**Отчёт обучающегося по практике**

Обучающийся: Артамонова Анастасия Юрьевна Группа: ПИН-44

Направление: 09.03.04 Программная инженерия

Образовательная программа: Программные технологии распределенной обработки информации

Вид практики: производственная(технологическая(проектно-технологическая))

Весенний семестр 2022/2023 учебного года

Место практики: ООО «Эксперт»

«Разработка мобильного приложения для просмотра расписания МИЭТ»

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ 3](#_Toc130412806)

[ВВЕДЕНИЕ 4](#_Toc130412807)

[1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ 6](#_Toc130412808)

[1.1. Анализ существующих решений 6](#_Toc130412809)

[1.2. Концептуальное описание предметной области МП ПР 10](#_Toc130412810)

[1.3. Схема данных приложения 14](#_Toc130412811)

[1.4. Алгоритм работы приложения 15](#_Toc130412812)

[Выводы по исследовательскому разделу 17](#_Toc130412813)

[2. КОНТРУКТОРСКИЙ РАЗДЕЛ 18](#_Toc130412814)

[2.1. Выбор языка программирования 18](#_Toc130412815)

[2.2. Выбор среды разработки 20](#_Toc130412816)

[2.3. Особенности разработки Android приложений 22](#_Toc130412817)

[2.4. Разработка мобильного приложения МП ПР 27](#_Toc130412818)

[Выводы по конструкторскому разделу 33](#_Toc130412819)

[Список литературы 34](#_Toc130412820)

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

МП ПР – мобильное приложение для просмотра расписания

ПО – программное обеспечение

ОС – операционная система

БД – база данных

UML – язык графического описания для объектного моделирования

ИЛМ – инфологическая модель

ДЛМ – даталогическая модель

IDE – интегрированная среда разработки

SDK – Software Development Kit (комплект для разработки программного обеспечения)

NDK – Native Development Kit (комплект для нативной разработки)

ВВЕДЕНИЕ

Студенты просматривают свое расписание по несколько десятков раз за день, чтобы узнать сколько у них сегодня пар, в каком кабинете, что предстоит сделать на неделю. Заходить в браузер, искать расписание на сайте вуза, вводить свою группу является слишком длинным и долгим путём для просмотра своего расписания. А если интернет долго грузит страницу или отключится совсем? Даже если фотография с расписанием сохранена на телефоне, искать ее среди тысячи других проблематично.

Также у студентов возникает необходимость обговорить какой-то интересующий их вопрос с преподавателем, расписание которого можно узнать, только подойдя к кафедре.

Кроме того, расписание может быть составлено не очень удобно как для преподавателя, так и для студентов, из-за чего появляется необходимость переноса пары на другое время.

С большим количеством предметов трудно управлять своим свободным временем, поэтому возможность добавления задач с изменяемым приоритетом, могли бы помочь студентам структурировать свою занятость.

Исходя из сформулированных требований, ни одно из ныне существующих средств нельзя назвать удовлетворительным решением обозначенной проблемы – быстрый просмотр расписания занятий МИЭТ. В результате, актуальной является задача создания нового ПО, позволяющего упростить взаимодействие с расписанием МИЭТ в рамках ВКР бакалавра по направлению 09.03.04 «Программная инженерия» [1] (в соответствии с ГОСТ 19.701-90 [2], 19.505-78 [3] и 19.201-79 [4]).

Тема данной работы: разработка мобильного приложения для просмотра расписания МИЭТ.

Цель данной работы: ускорить просмотр расписания с возможностью просмотра расписания других групп и преподавателей, поискам окна и добавлением предстоящих задач.

Программный продукт должен обеспечивать следующие возможности:

* авторизация пользователя;
* просмотр своего расписания даже при отсутствии интернета;
* добавление задач, у которых автоматически меняется приоритет;
* просмотр расписания нужного преподавателя;
* просмотр расписания другой группы;
* поиск окна для групп и преподавателя, при необходимости с аудиторией.

Отчет состоит из введения, двух разделов, заключениям и списка литературы.

Исследовательский раздел включает в себя обзор существующих решений в области сервисов персональной аналитики, выбор архитектуры разрабатываемого ПО, схему данных, алгоритм работы.

Конструкторский раздел содержит информацию о выборе языка программирования, средств и технологий разработки ПО.

1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

* 1. Анализ существующих решений

В настоящее время существует большое количество программных средств для просмотра расписания ВУЗов. Большинство из программных средств реализовано на мобильных платформах. Это позволяет минимизировать время доступа к расписанию, упростить процесс взаимодействия с ним.

Рассмотрены следующие программные решения, реализованные под мобильные устройства (телефон или планшет) с сенсорным экраном, а также сайт МИЭТ:

* расписание на сайте miet.ru;
* «Orioks» от разработчика MIET;
* «Расписание занятий - Sked» от разработчика SKED LLC;
* «Кампус» от разработчика Ryabushko inc.;
* «СтудЖурнал» от разработчика Roman Sytnyk.

Эти приложения являются популярными среди пользователей устройств под управлением OC Android. Данные программы предоставлены в открытом доступе и для их скачивания необходимо стандартное приложение «Play Маркет». Однако, ни в одном из предложенных программных решений нельзя просмотреть расписание МИЭТ. А для просмотра расписания на сайте необходимо больше времени, чем хотелось бы, а также необходим доступ в интернет.

* + 1. Расписание на сайте miet.ru

На сайте можно просмотреть расписание МИЭТ своей и других групп на сегодня и на неделю. Данные удобно расположены в виде таблицы. Для доступа к расписанию требуется доступ в интернет. По данным опроса студентов, данный способ просмотра расписания не очень удобен.

* + 1. Приложение «Orioks»

Данное приложение позволяет просматривать свое расписание на сегодня, завтра, выбранный день, а также можно просмотреть расписание на неделю, где помечается числитель или знаменатель. Расписание загружается с ОРИОКСа. Также можно просмотреть расписание других групп, закрепив их в меню поиска для дальнейшего быстрого доступа. Кроме того, в приложении можно увидеть текущие дисциплины и полученные баллы за них и за контрольные мероприятия. На данный момент приложение не работает: не выгружает расписание и текущие дисциплины.

Работоспособность приложения гарантируется на устройствах с операционной системой Android версии 5.0 и более поздней версией. Для работы программы требуется Интернет-соединение.

Язык интерфейса: русский.

ПО предоставляется в магазине приложений «Play Маркет» и его модификация технически невозможна.

Программа предоставляется бесплатно.

* + 1. Приложение «Расписание занятий - Sked»

Данное приложение позволяет просматривать свое расписание на каждый день, но нельзя посмотреть расписание на неделю. Расписание загружается с ОРИОКСа/сайта miet.ru. Кроме того, можно просмотреть расписание других групп и преподавателя. Также в поиске указываются все существующие преподаватели и группы. Приложение обеспечивает возможность добавление задач, которые видны всей группе. На данный момент приложение не работает.

Работоспособность приложения гарантируется на устройствах с операционной системой Android версии 5.0 и более поздней версией. Для работы программы требуется Интернет-соединение.

Язык интерфейса: русский.

ПО предоставляется в магазине приложений «Play Маркет» и его модификация технически невозможна.

Программа предоставляется бесплатно.

* + 1. Приложение «Кампус»

Данное приложение загружает расписание с сайта ВУЗов. Просмотреть можно расписание групп и преподавателей. К плюсам относится то, что можно добавить виджет расписания на рабочий стол. Также есть возможность получать объявления, которые, скорее всего, публикует ВУЗ, но на данный момент там расположена реклама. Кроме того, есть вкладка «Отзывы», где можно прочитать и оставить отзыв о преподавателе, а также просмотреть рейтинг. Отзыв формируется из оценки компетентности, справедливости оценивания, умения донести материал, актуальности материала и общей оценки, а также тегов, характеризующих преподавателя, которые помогают составить портрет о нем и не влияют на рейтинг, и самого комментария. Основным минусом является то, что МИЭТа в списке ВУЗов нет.

Работоспособность приложения гарантируется на устройствах с операционной системой Android версии 5.0 и более поздней версией. Для работы программы требуется Интернет-соединение.

Язык интерфейса: русский.

ПО предоставляется в магазине приложений «Play Маркет» и его модификация технически невозможна.

Программа предоставляется бесплатно.

* + 1. Приложение «СтудЖурнал»

В данном приложении расписание составляется вручную. Есть возможность разделить недели на числитель/знаменатель(четную/нечетную). Можно скопировать составленное расписание со знаменателя на числитель и наоборот. Карточка предмета состоит из названия, корпуса/аудитории, преподавателя, чек-бокса лекция/практика, что немного не удобно, так как МИЭТе есть еще и лабораторные работы, ссылки на урок и стрелок вверх/вниз, чтобы передвинуть предмет на пару вверх/вниз соответственно. Кроме того, есть вкладка заметки, где можно оставить небольшую запись, и домашнее задание, в которое добавляется предмет, название которого подгружается из расписания, дата и текст задания. Также можно настроить время пар и перемен между ними. Составленным расписание можно поделиться с одногруппниками.

Работоспособность приложения гарантируется на устройствах с операционной системой Android версии 5.0 и более поздней версией. Для работы программы требуется Интернет-соединение.

Язык интерфейса: русский.

ПО предоставляется в магазине приложений «Play Маркет» и его модификация технически невозможна.

Программа предоставляется бесплатно.

* + 1. Сравнение существующих решений

В результате анализа существующих решений составлена таблица 1.1, в которой перечислены ключевые параметры для сравнения продуктов между собой.

В результате анализа вышеперечисленных программ, составлен перечень функций, которыми должно обладать искомое решение:

* работоспособность приложения/сайта c расписанием МИЭТ;
* загрузка расписания;
* быстрый доступ к просмотру своего расписания/удобство работы;
* просмотр своего расписания без доступа в интернет;
* просмотр расписания преподавателя;
* поиск окна;

Таблица 1.1 - Сравнение существующих решений

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Программный продукт  Параметры | Расписание на сайте miet.ru [5] | «Orioks» [6] | «Sked» [7] | «Кампус» [8] | «Студ  Журнал» [9] | МП ПР |
| Версия ОС | Любая | Android 5.0 и более поздние версии | Android 5.0 и более поздние версии | Android 5.0 и более поздние версии | Android 5.0 и более поздние версии | Android 5.0 и более поздние версии |
| Возможность работы с расписанием МИЭТ | Да | Нет | Нет | Нет | Да | Да |
| Загрузка расписания | Да | Да | Да | Нет | Нет | Да |
| Удобство работы | Нет | Да | Да | Да | Нет | Да |
| Просмотр своего расписания без доступа в интернет | Нет | Нет | Нет | Да | Да | Да |

Продолжение таблицы 1.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Просмотр расписания преподавателя | Нет | Нет | Да | Да | Нет | Да |
| Поиск окна | Нет | Нет | Нет | Нет | Нет | Да |

Так как ни одно из существующих решений полностью не удовлетворяет описанным выше требованиям, то актуальной является задача создания нового программного обеспечения – МП ПР.

## Концептуальное описание предметной области МП ПР

Архитектура МП ПР описывается при помощи Use case диаграммы, ИЛМ и ДЛМ, которые направлены на облегчение разбора связей элементов сложного объекта. Данные диаграммы показывают, как система должна работать, скрывая детальную реализацию. UML – это язык графического описания для объектного моделирования в области разработки ПО, моделирования бизнес-процессов, системного проектирования и отображения организационных структур. [10]

Use case диаграмма МП ПР указана на рисунке 1.1

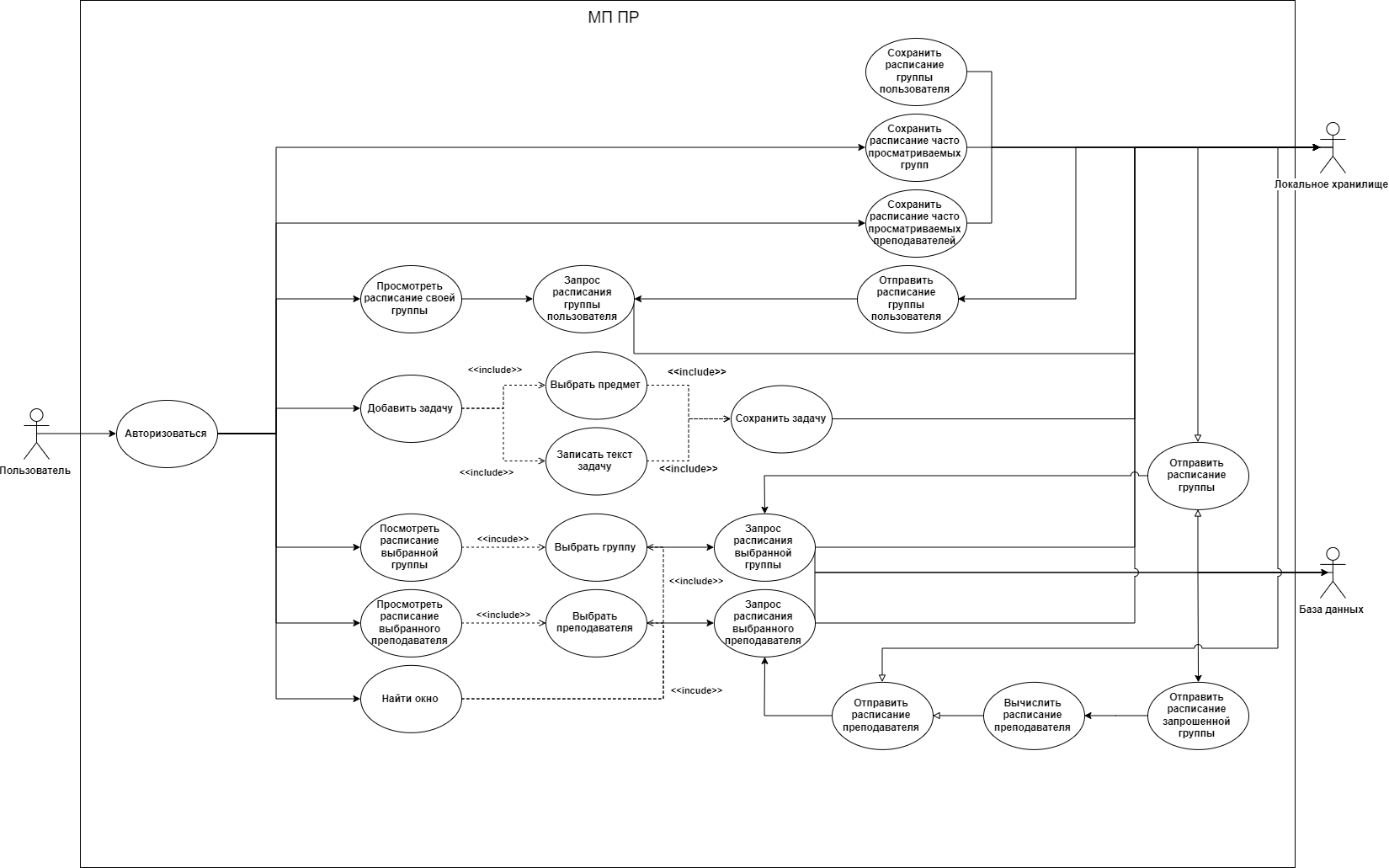


Рисунок 1.1 – Use case диаграмма

На схеме показаны сценарии, которые соответствуют требованиям, предъявляемым к программному обеспечению. Для начала работы приложения необходимо авторизоваться. При входе расписание пользователя сохраняется в локальном хранилище. Пользователь может просмотреть свое расписание. Оно получается из локальной базы данных и выводится пользователю на экран. Аналогично происходит с просмотром расписания другой группы, только пользователю необходимо выбрать группу. Чтобы просмотреть расписание преподавателя, расписание групп получаются с локального хранилища и формируется расписание преподавателя. Для нахождения окна необходимо указать группы и преподавателя, которым необходимо найти окно для переноса расписания. Для добавления задачи необходимо ввести название, текст задачи, предмет и дату сдачи задачи. Задача сохраняется в локальном хранилище. Расписание загружается либо из локального хранилища, если оно сохранено, либо из базы данных.

Инфологическая модель представлена на рисунке 1.2.

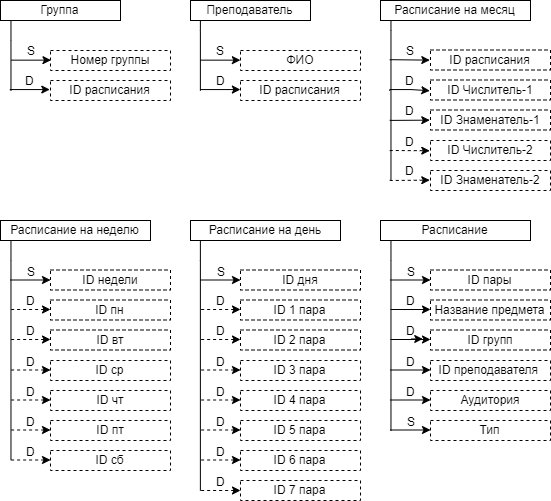


Рисунок 1.2 – ИЛМ БД

На инфологической модели предоставлены сущности с атрибутами:

* «Группа» со статическим атрибутом «Номер группы» и динамическим «ID расписания»;
* «Преподаватель» со статическим атрибутом «ФИО» и динамическим «ID расписания»;
* «Расписание на месяц» со статическим атрибутом «ID расписания» и динамическими «ID Числитель-1», «ID Знаменатель-1», «ID Числитель-2», «ID Знаменатель-2», причем ID последних двух недель могут отсутствовать, если расписание одинаковое для числителя 1(знаменателя 1) и числителя 2(знаменателя 2);
* «Расписание на неделю» со статическим атрибутом «ID недели» и динамическими для ID каждого дня недели, при этом ID дней может отсутствовать, так как в данный день отсутствуют занятия;
* «Расписание на день» со статическим атрибутом «ID дня» и динамическими для ID каждого пары, причем ID пар может отсутствовать;
* «Расписание» со статическими атрибутами «ID пары» и «Тип», который может иметь три значения: «Лекция», «Семинар», «Лабораторная», и динамическими «Название предмета», «ID групп», которых может быть несколько, «ID преподавателя» и «Аудитория».

Даталогическая модель представлена на рисунке 1.3.

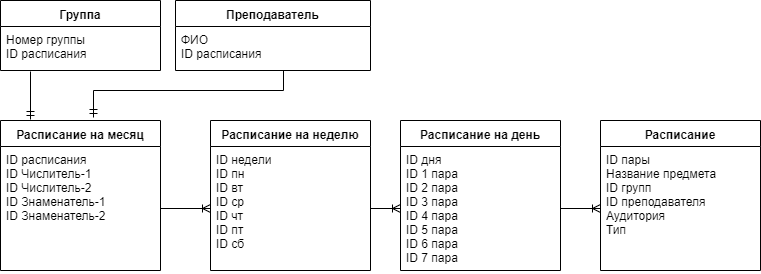


Рисунок 1.3 – ДЛМ БД

На даталогической модели представлены сущности с атрибутами и связь между ними. Так сущности «Группа» и «Преподаватель» имеют связь один к одному к сущности «Расписание на месяц». «Расписание на месяц» имеет связь один ко многим к «Расписание на неделю». «Расписание на неделю» имеет связь один ко многим к «Расписание на день». «Расписание на день» имеет связь один ко многим к «Расписание».

## Схема данных приложения

Схема данных представлена на рисунке 1.4. В течение работы приложения происходит постоянное взаимодействие пользователя с базой данных.

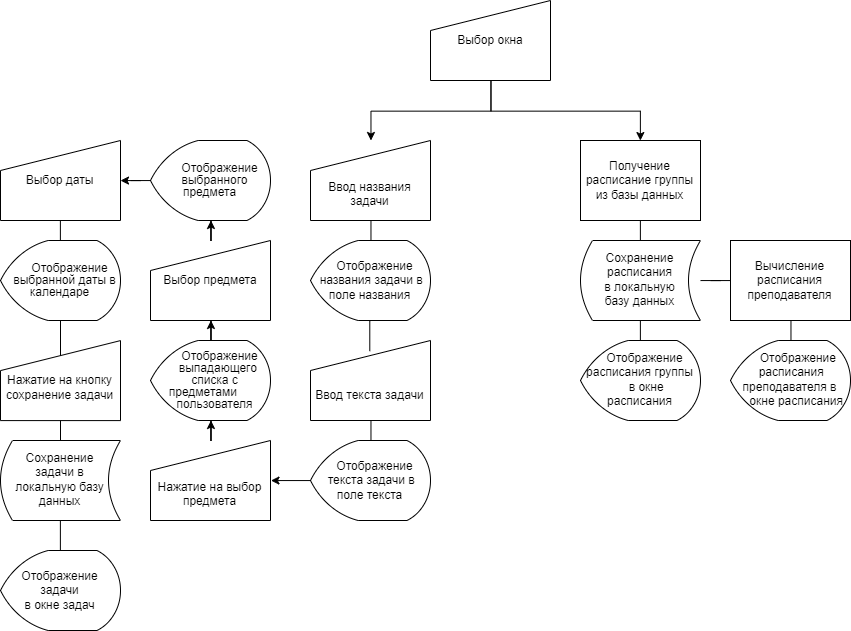
На схеме видно, что состояния системы при взаимодействии с пользователем – это в основном демонстрация пользователю окон с необходимым ему расписанием. Также при нажатии кнопок «Поиск окна» и «Задачи» пользователю демонстрируется возможные окна для переноса расписания для выбранных групп и преподавателя, и добавленные задачи соответственно.

Рисунок 1.4 – Схема данных приложения

## Алгоритм работы приложения

Схема алгоритма представлена на рисунке 1.5. Перед началом работы приложения необходимо авторизоваться через систему ОРИОКС. Далее пользователь может взаимодействовать с функционалом приложения. Просматривать свое расписание на сегодня и другие дни. При нажатии на кнопки «Поиск» появится окно для поиска расписания групп, преподавателей и поиска окон для переноса занятия, где при вводе номера группы, фамилии преподавателя и номера групп и фамилии преподавателя соответственно, приложение отправит запрос в базу данных и представит необходимые данные пользователю. При нажатии кнопки «Задачи» пользователь сможет просмотреть существующие задачи, а также добавить новую задачу, указав название предмета, срок выполнения задачи и саму задачу. Чтобы вернуться к просмотру своего расписания пользователю необходимо нажать кнопку «Расписание».

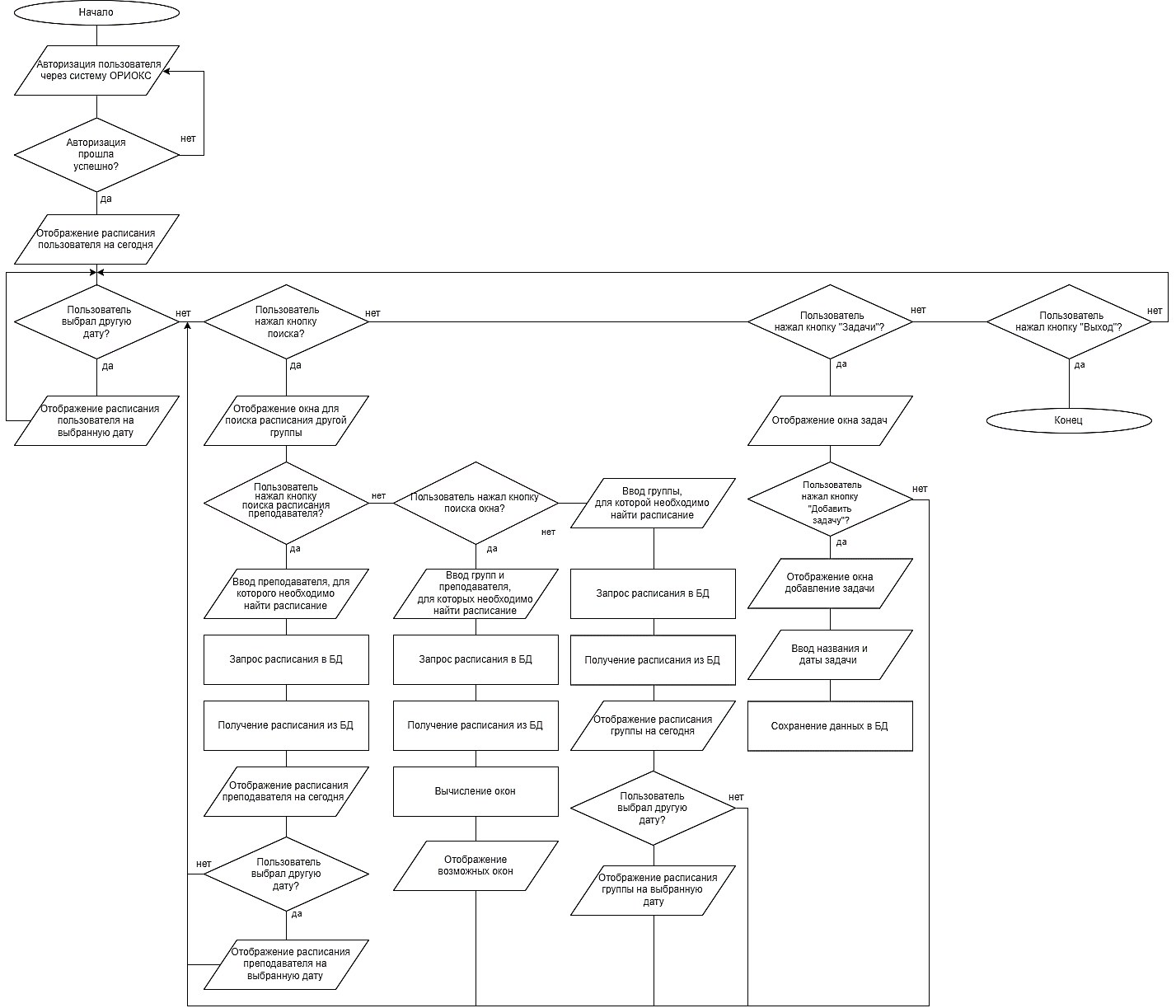


Рисунок 1.5 – Схема алгоритма работы приложения

# Выводы по исследовательскому разделу

В результате исследования предметной области и приложений-аналогов, принято решение о разработке собственного мобильного приложения, обладающего возможностью просмотра своего расписания, расписания других групп и преподавателей, а также поиск окна для переноса занятия и добавление задач. Сформулирован перечень требований к функционалу разрабатываемого приложения.

Мобильное приложения для просмотра расписания МИЭТ реализуется для мобильных устройств c операционной системой Android. Для облегчения этапа написания кода спроектированы Use case диаграммы, а также ИЛМ и ДЛМ. Разработаны схема данных, которая призвана облегчить понимание логических связей по данным между рабочими экранами приложения и схема алгоритма, на которой подробно описан процесс работы приложения.

МП ПР прошло апробацию путем опубликования тезисов доклада: Артамонова А. Разработка мобильного приложения для просмотра расписания МИЭТ. Актуальные проблемы информатизации в цифровой экономике и научных исследованиях. Ⅲ Научно-практическая конференция с международным участием: тезисы докладов. М.: МИЭТ, 2022. 8 с.

1. КОНТРУКТОРСКИЙ РАЗДЕЛ

## Выбор языка программирования

Выбор языка программирования основывался на сравнении доступных средств и инструментов языка и требований к мобильному приложению.

Перечень требований, которые могут повлиять на выбор языка программирования:

* наличие транслятора под платформу Android;
* наличие документации;
* наличие бесплатных инструментов для разработки;
* наличие декларативного фреймворка;
* наличие возможности контроля потоком данных;

Так как приложение реализовано на мобильной платформе проводился анализ только тех языков, которые пригодны для разработки нативных приложений. В качестве платформы выбрана операционная система Android, так как его доля на мировом рынке составляет 75% [11].

* + 1. Язык С++

Язык C++ является [компилируемым](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), статически типизированным языком программирования общего назначения. Он поддерживает основные парадигмы программирования, а также имеет большую стандартную библиотеку. Язык C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков.

С++ позиционируется как высокопроизводительный язык, что важно если программный продукт достаточно тяжеловесный.

Для разработки Android-приложения существует Android NDK. С помощью данного набора средств можно написать библиотеки на С++, которые впоследствии можно подключить к приложению с помощью Java, так как Android NDK не работает с графическим интерфейсом.

* + 1. Язык С#

Язык С# является объектно-ориентированным языком общего назначения. Язык имеет статическую типизацию, поддерживает полиморфизм и перегрузку операторов.

Для разработки Android-приложения существует Xamarin, фреймворк для кроссплатформенной разработки мобильных приложений с использованием языка C#. На данный момент Xamarin SDK, включая библиотеки и инструменты командной строки, стал открытым и доступным для всех. Но приложения, создаваемые с помощью фреймворка, обычно больше, чем нативные. И так как Xamarin сравнительно новый инструмент возникают сложности с интеграцией сторонних ресурсов.

* + 1. Язык Python

Язык Python является высокоуровневым языком программирования общего назначения с динамической типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика за счет читаемости кода. Синтаксис языка минималистичен, за счет чего редко возникает необходимость обращаться к документации.

Для разработки Android приложений существует два фреймворка Kivy и BeeWare Project. Kivy – это библиотека Python, предназначенная для разработки кроссплатформенных GUI приложений. Его недостатком является отсутствие нативного набора инструментов пользовательского интерфейса. BeeWare Project предлагает набор инструментов, которые можно использовать для написания мобильных и десктопных приложений. Но данный фреймворк находится в разработке, поэтому часто происходит изменения API, ошибки и отсутствия необходимых функций.

* + 1. Язык Java

Язык Java является строго типизированным объектно-ориентированным языком программирования общего назначения. Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация Java-машины. Java в разработке считается фундаментальным, а также поддерживается многими популярными средами разработки, такие как Eclipse, Netbeans и JetBrains.

* + 1. Язык Kotlin

Язык Kotlin является статически типизированным, объектно-ориентированным языком программирования, работающий поверх Java Virtual Machine. У языка Kotlin более быстрая компиляция и лучшая поддержка языка в IDE. Kotlin полностью совместим с Java, что позволяет для существующего Android-приложения внедрять новые функции на Kotlin без переписывания приложения целиком.

* + 1. Сравнение языков программирования

Чтобы наглядно продемонстрировать возможности рассмотренных языков программирования составлена таблица 1.2.

Таблица 2.1 - Сравнение языков программирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Язык  Критерий | С++ [12] [13] | С# [14] [15] | Python [16] [17] | Java [18] | Kotlin [19] |
| Наличие транслятора под платформу Android | Нет | Да | Нет | Да | Да |
| Наличие документации | Да | Да | Да | Да | Да |
| Наличие бесплатных инструментов для разработки | Да | Да | Да | Да | Да |
| Наличие декларативного фреймворка | Нет | Нет | Нет | Нет | Да |
| Наличие возможности контроля потоком данных | Нет | Нет | Нет | Нет | Да |

В результате сравнения для разработки МП ПР выбран язык программирования Kotlin, так как в нем присутствует транслятор под платформу Android и под Kotlin существует декларативный фреймворк Jetpack Compose, который дает возможность контроля потоком данных.

## Выбор среды разработки

После выбора языка необходимо выбрать среду разработки. К основным требованиям относится:

* наличие эмулятора для запуска Android-приложения;
* наличие бесплатных инструментов для разработки;
* сборка приложений, основанная на Gradle;
* инструменты для работы с UI;
* инструменты для тестирования.

JetBrains предоставляет официальный плагин Kotlin для двух интегрированных сред разработки (IDE): IntelliJ IDEA и Android Studio.

* + 1. Среда разработки IntelliJ IDEA

IntelliJ IDEA – IDE для языков JVM, предназначенная для максимальной производительности разработчиков. Она обеспечивает интеллектуальное завершение кода, статистический анализ кода и рефакторинг. Среда предлагает различные инструменты автоматизации сборки, включая Maven, Gradle, Gant, NPM и Grunt, которые помогают членам команды выполнять такие действия, как компиляция кода, упаковка, тестирование и многое другое.

* + 1. Среда разработки Android Studio

Android Studio – официальная среда разработки приложений для Android. Помимо мощного редактора кода и инструментов разработчика, IDE предлагает еще больше функций, повышающих продуктивность разработчика, таких как:

* гибкая система сборки на основе Gradle;
* быстрый и многофункциональный эмулятор;
* применение изменений для push-кода и изменений ресурсов в работающем приложении без перезапуска приложения;
* шаблоны кода и интеграция с GitHub;
* обширные инструменты и фреймворки для тестирования;
* инструменты Lint для выявления проблем с производительностью.

Данная среда разработки доступна для Windows, macOS и GNU/Linux.

* + 1. Сравнение сред разработки

По итогам изучения возможностей различных сред разработки составлена таблица 1.3.

Таблица 2.2 - Сравнение сред разработки

|  |  |  |
| --- | --- | --- |
| Среда разработки  Критерий | IntelliJ IDEA [20] | Android Studio [21] |
| Наличие эмулятора для запуска Android-приложения | Нет | Да |

Продолжение таблицы 2.2.

|  |  |  |
| --- | --- | --- |
| Наличие бесплатных инструментов для разработки | Да | Да |
| Сборка приложений, основанная на Gradle | Да | Да |
| Инструменты для работы с UI | Нет | Да |
| Инструменты для тестирования | Да | Да |

В результате сравнения, выбрана интегрированная среда разработки Android Studio, которая является официальной и рекомендованной средой разработки под OC Android, так как в ней, в отличие от IntelliJ IDEA, имеет эмулятор для запуска Android-приложения и инструменты для работы с UI.

* 1. Особенности разработки Android приложений

Для разработки мобильного приложения для просмотра расписания МИЭТ использовались инструменты Android Studio, а также Compose Navigation и Jetpack Compose.

В Jetpack Compose построение интерфейса основывается на графе Composable функций. После построения граф изменяется на основе состояний, при этом Jetpack Compose определяет какую часть графа необходимо, если нужно, изменить.

Compose Navigation необходим для переходов между экранами, с помощью построенного графа навигации.

Также для создания архитектуры приложения использовался паттерн MVC.

* + 1. Набор инструментов Jetpack Compose

Для создания интерфейса с помощью Jetpack Compose используются следующие компоненты.

1. Row [22]

Контейнер для размещения вложенных компонентов в строку. Функция Row принимает параметры:

* modifier, который позволяет настроить внешний вид и поведение компонента;
* verticalAlignment, который устанавливает выравнивание компонента по вертикали;
* horizontalArrangement, который устанавливает выравнивание компонента по горизонтали;
* content, который представляет вложенное содержимое.

Например, Row использовалось для размещения кнопок (рисунок 2.1).

Row(modifier = Modifier

.fillMaxSize(),

horizontalArrangement = Arrangement.SpaceBetween

){

ScheduleButton(navController, alpha1)

TaskButton(navController,alpha2)

}



Рисунок 2.1. – Кнопки

1. Column [23]

Контейнер для размещения вложенных компонентов в столбец. Функция Column принимает параметры:

* modifier, который позволяет настроить внешний вид и поведение компонента;
* verticalArrangement, который устанавливает выравнивание компонента по вертикали;
* horizontalAlignment, который устанавливает выравнивание компонента по горизонтали;
* content, который представляет вложенное содержимое.

Как один из примеров использования: расположение ФИО с группой в шапке (рисунок 2.2).

Column(modifier = Modifier

.padding(10.dp)

.weight(0.85f)

) {

Text(

name,

fontSize = 16.sp,

fontWeight = FontWeight.Normal,

fontFamily = centurygothic,

color = Color.White

)

Text(

group,

fontSize = 16.sp,

fontWeight = FontWeight.Normal,

fontFamily = centurygothic,

color = Color.White,

modifier = Modifier.alpha(0.9f)

)

}



Рисунок 2.2 – Шапка

1. Box [24]

Простой контейнер, который позволяет позиционировать вложенные компоненты. Функция Box принимает параметры:

* modifier, который позволяет настроить внешний вид и поведение компонента;
* contentAlignment, который устанавливает расположение компонента;
* propagateMinConstraints, который указывает, надо ли применять к содержимому ограничения по минимальным размерам
* content, который представляет вложенное содержимое.

В МП ПР Box использовался для создания кнопки поиска (рисунок 2.1)

Box(modifier = Modifier

.offset(0.dp, 30.dp)

.background(colorResource(R.color.orioks), CircleShape)

.fillMaxWidth(0.27f)

.aspectRatio(1f, true),

contentAlignment = Alignment.Center

)

{

Image(

painter = painterResource(R.drawable.search\_inactive),

contentDescription = "search",

modifier = Modifier

.align(Alignment.Center)

.offset(0.dp, -15.dp)

)

}

Объект Modifier используется для настройки внешнего вида компонентов. Основные и часто используемые его функции, и область их применения представлены в таблице 2.3.

Таблица 2.3 – Функции Modifier

|  |  |  |
| --- | --- | --- |
| Область применения | Функция | Назначение [25] |
| Любая | border() | Добавление границы |
| Любая | alpha() | Рисование компонента с прозрачностью |
| Любая | background() | Рисование компонента с заданным цветом фона |
| Любая | clip() | Рисование компонента в заданной форме |

Продолжение таблицы 2.3

|  |  |  |
| --- | --- | --- |
| Любая | padding() | Добавление дополнительного пространства вдоль края компонента |
| Любая | offset() | Смещение компонента |
| Любая | aspectRatio() | Изменение размера компонента в соответствии с заданным соотношением сторон |
| Любая | fillMaxSize() | Компонент принимает максимально возможный размер по ширине и высоте |
| Любая | fillMaxHeight() | Компонент принимает максимально возможный размер по высоте |
| Любая | fillMaxWidth() | Компонент принимает максимально возможный размер по ширине |
| ColumnScope/  RowScope | weight() | Размер высоты/ширины компонента пропорционален его отношению к другим взвешенным компонентам |
| Любая | clickable() | Настройка компонента для получения кликов через событие ввода или нажатия |

Таким образом Jetpack Compose значительно упрощает разработку пользовательского интерфейса, не ограничивая разработчика в возможностях создания интерфейса разнообразного дизайна. Кроме того, Jetpack Compose предоставляет возможность создания сложной анимации, что было практически невозможно сделать с помощью XML.

* + 1. Compose Navigation

Состояния необходимы для изменения отдельных компонентов на экране. Для переходов между окнами необходимо использовать Compose Navigation. Компонент Navigation обеспечивает поддержку приложений Jetpack Compose, перемещаясь между составными объектами, используя преимущества инфраструктуры и функций компонента навигации. [26]

Для начала создается граф навигации. Для этого задается navController, начальное окно, а также функция, отвечающая за отображение каждого экрана

NavHost(navController = navController,

startDestination = Screen.Auth.route

) {

composable(

route = Screen.Auth.route

){

AuthScreen(navController)

}

composable(

route = Screen.Schedule.route

){

ScheduleScreen(navController)

}

composable(

route = Screen.Task.route

){

TaskScreen(navController)

}

}

Также между экранами можно передавать агрументы. Для этого помимо пути в composable указывается название и тип аргумента.

composable(

route = Screen.TaskInfo.route,

arguments = listOf(

navArgument("name"){

type = NavType.StringType

}

)

){

val name = it.arguments?.getString("name")

TaskInfoScreen(navController, name)

}

Для перехода на другой экран в функции clickable объекта указывается путь перехода, а также опция, чтобы экраны не наслаивались друг на друга. [27]

modifier = Modifier

.clickable {

navController.navigate(Screen.Task.route) {

popUpTo(Screen.Task.route) {

inclusive = true

}

}

}

* + 1. Паттерн MVC

MVC (Model-View-Controller) — это паттерн, обычно используемый для реализации пользовательских интерфейсов, данных и управляющей логики. Он подчеркивает разделение между бизнес-логикой программного обеспечения и отображением. [28]

Три части паттерна проектирования MVC:

* Model: управляет данными и бизнес логикой;
* View: управляет макетом и отображением;
* Controller: направляет команды к Model и View.

Model определяет, какие данные должно содержать приложение и при изменении данных уведомляет View и Controller.

View определяет, как должны отображаться данные приложения.

Controller содержит логику, которая обновляет модель и представлений в ответ на ввод данных от пользователя.

* 1. Разработка мобильного приложения МП ПР
     1. Разработка интерфейса приложения

После авторизации в системе ОРИОКС, пользователь видит свое расписание на сегодня. При нажатии на дни в верхней части экрана свое расписание на другие дни. Чтобы посмотреть расписание на другую неделю необходимо раскрыть календарь, нажав кнопку вниз. Также над календарем указан тип недели. При нажатии на панель со своим именем и фамилией, откроется меню, чтобы выйти из аккаунта. При повторном нажатии выход в предыдущий экран.

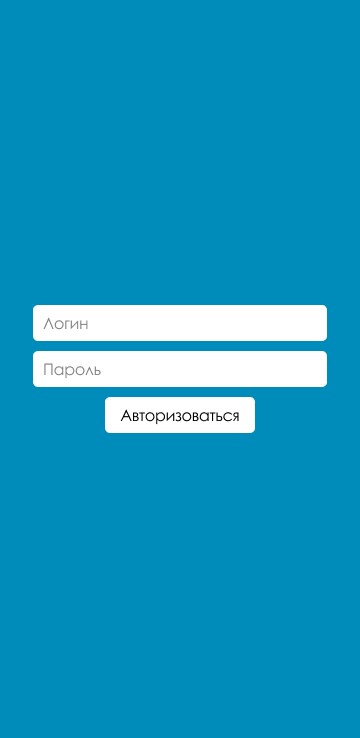


Рисунок 2.3 – Интерфейс приложения

При нажатии кнопки «Задачи» открывается окно задач, где можно просмотреть добавленные задачи и добавить новую. Задачи сортируются в порядке увеличения времени до задачи. При нажатии на задачу открывается окно просмотра задачи с комментарием. А в верхнем правом углу кнопка для редактирования задачи. Чтобы удалить задачу, необходимо смахнуть ее влево.

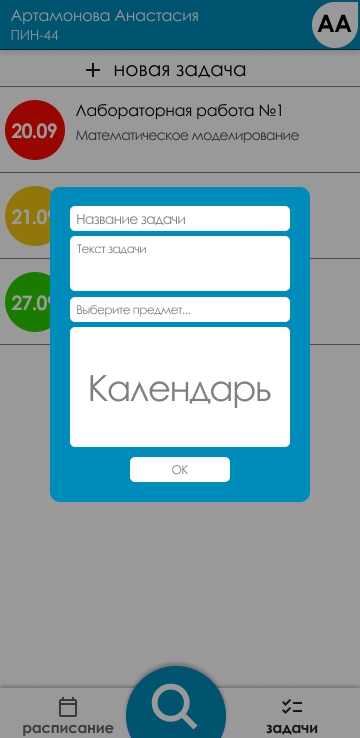
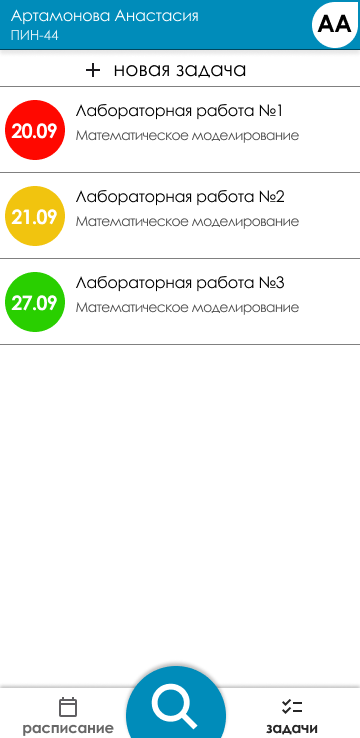


Рисунок 2.4 – Окно «Задачи»

При свайпе вверх или при нажатии кнопки «Поиск», открывается окно поиска, где предоставляется функционал поиска расписания группы, расписания преподавателя и окна. При нажатии кнопки поиск, происходит соответствующий поиск, а при свайпе вниз возвращение к предыдущему экрану задач или расписания. При нахождении в окне расписания другой группы или преподавателя, можно закрепить расписание для быстрого доступа.

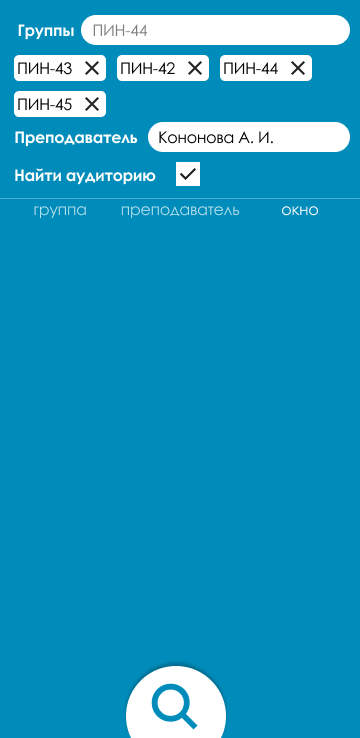
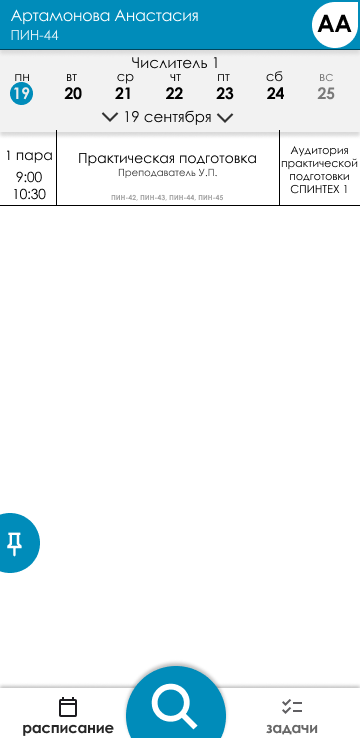


Рисунок 2.5 – Окно «Поиск»

* + 1. Разработка моделей приложения

Программная реализация МП ПР стоит из классов двух видов (классы для отрисовки графических интерфейсов активностей и классы-контейнеры), манифеста и файлов ресурсов.

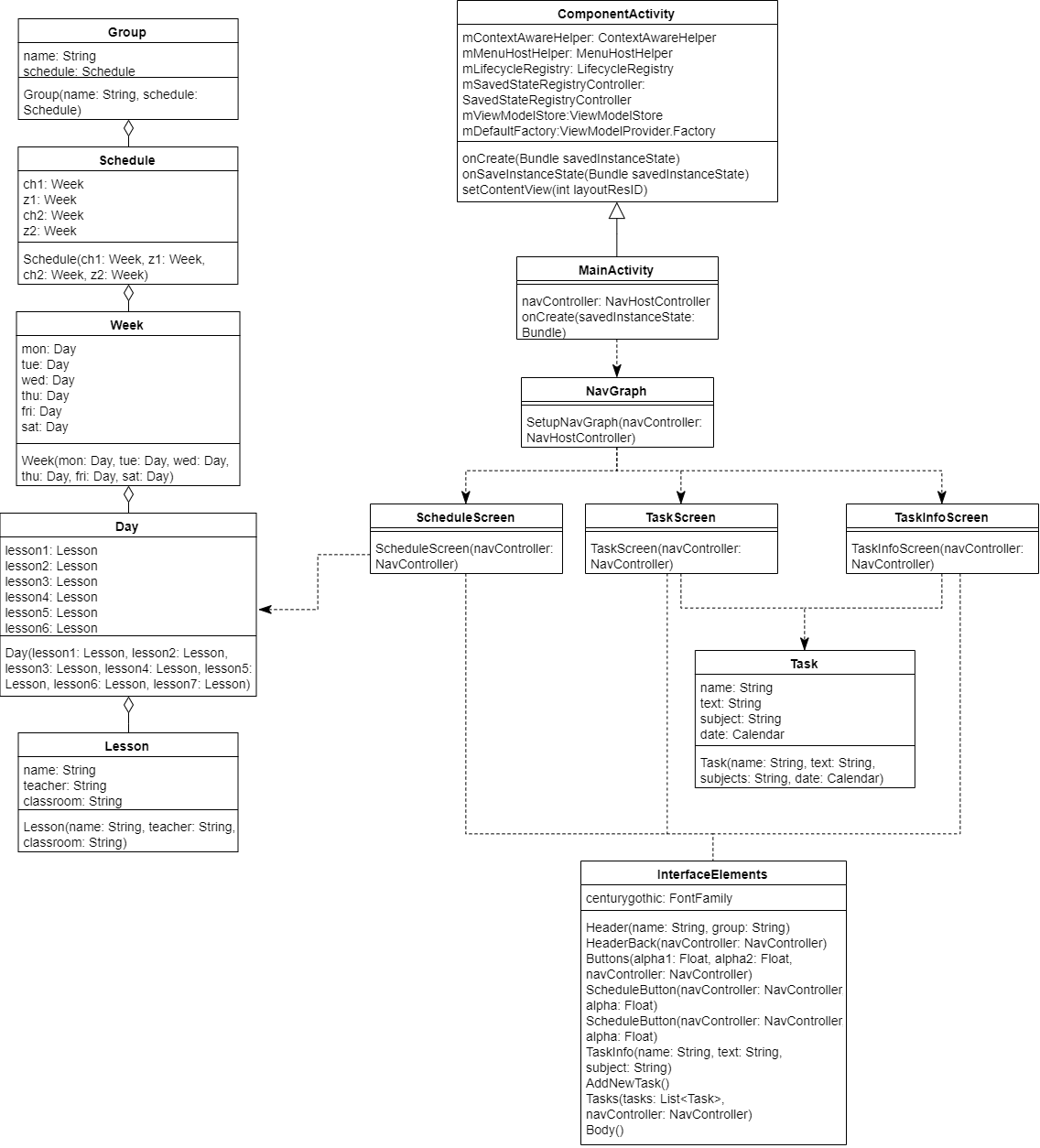
На рисунке 2.6 представлена обобщенная UML диаграмма классов мобильного приложения для просмотра.

Связи между классами показаны стрелками. Стрелка с ромбом означает связь агрегации, стрелка с концом треугольником – связь наследования, пунктирная стрелка – связь зависимости. [29] Например, TaskScreen и TaskInfoScreen используют объект класса Task в своих методах.

Классы Group, Schedule, Week, Day, Lesson, Task являются классами-моделями, которые реализуют логику работы приложения. Класс Task необходим для хранения и отображения задач, а Group, Schedule, Week, Day, Lesson для хранения и отображения расписания пользователя, других групп и преподователя.

Класс MainActivity, который наследуется от ComponentActivity, является основной активностью и будет первой отображаться при запуске приложения. [30]

Класс NavGraph представляет собой граф навигации для переходов между экранами.

Рисунок 2.6. – Диаграмма классов

Класс SheduleScreen отображает окно расписания пользователя и кнопки необходимые для перехода на дальнейшие экраны. Класс TaskScreen отображает созданные задачи пользователя, а также кнопку для создания новых задач. Класс TaskInfoScreen показывает полную информацию о выбранной задаче, а также дает возможность изменить ее.

Существуют класс элементов InterfaceElements, который используют все классы-окна.

Выводы по конструкторскому разделу

В качестве языка реализации МП ПР выбран язык программирования Kotlin, поскольку он отвечает выдвинутым требованиям. Для разработки мобильного приложения выбрана среда разработки Android Studio.

В ходе реализации МП ПР изучены особенности разработки под операционную систему Android. Обозначены составные части нативного Android-приложения. Разработаны экранные формы пользовательского интерфейса МП ПР. Сгенерирована UML-диаграмма классов приложения, демонстрирующая программную реализацию поставленной задачи.

# Список литературы

|  |  |
| --- | --- |
| [1] | А. А. Доронина, Р. А. Касимов и Е. Л. Федотова, Методические указания по подготовке выпускной квалификационной работы по направлению подготовки бакалавров 09.03.04 "Программная инженерия", Л. Г. Гагарина, Ред., Москва: МИЭТ, 2021. |
| [2] | ГОСТ 19.505-79. Руководство оператора. Требования к содержанию и оформлению. |
| [3] | ГОСТ 19.201-78. Техническое задание, требования к содержанию и оформлению. |
| [4] | ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. |
| [5] | «Расписание,» [В Интернете]. Available: https://miet.ru/schedule. |
| [6] | «Приложение GooglePlay - "Orioks",» [В Интернете]. Available: https://play.google.com/store/apps/details?id=ru.eva.miet.orioks. |
| [7] | «Приложение GooglePlay - "Расписание занятий - SKED",» [В Интернете]. Available: https://play.google.com/store/apps/details?id=com.sked.core. |
| [8] | «Приложение GooglePlay - "Кампус",» [В Интернете]. Available: https://play.google.com/store/apps/details?id=ru.dewish.campus. |
| [9] | «Приложение GooglePlay - "СтудЖурнал",» [В Интернете]. Available: https://play.google.com/store/apps/details?id=com.romansytnyk.studentstudio. |
| [10] | «Unified Modeling Language,» [В Интернете]. Available: https://en.wikipedia.org/wiki/Unified\_Modeling\_Language. [Дата обращения: 18 02 2018]. |
| [11] | «ОС Android: статистика, оценка и перспективы для рекламодателей,» [В Интернете]. Available: https://www.byyd.me/ru/blog/2021/08/android/. |
| [12] | «С++,» [В Интернете]. Available: https://ru.wikipedia.org/wiki/C%2B%2B. |
| [13] | «Разработка кроссплатформенных мобильных приложений на языке C++,» [В Интернете]. Available: https://learn.microsoft.com/ru-ru/cpp/cross-platform/visual-cpp-for-cross-platform-mobile-development?view=msvc-170. |
| [14] | «C Sharp,» [В Интернете]. Available: https://ru.wikipedia.org/wiki/C\_Sharp. |
| [15] | «Подробно о Xamarin,» [В Интернете]. Available: https://habr.com/ru/post/188130/. |
| [16] | «Python,» [В Интернете]. Available: https://ru.wikipedia.org/wiki/Python. |
| [17] | «Мобильная разработка на Python: обзор двух фреймворков,» [В Интернете]. Available: https://proglib.io/p/mobile-python. |
| [18] | «Java,» [В Интернете]. Available: https://ru.wikipedia.org/wiki/Java. |
| [19] | «Kotlin,» [В Интернете]. Available: https://ru.wikipedia.org/wiki/Kotlin. |
| [20] | «IDEs for Kotlin development,» [В Интернете]. Available: https://kotlinlang.org/docs/kotlin-ide.html. |
| [21] | «Meet Android Studio,» [В Интернете]. Available: https://developer.android.com/studio/intro. |
| [22] | «Jetpack Compose | Row,» [В Интернете]. Available: https://metanit.com/kotlin/jetpack/2.3.php. |
| [23] | «Jetpack Compose | Column,» [В Интернете]. Available: https://metanit.com/kotlin/jetpack/2.2.php. |
| [24] | «Jetpack Compose | Box,» [В Интернете]. Available: https://metanit.com/kotlin/jetpack/2.1.php. |
| [25] | «List of Compose modifiers | Android Developers,» [В Интернете]. Available: https://developer.android.com/jetpack/compose/modifiers-list. |
| [26] | «Navigating with Compose | Android Developers,» [В Интернете]. Available: https://developer.android.com/jetpack/compose/navigation. |
| [27] | «Навигация на Jetpack Compose by Google,» [В Интернете]. Available: https://habr.com/ru/post/594621/. |
| [28] | «MVC - MDN Web Docs Glossary: Definitions of Web-related,» [В Интернете]. Available: https://developer.mozilla.org/en-US/docs/Glossary/MVC. |
| [29] | «Построение диаграммы классов,» [В Интернете]. Available: https://flexberry.github.io/ru/gpg\_class-diagram.html. |
| [30] | «Создание и вызов Activity - Start Android,» [В Интернете]. Available: https://startandroid.ru/ru/uroki/vse-uroki-spiskom/58-urok-21-sozdanie-i-vyzov-activity.html#:~:text=MAIN%20показывает%20системе%2C%20что%20Activity,intent.. |

Руководитель практики от МИЭТ / Фёдоров А.Р/

Руководитель практики   
от организации / Гончаренко Г.А./

Обучающийся / Артамонова А. Ю/