCIPLINE
INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCIPLINE_CLASS.ID_GROUP
                        AND TRAINING.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING = TRAINING.ID_TRAINING
                        AND TEACHER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
                        AND TEACHER_TRAINING.ID_GROUP = TRAINING.ID_GROUP
INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACHER_TRAINING.ID_GROUP
                        AND SCHEDULE.CODE_DISCIPLINE = TEACHER_TRAINING.CODE_DISCIPLINE
                        AND SCHEDULE.ID_TRAINING = TEACHER_TRAINING.ID_TRAINING
INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACHER.ID_TEACHER

INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE
INNER JOIN DEGREE ON DEGREE.ID_DEGREE = TEACHER.ID_DEGREE
INNER JOIN CLASS ON CLASS.ID_GROUP = SCHEDULE.ID_GROUP
WHERE
DISCIPLINE_CLASS.ACADEMIC_YEAR = ''.$academic_year.''
AND CLASS.NAME_OF_GROUP = ''.$id_group.''
AND DAY.ID_DAY = ''.$id_day.''
AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = ''.$academic_period.''
ORDER BY

    TIME.ID_TIME;"
}
else if ($any_teacher == "true") {
    $id_teacher = (int)$_POST["id_teacher"];
    $sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>', TIME.TIME_END, '<br>',
        '(', COALESCE(SCHEDULE.SUB_GROUP,''), ')'), '()', '') AS TIME,
    CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(', LOWER(TRAINING.SHORT_NAME), ')')
    AS TRAINING,
    CLASS.NAME_OF_GROUP AS NAME,
    AUDIENCE.ID_AUDIENCE
FROM
DISCIPLINE
INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCIPLINE_CLASS.ID_GROUP
                        AND TRAINING.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING = TRAINING.ID_TRAINING
                        AND TEACHER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
                        AND TEACHER_TRAINING.ID_GROUP = TRAINING.ID_GROUP
INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACHER_TRAINING.ID_GROUP
                        AND SCHEDULE.CODE_DISCIPLINE = TEACHER_TRAINING.CODE_DISCIPLINE
                        AND SCHEDULE.ID_TRAINING = TEACHER_TRAINING.ID_TRAINING
INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACHER.ID_TEACHER
INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE
INNER JOIN CLASS ON DISCIPLINE_CLASS.ID_GROUP = CLASS.ID_GROUP
WHERE
DISCIPLINE_CLASS.ACADEMIC_YEAR = ''.$academic_year.''
AND TEACHER.ID_TEACHER= ''.$id_teacher.''
AND DAY.ID_DAY = ''.$id_day.''
AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = ''.$academic_period.''
ORDER BY

    TIME.ID_TIME;"
}
else {
    if ($role_id == 2) {
        $id_group = $_POST["id_group"];
        $sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>', TIME.TIME_END,
        '<br>', '(', COALESCE(SCHEDULE.SUB_GROUP,''), ')'), '()', '') AS TIME,
        CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(', LOWER(TRAINING.SHORT_NAME), ')')
        AS TRAINING,
        CONCAT(TEACHER.SURNAME, ' ', LEFT(TEACHER.NAME, 1), '. ', LEFT(TEACHER.PATRONYMIC, 1),

```

```

        '.', '<br>',
        DEGREE.SHORT_NAME) AS NAME,
        AUDIENCE.ID_AUDIENCE
FROM
    DISCIPLINE
    INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
    INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCIPLINE_CLASS.ID_GROUP
                        AND TRAINING.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
    INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING = TRAINING.ID_TRAINING
                        AND TEACHER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
                        AND TEACHER_TRAINING.ID_GROUP = TRAINING.ID_GROUP
    INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACHER_TRAINING.ID_GROUP
                        AND SCHEDULE.CODE_DISCIPLINE = TEACHER_TRAINING.CODE_DISCIPLINE
                        AND SCHEDULE.ID_TRAINING = TEACHER_TRAINING.ID_TRAINING
    INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
    INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
    INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACHER.ID_TEACHER
    INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE
    INNER JOIN DEGREE ON DEGREE.ID_DEGREE = TEACHER.ID_DEGREE
    INNER JOIN CLASS ON CLASS.ID_GROUP = SCHEDULE.ID_GROUP
WHERE
    DISCIPLINE_CLASS.ACADEMIC_YEAR = ".$academic_year."
    AND CLASS.NAME_OF_GROUP = ".$id_group."
    AND DAY.ID_DAY = ".$id_day."

    AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = ".$academic_period."
ORDER BY
    TIME.ID_TIME;"
}
else if ($role_id == 3) {
    $id_teacher = (int)$_POST["id_teacher"];
    $sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>', TIME.TIME_END, '<br>', '(',
        COALESCE(SCHEDULE.SUB_GROUP, ''), ')'), '()', '') AS TIME,

    CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(', LOWER(TRAINING.SHORT_NAME), ')')
    AS TRAINING,
    CLASS.NAME_OF_GROUP AS NAME,
    AUDIENCE.ID_AUDIENCE
FROM
    DISCIPLINE
    INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
    INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCIPLINE_CLASS.ID_GROUP
                        AND TRAINING.CODE_DISCIPLINE = DISCIPLINE_CLASS.CODE_DISCIPLINE
    INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING = TRAINING.ID_TRAINING
                        AND TEACHER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
                        AND TEACHER_TRAINING.ID_GROUP = TRAINING.ID_GROUP
    INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACHER_TRAINING.ID_GROUP
                        AND SCHEDULE.CODE_DISCIPLINE = TEACHER_TRAINING.CODE_DISCIPLINE
                        AND SCHEDULE.ID_TRAINING = TEACHER_TRAINING.ID_TRAINING
    INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
    INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
    INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACHER.ID_TEACHER
    INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE
    INNER JOIN CLASS ON DISCIPLINE_CLASS.ID_GROUP = CLASS.ID_GROUP
WHERE
    DISCIPLINE_CLASS.ACADEMIC_YEAR = ".$academic_year."
    AND TEACHER.ID_TEACHER= ".$id_teacher."
    AND DAY.ID_DAY = ".$id_day."
    AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = ".$academic_period."
ORDER BY
    TIME.ID_TIME;"
}
}

$result = mysqli_query($conn, $sql);
$json_array = array();
while ($row = mysqli_fetch_assoc($result))
{
    $json_array[] = $row;
}
echo json_encode(array('schedule' => $json_array), JSON_UNESCAPED_UNICODE, JSON_FORCE_OBJECT);
?>

```

3.6 Тестирование и отладка программного обеспечения

После разработки прототипа и описания классов активностей и фрагментов запускается приложение на смартфоне Xiaomi Redmi 5 PLUS (6.4 дюйм).

Сначала пользователь должен авторизоваться в системе. Процесс авторизации показан на рисунке 3.17.

После того как пользователь успешно авторизовался, ему открывается активити с фрагментом новостей. На рисунке 3.18 показана панель навигации для пользователя.

После нажатия пользователем на «Расписание группы» или «Расписание преподавателя», открывается новая деятельность, где пользователю нужно выбрать либо группу, либо ФИО преподавателя. Окно выбора для расписания группы и расписания преподавателя показаны на рисунках 3.19 и 3.20. После того, как пользователь выбрал нужное поле, откроется окно с расписанием конкретной группы или преподавателя. Итоговое расписание показано на рисунках 3.21 и 3.22.

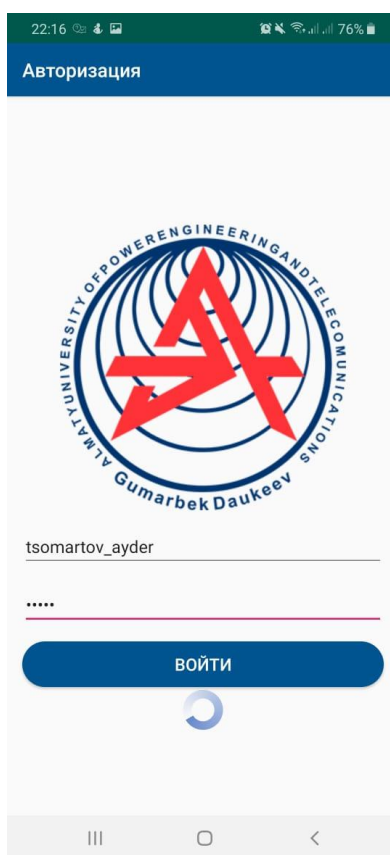


Рисунок 3.17 - Авторизация пользователя

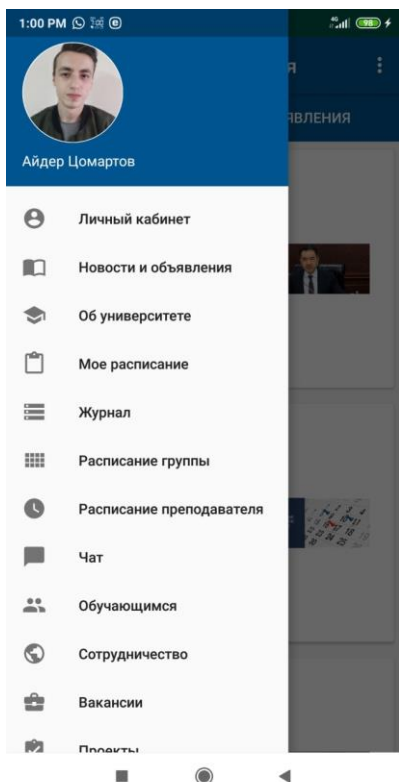


Рисунок 3.18 - Панель навигации

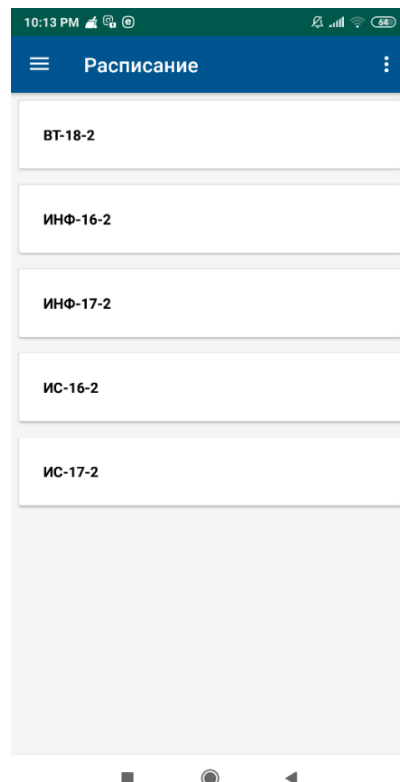


Рисунок 3.19 - Окно выбора расписания

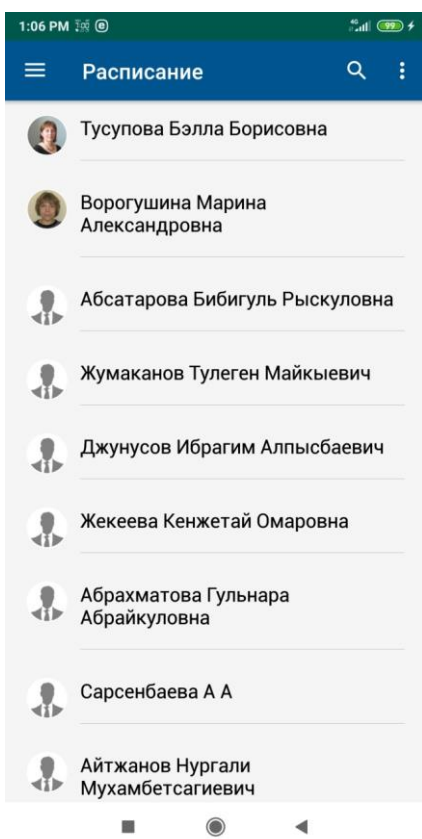


Рисунок 3.20 - Окно выбора преподавателя



Рисунок 3.21 - Итоговое расписание для группы



Рисунок 3.22 - Итоговое расписание для преподавателя

На рисунке 3.23 показан чат для обмена сообщениями между пользователями системы, в качестве хранилища используется Realtime Database. Далее на рисунке 3.24 имеется окно с информацией об университете. На рисунке 3.25 представлена информация об инфраструктуре университета. На рисунке 3.26 - учебные подразделения.

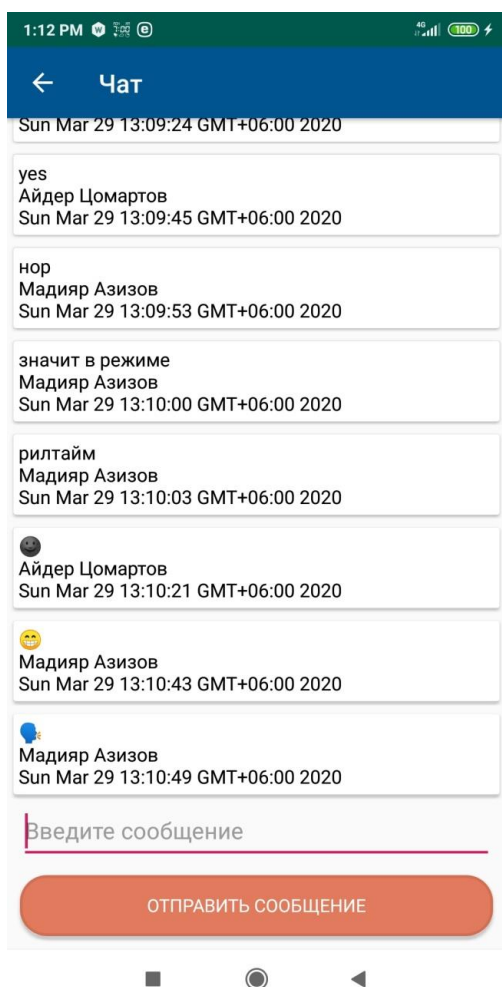


Рисунок 3.23 - Чат

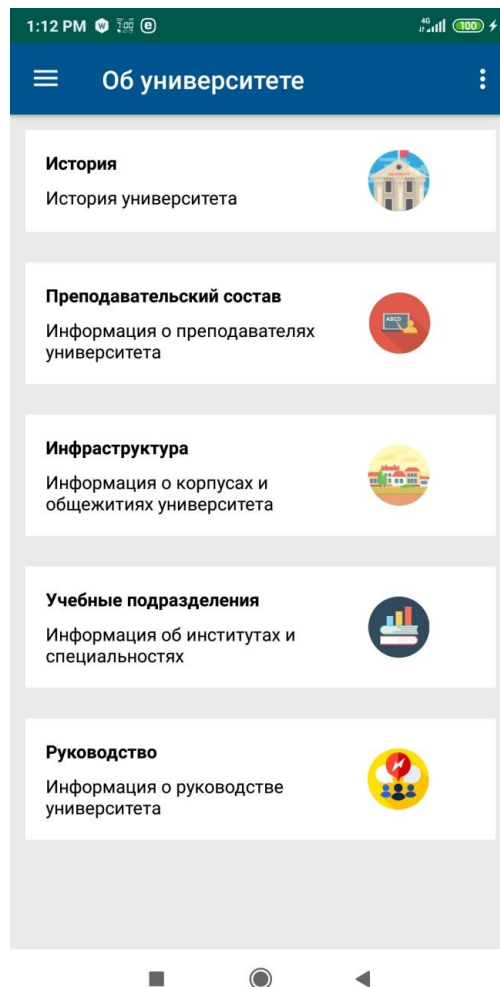


Рисунок 3.24 - Информация об университете

Кроме того, у пользователей имеется свой личный кабинет со своими функциями. Личный кабинет студента и преподавателя изображен на рисунках 3.27 и 3.28 соответственно. Одна из таких функций осуществляет доступ к индивидуальному плану студента - реализация на рисунке 3.29.

Имеется вкладка местоположение с Google-картой - показано на рисунке 3.30, обратная связь - рисунок 3.31, при нажатии на которую происходит интент в приложение с почтой - результат процесса предоставлен на рисунке 3.32.

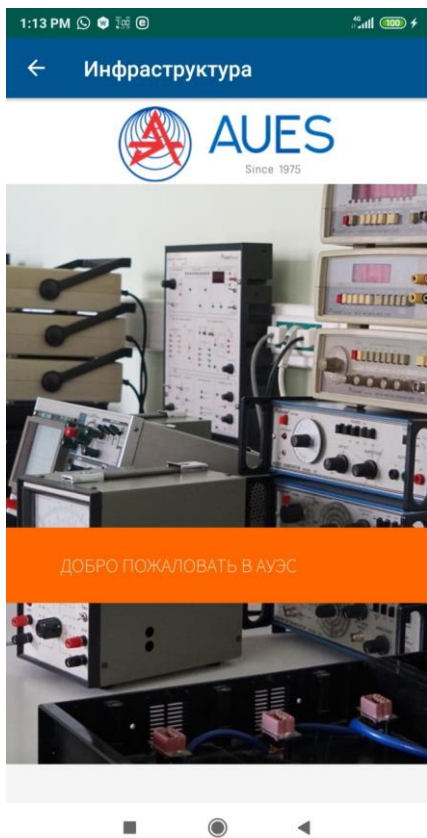


Рисунок 3.25 - Инфраструктура

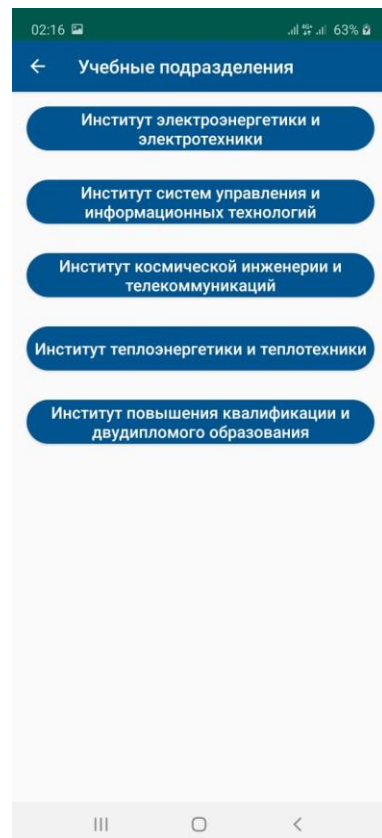


Рисунок 3.26 - Учебные подразделения

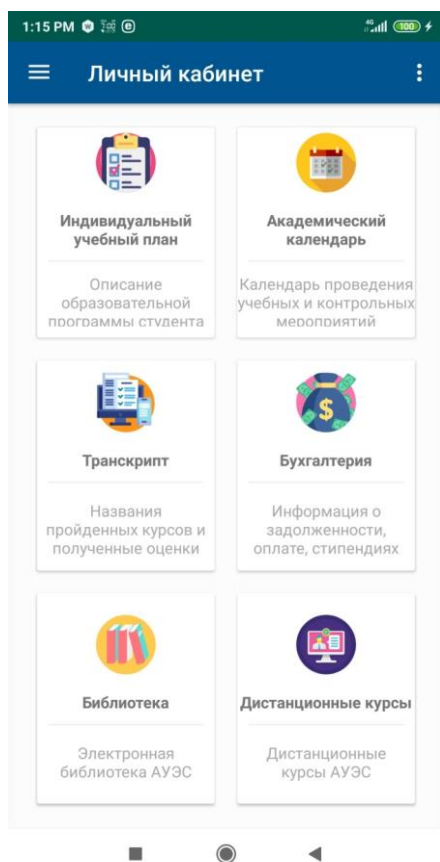


Рисунок 3.27 - Личный кабинет студента

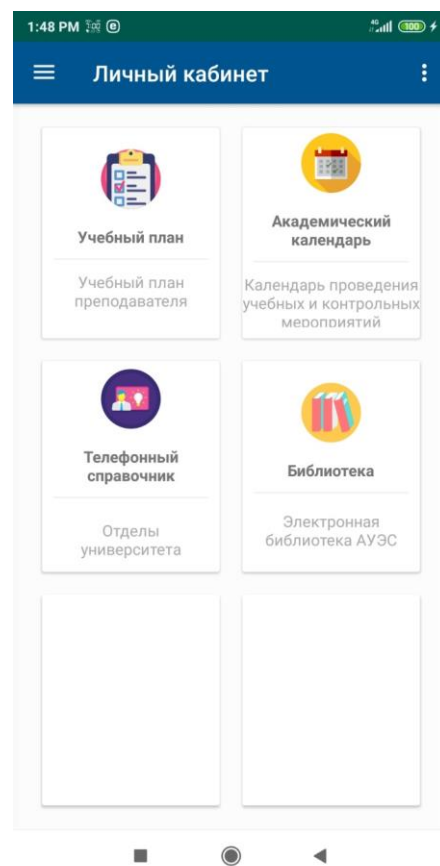


Рисунок 3.28 - Личный кабинет преподавателя

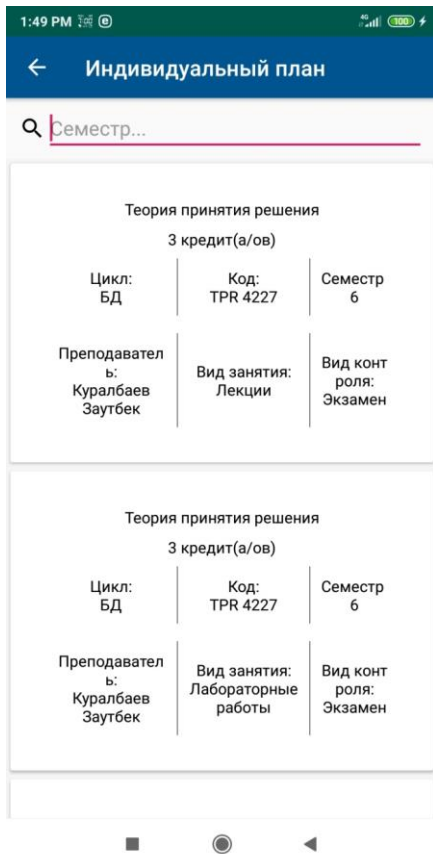


Рисунок 3.29 - Индивидуальный план

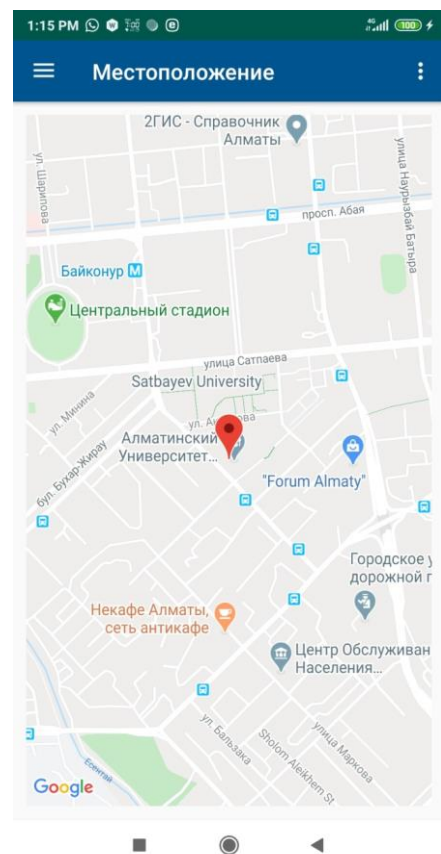


Рисунок 3.30 - Местоположение

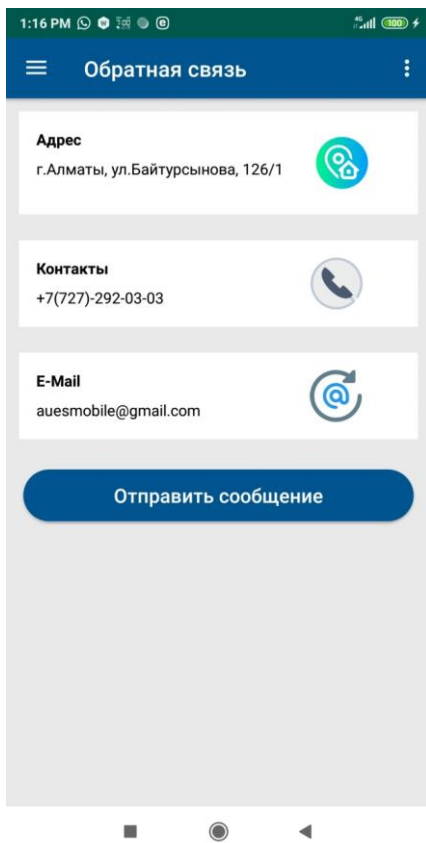


Рисунок 3.31 - Обратная связь

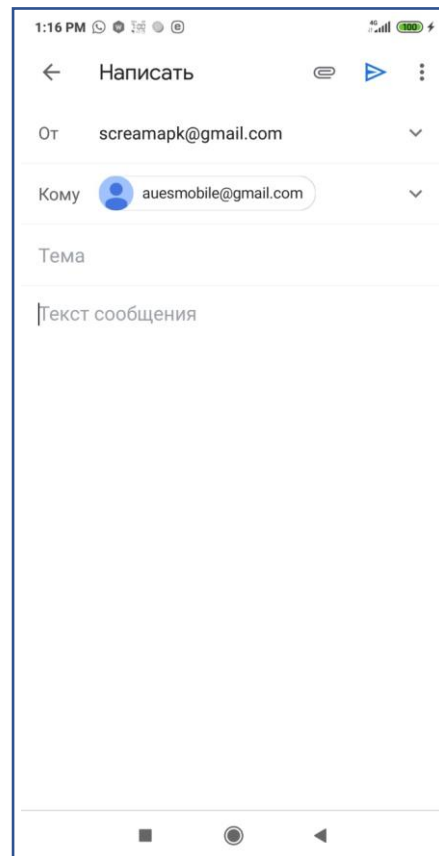


Рисунок 3.32 - Связь с администрацией

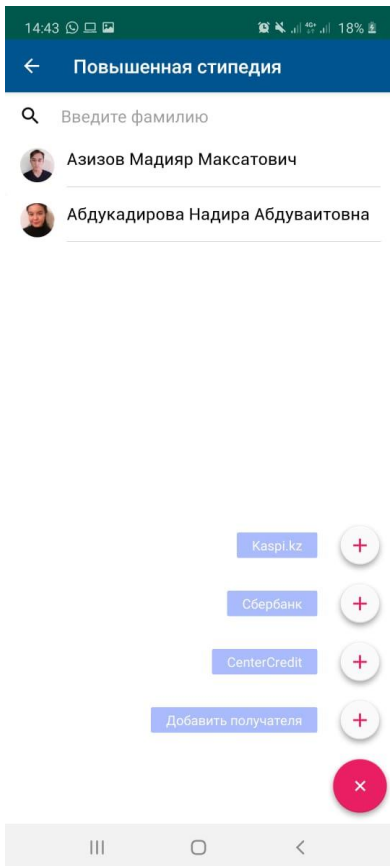


Рисунок 3.33 - Назначение стипендий Рисунок 3.34 - Логирование действий

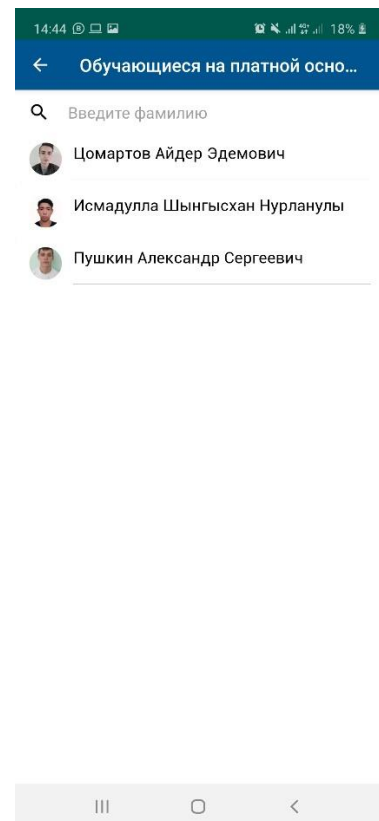
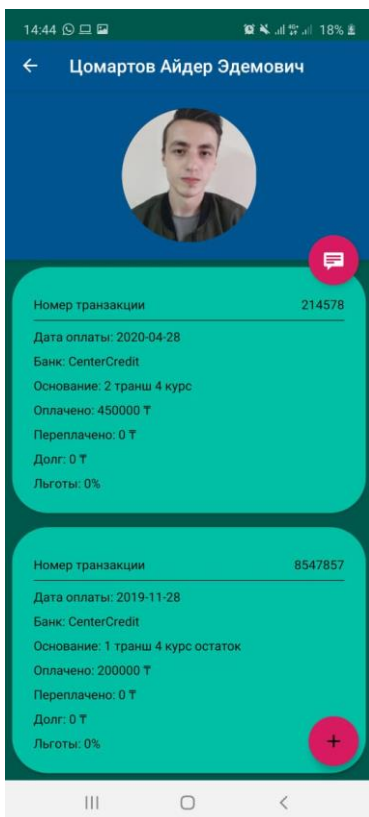


Рисунок 3.35 - Информация об оплате Рисунок 3.36 - Список «платников»

4 Экономическое обоснование проекта

Целью данного дипломного проекта является разработка мобильного приложения «AUES Mobile» на платформе Android. Объектами автоматизации в ходе проектирования являются процесс оперативного получения информации о расписании студентов и преподавателей, процесс мгновенного обмена и получения информации между пользователями системы. Кроме того, мобильное приложение «AUES Mobile» имеет функции, которые позволяют отслеживать студентам сведения об их задолженности, информацию об оплате и стипендиях. Основные преимущества данного мобильного приложения - это удобство, расписание и остальную информацию можно получить напрямую из телефона, а также скорость и простой интерфейс. Основной задачей экономического раздела является определение объема и трудоемкости разработки программного обеспечения, а также расчет сроков окупаемости.

4.1 Определение объема и трудоемкости разработки программного обеспечения

Общий объем (V_0) программного продукта определяется исходя из количества и объема функций, реализуемых программой [19]. За уточненный объем ПО (V_y) учитывается «строка исходного кода» как одна, если в ней содержится лишь один оператор, в данном случае:

$$V_y = 6000 \text{ строк}$$

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 трудоемкости ПО [19]:

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

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

T_n - нормативная трудоемкость;

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 трудоемкость T_n определяем по таблице В.1 приложения В [19]:

$$T_n = 194 \text{ часов.}$$

Коэффициент сложности рассчитывается по формуле [19]

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

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

n - количество учитываемых характеристик.

Согласно таблице 1 и Г.1 приложения Г [19] 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 ($K_c = 0,18$).

Следовательно,

$$K_c = 1 + \frac{0,06 + 0,18}{2} = 1,12.$$

Поправочный коэффициент, учитывающий степень использования готовых модулей, берется из таблицы 2 [19] и составляет $K_m = 0,7$, т.к в проекте задействованы стандартные библиотеки Android Studio, например Volley и Picasso

Коэффициент, учитывающий степень новизны, определяется по таблице 3 [1] и равен $K_n = 1,6$, т.к данный продукт не имеет доступных аналогов и создан на базе ОС Android. Значения коэффициентов удельных весов трудоемкости стадий берутся из таблицы 4 [19], 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 трудоемкости	K_c - коэффициент сложности	T_n - нормативная трудоемкость	K_n - коэффициент новизны
1	Техническое задание	0,11	1,12	194	1,6
2	Эскизный проект	0,09	1,12	194	1,6
3	Технический проект	0,11	1,12	194	1,6
4	Рабочий проект	0,55	1,12	194	1,6
5	Внедрение	0,14	1,12	194	1,6

Трудоемкость стадий ПО рассчитывается по следующим формулам [19]

$$(4.3) \quad \text{трудоемкость стадии ТЗ} \quad T_{уз} = T_H * d_3 * K_C * K_H$$

$$(4.4) \quad \text{трудоемкость стадии ЭП} \quad T_{уэ} = T_H * d_3 * K_C * K_H$$

$$(4.5) \quad \text{трудоемкость стадии ТП} \quad T_{ут} = T_H * d_m * K_C * K_H$$

$$(4.6) \quad \text{трудоемкость стадии РП} \quad T_{ур} = T_H * d_p * K_C * K_H * K_m$$

$$(4.7) \quad \text{трудоемкость стадии ВН} \quad T_{ув} = T_H * d_B * K_C * K_H$$

Тогда,

$$\begin{aligned} T_{уз} &= 194 * 1,12 * 0,11 * 1,6 = 38,24 \text{ часов;} \\ T_{уэ} &= 194 * 1,12 * 0,09 * 1,6 = 31,29 \text{ часов;} \\ T_{ут} &= 194 * 1,12 * 0,11 * 1,6 = 38,24 \text{ часов;} \\ T_{ур} &= 194 * 1,12 * 0,55 * 1,6 * 0,7 = 133,84 \text{ часов;} \\ T_{ув} &= 194 * 1,12 * 0,14 * 1,6 = 48,67 \text{ часов.} \end{aligned}$$

Общая трудоемкость будет равна

$$T_y = 38,24 + 31,29 + 38,24 + 133,84 + 48,67 = 290,28 \text{ часов} \approx 36 \text{ дней.}$$

4.2 Расчет затрат на разработку информационных технологий

Расчет полных затрат на разработку проектного решения осуществляется по формуле [19]:

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

где $Z_{фот}$ - общий фонд оплаты труда разработчиков, тенге;
 $Z_{сзи}$ - отчисления по социальному налогу, тенге;
 $Z_{сои}$ - социальное отчисление, тенге;
 M_i - затраты на материалы, тенге;
 P_{ci} - затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;
 $P_{ми}$ - затраты, связанные с эксплуатацией техники, тенге;
 $P_{нки}$ - затраты на научные командировки, тенге;
 $П_{зи}$ - прочие затраты, тенге;
 $P_{ни}$ - накладные расходы, тенге.

А - амортизационные отчисления;
ВОСМС - отчисления по ОСМС.

Размер фонда оплаты труда разработчиков рассчитывается по формуле [19]:

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

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

Основная заработная плата рассчитывается по следующей формуле [19]

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

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.

Т.к средней заработной плате в Казахстане в 2020 году составляет 203900 тенге, то мы используем ее в формуле:

$$Z_{oi} = 290,28 * \frac{203\,900}{22 * 8} = 336\,300 \text{ тенге.}$$

Дополнительная заработная плата составляет 10% от основной и рассчитывается по формуле [19]:

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

где H_d - коэффициент дополнительной заработной платы разработчиков.

Тогда,

$$Z_{\varphi i} = 336\,300 * \frac{15}{100} = 50\,445 \text{ тенге.}$$

Отчисления на ОСМС рассчитываем по формуле:

$$\text{ВОСМС} = \text{ФОТ} * 2\% \quad (4.12)$$

Тогда,

$$\text{ВОСМС} = 386745 * 2\% = 7734 \text{ тенге};$$

Социальный налог составляет (9,5%) (ст. 358 п.1 НК РК) от дохода работника, и рассчитывается по формуле [19]:

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

(4.13)

где ПО - пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются [19]:

$$\text{ПО} = \text{ФОТ} * 10\%.$$

(4.14)

Тогда,

$$\begin{aligned} \text{ПО} &= (336300 + 50445) * 10\% = 38674 \text{ тенге}; \\ Z_{czi} &= (386745 - 38674 - 7734) * 9,5\% - 12182 = 20150 \text{ тенге}. \end{aligned}$$

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

$$Z_{coi} = (\text{ФОТ} - \text{ПО}) * 3,5\%,$$

(4.15)

Тогда,

$$Z_{coi} = (386745 - 38674) * 3,5\% = 12182 \text{ тенге}.$$

Общая сумма налогов будет составлять:

$$\text{Налоги} = \text{СО} + \text{СН} + \text{ВОСМС} = 12182 + 20150 + 7734 = 40066 \text{ тенге}$$

Величина затрат на материалы определяется по формуле [19]:

$$M_i = \frac{Z_{ocn} * H_{mz}}{100} \%,$$

(4.16)

где H_{mz} - норма расхода материалов от основной заработной платы (3-5%).

Следовательно,

$$M_i = 336300 * \frac{5}{100} = 16815 \text{ тенге}.$$

Расходы по статье «Спецоборудование» (P_{ci}) включает в себя покупку аккаунта разработчика Google Play стоимостью 25\$, покупку дополнительной оперативной памяти 4 Гб стоимостью 30\$, а также расходы на приобретение компьютерной мыши стоимостью 10\$ и пакета интернет трафика на сумму 40\$

$$P_{ci} = 105\$ = 45\ 000 \text{ тенге.}$$

Амортизационные отчисления рассчитываются по формуле:

$$(4.16) \quad A = \frac{C_{\text{обор}} * H_a * N}{100 * 12 * t},$$

где H_a - норма амортизации (25 %);

$C_{\text{обор}}$ - первоначальная стоимость оборудования;

t - количество рабочих дней в месяце ($t = 22$ дней);

N - время использования персонального компьютера;

$$N = 290,28/8 \approx 36 \text{ дней.}$$

Тогда,

$$A = \frac{17\ 000 * 25 * 36}{100 * 12 * 22} = 579 \text{ тенге.}$$

Расходы по статье «Машинное время» (P_{mi}) включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машино-часах) на 100 строк исходного кода машинного времени (H_{mv}) в зависимости от характера решаемых задач и типа ПК (Приложение Д) [19]:

$$(4.17) \quad P_{mi} = C_{mi} * \left(\frac{V_{oi}}{100}\right) * H_{mv},$$

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

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

Тогда,

$$P_{mi} = 400 * \left(\frac{6\ 000}{100}\right) * 12 = 288\ 000 \text{ тенге.}$$

Расходы по статье «Научные командировки» ($P_{\text{нк}i}$) на конкретное ПО определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате [19]:

$$P_{\text{нк}i} = Z_{oi} * H_{\text{рнк}}/100 \quad (4.18)$$

где $H_{\text{рнк}}$ - норматив расходов на командировки в целом по организации в (%), равен 30% [19].

Тогда,

$$P_{\text{нк}i} = 336\,300 * \frac{30}{100} = 100\,890 \text{ тенге.}$$

Расходы по статье «Прочие затраты» (Π_{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 [19]:

$$\Pi_{zi} = Z_{oi} * H_{\text{пз}}/100 \quad (4.19)$$

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

Тогда,

$$\Pi_{zi} = 336\,300 * \frac{20}{100} = 67\,260 \text{ тенге.}$$

Затраты по статье «Накладные расходы» ($P_{\text{н}i}$), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также с расходами на общехозяйственные нужды ($P_{\text{н}i}$), относятся на конкретное ПО по нормативу ($H_{\text{нр}}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации [19]:

$$P_{\text{н}i} = Z_{oi} * H_{\text{нр}}/100 \quad (4.20)$$

где $P_{\text{н}i}$ - накладные расходы на конкретную ПО (тыс. тенге);

$H_{\text{нр}}$ - норматив накладных расходов в целом по организации в (%), равен 70%.

Следовательно,

$$P_{\text{н}i} = 336\,300 * \frac{70}{100} = 235\,410 \text{ тенге.}$$

Таблица 4.2 - Затраты на разработку информационных технологий

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{\text{ФОТ}}$	386 745	32,77
Социальный налог	$Z_{\text{сзи}}$	33 066	2,80
Социальное отчисление	$Z_{\text{сои}}$	12 182	1,03
Материалы	M_i	16 185	1,37
Спецоборудование	$P_{\text{си}}$	45 000	3,81
Машинное время	$P_{\text{ми}}$	288 000	24,40
Научные командировки	$P_{\text{нки}}$	100 890	8,54
Прочие затраты	$P_{\text{зи}}$	67260	5,69
Накладные расходы	$P_{\text{ни}}$	235 410	19,94
Амортизационные отчисления	A	579	0,04
Отчисления по ОСМС	ВОСМС	7734	0,65
Итого:	C_0	1 180 135	100

Цена разрабатываемого программного обеспечения рассчитывается по формуле [19]:

$$C_p = C_0 + \text{НДС}, \quad (4.21)$$

где C_0 - первоначальная цена.

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

$$C_{\text{п}} = C \times \left(1 + \frac{P}{100}\right), \quad (4.22)$$

где P - рентабельность (50%).

Подставив полученные значения, первоначальная цена согласно формуле составит:

$$C_0 = 1\,180\,135 * \left(1 + \frac{50}{100}\right) = 1\,770\,202 \text{ тенге.}$$

Цена готовой продукции рассчитывается по формуле:

$$C_p = C_0 + \text{НДС}, \quad (4.23)$$

где НДС - 12% от начальной цены готового продукта.

Рассчитаем НДС используя формулу (4.22)

$$\text{НДС} = 1\,770\,202 * 0,12 = 212\,424 \text{ тенге.}$$

Следовательно, конечная итоговая цена программного продукта составит:

$$C_p = 1\,770\,202 + 212\,424 = 1\,982\,626 \text{ тенге.}$$

4.3 Расчет динамических показателей эффективности капитальных вложений

Прибыль от приложения университету АУЭС и его разработчикам будет полностью зависеть от рекламы внутри приложения. В университете обучаются порядка 7000 студентов, допустим, что в среднем каждый день приложение будут посещать около 500 пользователей. В данном приложении будет присутствовать баннерная и всплывающая реклама. Сначала следует рассмотреть баннерную рекламу, она всегда будет показываться внизу приложения и не будет мешать пользователю взаимодействовать с программой. Средний доход от клика на баннерную рекламу составляет 5 тенге, допустим, что каждый пользователь в среднем кликает один раз на баннерную рекламу, значит доход с каждого активного пользователя на баннерной рекламе в день составит 5 тенге. Также в приложении будет присутствовать всплывающая реклама при входе в приложение, 2 секунды просмотра всплывающей рекламы стоит 1 тенге, в среднем всплывающая реклама длится 10 секунд, таким образом, допустим, что каждый пользователь заходит в приложение 2 раза в день, а значит что доход со всплывающей рекламы в день будет приносить 10 тенге с каждого пользователя, таким образом, общий доход со всех видов рекламы с одного пользователя составляет 15 тенге в день. Из этого сделаем вывод, что приложение будет приносить университету 7500 тенге ежедневно. Чтобы посчитать прибыль за год, исключим из счета 15 дней зимних каникул и 90 дней летних каникул. При расчете будем учитывать, что число активных пользователей приложения растет на 50 человек каждый год.

Таблица 4.3 - Исходные данные для оценки проекта

Инвестиционные затраты, тг	Значения P_t , тг.		
	1 год	2 год	3 год
$T = 0$			
1 982 626	1 950 000	2 145 000	2 340 000

Чистый дисконтированный доход (NPV)

Коэффициент дисконтирования рассчитывается по формуле

$$\alpha_t = \frac{1}{(1+r)^t} \quad (4.24)$$

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

$$PV = \sum_{t=1}^n \frac{P_t}{(1+r)^t} = \sum_{t=1}^n P_t * \alpha_t \quad (4.25)$$

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,

$$PV_1 = 1\,950\,000 * 0,8 = 1\,560\,000 \text{ тенге};$$

$$PV_2 = 2\,145\,000 * 0,64 = 1\,372\,800 \text{ тенге};$$

$$PV_3 = 2\,340\,000 * 0,512 = 1\,198\,080 \text{ тенге}.$$

Общая накопленная величина дисконтированных доходов будет равна

$$PV = 1\,560\,000 + 1\,372\,800 + 1\,198\,080 = 4\,130\,880 \text{ тенге}.$$

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

$$NPV = \sum_{t=1}^n \frac{CF_t}{(+r)^t} - I_0 \quad (4.26)$$

$$NPV = 4\,130\,880 - 1\,982\,626 = 2\,148\,254 > 0$$

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

$$PI = \sum_k \frac{P_k}{(1+r)^k} / I \quad (4.27)$$

Тогда,

$$PI = \frac{4\,130\,880}{1\,982\,626} = 2,08 > 1.$$

Дисконтированный срок окупаемости (DPP)

Срок окупаемости инвестиций - один из самых простых и широко распространен в мировой практике, не предполагает временной упорядоченности денежных поступлений [19].

Общая формула расчета показателя DPP имеет вид

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

(4.28)

где P_t - чистый денежный поток доходов

$$DPP = 1 + \frac{422\,626}{1\,372\,800} = 1,30 \text{ года.}$$

График окупаемости проекта

Заполняется таблица окупаемости проекта

Таблица 4.4 - Окупаемость проекта

Год	Доход, тенге	Расход, тенге	Прибыль/Чистый доход, тенге	Чистый доход нарастающим итогом, тенге
1	1 560 000	1 982 626	-422 626	-422 626
2	1 111 040	0	1 372 800	950 174
3	931 840	0	1 198 080	2 148 254

Строится график зависимости чистого дохода нарастающим итогом от года, данный график представлен на рисунке 4.1.

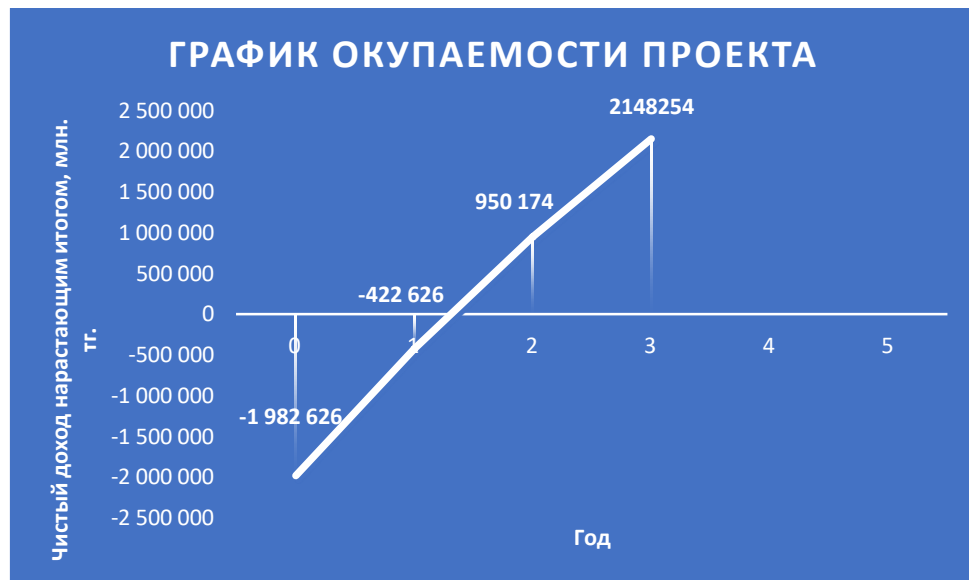


Рисунок 4.1 - График окупаемости проекта

Исходя из графика, можно определить, что срок окупаемости данного проекта приблизительно равен 1.3 года.

Заключение о принятии решения по проекту

В заключении следует отметить, что Чистый дисконтированный доход (NPV) проекта гораздо выше нуля и составляет 2 148 254 тенге. Исходя из расчетов, индекс прибыльности проекта (PI) > 1 и равняется 2,08, что также является достаточно высоким показателем. Срок окупаемости проекта составляет 1,30 года, а через 3 года доход от проекта будет составлять 2 148 254 тенге и будет расти с каждым годом, постоянно находясь в положительной динамике, за счет притока новых студентов.

5 Охрана труда и БЖД

5.1 Теоретическая часть

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

В ДИТ университета «АУЭС» выделяют следующие опасные факторы при работе в офисе:

Ухудшение зрения

Офисные работники всегда сталкиваются с проблемой ухудшения зрения на рабочем месте. Человеческий глаз не предназначен для полного рабочего дня за монитором. Для того, чтобы минимизировать риски ухудшения зрения ДИТ университета «АУЭС» использует большой ЖК экран с плоской поверхностью, норматив расстояния от глаз до монитора составляет 40 см. Также с сотрудниками регулярно проводится инструктаж, где рассказывается об упражнениях для поддержания зрения и оптимальном расстоянии от глаз до компьютера[20].

Стресс

Сотрудники, которые сталкиваются с большими требованиями и обязанностями, чем те, с которыми они могут справиться, страдают от повышенного уровня стресса, который может нанести ущерб эмоциональным и физическим реакциям работника, таким образом, создавая проблемы как для работника, так и для организации. Исследования связывают стресс на работе с неоднозначностью ролей и конфликтом ролей и показывают, что определенные факторы, такие как перегрузка работой и плохие условия труда, часто приводят к негативным последствиям для психического и физического здоровья. Согласно исследованиям, стресс является ключевым фактором низкой мотивации и морального духа, что приводит к низкой производительности, высокой текучести кадров, низкому уровню удовлетворенности работой, увеличению числа невыходов на работу и низкому качеству продуктов и услуг.[20] Поскольку стресс может напрямую влиять на организационную эффективность, иорданским компаниям необходимо определить основную причину стресса на работе и найти способы борьбы со стрессовыми факторами, которые влияют на удовлетворенность и производительность сотрудников.

В целях улучшения эмоционального и психологического состояния сотрудников, компания дает задания сотрудникам с максимальным сроком в одну неделю, также в офисе имеется комната отдыха и проводятся регулярные тимбилдинги.

Болезни позвоночника

Сидение в офисном кресле в течение длительного периода времени может вызывает боль в пояснице и приводит к болезням позвоночного отдела. Основная причина этого заключается в том, что сидение в офисном кресле или в целом представляет собой статическую позу, которая увеличивает напряжение в спине, плечах, руках и ногах, и, в частности, увеличивает давление на мышцы спины и позвоночные диски.[20] Для того, чтобы минимизировать риски возникновения такого рода проблем, университет «АУЭС» использует специальные ортопедические офисные стулья, которые полностью прилегают к позвоночнику и пояснице.

Распространение болезней

Учитывая, что работники проводят время друг с другом каждый день, для них становится неизбежным делиться вещами, даже болезнями. Простой кашель и простуда одного работника могут передаваться, когда человек каш-

ляет, чихает или разговаривает, потому что эта инфекция передается по воздуху [20].

Распространение инфекционных заболеваний создает серьезные риски для работников, поскольку это снижает их производительность и угрожает их физическому благополучию. Для устранения этого риска компания поощряет надлежащую гигиену для всех работников. Это включает прививку от инфекционных заболеваний, регулярное мытье рук и покрытие рта при кашле или чихании [20].

Кроме того, если коллега болен, его отпускают домой, чтобы сдержать болезнь. В случае возникновения вспышек у руководства имеются соответствующие планы по борьбе с пандемиями и обеспечению безопасности на рабочем месте.

5.2 Расчетная часть

5.2.1 Разработка системы естественного освещения офисного помещения

Необходимо произвести расчет естественного освещения кабинета в офисе ДИТ университета «АУЭС имени Г.Даукеева» в г.Алматы.

План офиса

В каждом кабинете работают по два человека. Общее число сотрудников - 8 человек. Площадь отсека для команды разработчиков составляет 15 м^2 , общая площадь офиса - 60 м^2 . Рабочее пространство для каждого сотрудника составляет 6 м^2 , что соответствует нормативным стандартам. Расстояние между рабочими столами составляет 2 метра, высота каждого стола - 0.7 метров. Угол наклона спинки стула составляет 30° , что позволяет не перегружать позвоночник и ослаблять нагрузку на него.



Рисунок 5.1 - Схема офисного помещения ДИТ

Таблица 5.1 - Исходные данные

Тип помещения	Параметры помещения			hок, м	hнок, м	Разряд зрит. Работ
	L, м	B, м	H, м			
Офисное помеще- ние	12	5	4	2,5	0,5	IV, б
Световой пояс	Hзд, М	Расст. до ближ. здаия P, м	Коэффициенты отражения			
			$\rho_{пот}$	$\rho_{ст}$	$\rho_{пол}$	
г.Алматы	30	18	50	30	10	

5.2.1.1 Расчет естественного освещения

Помещения с постоянным пребыванием людей должны иметь, как правило, естественное освещение.

Расчет естественного освещения заключается в определении площади световых проемов.

Общая площадь окон определяется по формуле (1.1) для бокового освещения:

$$100 \frac{S_o}{S_n} = \frac{e_n K_3 \eta_o}{\tau_o r_1} K_{30} \rightarrow S_o = \frac{S_n \cdot e_n K_3 \eta_o K_{30}}{100 \tau_o r} \quad (5.1)$$

где:

S_o - площадь световых проемов при боковом освещении, м²;

S_n - площадь пола помещения, м²;

$$S_n = L \cdot B = 12 \cdot 5 = 60 \text{ м}^2$$

e_n - нормируемое значение КЕО;

K_3 - коэффициент запаса, принимают по таблице 3.11 [21]. $K_3=1,2$;

η_o - световая характеристика окон, принимают по таблице 3.2 [21];

По условию ширина помещения ($B = 5$ м), следовательно, мы можем применить одностороннее освещение, то есть ($l = 5$ м):

$$L/l=12/5=2,4 \text{ м}$$

$$h_{расч} = h_{ок} + h_{нок} - h_{рп} = 2,5 + 0,5 - 0,5 = 2,5 \text{ м} \quad (5.2)$$

где $h_{рп} = 0,5$ м - высота рабочей поверхности

$$l/h_{расч}=5/2,5=2 \text{ м}$$

Следовательно, $\eta_o=9,1$ (по методу линейной интерполяции);

τ_o - общий коэффициент светопропускания, определяют по формуле:

$$\tau_o = \tau_1 \tau_2 \tau_3 \tau_4,$$

где τ_1 - коэффициент светопропускания материала, $\tau_1=0,8$ (для стекла оконного листового двойного);

τ_2 - коэффициент, учитывающий потери света в переплетах светопроема $\tau_2=0,6$ (переплеты деревянные двойные отдельные);

τ_3 - коэффициент, учитывающий потери света в несущих конструкциях $\tau_3=0,8$ (железобетонные и деревянные формы и арки);

τ_4 - коэффициент, учитывающий потери света в солнцезащитных устройствах, $\tau_4=1$ (убирающиеся регулируемые жалюзи и шторы);

$$\tau_o = \tau_1 \tau_2 \tau_3 \tau_4 = 0,8 * 0,6 * 0,8 * 1 = 0,384 \quad (5.3)$$

Расстояние от расчетной точки до наружной стены составит $B = 5$ м.

Отношение расстояния расчетной точки от наружной стены к глубине помещения:

$$B/l = 5/5 = 1 \text{ м} \quad (5.4)$$

Отношение глубины помещения к высоте от уровня условной рабочей поверхности верха окна:

$$\begin{aligned} l/h_{\text{расч}} &= 5/2,5 = 2 \text{ м} \\ \frac{P_{\text{пот}} + P_{\text{ст}} + P_{\text{пол}}}{3} &= \frac{50 + 30 + 10}{3} = 30 \% \\ L/l &= 12/5 = 2,4 \text{ м} \\ r_1 &= 1,5 \end{aligned} \quad (5.5)$$

r_1 - коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к зданию;

$K_{\text{зд}}$ - коэффициент, учитывающий затемнение окон противостоящими зданиями, принимают по таблице 3.8 [21]:

$$\begin{aligned} P/H_{\text{зд}} &= 18/30 = 0,6 \\ K_{\text{зд}} &= 1,64 \text{ (по методу линейной интерполяции)} \end{aligned} \quad (5.6)$$

Нормированные значения КЕО e_n для зданий, располагаемых в различных районах, следует определять по формуле:

$$e_N = e_n \cdot m_N = 1,5 \cdot 0,7 = 1,05 \quad (5.7)$$

где N - номер группы обеспеченности естественным светом по таблице 3.1[1];

e_n - значения КЕО по таблице 3.12 для IV(б) разряда зрит. работ при естественном боковом освещении;

m_N - коэффициент светового климата, табл.3.1 [21];
Подставим все значения в расчетную формулу:

$$S_o = \frac{S_n \cdot e_n \cdot K_3 \cdot \eta_o \cdot K_{3d}}{100 \tau_o r} \quad (5.8)$$

$$S_o = \frac{60 \cdot 1,05 \cdot 1,2 \cdot 9,1 \cdot 1,64}{100 \cdot 0,384 \cdot 1,5} = 19,58775 \text{ м}^2$$

$$l_{ок} = \frac{S_o}{h_{ок}} = 19,58775 / 2,5 = 7,8351 \text{ м}. \quad (5.9)$$

Таким образом, площадь светового проема составит 17,341 м² (7,8351 x 2,5) (см. рисунок 1)

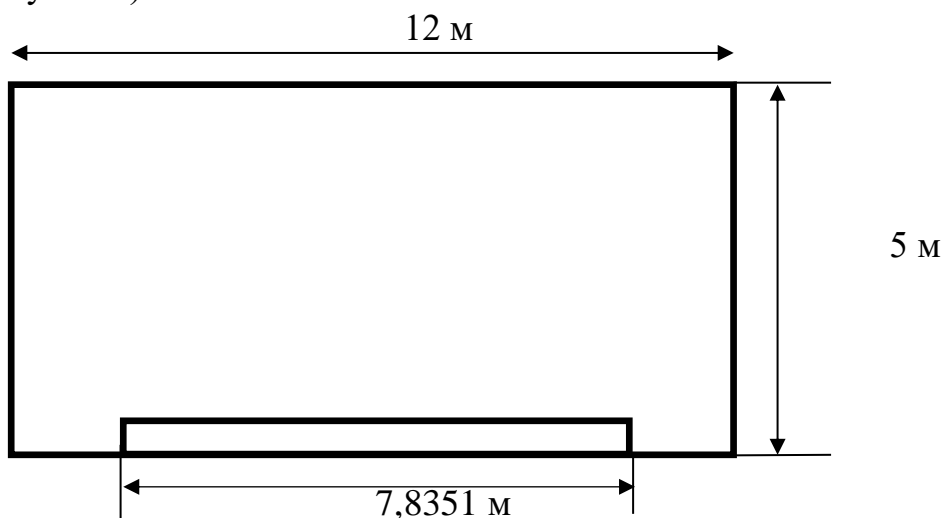


Рисунок 5.2 - Схема помещения при естественном освещении

Вывод: в данном задании была рассчитана площадь боковых световых проемов учебной лаборатории, необходимой для создания нормируемой освещенности на рабочих местах для разряда зрительной работы IV(б). Для помещения размерами 12 x 5 x 4 м³ значение для площади одного светового проема получилось равным 19,58775 м², где высота окна $h_{ок} = 2,5$ м, а длина окна $l_{ок} = 7,8351$ м.

5.2.1.2 Расчет искусственного освещения

Для расчета искусственного освещения используют один из трех методов: по коэффициенту использования светового потока, точечный и метод удельной мощности.

При расчете общего равномерного освещения основным является метод использования светового потока, создаваемого источником света, и с учетом отражения от стен, потолка, пола.

Расчет освещения начинают с выбора типа светильника, который принимается в зависимости от условий среды и класса помещений по взрывопожароопасности[22].

5. 2. 1. 3 Расчет освещенности точечным методом

Разряд зрительной работы IV (б), поэтому нормируемая освещенность (по таблице 3.12) $E_n = 200$ лк (при системе общего освещения).

Определение расчетной высоты подвеса:

$$h_{расч} = H_{помещения} - H_{свеса} - H_{р.п.}, \quad (5.10)$$

где $H_{свеса} = 0,3$ - высота свеса ламп, м;

$H_{р.п.} = 0,5$ - расстояние рабочей поверхности над полом, м;

$H_{помещения} = 4$ - высота помещения, м.

$$h_{расч} = 4 - 0,3 - 0,5 = 3,2 \text{ м}$$

Определение наивыгоднейшего расстояния между светильниками (Z): Расстояния между соседними светильниками L определяется как $L = \lambda * h$. Рекомендуемое значение λ равно 0,5-1,5 для равномерного освещения.

$$L_a = 0,9375 * 3,2 = 3 \text{ м}$$

$$l_a = 0,9375 * 3,2/2 = 1,5 \text{ м}$$

$$L_b = 0,9375 * 3,2 = 3 \text{ м}$$

$$l_b = 0,9375 * 3,2/3 = 1 \text{ м.}$$

Исходя из габаритов помещения, используем 2 ряда по 4 светильников в каждом. Расстояние между светильниками 3 метра, между рядами 3 метра, от стены до ряда 1,5 метра, от боковых стен 1 метр.

Выберем контрольную точку А (рисунок 3). Из (1)

$$E = \frac{\Phi \cdot \mu \cdot \sum_{i=1}^4 E_y}{1000 \cdot K_3}$$

(5.11)

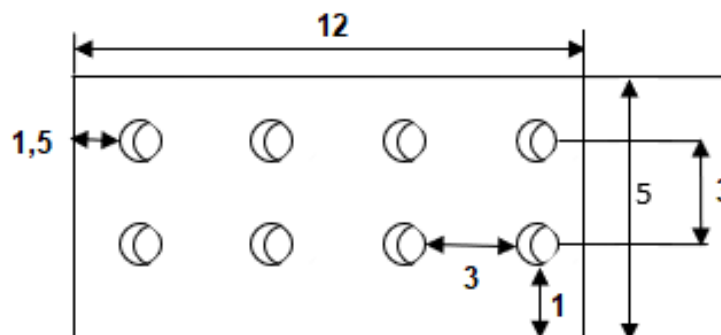


Рисунок 5.3 - Расчетная схема точечного метода

Величина условной освещенности зависит от светораспределения светильника и геометрических размеров: расстояния от точки до проекции освещающего ее светильника ($d = \sqrt{b^2 + c^2}$) и высоты расположения светильника над освещаемой поверхностью (h) (рисунок 3).

$$E_y = \frac{I_\alpha \cdot \cos^3 \alpha}{h^2}$$

(5.12)

где I_α - сила света, зависит от типа источника и угла падения света.

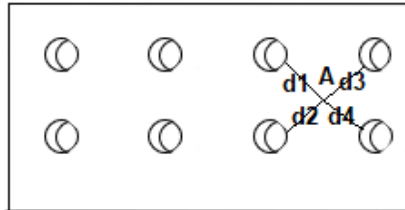


Рисунок 5.4 - Выбор точки

1,2,3,4-я лампы $d = 2,12$ м;

$$\operatorname{tg} \alpha_{1,2,3,4} = \frac{d_{1,1,2,3,4}}{h_{расч}} = \frac{2,12}{3,2} = 0 \rightarrow \alpha_{1,2,3,4} = 33,52^\circ; \quad (5.13)$$

$$\cos^3 \alpha_1 = 0,579$$

$$E_y = \frac{300 \cdot 0,579}{3,2^2} = 16,96$$

лк;

(5.14)

$$E_{уобщ} = \frac{300 \cdot 0,579}{3,2^2} * 4 = 67,85 \text{ лк}$$

Тип лампы - LINER/S DR LED 1500 W 4000K, мощность 33 Вт, световой поток 2850 лм.

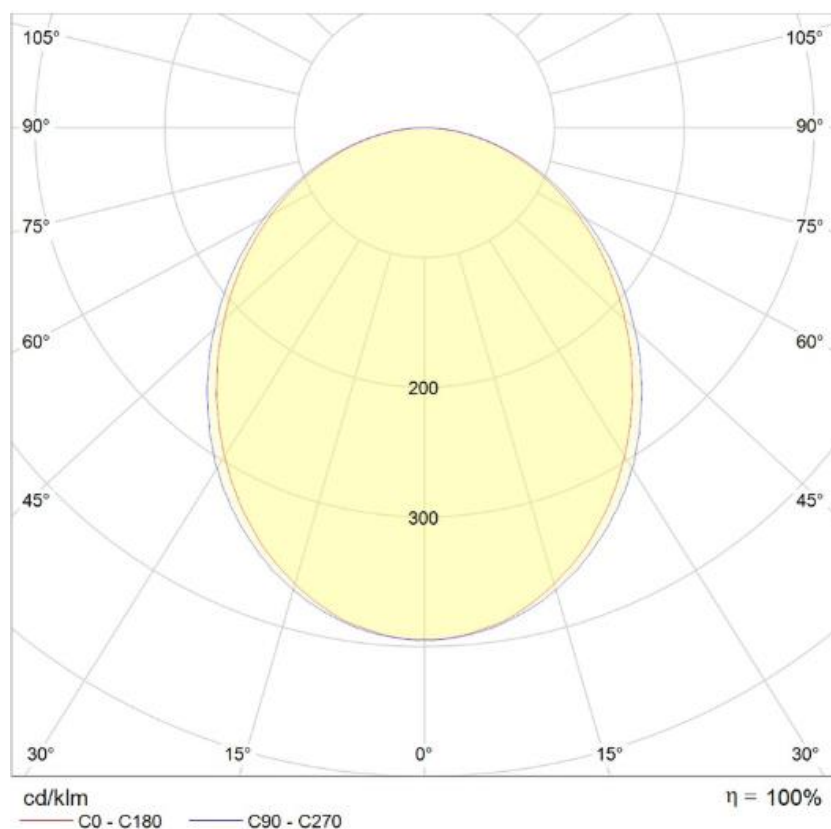


Рисунок 5.5 - КСС

$$E_{\Gamma} = \frac{2850 \cdot 1,2 \cdot 67,85}{1000 \cdot 1,1} = 210,95 \quad (5.15)$$

лк

Вывод: Освещенность на рабочем месте считается достаточной, если $E_{\Gamma} > E_{н}$; $E_{н} = 200$ лк (таблица 3.12). В данном случае условие выполняется, т.е. $E_{\Gamma} = 210,95$ лк $>$ 200 лк.

Нужно произвести расчет тепловых нагрузок помещению офиса ДИТ университета «АУЭС». На каждое помещение выделено по одному кондиционеру по принципу мульти-сплит системы.

Исходные данные

Город: Алматы;

Параметры помещения (Д x Ш x В), м: 12 x 5 x 3,2;

Данные по оборудованию: 6 шт.;

Мощность $P_{об}$, 0,5 кВт/ч;

КПД η 0,75;

Данные по ист. света: мощ. N ос.уст., 60 Вт/м²;

Вид ист. св. люминесцентные лампы

Число сотрудников, из них: мужчины 4

женщины 4

Окна: кол-во 2/2

площадь 1 окна, 4 м²

расположение С/Ю

Вид жалюзи, дерев. переплет, загрязнение незначительное

Расчетное время суток, ч.: 11-12

Температура в помещении, °С: летом 26
зимой 18

Вид положения работы: сидя

5.2.2 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)[22].

5.2.2.1 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[22]:

$$Q_{огр} = V_{пом} \cdot X_o \cdot (t_{Нрасч} - t_{Врасч}), \text{ Вт} \quad (5.16)$$

где $V_{пом}$ - объем помещения, м³:

$$V_{пом} = 12 * 5 * 3,2 = 128 \text{ м}^3; \quad (5.17)$$

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

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

$t_{Нрасч}$ - наружная температура. Для холодного периода - средняя температура самого холодного месяца в 13 часов, для теплого периода - средней температуре самого жаркого месяца в 13 часов.

$t_{\text{врасч}}$ - внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам [22].

Для теплого времени года

$$t_{\text{Нрасч}} = 31,7 \text{ }^{\circ}\text{C}$$

$$t_{\text{Врасч}} = 26 \text{ }^{\circ}\text{C}$$

$$Q_{\text{огр}} = 128 * 0,39 * (31,7 - 26) = 284,544 \text{ Дж}$$

Для холодного времени года

$$t_{\text{Нрасч}} = -22 \text{ }^{\circ}\text{C}$$

$$t_{\text{Врасч}} = 18 \text{ }^{\circ}\text{C}$$

$$Q_{\text{огр}} = 128 * 0,39 * |(-22 - 18)| = 1996,8 \text{ Дж}$$

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[22].

Теплопоступление от солнечного излучения через остекление определяется по формуле:

$$Q_p = (q^I F_0^I + q^{II} F_0^{II}) \cdot \beta_{\text{с.з.}} \quad (5.18)$$

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 [2]:

$$\beta_{\text{с.з.}} = 0,15$$

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

$$Q_p = q^I F_0 \cdot \beta_{\text{с.з.}} = (q_{\text{вп}} + q_{\text{вр}}) \cdot K_1^C \cdot K_2 \cdot \beta_{\text{с.з.}} \cdot n \cdot S_o, \text{ Вт} \quad (5.19)$$

$q_{\text{вп}}; q_{\text{вр}}$ - тепловые потоки от прямой и рассеянной радиации, Вт/м². По таблице 1 [21] для широты в 50⁰с. ш. до полудня в 11-12 ч. при расположении Ю:

$$q_{\text{вп}} = 271 \text{ Вт/м}^2; q_{\text{вр}} = 87 \text{ Вт/м}^2;$$

$F_0 = nS_o = 2 \cdot 4 = 8 \text{ м}^2$ - площадь светового проема (n - число окон; S_o - площадь 1 окна);

K_1 - коэффициент затемнения остекления переплетами (K_1^C - для облученных проемов). По таблице 3 [22]:

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

K_2 - коэффициент загрязнения остекления. По таблице 4 [22]:

$$K_2 = 0,95.$$

Тогда:

$$Q_p = (271+87)*0,75*0,95*0,15*8 = 306,09 \text{ Дж}$$

По таблице 1 [22] для широты в 50⁰с.ш. до полудня в 11-12 ч. при расположении С:

$$q_{\text{вп}} = 0 \text{ Вт/м}^2; q_{\text{вр}} = 60 \text{ Вт/м}^2;$$

$F_o = nS_o = 2*4=8 \text{ м}^2$ - площадь светового проема (n - число окон; S_o - площадь 1 окна);

Тогда:

$$Q_p = (0 + 60)*0,75*0,95*0,15*8 = 51,3 \text{ Дж}$$

Тогда общее тепlopоступление солнечного излучения с обеих окон равно:

$$Q_p = 306,09 + 51,3 = 357,39 \text{ Дж}$$

5.2.2.2 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[22].

Тепlopоступления 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 26⁰С 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:

$$Q_{\text{л}}^{\text{я}} = 61 * 3 + 61 * 3 * 0,85 = 338,55 \text{ Дж}$$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{о}} = 102 * 3 + 102 * 3 * 0,85 = 566,1 \text{ Дж}$$

По таблице [22] зимой при 18⁰С 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:

$$Q_{\text{л}}^{\text{я}} = 85 * 3 + 85 * 3 * 0,85 = 471,75 \text{ Дж}$$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{о}} = 105 * 3 + 105 * 3 * 0,85 = 582,75 \text{ Дж}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле[22]:

$$Q_{осв} = \eta \cdot N_{осв} \cdot F_{пол}, \text{ Вт} \quad (5.20)$$

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

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

$F_{пол}$ - площадь пола: $F_{пол} = 10 \cdot 4 = 40 \text{ м}^2$

Тогда:

$$Q_{осв} = 0,5 \cdot 60 \cdot 40 = 1200 \text{ Дж}$$

Тепло, выделяемое производственным оборудованием, определяется по формуле:

$$Q_{об} = N_{уст} \cdot K \quad (5.21)$$

$$Q_{об} = 0,5 \cdot 0,75 \cdot 6 = 2,25 \text{ кДж}$$

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

$$Q_{орг} = 0,3 \cdot 2,25 = 675 \text{ Дж.}$$

5.2.2.3 Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс теплоотсуплений в помещении:

$$Q_{вс\acute{a}} = Q_{\delta} + Q_{\acute{e}}^y + Q_{\acute{m}\grave{a}} + Q_{\acute{a}} + Q_{\acute{\delta}\grave{a}} + Q_{\acute{a}\grave{\delta}} \quad (5.22)$$

Лето:

$$Q_{изб} = 284,55 + 357,39 + 338,55 + 1200 + 2250 + 675 = 5105,49 \text{ Дж}$$

Зима:

$$Q_{изб} = 1996,8 + 357,39 + 471,55 + 1200 + 2250 + 675 = 6950,84 \text{ Дж}$$

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплонапряженность воздуха по формуле[22]:

$$Q_H = Q_{изб.зима} \cdot 860 / V_{пом} = 6950,84 \cdot 860 / 128 = 46,7 \text{ ккал/м}^3.$$

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

Определение количества воздуха, необходимое для поступления в помещение:

$$L = Q_{изб} * 860 / (C * \Delta t * \gamma) = 6950,84 * 860 / (0,24 * 1000 * 8 * 1,206) = 2582 \text{ м}^3/\text{час}$$

где $C=0,24$ ккал/(кг $^{\circ}$ С) - теплоемкость воздуха,
 $\gamma=1,206$ кг/м 3 - удельная масса приточного воздуха.

Определение кратности воздухообмена[22]:

$$n = L / V_{пом} = 2582 / 128 = 20,17 \text{ час}^{-1}.$$

5.2.2.4 Выбор кондиционера. Схема расположения

Исходя из полученных данных, выберем кондиционер мульти-сплит системы настенного типа.

Кондиционер SAMSUNG AR5500M



Рисунок 5.6 - Кондиционер SAMSUNG AR5500M

Таблица 5.2 - Основные характеристики

Разновидность	комплект
Площадь помещения	143 м 2
Напряжение	220 В
Мощность охлаждения	14.3 кВт
Потребляемая мощность	6.3 кВт

при охлаждении	
Режим обогрева	
Мощность обогрева	17.5 кВт
Потребляемая мощность при обогреве	5 кВт
Режим вентиляции	

Продолжение таблицы 5.2

Расход воздуха	5000 м ³ /ч
Макс. уровень шума (внутр. блок)	46 дБ
Макс. уровень шума (внеш. блок)	55 дБ
Хладагент	R407
Ширина (внутр. блок)	1120 мм
Глубина (внутр. блок)	605 мм
Высота (внутр. блок)	280 мм
Вес (внутр. блок)	70 кг
Ширина (внеш. блок)	860 мм
Глубина (внеш. блок)	325 мм
Высота (внеш. блок)	1300 мм
Вес (внеш. блок)	110 кг
Страна-производитель	Южная Корея

Заключение

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

Список использованной литературы

1 Гриффитс Дон, Гриффитс Дэвид Head First. Программирование для Android. - СПб.: Питер, 2016. - 704 с.: ил. - (Серия «Head First O'Reilly»).

2 Хорстманн К.С. Java. Библиотека Профессионала. 9-е изд. - СПб.: Питер, 2014. - 860 с.: ил. - (Серия «Библиотека Профессионала»).

3 Кригель, Алекс, Трухнов, Борис. SQL. Библия пользователя, 2-е издание.: Пер. с англ. - М.: ООО «И.Д. Вильямс», 2010. - 752 с.: ил. - Парал. тит. англ.

4 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. 2-е изд. - СПб.: Питер, 2014. - 544 с.: ил. - (Серия «Бестселлеры O'Reilly»). 6 Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие - М.: Postgress Professional, 2017. - 256 с.

5 Приложение НИЯУ МИФИ <https://play.google.com/store/apps/details?id=ru.mephi.mephi.app001&hl=ne>

6 Приложение МГИМО <https://play.google.com/store/apps/details?id=com.genisoft.launcher55193&hl=ru>

7 Приложение СФУ <https://play.google.com/store/apps/details?id=ru.app-rii.sfu&hl=ru>

8 Статья «Key Database Features» <https://www.oracle.com/technetwork/database/options/index.html>

9 Статья «The Main Features of MySQL» <https://dev.mysql.com/doc/refman/8.0/en/features.html>

10 Статья «MySQL vs Oracle» <https://www.educba.com/mysql-vs-oracle/>

11 Статья «Use Case Diagram» <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>

12 Статья «XML: Pros and Cons» <https://www.eetimes.com/xml-pros-and-cons/#>

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

14 Статья «Pros and Cons of Java» <https://www.quora.com/What-are-some-pros-and-cons-of-Java>

15 Статья «The Main Features of MySQL» <https://dev.mysql.com/doc/refman/8.0/en/features.html>

16 Статья «Key Database Features» <https://www.oracle.com/technetwork/database/options/index.html>

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

18 Официальная документация по NavigationView на сайте <https://developer.android.com/reference/android/support/design/widget/NavigationView>

19 Г.Ш.Боканова. 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.

20 Статья Кейт Инствуд «25 Types of Hazards in the Workplace And How To Prepare» на сайте <https://www.opensourcedworkplace.com/news/25-types-of-hazards-in-the-workplace-and-how-to-prepare>

21 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. - Алматы: АИЭС, 2002. - 30 с.

22 СНиП РК 2.04-01-2001. Общие строительные нормы и правила устройства систем вентиляции и кондиционирования воздуха. Административные и бытовые здания.

23 Статья «The Main Features of MySQL» на официальном сайте <https://dev.mysql.com/doc/refman/8.0/en/features.html>

24 Статья «Firebase RealTime Database with Operations in Android with Examples» на сайте <https://www.geeksforgeeks.org/firebase-realtime-database-with-operations-in-android-with-examples/>

25 Официальная документация по NavigationView на сайте <https://developer.android.com/reference/android/support/design/widget/NavigationView>

26 ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы

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

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

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

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

31 Статья Раида Масадиха «Factors that affect Employees Job Satisfaction and Performance to Increase Customers' Satisfactions» на сайте https://www.researchgate.net/publication/332471132_Factors_that_affect_Employees_Job_Satisfaction_and_Performance_to_Increase_Customers'_Satisfactions

32 <http://base.garant.ru/2306019/38d0e20d10a9099ed1e190abf152a12a/>

33 Пособие 2.91 к СНиП 2.04.05-91 «Расчет поступления теплоты солнечной радиации в помещения»

34 Статья «Поступление теплоты в помещения гражданских и производственных зданий» на сайте https://www.hvac-school.ru/biblioteka/tepl_balans/metodika_rascheta/teploti_pomeshhen/

35 Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат - Алматы: АИЭС, 2009. - 20 с.

36 СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования. - М.: Стройиздат, 2002.

37 СНиП II - 4 - 79. Естественное и искусственное освещение. Нормы проектирования.-М.: Стройиздат, 1980.- 48 с.

38 ГОСТ Р ИСО/МЭК 12119-2000, «Информационная технология. Пакеты программ. Требование к качеству и тестирование»

39 ГОСТ 28195-89 «Оценка качества программных средств. Общие положения».

40 Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. - М.: Издательский дом «Вильямс», 2014г. - 1136 с.

41 Международный стандарт ISO/IEC 9126-1:2001. Программирование. Качество продукта. Часть 1. Модель качества.

42 Липаев В. В. Технико-экономическое обоснование проектов сложных программных систем. - М.: СИНТЕГ, 2014, - 284 с.

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 Market. 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.

Продолжение приложения А

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 расписания. Также приложение будет включать общую информацию о структуре университета. Кроме того, необходимо создать облачный чат для получения новостей, связанных с форс-мажорными обстоятельствами.

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 данных, которая предназначена для хранения данных в структурах, нацеленных на расписание;
- подсистема облачного чата, для хранения и загрузки данных.
- Источниками данных для Системы должны быть:
- подсистема «Расписание» (СУБД MySQL);
- подсистема «Облачный чат» (Firebase Realtime Database от Google);

Продолжение приложения А

- общая информация о структуре университета (СУБД MySQL);
 - локализация, местоположение Google-карты.
- 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 сообщений - отображаются сообщения в чате от студентов и администрации
- 2) просмотр расписания - отображается информация о расписании занятий той группы, в которой обучается пользователь «Студент» по дням. Данная информация включает в себя время начала и конца занятия, номер занятия, дисциплину, вид занятия, преподавателя, его степень, корпус и аудиторию, в которой проходит то или иное занятие.
- 3) связь с руководством данная - страница будет включать в себя контактные данные, т.е. адрес университета, телефон для связи, электронную почту и местоположение Яндекс карты. Необходимо организовать обработку события нажатия на кнопку, при котором открывается приложение для работы с электронной почтой, адрес доставки автоматически заполняется. Работа по отправке почтового сообщения ложится на приложение по работе с почтой, установленное на мобильное устройство пользователя.

Требования к интерфейсу приложения:

- 1) вкладка Главная
 - чат.
- 2) Вкладка зависит от авторизации пользователя
 - учебный план;
 - индивидуальный учебный план.

С точки зрения пользователя «Преподаватель»:

- 1) просмотр расписания - отражается информация о расписании занятий преподавателя по дням. Данная информация включает в себя время нача-

ла и конца занятия, номер занятия, дисциплину, вид занятия, группу, корпус и аудиторию, в которой проходит то или иное занятие;

2) связь с руководством - данная страница будет включать в себя контактные данные, т.е. адрес университета, телефон для связи, электронную почту и местоположение Яндекс карты. Необходимо организовать обработку события нажатия на кнопку, при котором открывается приложение для работы с электронной почтой, адрес доставки автоматически заполняется.

Продолжение приложения А

Работа по отправке почтового сообщения ложится на приложение по работе с почтой, установленное на мобильное устройство пользователя.

С точки зрения пользователя «Сотрудник»:

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 Просмотр электронного журнала.

2) просмотр сообщений - отображаются сообщения в чате от студентов и администрации

3) телефонный справочник - отображается информация о внутренних номерах.

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.

Для облачного чата необходимо использовать модель BaaS (Backend as A Service) для связывания мобильного приложения с облачным хранилищем через API. В качестве СУБД взять облачную СУБД класса NoSQL мобильной платформы Firebase, будут использованы базы данных реального времени, в котором данные будут обрабатываться в режиме реального времени.

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. CloudActivity

```
package com.example.aues;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
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.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.ArrayList;
import java.util.Date;

public class CloudActivity extends AppCompatActivity {
    SharedPreferences mSharedPreferences = null;
    private static int MAX_MESSAGE_LENGTH = 150;
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference("messages");

    EditText mEditTextMessage;
    Button mSendButton;
    RecyclerView mMessagesRecycler;
    ArrayList<String> messages = new ArrayList<>();
    String auth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cloud);
        setTitle("Чат");
        mSharedPreferences = PreferenceManag-
```



```

er.getDefaultSharedPreferences(getBaseContext());
mEditTextMessage = (EditText)findViewById(R.id.message_input);
mSendButton = (Button)findViewById(R.id.send_message_b);
mMessagesRecycler = (RecyclerView)
findViewById(R.id.messages_recycler);
mMessagesRecycler.setLayoutManager(new LinearLayoutManager(
er(this));

```

Продолжение приложения Б

```

final DataAdapter dataAdapter = new DataAdapter(this, messages/*,
authors, dates*/);

mMessagesRecycler.setAdapter(dataAdapter);
mSendButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String msg = mEditTextMessage.getText().toString();
        Date now = new Date();
        String date = now.toString();
        auth = mSharedPreferences.getString("alias", "");
        if (msg.equals("")) {
            Toast.makeText(getApplicationContext(), "Введите
сообщение", Toast.LENGTH_SHORT).show();
            return;
        }

        if (msg.length() > MAX_MESSAGE_LENGTH) {
            Toast.makeText(getApplicationContext(), "Слишком
длинное сообщение", Toast.LENGTH_SHORT).show();
            return;
        }

        myRef.push().setValue(msg + "\n" + auth + "\n" +
date);

        // myRefAuth.push().setValue(auth);
        // myRefDate.push().setValue(date);
        mEditTextMessage.setText("");
    }
});

myRef.addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {
        String msg = dataSnapshot.getValue(String.class);
        messages.add(msg);
        dataAdapter.notifyDataSetChanged();
        mMessagesRecycler.smoothScrollToPosition(messages.size());
    }

    @Override
    public void onChildChanged(@NonNull DataSnapshot dataSnapshot,
@Nullable String s) {

```

```

        }

        @Override
        public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {

        }

        @Override
        public void onChildMoved(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}
}

```

Продолжение приложения Б

Листинг 2. Cooperation

```

package com.example.aues;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.TargetApi;
import android.os.Build;
import android.os.Bundle;
import android.webkit.WebResourceRequest;
import android.webkit.WebView;
import android.webkit.WebViewClient;

public class Cooperation extends AppCompatActivity {

    private WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_cooperation);
        setTitle("Сотрудничество");
        webView = findViewById(R.id.webView1);
        // включаем поддержку JavaScript
        webView.getSettings().setJavaScriptEnabled(true);
        // указываем страницу загрузки
        webView.loadUrl("https://ic.aues.kz/ru");
        webView.setWebViewClient(new Cooperation.MyWebViewClient());
    }

    private class MyWebViewClient extends WebViewClient {
        @TargetApi(Build.VERSION_CODES.N)
    }
}

```

```

@Override
public boolean shouldOverrideUrlLoading(WebView view, WebRe-
sourceRequest request) {
    view.loadUrl(request.getUrl().toString());
    return true;
}

// Для старых устройств

```

Продолжение приложения Б

```

@Override
public boolean shouldOverrideUrlLoading(WebView view, String
url) {
    view.loadUrl(url);
    return true;
}
@Override
public void onBackPressed() {
    if(webView.canGoBack()) {
        webView.goBack();
    } else {
        super.onBackPressed();
    }
}
}

```

Листинг 3. Example Adapter Individual Plan

```

package com.example.aues;

import android.content.Context;
import android.text.Html;
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 ExampleAdapterIndividualPlan extends RecyclerView.
Adapter<ExampleAdapterIndividualPlan.ExampleViewHolder> {
    private Context mContext;
    private ArrayList<ExampleItemIndividualPlan> mExampleList;
    private OnItemClickListener mListener;

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

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

    public ExampleAdapterIndividualPlan(Context context, Ar-

```

```

rayList<ExampleItemIndividualPlan> exampleList) {
    mContext = context;
    mExampleList = exampleList;
}

@Override
public ExampleViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {

```

Продолжение приложения Б

```

    View v = LayoutInflater.from(mContext).inflate(R.layout.example_item_individual_plan, parent, false);
    return new ExampleViewHolder(v);
}
@Override
public void onBindViewHolder(ExampleViewHolder holder, int position) {
    ExampleItemIndividualPlan currentItem = mExampleList.get(position);

    String cycle = currentItem.getCycle();
    String code = currentItem.getCode();
    String discipline = currentItem.getDiscipline();
    String credits = currentItem.getCredits();
    String teacher = currentItem.getTeacher();
    String training = currentItem.getTraining();
    String control = currentItem.getControl();

    holder.mTextViewCycle.setText("Цикл:" + System.getProperty("line.separator") + cycle);
    holder.mTextViewCode.setText("Код:" + System.getProperty("line.separator") + code);
    holder.mTextViewDiscipline.setText(discipline);
    holder.mTextViewCredits.setText(credits + " кредит(а/ов)");
    holder.mTextViewTeacher.setText("Преподаватель:" + System.getProperty("line.separator") + teacher);
    holder.mTextViewControl.setText("Вид контроля:" + System.getProperty("line.separator") + control);
    holder.mTextViewTraining.setText("Вид занятия:" + System.getProperty("line.separator") + training);
}

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

public void filterList(ArrayList<ExampleItemIndividualPlan> filteredList) {
    mExampleList = filteredList;
    notifyDataSetChanged();
}

public class ExampleViewHolder extends RecyclerView.ViewHolder {
    public TextView mTextViewCycle;

```

```

public TextView mTextViewCode;
public TextView mTextViewDiscipline;
public TextView mTextViewCredits;
public TextView mTextViewTeacher;
public TextView mTextViewTraining;
public TextView mTextViewControl;

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

```

Продолжение приложения Б

```

        mTextViewCycle = itemView.findViewById(R.id.cycle);
        mTextViewCode =
itemView.findViewById(R.id.code_discipline);
        mTextViewDiscipline =
itemView.findViewById(R.id.name_of_discipline);
        mTextViewCredits = itemView.findViewById(R.id.credits);
        mTextViewTraining =
itemView.findViewById(R.id.name_of_training);
        mTextViewTeacher = itemView.findViewById(R.id.teacher);
        mTextViewControl = itemView.findViewById(R.id.control);

        itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (mListener != null) {
                    int position = getAdapterPosition();
                    if (position != RecyclerView.NO_POSITION) {
                        mListener.onItemClick(position);
                    }
                }
            }
        });
    }
}
}
}

```

Листинг 4. Example Adapter Schedule

```

package com.example.aues;

import android.content.Context;
import android.text.Html;
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 ExampleAdapterSchedule extends RecyclerView-
Adapter<ExampleAdapterSchedule.ExampleViewHolder> {

```

```

private Context mContext;
private ArrayList<ExampleItemSchedule> mExampleList;
private OnItemClickListener mListener;

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

```

Продолжение приложения Б

```

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

public ExampleAdapterSchedule(Context context, Ar-
rayList<ExampleItemSchedule> exampleList) {
    mContext = context;
    mExampleList = exampleList;
}

@Override
public ExampleViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
    View v = LayoutInflater.
from(mContext).inflate(R.layout.example_item_schedule, parent,
false);
    return new ExampleViewHolder(v);
}

@Override
public void onBindViewHolder(ExampleViewHolder holder, int posi-
tion) {
    ExampleItemSchedule currentItem = mExampleList.get(position);

    String time = currentItem.getTime();
    String training = currentItem.getTraining();
    String teacher = currentItem.getTeacher();
    String audience = currentItem.getAudience();

    holder.mTextViewTime.setText(Html.fromHtml(time));
    holder.mTextViewTraining.setText(Html.fromHtml(training));
    holder.mTextViewTeacher.setText(Html.fromHtml(teacher));
    holder.mTextViewAudience.setText(audience);
}

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

public class ExampleViewHolder extends RecyclerView.ViewHolder {
    public TextView mTextViewTime;
}

```

```

public TextView mTextViewTraining;
public TextView mTextViewTeacher;
public TextView mTextViewAudience;

public ExampleViewHolder(View itemView) {
    super(itemView);
    mTextViewTime = itemView.findViewById(R.id.time);
    mTextViewTraining = itemView.findViewById(R.id.disc);
    mTextViewTeacher = itemView.findViewById(R.id.teacher);

```

Продолжение приложения Б

```

    mTextViewAudience = itemView.findViewById(R.id.audience);

    itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mListener != null) {
                int position = getAdapterPosition();
                if (position != RecyclerView.NO_POSITION) {
                    mListener.onItemClick(position);
                }
            }
        }
    });
}
}
}
}
}

```

Листинг 5. Feedback

```

package com.example.aues;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

public class feedback extends Fragment {
    Context context;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View rootView = inflater.inflate(R.layout.fragment_feedback,
            container, false);
    }
}

```

```

        context = rootView.getContext();
        Button button_send_msg = (Button) rootView.findViewById(R.id.send_msg);
        button_send_msg.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) {
                String[] addresses = new String[1];
                addresses[0] = "auesmobile@gmail.com";
                Intent intent = new Intent(Intent.ACTION_SENDTO);
                intent.setData(Uri.parse("mailto:"));
                Продолжение приложения Б

                intent.putExtra(Intent.EXTRA_EMAIL, addresses);
                startActivity(Intent.createChooser(intent, "Send
Email"));
            }
        });
        return rootView;
    }
}

```

Листинг 6. Geo-position

```

package com.example.aues;

import android.content.Context;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

public class GeoPosition extends Fragment implements
OnMapReadyCallback {
    GoogleMap map;

    Context context;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View rootView = inflater.inflate(R.layout.fragment_geo_position, container, false);

        SupportMapFragment mapFragment = (SupportMapFragment)

```



```

this.getChildFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync(this);

context = rootView.getContext();
return rootView;
}

@Override
public void onMapReady(GoogleMap googleMap) {
    Продолжение приложения Б

    map = googleMap;
    LatLng AUES = new LatLng(43.234521, 76.930922);
    map.addMarker(new MarkerOptions().position(AUES).title("АУЭС"));
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(AUES, 15));
    // map.animateCamera(CameraUpdateFactory.zoomTo(10.0f));
}
}

```

Листинг 7. Individual Plan

```

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.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;

public class IndividualPlan extends AppCompatActivity {

```

```

private RecyclerView mRecyclerView;
private ExampleAdapterIndividualPlan mExampleAdapter;
private ArrayList<ExampleItemIndividualPlan> mExampleList;
private RequestQueue mRequestQueue;
private EditText mEditText;
ArrayList<ExampleItemIndividualPlan> filteredList;
String url =
"https://auesmobiledatabase.000webhostapp.com/individual_plan.php";
String id_student;
SharedPreferences mSharedPreferences;

```

Продолжение приложения Б

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_individual_plan);
    setTitle("Индивидуальный план");

    mRecyclerView = findViewById(R.id.recycler_view_iup);
    mRecyclerView.setHasFixedSize(true);
    mRecyclerView.setLayoutManager(new LinearLayoutManager(
IndividualPlan.this));

    mExampleList = new ArrayList<>();

    mRequestQueue = Volley.newRequestQueue(IndividualPlan.this);

    mEditText = findViewById(R.id.search_semester);
    mEditText.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start,
int count, int after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int
before, int count) {

        }

        @Override
        public void afterTextChanged(Editable s) {
            filter(s.toString());
        }
    });

    mSharedPreferences= PreferenceManag-
er.getDefaultSharedPreferences(getApplicationContext());
    id_student = mSharedPreferences.getString("id_student", "");

    parseJSON();

}
private void filter(String text) {
    filteredList = new ArrayList<>();

```

```

        for (ExampleItemIndividualPlan item : mExampleList) {
            if
(item.getCredits().toLowerCase().contains(text.toLowerCase())) {
                filteredList.add(item);
            }
        }
        mExampleAdapter.filterList(filteredList);
    }
}
private void parseJSON() {

```

Продолжение приложения Б

```

StringRequest request = new StringRequest(Request.Method.POST,
url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonObject = new JSONOb-
ject(response);
                    JSONArray jsonArray = jsonOb-
ject.getJSONArray("individual_plan");
                    for (int i = 0; i < jsonArray.length();
i++) {
                        JSONObject individual_plan = jso-
nArray.getJSONObject(i);
                        String cycle = individu-
al_plan.getString("CYCLE");
                        String code = individu-
al_plan.getString("DISCIPLINE");
                        String discipline = individu-
al_plan.getString("NAME_DISCIPLINE");
                        String credits = individu-
al_plan.getString("CREDITS");
                        String training = individu-
al_plan.getString("TRAINING");
                        String teacher = individu-
al_plan.getString("TEACHER");
                        String control = individu-
al_plan.getString("CONTROL");

                        mExampleList.add(new ExampleItemIndi-
vidualPlan(cycle, code, discipline, credits, training, teacher, con-
trol));
                    }

                    mExampleAdapter = new ExampleAdapterIndi-
vidualPlan(IndividualPlan.this, mExampleList);
                    mRecyclerView.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 AuthFail-
ureError {

```

Продолжение приложения Б

```

        Map<String, String> params = new HashMap<String, String>();
        params.put("id_student", id_student);

        return params;
    }
};

mRequestQueue.add(request);
}
}

```

Листинг 8. Info University

```

package com.example.aues;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.RelativeLayout;

public class info_university extends Fragment {
    Context context;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup con-
tainer,
                               Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View rootView = inflat-
er.inflate(R.layout.fragment_info_university, container, false);
        context = rootView.getContext();

        RelativeLayout button = (RelativeLayout) root-
View.findViewById(R.id.layout);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context, train-
ingUnits.class);
                startActivity(intent);
            }
        });
    }
}

```

```

    }
    });
    RelativeLayout teacher = (RelativeLayout) rootView.findViewById(R.id.teacher_staff);
    teacher.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(context, teachers_staff.class);
            startActivity(intent);
        }
    });
    RelativeLayout history = (RelativeLayout) rootView.findViewById(R.id.history);
    history.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(context, History.class);
            startActivity(intent);
        }
    });
    RelativeLayout infrastructure = (RelativeLayout) rootView.findViewById(R.id.infra);
    infrastructure.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(context, infrastructure.class);
            startActivity(intent);
        }
    });
    RelativeLayout leadership = (RelativeLayout) rootView.findViewById(R.id.leadership_layout);
    leadership.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(context, leadership.class);
            startActivity(intent);
        }
    });
    return rootView;
}
}

```

Продолжение приложения Б

Листинг 9. Monday

```

package com.example.aues;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

```

```
import android.preference.PreferenceManager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
```

```
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
```

Продолжение приложения Б

```
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 monday extends Fragment{
    SharedPreferences mSharedPreference;
    int role_id;
    String id_group;
    String id_day;
    String id_teacher;
    Boolean any_group = false;
    Boolean any_teacher = false;

    public monday(String day, String group) {
        id_day = day;
        id_group = group;
    }

    public monday(String day, int teacher) {
        id_day = day;
        id_teacher = Integer.toString(teacher);
    }

    public monday(String day, String group, Boolean any_group) {
        id_day = day;
        id_group = group;
        this.any_group = any_group;
    }

    public monday(String day, int teacher, Boolean any_teacher) {
        id_day = day;
        id_teacher = Integer.toString(teacher);
        this.any_teacher = any_teacher;
    }
}
```

```

Context context;

private TextView empty;
private RecyclerView mRecyclerView;
private ExampleAdapterSchedule mExampleAdapter;
private ArrayList<ExampleItemSchedule> mExampleList;
private RequestQueue mRequestQueue;
String url =
"https://auesmobiledatabase.000webhostapp.com/schedule.php";

```

Продолжение приложения Б

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
// Inflate the layout for this fragment
View rootView = inflater.inflate(R.layout.fragment_monday,
container, false);
context = rootView.getContext();
mRecyclerView = rootView.findViewById(R.id.recycler_view_schedule_monday);
mRecyclerView.setHasFixedSize(true);
mRecyclerView.setLayoutManager(new LinearLayoutManager(context));
empty = (TextView)rootView.findViewById(R.id.empty_day);
mExampleList = new ArrayList<>();

mSharedPreferences= PreferenceManager.getDefaultSharedPreferences(getActivity().getBaseContext());
role_id = mSharedPreferences.getInt("role_id", 0);

mRequestQueue = Volley.newRequestQueue(context);
parseJSON();
return rootView;
}

private void parseJSON() {
StringRequest request = new StringRequest(Request.Method.POST,
url,
new Response.Listener<String>() {
@Override
public void onResponse(String response) {
try {
if (response.equals("{\"schedule\":[]}"))
{
empty.setVisibility(View.VISIBLE);
}
else {
JSONObject jsonObject= new JSONObject(response.toString());
JSONArray jsonArray = jsonObject.getJSONArray("schedule");
for (int i = 0; i < jsonArray.length(); i++) {
JSONObject schedule = jsonObject.getJSONArray(i);
}
}
}
}
}
}
}

```

```

ule.getString("TIME");
ule.getString("TRAINING");
ule.getString("NAME");
ule.getString("ID_AUDIENCE");
String time = sched-
String training = sched-
String teacher = sched-
String audience = sched-

```

Продолжение приложения Б

```

        mExampleList.add(new ExampleItemSchedule(
            time, training, teacher, audience));
    }
}

mExampleAdapter = new ExampleAdapterSchedule(
    context, mExampleList);
mRecyclerView.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("id_day", id_day);
    params.put("role_id", Integer.toString(role_id));
    params.put("any_group", Boolean.toString(any_group));
    params.put("any_teacher", Boolean.toString(any_teacher));
    if (any_group == false && any_teacher == false) {
        if (role_id == 2) {
            params.put("id_group", id_group);
        }
        else if (role_id == 3) {
            params.put("id_teacher", id_teacher);
        }
    }
    else if (any_group) {
        params.put("id_group", id_group);
    }
    else if (any_teacher) {

```



```

        params.put("id_teacher", id_teacher);
    }

    return params;
};

mRequestQueue.add(request);
}

```

Продолжение приложения Б

```

}

```

Листинг 10. Schedule

```

package com.example.aues;

import android.content.SharedPreferences;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import com.google.android.material.tabs.TabLayout;
import androidx.fragment.app.Fragment;
import androidx.viewpager.widget.ViewPager;

import android.preference.PreferenceManager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class schedule extends Fragment {
    SharedPreferences mSharedPreferences = null;
    int role_id;
    String group_id;
    int id_teacher;
    Boolean any_group = false;
    Boolean any_teacher = false;

    public schedule(String group) {
        group_id = group;
    }

    public schedule(int teacher) {
        id_teacher = teacher;
    }

    public schedule(String group, Boolean any_group) {
        group_id = group;
        this.any_group = any_group;
    }

    public schedule(int teacher, Boolean any_teacher) {
        id_teacher = teacher;
        this.any_teacher = any_teacher;
    }
}

```

```

private TabAdapter adapter;
private TabLayout tableLayout;
private ViewPager viewPager;

@Nullable
@Override
public View onCreateView(@NonNull LayoutInflater inflater,
@Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
    // Inflate the layout for this fragment

```

Продолжение приложения Б

```

View view = inflater.inflate(R.layout.fragment_schedule, con-
tainer, false);
viewPager = view.findViewById(R.id.viewpager_schedule);
tableLayout = view.findViewById(R.id.result_tabs_schedule);
adapter = new TabAdapter(getFragmentManager());

mSharedPreferences = PreferenceManag-
er.getDefaultSharedPreferences(getActivity());
role_id = mSharedPreferences.getInt("role_id", 0);
if (role_id == 2 && any_group == false && any_teacher ==
false) {
    adapter.addFragment(new monday("1", group_id), "Понедель-
ник");
    adapter.addFragment(new monday("2", group_id), "Вторник");
    adapter.addFragment(new monday("3", group_id), "Среда");
    adapter.addFragment(new monday("4", group_id), "Четверг");
    adapter.addFragment(new monday("5", group_id), "Пятница");
    adapter.addFragment(new monday("6", group_id), "Суббота");
} else if (role_id == 3 && any_group == false && any_teacher
== false) {
    adapter.addFragment(new monday("1", id_teacher),
        "Понедельник");
    adapter.addFragment(new monday("2", id_teacher),
        "Вторник");
    adapter.addFragment(new monday("3", id_teacher), "Среда");
    adapter.addFragment(new monday("4", id_teacher),
        "Четверг");
    adapter.addFragment(new monday("5", id_teacher),
        "Пятница");
    adapter.addFragment(new monday("6", id_teacher),
        "Суббота");
    } else if (any_group) {
    adapter.addFragment(new monday("1", group_id, true), "По-
недельник");
    adapter.addFragment(new monday("2", group_id, true),
        "Вторник");
    adapter.addFragment(new monday("3", group_id, true), "Сре-
да");
    adapter.addFragment(new monday("4", group_id, true), "Чет-
верг");
    adapter.addFragment(new monday("5", group_id, true), "Пят-
ница");
    adapter.addFragment(new monday("6", group_id, true), "Суб-
бота");
}

```

```

        } else if (any_teacher) {
adapter.addFragment(new monday("1", id_teacher, true),
        "Понедельник");
adapter.addFragment(new monday("2", id_teacher, true),
        "Вторник");
adapter.addFragment(new monday("3", id_teacher, true),
        "Среда");
adapter.addFragment(new monday("4", id_teacher, true),
        "Четверг");

```

Продолжение приложения Б

```

adapter.addFragment(new monday("5", id_teacher, true), "Пятница");
adapter.addFragment(new monday("6", id_teacher, true),
"Суббота");
    }

    viewPager.setAdapter(adapter);
    tableLayout.setupWithViewPager(viewPager);
    return view;
}
}

```

Листинг 11. Schedule For Group

```

package com.example.aues;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

import android.content.Intent;
import android.os.Bundle;

import static com.example.aues.Groups.EXTRA_TITLE;

public class schedule_for_group extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_schedule_for_group);

        Intent intent = getIntent();
        final String group_name = intent.getStringExtra(EXTRA_TITLE);
        setTitle("Расписание группы " + group_name);
        Fragment fragment = null;
        FragmentManager fm = getSupportFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();
        fragment = new schedule(group_name, true);
        ft.replace(R.id.frag, fragment);
        ft.commit();
    }
}

```

Листинг 12. Schedule For Teacher

```
package com.example.aues;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;

    Продолжение приложения Б

import android.content.Intent;
import android.os.Bundle;

import static com.example.aues.Teachers.EXTRA_ID;
import static com.example.aues.Teachers.EXTRA_TEACHER_NAME;

public class schedule_for_teacher extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_schedule_for_teacher);

        Intent intent = getIntent();
        final int teacher_id = Integer.parseInt(intent.getStringExtra(EXTRA_ID));
        final String teacher_name = intent.getStringExtra(EXTRA_TEACHER_NAME);
        setTitle(teacher_name);
        Fragment fragment = null;
        FragmentManager fm = getSupportFragmentManager();
        FragmentTransaction ft = fm.beginTransaction();
        fragment = new schedule(teacher_id, true);
        ft.replace(R.id.frag_teacher, fragment);
        ft.commit();
    }
}
```

Листинг 13. Teachers Staff

```
package com.example.aues;

import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentTransaction;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.TargetApi;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebResourceRequest;
import android.webkit.WebView;
```

```

import android.webkit.WebViewClient;
import android.widget.Button;

public class teachers_staff extends AppCompatActivity {
    private WebView webView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_teachers_staff);
        setTitle(R.string.teaching_staff);

        Продолжение приложения Б

        webView = findViewById(R.id.webView);
        // включаем поддержку JavaScript
        webView.getSettings().setJavaScriptEnabled(true);
        // указываем страницу загрузки
        webView.loadUrl("https://aues.kz/ru/site/pps");
        webView.setWebViewClient(new MyWebViewClient());
    }

    private class MyWebViewClient extends WebViewClient {
        @TargetApi(Build.VERSION_CODES.N)
        @Override
        public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {
            view.loadUrl(request.getUrl().toString());
            return true;
        }

        // Для старых устройств
        @Override
        public boolean shouldOverrideUrlLoading(WebView view, String url) {
            view.loadUrl(url);
            return true;
        }
    }
    @Override
    public void onBackPressed() {
        if(webView.canGoBack()) {
            webView.goBack();
        } else {
            super.onBackPressed();
        }
    }
}

```

Листинг 14. Individual Plan

```

<?php
require "int.php";
mysqli_set_charset($conn, 'utf8');
$id_student = (int)$_POST["id_student"];
$sql = "SELECT CYCLE.SHORT_NAME AS CYCLE, INDIVIDUAL_PLAN.CODE_DISCIPLINE AS DISCIPLINE, DISCIPLINE.NAME_OF_DISCIPLINE AS NAME_DISCIPLINE, DISCIPLINE_CLASS.CREDITS AS CREDITS, TRAIN-

```

```

ING.NAME_OF_TRAINING AS TRAINING, CONCAT(TEACHER.SURNAME, ' ', TEACH-
ER.NAME, ' ', TEACHER.PATRONYMIC) AS TEACHER, DISCI-
PLINE_CLASS.FORM_OF_CONTROL AS CONTROL FROM INDIVIDUAL_PLAN JOIN TEACH-
ER_TRAINING ON INDIVIDUAL_PLAN.ID_GROUP = TEACHER_TRAINING.ID_GROUP AND
INDIVIDUAL_PLAN.CODE_DISCIPLINE = TEACHER_TRAINING.CODE_DISCIPLINE AND
INDIVIDUAL_PLAN.ID_TRAINING = TEACHER_TRAINING.ID_TRAINING JOIN
TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACHER.ID_TEACHER
JOIN TRAINING ON

```

Продолжение приложения Б

```

TEACHER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
AND TEACHER_TRAINING.ID_GROUP = TRAINING.ID_GROUP AND TEACH-
ER_TRAINING.ID_TRAINING = TRAINING.ID_TRAINING
JOIN DISCIPLINE_CLASS ON DISCIPLINE_CLASS.CODE_DISCIPLINE =
TRAINING.CODE_DISCIPLINE AND DISCIPLINE_CLASS.ID_GROUP = TRAIN-
ING.ID_GROUP
JOIN DISCIPLINE ON DISCIPLINE_CLASS.CODE_DISCIPLINE = DISCI-
PLINE.CODE_DISCIPLINE
JOIN CYCLE ON CYCLE.ID_CYCLE = DISCIPLINE.ID_CYCLE
WHERE INDIVIDUAL_PLAN.ID_STUDENT = '". $id_student.'" ORDER BY
DISCIPLINE DESC;";
$result = mysqli_query($conn, $sql);
$json_array = array();
while ($row = mysqli_fetch_assoc($result))
{
    $json_array[] = $row;
}
echo json_encode(array('individual_plan' => $json_array),
JSON_UNESCAPED_UNICODE, JSON_FORCE_OBJECT);
mysqli_close($conn);
?>

```

Листинг 15. Schedule

```

<?php
require "int.php";
$id_day = (int)$_POST["id_day"];
$role_id = (int)$_POST["role_id"];
$any_group = $_POST["any_group"];
$any_teacher = $_POST["any_teacher"];
$year = date("Y");
$academic_year = "";
$academic_period = "";
$month = date("m");
$sql = "";
if ($month >=8) {
    $year = date("Y") + 1;
    $academic_year = date("Y") . "-" . $year;
    $academic_period = 1;
}
else if ($month < 8) {
    $year = date("Y") - 1;
    $academic_year = $year . "-" . date("Y");
    $academic_period = 2;
}

```

```

    }

    if ($any_group == "true") {
        $id_group = $_POST["id_group"];
        $sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>',
TIME.TIME_END, '<br>', '(', COALESCE(SCHEDULE.SUB_GROUP, ''), ')'),
'(',')' AS TIME,
CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(', LOWER
(TRAINING.SHORT_NAME), ')) AS TRAINING,

```

Продолжение приложения Б

```

CONCAT(TEACHER.SURNAME, ' ', LEFT(TEACHER.NAME, 1), '. ',
LEFT(TEACHER.PATRONYMIC, 1), '.', '<br>',
DEGREE.SHORT_NAME) AS NAME,
AUDIENCE.ID_AUDIENCE
FROM
DISCIPLINE
INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCI-
PLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCI-
PLINE_CLASS.ID_GROUP
AND TRAINING.CODE_DISCIPLINE = DIS-
CIPLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING =
TRAINING.ID_TRAINING
AND TEACH-
ER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
AND TEACH-
ER_TRAINING.ID_GROUP = TRAINING.ID_GROUP
INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACH-
ER_TRAINING.ID_GROUP
AND SCHEDULE.CODE_DISCIPLINE =
TEACHER_TRAINING.CODE_DISCIPLINE
AND SCHEDULE.ID_TRAINING = TEACH-
ER_TRAINING.ID_TRAINING
INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACH-
ER.ID_TEACHER
INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE
INNER JOIN DEGREE ON DEGREE.ID_DEGREE = TEACHER.ID_DEGREE
INNER JOIN CLASS ON CLASS.ID_GROUP = SCHEDULE.ID_GROUP
WHERE
DISCIPLINE_CLASS.ACADEMIC_YEAR = ".$academic_year."
AND CLASS.NAME_OF_GROUP = ".$id_group."
AND DAY.ID_DAY = ".$id_day."
AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = ".$academic_period."
ORDER BY
TIME.ID_TIME;";
    }
    else if ($any_teacher == "true") {
        $id_teacher = (int)$_POST["id_teacher"];
        $sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>',
TIME.TIME_END, '<br>', '(', COALESCE(SCHEDULE.SUB_GROUP, ''), ')'),
'(',')' AS TIME,

```

```

        CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(' , LOWER
        (TRAINING.SHORT_NAME), ')') AS TRAINING,
        CLASS.NAME_OF_GROUP AS NAME,
        AUDIENCE.ID_AUDIENCE
FROM
    DISCIPLINE
    INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCI-
    PLINE_CLASS.CODE_DISCIPLINE
    INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCI-
    PLINE_CLASS.ID_GROUP

```

Продолжение приложения Б

```

        AND TRAINING.CODE_DISCIPLINE = DIS-
        CIPLINE_CLASS.CODE_DISCIPLINE
        INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING =
        TRAINING.ID_TRAINING
        AND TEACH-
        ER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
        AND TEACH-
        ER_TRAINING.ID_GROUP = TRAINING.ID_GROUP
        INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACH-
        ER_TRAINING.ID_GROUP
        AND SCHEDULE.CODE_DISCIPLINE =
        TEACHER_TRAINING.CODE_DISCIPLINE
        AND SCHEDULE.ID_TRAINING = TEACH-
        ER_TRAINING.ID_TRAINING
        INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
        INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
        INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACH-
        ER.ID_TEACHER
        INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE
        INNER JOIN CLASS ON DISCIPLINE_CLASS.ID_GROUP = CLASS.ID_GROUP
WHERE
    DISCIPLINE_CLASS.ACADEMIC_YEAR = ".$academic_year."
    AND TEACHER.ID_TEACHER= ".$id_teacher."
    AND DAY.ID_DAY = ".$id_day."
    AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = ".$academic_period."
ORDER BY
    TIME.ID_TIME;";
    }
    else {
        if ($role_id == 2) {
            $id_group = $_POST["id_group"];
            $sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>',
            TIME.TIME_END, '<br>', '(' , COALESCE(SCHEDULE.SUB_GROUP, ''), ')'),
            '()', '') AS TIME,
            CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(' , LOW-
            ER(TRAINING.SHORT_NAME), ')') AS TRAINING,
            CONCAT(TEACHER.SURNAME, ' ', LEFT(TEACHER.NAME, 1), '. ',
            LEFT(TEACHER.PATRONYMIC, 1), '.', '<br>',
            DEGREE.SHORT_NAME) AS NAME,
            AUDIENCE.ID_AUDIENCE
FROM
    DISCIPLINE
    INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCI-
    PLINE_CLASS.CODE_DISCIPLINE

```



```

INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCI-
PLINE_CLASS.ID_GROUP
AND TRAINING.CODE_DISCIPLINE = DIS-
CIPLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING =
TRAINING.ID_TRAINING
AND TEACH-
ER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
AND TEACH-
ER_TRAINING.ID_GROUP = TRAINING.ID_GROUP

```

Продолжение приложения Б

```

INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACH-
ER_TRAINING.ID_GROUP
AND SCHEDULE.CODE_DISCIPLINE =
TEACHER_TRAINING.CODE_DISCIPLINE
AND SCHEDULE.ID_TRAINING = TEACH-
ER_TRAINING.ID_TRAINING
INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACH-
ER.ID_TEACHER
INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE
INNER JOIN DEGREE ON DEGREE.ID_DEGREE = TEACHER.ID_DEGREE
INNER JOIN CLASS ON CLASS.ID_GROUP = SCHEDULE.ID_GROUP
WHERE
DISCIPLINE_CLASS.ACADEMIC_YEAR = '$academic_year.'"
AND CLASS.NAME_OF_GROUP = '$id_group.'"
AND DAY.ID_DAY = '$id_day.'"
AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = '$academic_period.'"
ORDER BY
TIME.ID_TIME;";
}
else if ($role_id == 3) {
$id_teacher = (int)$_POST["id_teacher"];
$sql = "SELECT REPLACE(CONCAT(TIME.TIME_START, '<br>',
TIME.TIME_END, '<br>', '(', COALESCE(SCHEDULE.SUB_GROUP, ''), ')),
'()', '') AS TIME,
CONCAT(DISCIPLINE.NAME_OF_DISCIPLINE, '<br>', '(', LOW-
ER(TRAINING.SHORT_NAME), ')) AS TRAINING,
CLASS.NAME_OF_GROUP AS NAME,
AUDIENCE.ID_AUDIENCE
FROM
DISCIPLINE
INNER JOIN DISCIPLINE_CLASS ON DISCIPLINE.CODE_DISCIPLINE = DISCI-
PLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TRAINING ON TRAINING.ID_GROUP = DISCI-
PLINE_CLASS.ID_GROUP
AND TRAINING.CODE_DISCIPLINE = DIS-
CIPLINE_CLASS.CODE_DISCIPLINE
INNER JOIN TEACHER_TRAINING ON TEACHER_TRAINING.ID_TRAINING =
TRAINING.ID_TRAINING
AND TEACH-
ER_TRAINING.CODE_DISCIPLINE = TRAINING.CODE_DISCIPLINE
AND TEACH-
ER_TRAINING.ID_GROUP = TRAINING.ID_GROUP

```

```

INNER JOIN SCHEDULE ON SCHEDULE.ID_GROUP = TEACH-
ER_TRAINING.ID_GROUP
                                AND SCHEDULE.CODE_DISCIPLINE =
TEACHER_TRAINING.CODE_DISCIPLINE
                                AND SCHEDULE.ID_TRAINING = TEACH-
ER_TRAINING.ID_TRAINING
INNER JOIN TIME ON SCHEDULE.ID_TIME = TIME.ID_TIME
INNER JOIN DAY ON DAY.ID_DAY = SCHEDULE.ID_DAY
INNER JOIN TEACHER ON TEACHER_TRAINING.ID_TEACHER = TEACH-
ER.ID_TEACHER
INNER JOIN AUDIENCE ON SCHEDULE.ID_AUDIENCE = AUDIENCE.ID_AUDIENCE

```

Продолжение приложения Б

```

INNER JOIN CLASS ON DISCIPLINE_CLASS.ID_GROUP = CLASS.ID_GROUP
WHERE
DISCIPLINE_CLASS.ACADEMIC_YEAR = ".$academic_year."
AND TEACHER.ID_TEACHER= ".$id_teacher."
AND DAY.ID_DAY = ".$id_day."
AND DISCIPLINE_CLASS.ACADEMIC_PERIOD = ".$academic_period."
ORDER BY
TIME.ID_TIME;";
}
}

```

```

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

```

?>

ПРИЛОЖЕНИЕ В
Акт внедрения

ПРИЛОЖЕНИЕ Г
Авторское свидетельство

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ  РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ
МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

ҚУӘЛІК
2020 жылғы «27» ақпан № 8425

Автордың (арзын) аты, оты, әкесінің аты (егер-ол жеке басын куәландыратын құжатта көрсетілсе):
ТҮСУПОВА БАТІА БҒЯЛЕВНА

Авторлық құқық объектісі: **3EM-ге арналған бұлааралма**

Объектінің атауы: **Мәби қымыс приложенис «AUES mobile»**

Объектіні алынған күні: **18.06.2019**



Ақпарат табу үшін мына Интернет мекендігінен қараңыз:
"Авторлық құқық" Бөліміне: www.kazpatent.kz

Подлинность документа можно проверить на сайте www.kazpatent.kz
в разделе «Авторское право» ИФРР Государственной АИДР РК

Подписано ЗЦП Абулкаиров Н.А.

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):

ТРАСУПОВА БЕЗЗА БИЯЗЕВНА

Вид объекта авторского права: **программа для ЭВМ**

Название объекта: **Мобильное приложение «AUES mobile»**

Дата создания объекта: **18.06.2019**



Срок: бессрочный <http://www.kazpatent.kz> или онлайн
"Авторское право" Белаяков, Айдаровы Толық <http://topright.kazpatent.kz>

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

Подписано ЭЦП

Абулкаиров Н.А.