

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование кафедры)

02.03.03 Математическое обеспечение и администрирование  
информационных систем

(код и наименование направления подготовки, специальности)

Технология программирование

(направленность (профиль)/специализация)

## БАКАЛАВРСКАЯ РАБОТА

на тему Разработка мобильного приложения для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО "ТГУ"

Студент

М.В. Коннов

(И.О. Фамилия)

\_\_\_\_\_ (личная подпись)

Руководитель

А.Б. Кузьмичев

(И.О. Фамилия)

\_\_\_\_\_ (личная подпись)

Консультанты

А.В. Москалюк

(И.О. Фамилия)

\_\_\_\_\_ (личная подпись)

**Допустить к защите**

Заведующий кафедрой \_\_\_\_\_

А.В. Очеповский

(ученая степень, звание, И.О. Фамилия)

\_\_\_\_\_ (личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование кафедры)

УТВЕРЖДАЮ  
Зав. кафедрой Прикладная математика  
и информатика  
\_\_\_\_\_ А.В. Очеповский  
(подпись) (И.О. Фамилия)  
« \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.

**ЗАДАНИЕ**

**на выполнение бакалаврской работы**

Студент Коннов Максим Владимирович

1. Тема Разработка мобильного приложения для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО "ТГУ"
2. Срок сдачи студентом законченной выпускной квалификационной работы 01.07.2019 г.
3. Исходные данные к выпускной квалификационной работе текущий процесс публикации расписания ФГБОУ ВО «ТГУ», информация о сервисах планирования событий: Google Calendar, Yandex календарь, Windows Live.
4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов) Анализ существующих систем планирования событий, проектирование мобильного приложения с интеграцией расписания ФГБОУ ВО «ТГУ» разработка приложения».
5. Ориентировочный перечень графического и иллюстративного материала блок-схемы; графики и диаграммы, поясняющие результат работы приложения
6. Консультанты по разделам А. В. Москалюк
7. Дата выдачи задания «15» декабря 2018г.

Руководитель выпускной  
квалификационной работы

\_\_\_\_\_ (подпись)

А.Б. Кузьмичев  
(И.О. Фамилия)

Задание принял к исполнению

\_\_\_\_\_ (подпись)

М.В. Коннов  
(И.О. Фамилия)

## АННОТАЦИЯ

Объем 50 с., 10 рис., 10 таблиц, 25 источников (в т.ч. 5 источников на английском языке).

Работа состоит из введения, 3 глав, заключения и приложения.

Объектом является мобильное приложения для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО «ТГУ».

Целью бакалаврской работы является разработка мобильного приложения для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО «ТГУ».

Актуальность работы заключается в том, что в ФГБОУ ВО «ТГУ» отсутствует мобильное приложение для планирования событий с отображением расписания учебных занятий. Модуль оповещения и интеграция сервисов планирования событий, позволит удобнее планировать своё время, совмещать его с расписанием занятий.

Мобильное приложения для планирования событий с интеграцией расписания учебных занятий должна будет получать данные с web-сервера ТГУ, обрабатывать их, и выводить пользователю на экране мобильного приложения. Модуль оповещения должен, получать данные с сторонних сервисов планирования событий, таких как Google Calendar, и отображать их в расписании пользователя. Платформа реализации системы будет Android Studio. Проведя анализ языков программирования, интегрированной среды разработки приложений, был сделан вывод, что наилучшим выбором будет язык Java, интегрированной среды разработки Android Studio.

В заключении формируется вывод по разработке мобильного приложения для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО «ТГУ».

## **ABSTRACT**

Volume 50 p., 10 pic., 10 tables, 25 sources (including 5 sources in English).  
The work consists of an introduction, three chapters, a conclusion and an appendix.  
The object is a mobile application for scheduling events with the integration of the timetable of the training sessions of the FGBOU VO "TSU".  
The aim of the bachelor's work is the development of a mobile application for event planning with the integration of the timetable of the training sessions of the FGBOU VO "TSU".

The urgency of the work lies in the fact that there is no mobile application for scheduling events in the TSGUU VO TSU with the schedule of training sessions displayed. The notification module and integration of event planning services will make it easier to plan your time, combine it with the schedule of lessons.

Mobile applications for scheduling events with the integration of the schedule of training sessions will have to receive data from the TSU web server, process them, and display them to the user on the screen of the mobile application. The alert module should receive data from third-party event planning services, such as Google Calendar, and display them in the user's schedule. Platform implementation of the system will be Android Studio. After analyzing the programming languages, the integrated application development environment, it was concluded that the best choice would be the Java language, the integrated development environment of Android Studio.

In conclusion, the conclusion is drawn on the development of a mobile application for event planning with the integration of the schedule of the training sessions of the FGBOU VO "TSU".

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	8
1 АНАЛИЗ СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ ПЛАНИРОВАНИЯ СОБЫТИЙ.....	10
1.1 Анализ предметной области на примере ФГБОУ ВО «ТГУ» с точки зрения разработки мобильного приложения для планирования событий...10	
1.2 Анализ системы публикация расписания ФГБОУ ВО «ТГУ».....	12
1.3 Анализ существующих систем планирования событий.....	15
1.4 Выработка требований к проектируемому приложению.....	16
1.5 Постановка задачи на разработку мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».....	19
2.1 Выбор платформы реализации мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».....	21
2.2 Разработка новой технологии мобильного приложения для планирования событий.....	22
2.3 Проектирование архитектуры мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».....	26
2.4 Разработка диаграммы последовательности по прецеденту «получение расписания групп».....	28
2.5 Разработка алгоритма модуля уведомления о важных событиях.....	29
2.6 Обоснование выбора алгоритма классификации событий.....	31
2.7 Разработка диаграммы классов мобильного приложения.....	34
3 РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ.....	38
3.1 Выбор средств реализации мобильного приложения.....	38
3.2 Реализация основных принципов работы мобильного приложения.....	41
3.3 Разработка алгоритма классификации.....	44
3.3 Разработка диаграммы развертывания мобильного приложения.....	46
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	50
ПРИЛОЖЕНИЕ А.....	53

## ВВЕДЕНИЕ

С течением времени прогресс и информационные технологии в частности стремительно развивается. Соответственно должны развиваться и корпоративные технологии, чтобы не потерять актуальность и востребованность. Особенно информационные системы публикации корпоративных данных, таких как расписание учебных занятий в ВУЗах.

Актуальность работы заключается в том, что автоматизации процесса планирования событий позволяет упростить процесс ориентирования во времени, благодаря четкому планированию графика, а интеграция расписания учебных занятий значительно сокращает время, необходимое на получение данных расписания, и позволяет предоставлять информацию в требуемой форме, например, мобильного приложения, которого на данный момент нет во ФГБОУ ВО «ТГУ».

Новизна данной работы заключается в том, что ранее в ТГУ не существовало единой системы публикации расписания и, реализуя в этой дипломной работе мобильное приложение с интеграцией расписания учебных занятий, она будет спроектирована, разработана и внедрена во ФГБОУ ВО «ТГУ».

Объектом работы мобильное приложение для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».

Предметом работы является технология разработки мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».

Целью выпускной квалификационной работы является разработка мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ». Для достижения цели необходимо выполнить следующие задачи:

- проанализировать текущую систему публикации расписания учебных занятий ФГБОУ ВО «ТГУ»;

- изучить современные сервисы для планирования событий;
- составить список требований к проектируемому приложению;
- спроектировать мобильное приложение для планирования событий с интеграцией учебных занятий ВУЗа;
- изучить технологии разработки мобильных приложений;
- разработать мобильное приложение для планирования событий с интеграцией расписания учебных занятий ВУЗа на основе полученных знаний и проектируемой модели приложения.

# **1 АНАЛИЗ СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ ПЛАНИРОВАНИЯ СОБЫТИЙ**

## **1.1 Анализ предметной области на примере ФГБОУ ВО «ТГУ» с точки зрения разработки мобильного приложения для планирования событий**

Тольяттинский государственный университет – высшее учебное заведение, созданное в 2001 году решением Правительства России на базе Тольяттинского политехнического института и Тольяттинского филиала Самарского государственного педагогического университета.

ТГУ – крупнейший и старейший государственный вуз города, осуществляющий подготовку по гуманитарно-педагогическому, техническому и естественнонаучному направлениям, а также один из немногих российских вузов имеющий военную кафедру и учебный военный центр.

Главный корпус университета расположен по адресу: 445020, г. Тольятти, ул. Белорусская, 14 (центральный кампус). В сети Интернет существует сайт университета расположенный по адресу: <http://www.tltsu.ru> Тольяттинский государственный университет объединяет институтов:

1. Архитектурно-строительный институт.
2. Гуманитарно-педагогический институт.
3. Институт изобразительного и декоративно-прикладного искусства.
4. Институт машиностроения.
5. Институт математики, физики и информационных технологий.
6. Институт права.
7. Институт энергетики и электротехники.
8. Институт химии и инженерной экологии.
9. Институт физической культуры и спорта.
10. Институт финансов, экономики и управления.
11. Институт военного обучения.



В структуру ТГУ входят 47 кафедр и работают более 2000 преподавателей и сотрудников, среди которых 100 докторов и более 450 кандидатов наук. В среднем во ФГБОУ ВО «ТГУ» учится 12 000 студентов.

С точки зрения организации учебного процесса каждому ВУЗу необходимо публиковать расписание учебных занятий, и наиболее эффективно эту работу будет выполнять система публикации расписания, в состав которой войдет мобильное приложение с выводом данных расписания учебных занятий. Роль этого приложения заключается в организации подключения к серверу системы публикации расписания ФГБОУ ВО «ТГУ», отправке запроса данных, необходимых для публикации расписания, получение самих данных, их преобразования в вид структуры мобильного приложения, отображение данных.

С точки зрения студентов, быстрый доступ к актуальному расписанию занятий играет важную роль. Благодаря ему, студент имеет сведения о месте и времени проведения занятий. Так же это даёт возможность быть всегда в курсе изменений и рассчитывать своё время.

Не менее важную роль имеет доступ к актуальному расписанию для преподавателей. Имея быстрый доступ к актуальному расписанию занятий, преподаватель может планировать своё время, оценивать нагрузку, имеет сведения об аудитории и группе, что в свою очередь даёт ему больше времени, чтобы подготовиться к предстоящему занятию.

Для других сотрудников ВУЗа, лаборантов, это так же необходимо для того, чтобы своевременно подготавливать аудитории к проведению занятий.

Расписание учебных занятий во ФГБОУ ВО «ТГУ» создается отделом учебно-методического управления и диспетчерской службой, а затем утверждается заместителем ректора по развитию учебного процесса. В момент составления расписания занятий, используется технология «Галактика Расписание учебных занятий», которая является дополнением ERP-системы «Галактика», позволяющая исключать ошибки, связанные с человеческим

фактором, во время создания расписания. Эта часть процесса составления расписания слишком сложна и велика, что в рамках данного проектирования рассматриваться не будет. Таким образом, данные, получаемые вследствие работы системы «Галактика Расписание учебных занятий» выступают в роли входных данных. Для публикации расписания учебных занятий, в едином информационном пространстве ФГБОУ ВО «ТГУ», экспортируются в MS Excel [2] сотрудником диспетчерской службы. Расписание составляется на 2 недели вперёд и публикуется на сайте университета.

Исходя из всего вышесказанного, можно сделать вывод, что подобная система публикации расписания не эффективной и даёт возможность для появления альтернатив. В данной дипломной работе будет рассмотрена технология разработки мобильного приложения, которая является частью организации учебного процесса ФГБОУ ВО «ТГУ», и на основании полученной технологии затем должна быть создана система публикации расписания занятий во ФГБОУ ВО «ТГУ».

## **1.2 Анализ системы публикация расписания ФГБОУ ВО «ТГУ»**

Чтобы проанализировать текущую систему публикации расписания, рассмотрим подробно процесс получения данных расписания.

Существует сайт университета, на котором публикуются табличные документы в формате xls с данными о расписании занятий студентов и преподавателей. Вид документа представляет собой большую сложную таблицу, в которой размещены данные о расписании занятий всего института по группам.

Алгоритм действий для просмотра расписания занятий для студента представлен на рисунке 1:

1. Открыть приложение браузер.
2. В адресной строке браузера набрать URL-адрес сайта ФГБОУ ВО «ТГУ» <https://www.tltsu.ru>.
3. На главной странице сайта перейти по ссылке «Студентам».

4. На странице «Студентам» перейти по ссылке «Расписание занятий».
5. На странице «Расписание занятий» выбрать необходимую неделю и свой институт.
6. Получить файл с расширением xls, в название которого входит название института и номер недели, и открыть его с помощью предустановленного приложения для просмотра табличных файлов.
7. Найти необходимый курс обучения.
8. Найти необходимую учебную группу.
9. Просмотреть расписание.

Для получения расписания занятий, студенту необходимо совершить 9 действий, чтобы посмотреть своё расписание занятий с мобильного устройства.

Из текущей модели получения расписания выделен ряд минусов:

- для просмотра расписания нужно получать файлы с расширением xls;
- чтобы найти требуемое расписание нужно выполнить 9 шагов;
- зачастую, на мобильных устройствах пользователей отсутствуют приложения для просмотра файлов с расширением xls;
- в файле с расписанием неудобно ориентироваться и нет возможности быстро находить информацию об учебных занятиях;
- нет возможности быстро получить информации о расписании по аудиториям.

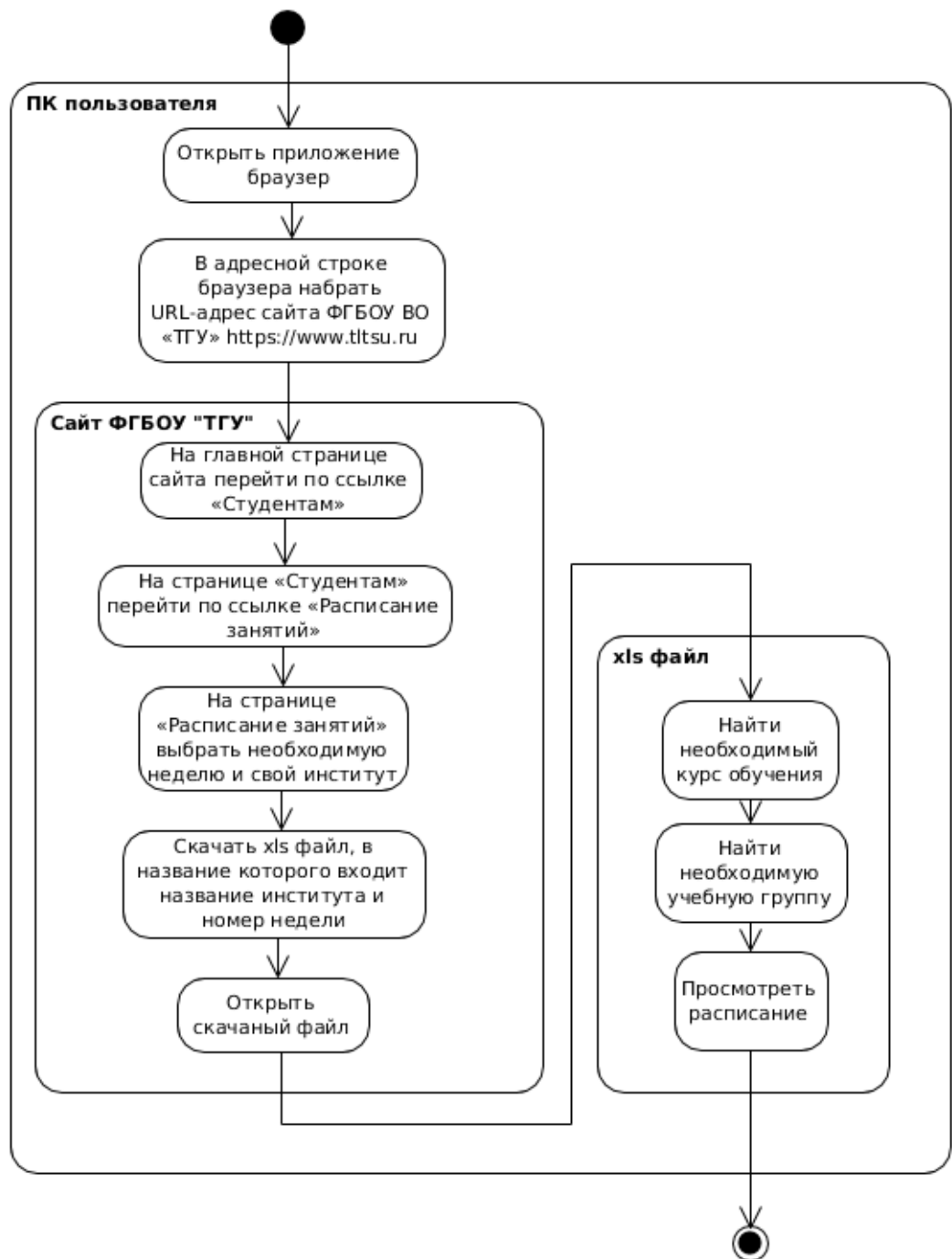


Рисунок 1 – Алгоритм получения данных расписания с точки зрения студента

Таким образом, резюмируя всё вышесказанное можно сделать вывод, что такая рядовая задача, как просмотр расписания учебных занятий во ФГБОУ ВО «ТГУ» с мобильного устройства отнимает сравнительно много времени и является процессом сложным и неудобным для выполнения, который следует автоматизировать и перевести в более удобную форму.

### 1.3 Анализ существующих сервисов планирования событий

Перед выработкой требований к проектируемому мобильному приложению следует изучить существующие программы, реализующие возможность планировать свой график. Основные сервисы для планирования событий — это различные органайзеры, такие как: Яндекс Календарь [9], Google Календарь [10], Календарь Windows Live [11].

Анализ систем планирования событий будет производиться критериям наличия API для интеграции данных в проекты сторонних разработчиков и способу передачи данных. Анализ систем планирования событий представлен в таблице 1.

Таблица 1 – Анализ систем планирования событий.

Название системы планирования событий	Яндекс календарь	Google Calendar	Windows Live
Наличие API	0	1	1
Библиотеки для импорта данных	0	1	0
Итого	0	2	1

Яндекс календарь не обладает открытым прикладным программным интерфейсом, что не позволяет использовать данные в проектах сторонних разработчиков, но в нём реализована возможность импорта данных в формате ICS.

Google календарь обладает прикладным программным интерфейсом для интеграции данных в проекты сторонних разработчиков. Кроме того, компанией предоставлены обширный выбор библиотек для упрощения работы с процессом импорта данных

В календаре Windows Live есть возможность получить данные о событиях в формате обмена данными – json [13]. Однако, этот органайзер не

предоставляет готовых библиотек для упрощения работы с процессом импорта данных.

Таким образом, основываясь на критерии интеграции данных, для реализации импорта данных с сервиса Google Calendar принято решение использовать библиотеки, предоставленные компанией Google.

#### **1.4 Выработка требований к проектируемому приложению**

Проектируемая технология планирования событий будет представлять из себя мобильное приложение, которое будет на основе входных и вводимых пользователем данных формировать календарь событий из совокупности данных.

Для разработки требований [2] возьмем систему классификации требований FURPS+ [14].

В ходе реализации практической части этой бакалаврской работы, будет разработано мобильное приложение для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО «ТГУ» и событий из органайзера Google календарь. Цель создания приложения: обеспечение быстрого и удобного доступа к информации расписания студентов и преподавателей.

В разрабатываемом приложении будет реализован следующий алгоритм получения данных о расписании занятий ФГБОУ ВО «ТГУ».

При первом запуске приложения:

1. Открыть приложение.
2. Указать институт и группу.
3. Просмотреть расписание.

При последующих запусках пункт 2 пропускается. Очевидные плюсы этой модели:

- быстрый и удобный доступ к информации о расписании занятий;
- отсутствие необходимости иметь предустановленное приложения для просмотра табличных файлов;

- возможность кэширования данных, для просмотра расписания даже при отсутствии подключения к сети Интернет;
- удобное представление данных;
- запуск узкоспециализированного мобильного приложения тратит намного меньше ресурсов устройства, чем программа для просмотра xls-файлов;
- возможность просмотреть расписание в конкретных аудиториях.

Перечень требований к мобильному приложению для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ» приведен в таблице 2, где в столбце «Статус», «О» означает «Одобренные», «П» - «Предложенное», а в столбце полезность, «К» означает «Критичное», «В» - Важное, «П» - «Полезное».

Таблица 2 – Требования к мобильному приложению

№	Требование	Статус	Полезность	Обоснование выбора
Functionality – Функциональные требования				
1.	Отображение списка преподавателей, групп, аудиторий, кафедр, институтов и сведения по каждому из них.	О	К	В приложении должно быть реализовано хранение информации о преподавателях, группах, аудиториях, кафедрах и институтах
2.	Отображение расписания и событий со стороннего органайзера	О	В	В приложении должна быть реализована возможность интегрировать данные из сторонних органайзеров для комплексного планирования времени
3.	Создание записи о событиях	О	К	В приложении должна быть реализована функция создания записи о событиях
4.	Управление настройками приложения	О	В	В приложении должна быть реализована панель настроек

Продолжение таблицы 2

5.	Уведомление об изменении расписания	О	К	Приложение должно оповещать об изменениях в расписании
6.	Уведомление о предстоящем событии	П	В	В приложении должна быть реализована функция уведомления о предстоящем событии
Usability – Требования к удобству использования				
7.	Обеспечение возможности интерактивной справки	П	П	Приложение должно иметь инструмент для помощи пользователям
8.	Читаемые шрифты	П	В	Весь текст приложения должен быть удобен для чтения
9.	Комфортная цветовая гамма	П	В	Цветовая гамма приложения не должна вызывать отторжения и избыточно напрягать глаза пользователя
Reliability – Требования к надежности				
10.	Реализованное приложение должно проверять входные данные на наличие ошибок.	П	К	Приложение должно анализировать входные данные на наличие ошибок и уметь обрабатывать их
Performance – Требования к производительности				
11.	Время запуска приложения на устройстве не должно превышать 5 секунд	О	К	Приложение должно иметь минимальную по времени реакцию для ответа на действия пользователя
12.	Обработка запроса одного пользователя не должна превышать 1 секунду	О	К	
Supportability - Требование к поддержке				



Продолжение таблицы 2

13.	При реализации приложения, использовать контроль версий	О	В	Приложение должно поддерживаться системой контроля версий для простой работы над проектом несколькими людьми одновременно
14.	При реализации приложения, создавать проектную документацию	О	В	Приложение должно иметь документацию для быстрого ввода в курс дела новых людей в проекте
Interface requirements – Требование к интерфейсу				
15.	Приложение должно получать данные из сторонних сервисов для планирования событий	О	К	Мобильные приложения реализуют получение данных со сторонних ресурсов
Physical requirements – Физические требования				
16.	Наличие интернет соединения	О	К	На устройстве должно присутствовать интернет подключение для корректной работы и выкачки данных

Таким образом, на основе анализа необходимо реализовать 6 функциональных требований и 10 не функциональных ограничений, представленных в таблице. Из функциональных требований самыми важными являются, отображение списков, событий и расписания. Из не функциональных самым приоритетным является - время запуска приложения не должно превышать более 5 секунд.

### **1.5 Постановка задачи на разработку мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ»**

Целью работы является разработка мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».

Приложение должно отвечать всем важным требованиям, выдвинутым в разделе 1.4 по системе классификации требований FURPS+.

Задачами на разработку мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ» являются:

1. Разработка технологии мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».
2. Разработка архитектуры мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».
3. Разработка взаимодействия между элементами мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ» посредством диаграммы последовательности.
4. Разработка диаграммы классов мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».
5. Разработка модуля создания списка событий.
6. Разработка модуля уведомления.
7. Разработка математической модели.
8. Выбор средств реализации мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».
9. Выбор платформы реализации мобильного приложения, языка программирования и интегрированной среды разработки.
10. Описание основных принципов работы мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».

По итогу в этой главе, был произведен анализ системы публикации расписания ФГБОУ ВО «ТГУ», анализ систем планирования событий, а также произведена выработка требований к будущему приложению и осуществлена постановка задачи на разработку мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ».

## 2 ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

### 2.1 Выбор платформы реализации мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ»

На основании отчёта аналитической компании Statcounter об количестве операционных систем на устройствах пользователей (<http://gs.statcounter.com/os-market-share>), доля Android [3] занимает 76.53% всего рынка мобильных операционных систем в мае месяце 2018 года, доля IOS (Iphone OS) составляет 18.97% рынка, остальные 4.5% поделены между Windows Phone, BlackBerry OS и прочими малоиспользуемыми мобильными операционными системами. Исходя из статистики, наибольшее количество устройств в мире работают с операционной системой Android.

На рисунке 2 представлена статистика использования операционных систем с официального источника Statcounter, где первый столбец диаграммы – данные использования мобильной операционной системы Android в мире.

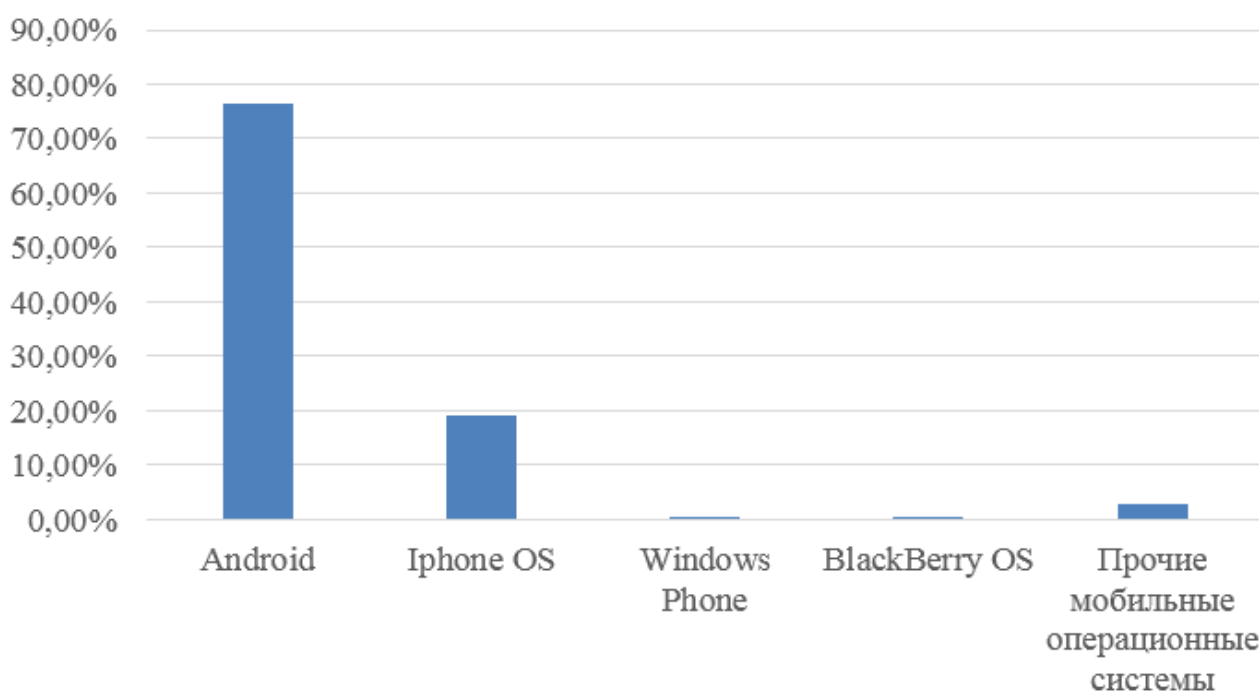


Рисунок 2 – Статистика использования операционных систем

Таким образом, выбор платформы реализации падает на операционную систему Android, ввиду наибольшего распространения этой системы на мобильных устройствах пользователей.

## **2.2 Разработка новой технологии мобильного приложения для планирования событий**

Диаграмма вариантов использования используется в данной выпускной квалификационной работе для того, чтобы смоделировать функциональные требования к проектируемой системе.

Так как пользователями являются студенты, преподаватели и лаборанты, то варианты использования данной системы для них только один — получение сведений о событиях.

Актеры:

1. Пользователь – человек, использующий мобильное приложение для планирования событий.
2. Таймер – класс, инициирующий некоторые события в приложении (Например, оповещение о предстоящем событии).

Прецеденты разрабатываемого приложения описаны в таблицах 3,4,5 ,6.

Таблица 3 — Описание прецедентов

ID	Название прецедента	Описание
1.	Отображение списка институтов, преподавателей, аудиторий, групп, кафедр.	функция, которая будет высылать на запрос пользователя список институтов, преподавателей, аудиторий, групп, кафедр.
2.	Отображение расписания групп, преподавателей, аудиторий	функция, которая будет высылать на запрос пользователя расписание аудиторий, преподавателей и групп

Продолжение таблицы 3

3.	Проверка изменения расписания	функция, которая будет проверять данные о конкретной группе или преподавателе на изменение расписания
4.	Создание записи о событии в календаре	Функция, которая будет создавать запись о событии и сохранять его в локальной памяти устройства
5.	Получение списка событий из сервисов планирования событий	Функция, которая будет высылать запрос на получение списка событий из сервисов планирования событий
6.	Сортировка событий по времени и приоритету	Функция, которая будет сортировать данные из разных источников в порядке наступления событий и приоритету события
7.	Настройка уведомлений, сохранения и импорта данных со сторонних систем планирования событий	Функция, которая будет вызывать экран активности с панелью настроек управления приложением.
8.	Присваивание атрибута «приоритет» полученным событиям	Функция, которая будет автоматически присваивать новым событиям приоритет, путём математической обработки данных
9.	Оповещение о приближающемся событии	Функция, которая будет оповещать пользователя о предстоящем событии за определённое время до наступления времени события.

Из 9 представленных в таблице прецедентов, 3 (Оповещение о приближающемся событии, Присваивание атрибута «приоритет» полученным событиям, Сортировка событий по времени и приоритету) происходят по инициативе актёра – таймер и 6 по инициативе пользователя мобильного приложения.

Таблица 4 – Прецедент «Получение расписания групп»

Прецедент: Отображение расписания групп, преподавателей, аудиторий
ID: 2
Краткое описание: Реализация функции получения расписания групп.
Главные актеры: 1. Пользователь
Второстепенные актеры:
Предусловие: Прецедент начинается по инициативе пользователя
Основной поток: 1. Пользователь запрашивает расписание групп. 2. Приложение выводит список институтов. 3. Пользователь выбирает необходимый институт. 4. Приложение выводит список групп института. 5. Пользователь выбирает необходимую группу. 6. Приложение выводит данные о расписании учебных занятий.
Постусловие: Приложение получило и обработало данные с сервера
Альтернативные потоки: Нет

Таблица 5 – Прецедент «Получение списка событий из календаря Google»

Прецедент: Получение списка событий из календаря Google
ID: 8
Краткое описание: функция, которая будет высылать запрос на получение списка событий в календаре Google
Главные актеры: 1. Пользователь
Второстепенные актеры:
Предусловие: Прецедент начинается по инициативе пользователя

Продолжение таблицы 5

<p>Основной поток:</p> <ol style="list-style-type: none"> <li>1. Приложение отправляет запрос на сервер на получение данных.</li> <li>2. Приложение обрабатывает данные, создавая объект коллекции списка событий в календаре.</li> <li>3. Приложение интегрирует в список событий полученные данные.</li> </ol>
<p>Постусловие:</p> <p>Приложение произвело проверку на изменения в расписании</p>
<p>Альтернативные потоки:</p> <p>Нет</p>

Таблица 6 – Прецедент «Создание уведомление для пользователей»

<p>Прецедент: Оповещение о приближающемся событии</p>
<p>ID: 16</p>
<p>Краткое описание: функция, которая будет оповещать пользователя о предстоящем событии за определённое время до наступления времени события</p>
<p>Главные актеры:</p> <ol style="list-style-type: none"> <li>1. Таймер</li> </ol>
<p>Второстепенные актеры:</p>
<p>Предусловие:</p> <p>Прецедент начинается по наступлении определённого времени</p>
<p>Основной поток:</p> <ol style="list-style-type: none"> <li>1. Таймер приложения срабатывает за фиксированный промежуток времени до наступления события.</li> <li>2. Приложение создаёт уведомление о скором наступлении события.</li> </ol>
<p>Постусловие:</p> <p>Приложение создало уведомление о скором наступлении события</p>
<p>Альтернативные потоки:</p> <p>Нет</p>

Диаграмма вариантов использования отображает функциональные возможности проектируемого мобильного приложения для публикации расписания. Например, актер «пользователь» инициирует вариант использования «Просмотр расписания группы», целью которого является получить данных о том какие есть преподаватели в конкретном институте.

На рисунке 3 представлена диаграмма вариантов использования проектируемого приложения.

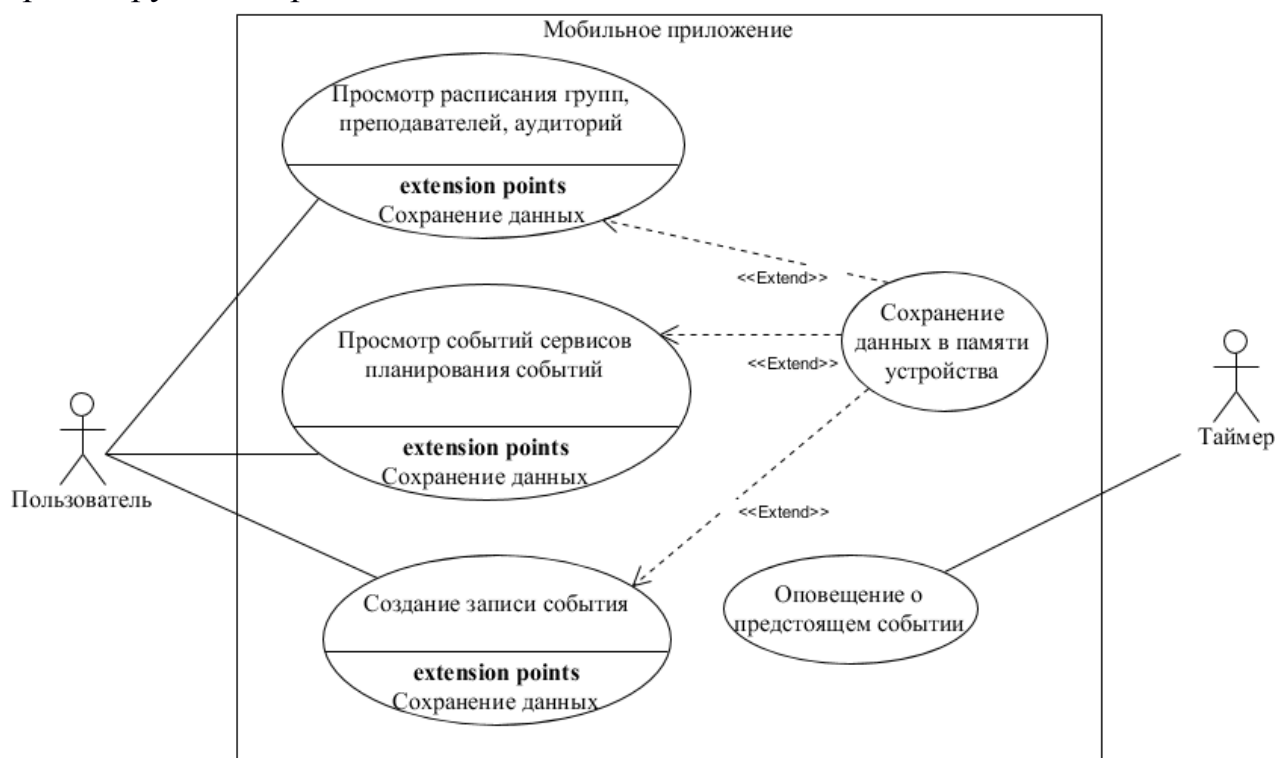


Рисунок 3 – Диаграмма вариантов использования проектируемого мобильного приложения

Таким образом, в данном пункте были составлены основные прецеденты, описаны самые важные из них, выделены 2 актера, и составлена диаграмма прецедентов в которой было показано, какой актёр взаимодействует с каким прецедентом.

### 2.3 Проектирование архитектуры мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ»

Мобильное приложение для планирования событий будет реализовано как клиент в архитектуре клиент-сервер. Клиент будет получать данные из нескольких источников: web-сервер приложений системы публикации расписания ФГБОУ ВО «ТГУ», Google календарь и локальной памяти устройства.



Первое, из основных преимуществ такой модели это — возможность отображения данных даже, если один из источников будет недоступен. Например, при отсутствии соединения с Интернет, приложение будет отображать все событий из локальной памяти устройства.

Второе преимущество это - возможность комбинирования событий из разных источников в единую модель публикации событий с возможностью их сортировки.

Приложение будет состоять из 3 компонентов:

1. Классы управления приложением.
2. Классы прикладного программного интерфейса системы публикации расписания учебных занятий ФГБОУ ВО «ТГУ».
3. Класс прикладного программного интерфейса Google календарь.

Диаграмма компонентов мобильного приложения для планирования событий с интеграцией данных представлена на рисунке 4.

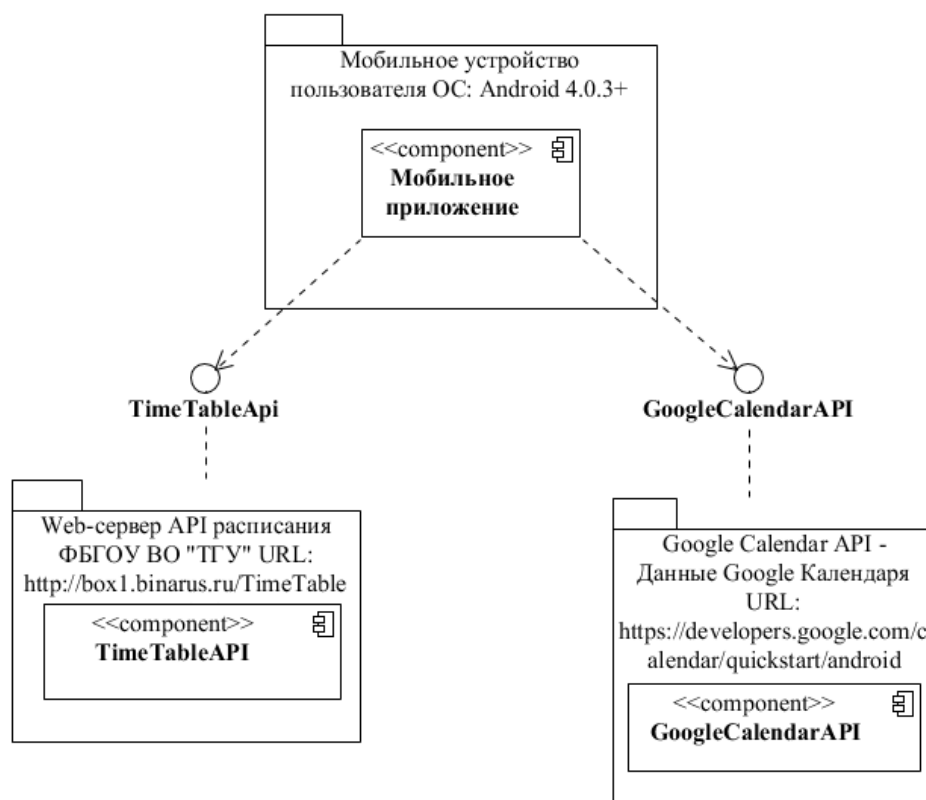


Рисунок 4 – Диаграмма компонентов проектируемого мобильного приложения

В представленной на рисунке 4 диаграмме компонентов видно, что приложение имеет 2 внешних источника данных, это — API Google календарь и API расписания ФГБОУ ВО «ТГУ», а все функции по управлению данными реализуются в компоненте «Мобильное приложение».

#### 2.4 Разработка диаграммы последовательности по прецеденту «Отображение расписания групп, преподавателей, аудиторий»

В соответствии с функциями системы необходимо реализовать диаграмму последовательности функцию отображения расписания. Для этого, системе нужно выполнить следующий алгоритм действий:

1. Пользователь запрашивает расписание групп.
2. Приложение выводит список институтов.
3. Пользователь выбирает необходимый институт.
4. Приложение выводит список групп института.
5. Пользователь выбирает необходимую группу.
6. Приложение выводит данные о расписании учебных занятий.

Процесс получения расписания представлен на рисунке 5.



Рисунок 5 – Диаграмма последовательности по прецеденту «Отображение расписания групп, преподавателей, аудиторий»

Построенная диаграмма последовательности отображает динамический аспект объекта автоматизации и позволяет определить порядок реализации функций, что в свою очередь позволяет разработчику создать мобильное приложение так, как оно было задумано.

### **2.5 Разработка алгоритма модуля уведомления о важных событиях**

Перед реализацией приложения необходимо решить проблему классификации уведомлений по атрибуту «приоритет». Проблема состоит в том, что есть необходимость уведомлять пользователя далеко не о всех событиях, но при этом уведомлять о событиях с атрибутом «важное». Вручную расставлять приоритеты на все события отнимет у пользователя слишком много времени, а игнорирование этого атрибута приведёт к тому, что приложение будет уведомлять пользователя о всех, даже малозначительных событиях

После получения всех данных для построения списка событий приложение по умолчанию производит сортировку по их времени начала в порядке убывания. Для этого приложение формирует полный список событий на день из всех источников данных, сравнивает время их начала и первым пунктом устанавливается событие с самым ранним временем начала события и так далее.

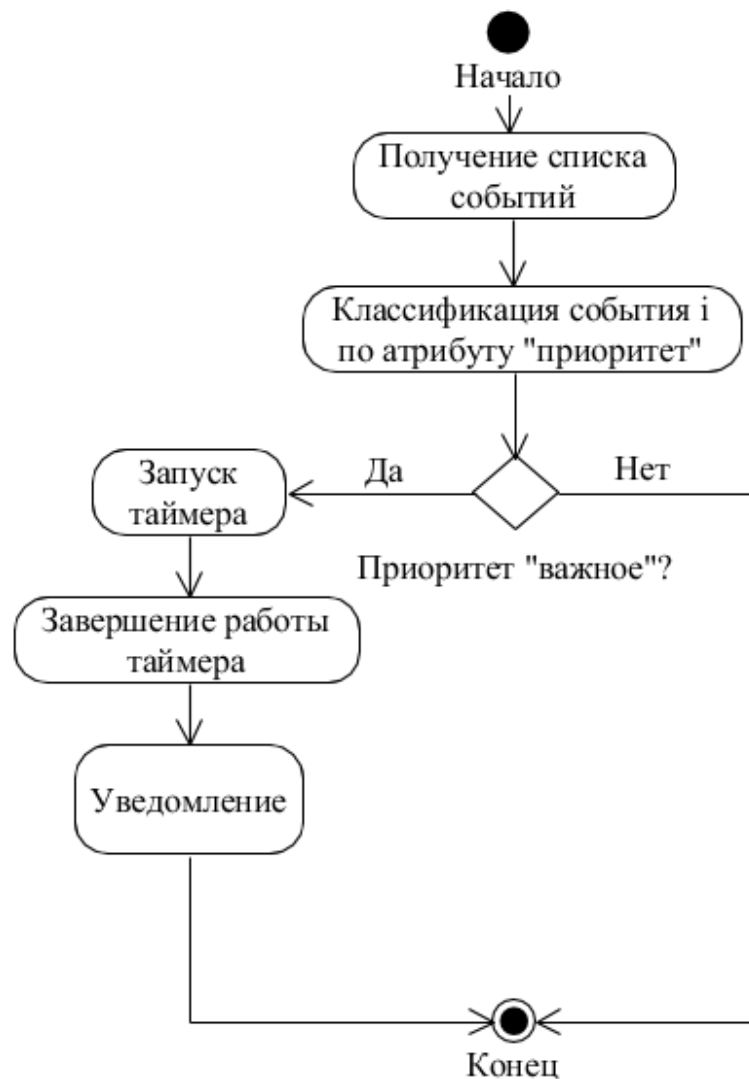
Алгоритм модуля оповещения будет реализован следующим образом:

1. Получение полного списка событий в текущем месяце.
2. Классификация в цикле каждого события по атрибуту «приоритет».
3. Запуск таймера на оповещение перед событиями с приоритетом «очень важное» и «важное».
4. Уведомления по окончанию работы таймера.

Полученные данные приложение будет приводить к общему виду представления данных приложения. Разрабатываемая структура полей события:

- название события;
- дата начала события;
- время начала события;
- дата окончания события;
- время окончания события;
- место события;
- описание события;
- приоритет.

Пункт «приоритет» будет списочным, а значит будет иметь заранее подготовленные значения поля: очень важное, важное, обычное, желательное, не важно. Блок-схема алгоритма представлена на рисунке 6.



## Рисунок 6 – Алгоритм модуля уведомления

Таким образом, сформирован алгоритм оповещения о важных событиях, который требует реализации алгоритма классификации событий для экономии времени пользователей.

### **2.6 Обоснование выбора алгоритма классификации событий**

В реализации модуля оповещения о важных событиях возникает проблема классификации событий по важности. Для реализации классификатора лучше всего прибегнуть к использованию одного из алгоритмов классификации данных, который будет автоматически устанавливать атрибут «приоритет» каждому новому событию, который впоследствии можно будет изменить вручную, при необходимости.

Выбор алгоритма классификации. Ввиду того, что приложение должно будет принимать решение о присваивании атрибута самостоятельно, разумно использовать алгоритм, который будет принимать решения исходя из обучающей выборки событий с уже проставленными метками приоритета. Поэтому нужно использовать алгоритм, реализующий дерево принятия решений. Существуют несколько алгоритмов, решающих эту задачу. Например, C4.5 [4] и CART [4].

Так как алгоритм CART имеет недостаток - деревья решений, созданные алгоритмом, нестабильны: результат, полученный на одной выборке, бывает не воспроизводим на другой, был выбран алгоритм - C4.5.

Исходными данными задачи классификации является множество примеров событий  $T$ . Элементы этого множества событий описывается атрибутами полей, используемых для хранения данных в приложении для планирования событий. Пусть метка класса принимает значения атрибута «приоритет»: очень важное, важное, обычное, желательное, не важное.

Задача алгоритма заключается в построении дерева из множества примеров событий  $T$  из заранее подготовленной выборки событий. Процесс

построения дерева будет происходить сверху вниз. Сначала создается корень дерева, а затем потомки корня и листья.

На первом шаге дерево не имеет узлов (только корень) и множество событий  $T$  (ассоциированное с корнем). Необходимо разбить исходное множество на подмножества. Для этого нужно выбрать один из атрибутов в качестве проверки. Тогда в результате разбиения получаются  $n$  (по числу значений атрибута, в данном случае  $n=5$ ) подмножеств  $i$ , соответственно, создаются 5 потомков корня, которым ставится в соответствие свое подмножество, получаемое при разбиении множества событий  $T$ . Затем эта процедура рекурсивно применяется ко всем потомкам корня и т.д.

Очевидно, что из  $m$  (по числу атрибутов) возможных вариантов, требуется выбрать самый подходящий.

Допустим, мы имеем проверку  $X$  (в качестве проверки может быть выбран любой атрибут, например, наименование события – поле name), принимающая 5 значений: A1-очень важное, A2-важное, A3-обычное, A4-желательное, A5-не важное. Тогда разбиение  $T$  по проверке  $X$  даст нам подмножества  $T_1, T_2 \dots T_5$ , при  $X$  равном соответственно A1, A2 ... A $n$ . Единственная доступная информация – распределение классов во множестве  $T$  и его подмножествах, которые получаются при разбиении по  $X$ . Это и играет роль при определении критерия.

Пусть  $freq(C_j, S)$  – количество примеров событий из некоторого множества  $S$ , которые относятся к тому же классу  $C_j$ . Тогда вероятность того, что случайно выбранный пример из множества  $S$  будет принадлежать к классу  $C_j$ , согласно теории информации, количество содержащейся в сообщении информации, зависит от ее вероятности  $\log_2(1/P)$ .

Где  $P$  – количество примеров, которые относятся к определённому классу согласно выбранному критерию разбиения.

Выражение оценивает среднее количество информации, которое необходимо для определения класса примера из множества событий  $T$ . В

терминологии теории информации выражение (1) является энтропией множества  $T$ .

$$Info(T) = \sum_{j=1}^k \frac{freq(C_j, T)}{|T|} * \log_2 \left( \frac{freq(C_j, T)}{|T|} \right) \quad (1)$$

Где  $Info(T)$  – среднее количество информации, необходимое для классификации события,

$j$  – порядковый номер события во множестве  $T$ ,

$k$  – общее количество событий в выборке,

$freq(C_j, T)$  – количество событий, относящихся к классу  $C_j$ ,

$C_j$  – один из атрибутов приоритета,

$T$  – множество событий выборки.

Ту же оценку, но уже после разбиения множества событий  $T$  по критерию  $X$ , дает следующее выражение:

$$Info_X(T) = \sum_{i=1}^n \frac{|T_i|}{|T|} * Info(T_i) \quad (2)$$

Где  $Info_X(T)$  – оценка количества информации после разбиения,

$X$  – текущий атрибут классификации,

$i$  – идентификатор события,

$n$  – общее количество событий в узле,

$T_i$  – событие выборки с номером  $i$ ,

$T$  – множество событий выборки,

$Info(T_i)$  - среднее количество информации, необходимое для классификации события  $i$ .

Тогда критерием для выбора атрибута будет являться формула:

$$Gain(X) = Info(T) - Info_X(T) \quad (3)$$

Где  $Gain(X)$  – критерий для выбора атрибута, выраженный в битах,

$X$  – текущий атрибут классификации,

$Info(T)$  – среднее количество информации, необходимое для классификации события,

$Info_x(T)$  – количество информации, получаемое при выборе текущего атрибута.

Критерий Gain рассчитывается для всех независимых переменных, после чего выбирается переменная с максимальным значением Gain.

Необходимо выбрать такую переменную, чтобы при разбиении по ней один из классов имел наибольшую вероятность появления. Это возможно в том случае, когда энтропия  $Info_x$  имеет минимальное значение и, соответственно, критерий  $Gain(X)$  достигает своего максимума.

Критерий (3) считается для всех атрибутов. Выбирается атрибут, который будет максимальным для данного выражения. Этот атрибут станет проверкой в текущем узле, а потом по этому атрибуту будет производиться дальнейшее построение дерева.

Такие же операции можно применяться к полученным подмножествам  $T_1, T_2 \dots T_5$  и продолжаютя рекурсивно до тех пор, пока в узле не останутся примеры только из одного класса.

Таким образом, текущая математическая модель позволяет построить дерево принятия решений, которое будет классифицировать событий по атрибуту «приоритет» автоматически на основе данных выборки событий.

## 2.7 Разработка диаграммы классов мобильного приложения

Разработка диаграммы последовательности позволила выделить пакеты и определить отношения между ними в соответствии с особенностями предметной области. В таблице 7 представлены классы и их роли в проектируемой системе.

Таблица 7 - Описание ролей классов в системе

Название класса	Роль класса в системе
Классы отображения экранов активностей	
ScreenSelector	Отображает элементы управления для переходов между экранами активности приложения



<b>Название класса</b>	<b>Роль класса в системе</b>
InstList	Отображает список институтов университета, позволяет их выбирать
GroupList	Отображает список групп института, позволяет их выбирать
SheduleList	Отображает расписание групп\преподавателей
KafList	Отображает список кафедр института, позволяет их выбирать

Продолжение таблицы 7

TeachList	Отображает список преподавателей
EventList	Отображает список событий
AppSettings	Отображает панель настроек приложения
Классы – сущности	
Insts	Описывает данные институтов
Groups	Описывает данные групп
Shedule	Описывает данные расписания
Kafs	Описывает данные кафедр
Teachers	Описывает данные преподавателей
Events	Описывает события в календаре
Notification	Описывает данные уведомления
Классы обработки данных	
GoogleQuickStart	Получает данные событий из календаря Google
Timer	Контролирует время выдачи уведомлений
Combiner	Формирует данные событий из разных источников в единый список
C45Classifier	Классифицирует список событий по атрибуту «приоритет»

Класс Shedule – класс, который реализует:

1. Получение данных расписания в формате json;
2. Разбирает его на поля и значения, формирует объекты в памяти приложения;
3. Создает коллекцию объектов для вывода расписания на экран;
4. Записывает расписание в кеш.

Класс ScreenSelector – класс, который реализует генерацию разметки кнопок выбора расписания групп, преподавателей или аудиторий, впоследствии перенаправляет на список выбранного расписания.

Класс C45Classifier – класс, реализующий классификацию новых событий по атрибуту «приоритет». Класс Timer – реализует все события,

происходящие по заданному времени. В таблице 8 приведена спецификация класса Shedule.

Таблица 8 — Спецификация класса Shedule

Именованние	Назначение
doInBackground	Создаёт соединение с сервером
getJSONFile	Выкачивает данные расписания с сервера с API
parseJson	Выполняет разбор данных из json-файла и создаёт коллекцию объектов
writeToMemory	Сохраняет коллекцию объектов расписания в локальную память устройства

На рисунке 7 изображена диаграмма классов, которые реализуют 2 из основных функций разрабатываемого мобильного приложения — получение и обработка данных с сервера и преобразование данных в форму, пригодную для вывода на экране приложения.

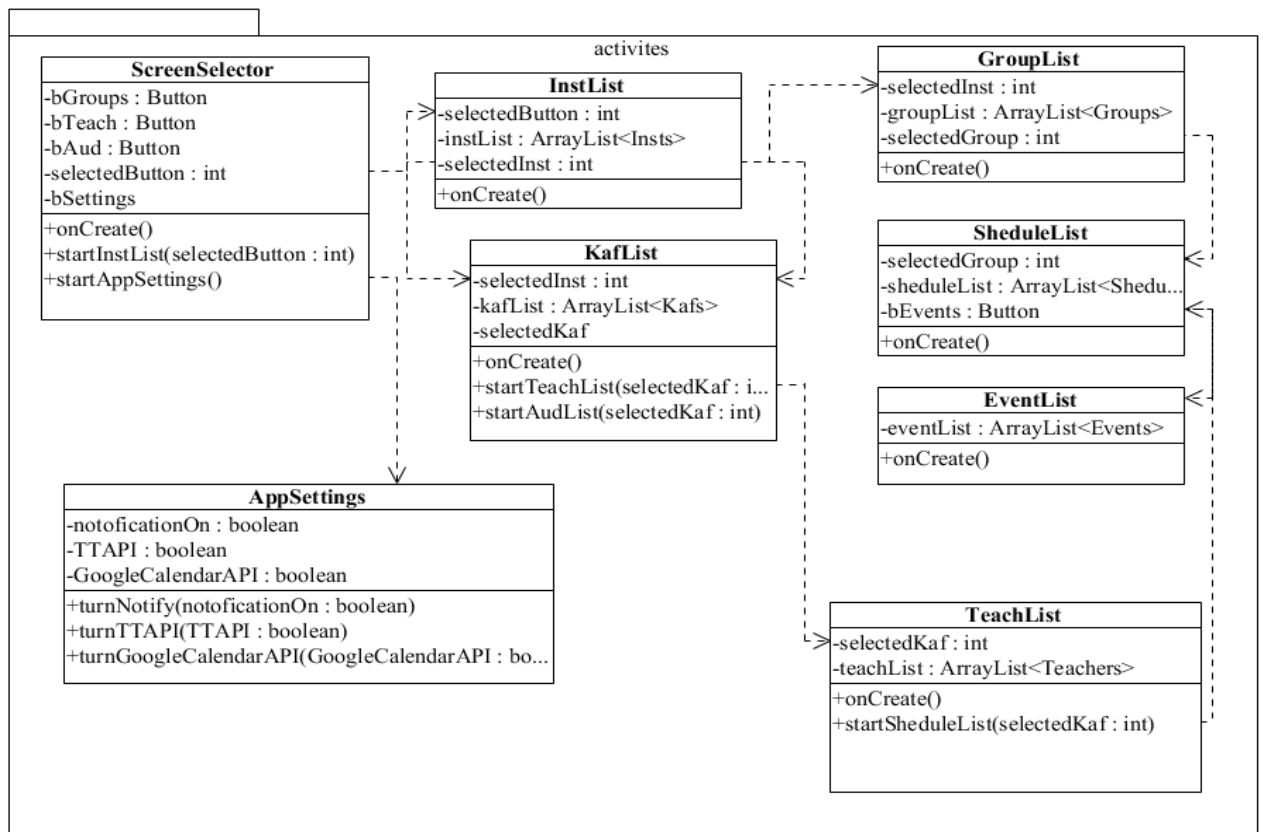


Рисунок 7 – Диаграмма классов стереотипа «Классы отображения экранов активностей»

Классы Insts, Groups, Shedule, Kafs, Teachers, Events и Notifications являются классами сущностями, то есть содержат поля описывающие их методы-конструкторы.

Класс GoogleQuickStart – класс использующий стандартные библиотеки GoogleCalendar API и служит для получения данных из календаря Google.

Классы Timer, Combiner и C45Classifier – являются классами, которые запускают фоновые процессы обработки данных внутри приложения и содержат алгоритмы обработки таких данных как время, дата, место и приоритет событий.

Таким образом, были спроектированы основные классы и методы для реализации мобильного приложения.

В итоге этой главы были построены диаграммы: вариантов использования, последовательности, компонентов, классов. Спроектированы модули формирования списков, присваивания атрибута «приоритет» и оповещения. Изъяснён основной алгоритм сортировки данных, получаемых из различных источников. Описаны основные классы приложения и методы класса Shedule.

### 3 РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

#### 3.1 Выбор средств реализации мобильного приложения

Был произведён сравнительный анализ наиболее распространённых языков разработки мобильных приложений, среди которых: Java, C#.

Java - объектно-ориентированный язык программирования, где выполнение кода осуществляется с помощью виртуальной Java-машины [23].

C# - объектно-ориентированный язык программирования. Разработан в 1998—2001 годах инженерами компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота для разработки приложений платформы Microsoft - .NET Framework.

Результаты анализа языков программирования приведены в таблице 9.

Таблица 9 - Сравнительный анализ языков программирования

Функции и возможности	Языки программирования	
	Java	C#
Производительность кода и требовательность к ресурсам	5	6
Наличие библиотек	7	5
Удобство отладки	7	4
Читаемость языка и анализ синтаксиса	6	7
Удобство сборки	7	5
Опыт работы	6	1
Итого:	38	27

В итоге сравнительного анализа, был выбран язык объектно-ориентированного программирования – Java. Данный язык достаточно удобен и прост в реализации написания программного кода приложений. Кроме того, он имеет большой набор различных библиотек, способствующих решению большинства типизированных и узкоспециализированных задач. Очень важную роль играет опыт разработки на этом языке, потому что благодаря нему процесс

разработки мобильного приложения для планирования событий с интеграцией расписания ФГБОУ ВО «ТГУ» будет реализован быстрее.

Выбор интегрированной среды разработки для реализации приложения.

Интегрированная среда разработки — важный инструмент, используемый программистами при разработке программного обеспечения, поэтому правильный выбор среды разработки упростит для программиста реализацию написания кода программы благодаря средствам анализа создаваемого кода в реальном времени и уведомлениям отладчика об ошибках на этапе компиляции программного кода. Выбор IDE проводится на основе сравнительного анализа популярных интегрированных средств разработки, работающих с языком Java, т. к. он был выбран ранее в пункте 3.1. Среди популярных интегрированных средств разработки были отобраны для анализа NetBeans, Eclipse, IntelliJIDEA, Android Studio.

NetBeans - свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и других.

Eclipse - свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

IntelliJ IDEA — интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains. Но в виду того что есть среда Android Studio, основанная на IntelliJ IDEA, анализ будет проводиться только с Android Studio.

Android Studio - Android Studio — это интегрированная среда разработки для работы с платформой Android на языках Java, Kotlin.

Для того, чтобы выбрать более подходящую среду разработки, проведем выбор по следующим критериям:

- производительность — к производительности можно отнести, насколько быстро у IDE скорость сборки проектов, их обработка и компиляция;
- совместимость плагинов и библиотек — к совместимости относится то, насколько корректно работают плагины и библиотеки в данных IDE;
- обновление документации - то, насколько своевременно обновляется документация к данным средам разработки;
- опыт работы – личный опыт работы программиста в IDE.

Результаты сравнительного анализ интегрированных сред разработки представлены в виде таблицы 10, с перечнем критериев (слева), названием сред разработки сверху и таблицей субъективной оценки (по центру).

Таблица 10 — Сравнительный анализ интегрированных сред разработки

Интегрированные среды разработки	NetBeans	Eclipse	Android Studio
Производительность	7	7	6
Совместимость плагинов и библиотек	6	5	8
Обновление документации	5	4	7
Опыт работы	6	3	8
Итого:	27	19	29

По итогам анализа сравнения IDE, была выбрана интегрированная среда разработки – Android Studio. Эта IDE является достаточно удобной при разработке мобильных приложений в виду её узконаправленной специализации именно на разработку мобильных приложений для мобильных устройств. Кроме того, она имеет большой набор функций для создания графического интерфейса приложения и тестирования разрабатываемого ПО прямо во время разработки, благодаря встроенным XML редактору и средствам эмуляции операционной системы Android. Наиболее важную роль играет опыт разработки в этой IDE, потому что именно благодаря нему процесс разработки будет реализован быстрее и удобнее для разработчика.

### **3.2 Реализация основных принципов работы мобильного приложения**

Основная задача, которая стоит перед реализуемым программным обеспечением - это публикация данных событий на экране мобильного устройства.

Выделим основные функции системы:

- Загрузка данных с сервера расписания ФГБОУ ВО «ТГУ» в формате json;
- Преобразование данных в пригодную для вывода форму;
- Вывод данных расписания на экран приложения;
- Сохранения данных расписания о выбранной группе/преподавателе;
- Проверка на наличие изменений в расписании;

Реализация функции загрузки данных расписания учебных занятий.

Реализация классов, находящихся в пакете component. Эти классы отвечают за загрузку и обработку данных из jsonAPI и преобразование их форму, необходимую для вывода данных. Все они работают по похожему алгоритму. Единственная разница состоит в том, что они используют данные из разных json API.

Алгоритм реализации классов пакета component:

- Открытие сетевого соединения с сервером, откуда выкачиваются данные в формате json;
- Создание объекта, в котором размещаются данные json в не изменённом виде;
- Обработка данных и приведении их в необходимую для отображения форму;
- Запись данных в объекты класса.



На рисунке 8 представлена блок-схема алгоритма реализации классов пакета component.



Рисунок 8 – Блок-схема парсинга данных

Далее происходит реализация обработки полученных данных в формате json стандартными средствами работы с json. Все поля, относящиеся к данному виду, имеют постфиксы, указывающие на тип данных. Например, все поля, оканчивающиеся на “Name” являются указателем на название содержимого, хранимого в поле, а поля с постфиксом “Id” хранят в себе идентификатор объекта в json.

На примере, происходит разбор данных json формата стандартными средствами java для работы с данными в формате json.

Для того чтобы преобразовать данные из формата json в поля необходимо создать JSONObject, в который помещается содержимое полученного файла и получить доступ к его полям и их значениям через их имена. В данном случае здесь описаны названия групп института и их идентификатор:

- groupInstName;

- groupInstId.

Полученные данные помещаются в объекты коллекции ArrayList и могут быть выведены на экран приложения или обработаны в дальнейшем.

ArrayList — одна из самых используемых коллекций. ArrayList инкапсулирует в себе обычный массив, длина которого автоматически увеличивается при добавлении новых элементов. Так как ArrayList использует массив, то время доступа к элементу по индексу минимально (В отличие от LinkedList). При удалении произвольного элемента из списка, все элементы, находящиеся «правее» смещаются на одну ячейку влево, при этом реальный размер массива не изменяется. Если при добавлении элемента, оказывается, что массив полностью заполнен, будет создан новый массив размером  $(n * 3) / 2 + 1$ , в него будут помещены все элементы из старого массива + новый, добавляемый элемент.

Для того чтобы выводить данных нужно создать xml модель экрана активности в классе InstList, она должна включать в себя элемент TextView, который содержит в себе название экрана активности и компонент ListView для отображения списка данных коллекции. Xml код экрана активности должен начинаться с маленькой буквы, иметь префикс “activity” и полностью повторять имя класса, запускающим эту активность, но в другой нотации — все слова отделяются символом нижнее подчеркивание («\_») и начинаются с маленькой буквы. Файл экрана активности должен иметь расширение xml. Например, xml-файл класса InstList будет называться activity\_inst\_list.xml.

Реализованы 3 параметризованных контейнера:

1. LinearLayout – хранящий в себе остальные контейнеры и располагающий в себе все остальные элементы разметки в виде вертикального списка.
2. TextView – хранящий в себе название экрана активности.
3. ListView – являющийся контейнером для отображения обработанных данных в виде списка.

Для того чтобы реализовать привязку списка к контейнеру ListView, необходимо использовать класс адаптер. В нашем случае это класс создаётся объект класса ArrayAdapter, в параметрах которого указываются: разметка одной строки xml-модели списка и название коллекции, содержащий объекты класса. А затем необходимо назначить адаптер xml-контейнеру методом setAdapter.

Таким образом, в этом пункте была рассмотрена реализация нескольких основных функций разрабатываемого приложения, на основе которых будет строиться многосложная структура классов создаваемого ПО и дополняться дополнительным функционалом.

### **3.3 Разработка алгоритма классификации**

Алгоритм класса-классификатора состоит из следующих действий:

1. Создание корня дерева принятия решений.
2. Выбор атрибута классификации.
3. Разбиение элементов выборки по атрибуту «Приоритет».
4. Оценка качества разбиение (значение Gain).
5. Повторение процедуры до достижения максимального значения Gain.

На рисунке 9 представлен алгоритм классификации событий по атрибуту «Приоритет».

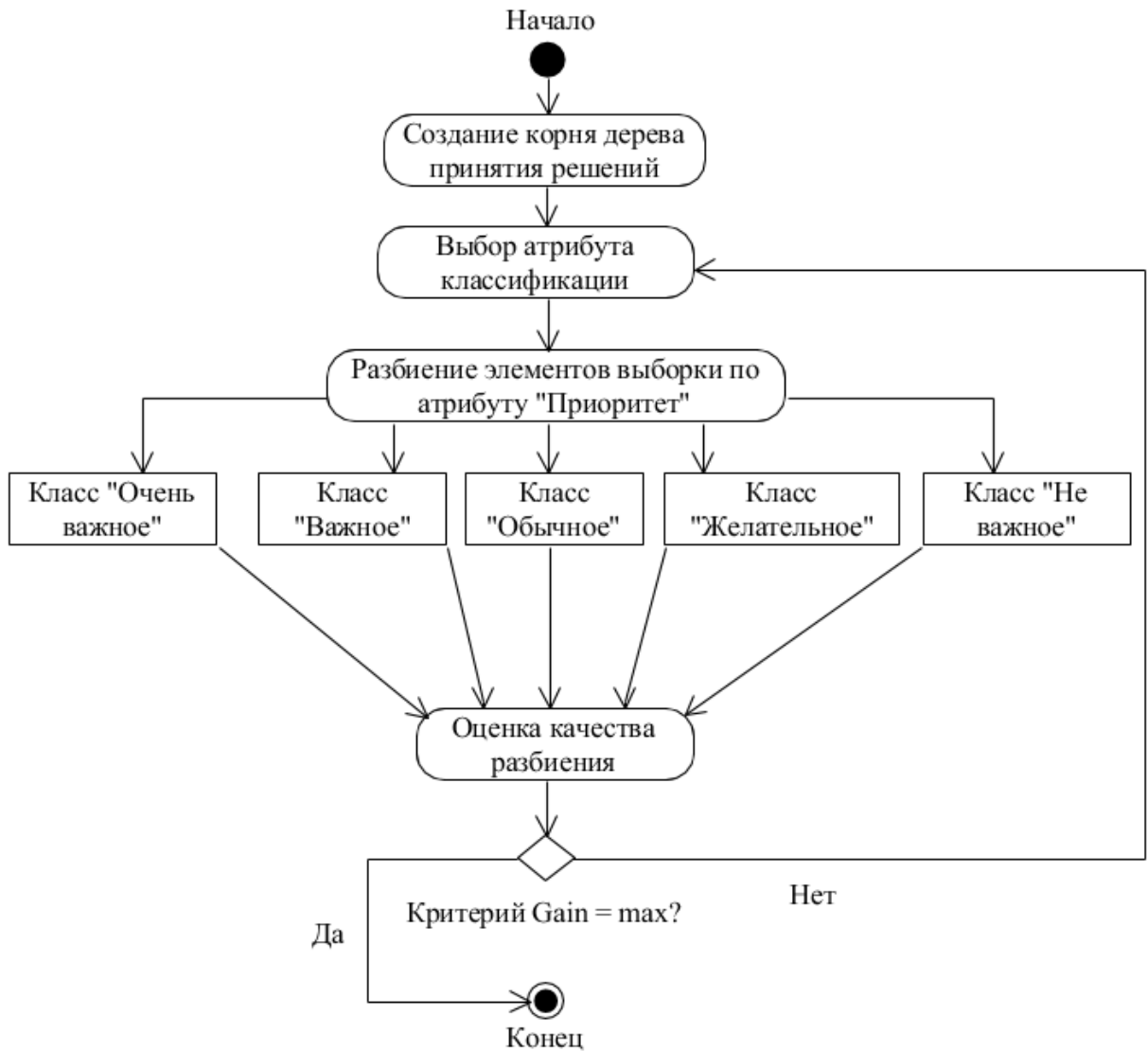
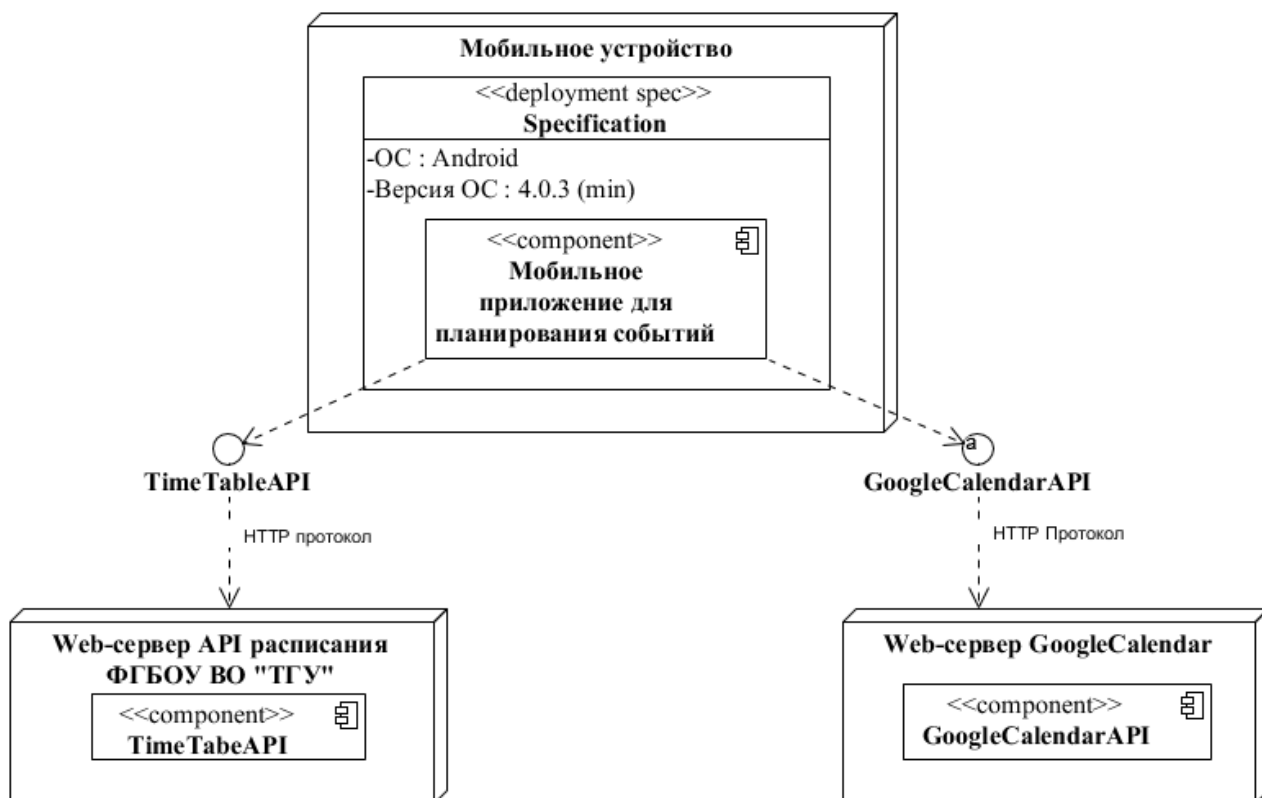


Рисунок 9 – Алгоритм метода класса C45Classifier

Таким образом, алгоритм построения дерева принятия решений адаптирован под решение задачи классификации событий по атрибуту «Приоритет», работающий до достижения максимального значения качества разбиения Gain.

### 3.3 Разработка диаграммы развертывания мобильного приложения

Для корректной работы приложения его необходимо установить на устройство с операционной системой Android OS минимум 4.0.3 версии, ввиду особенностей набора средств разработки SDK. Диаграмма развёртывания мобильного приложения для планирования событий представлена на рисунке



10.

Рисунок 10 – диаграмма развёртывания мобильного приложения для планирования событий

Из диаграммы видно, что приложение будет работоспособным на мобильном устройстве на операционной системе Android версией не менее 4.0.3 с наличием подключения к сети Интернет.

Инструкция по установке мобильного приложения для планирования событий:

1. Загрузить установочный файл StealthApp.apk на мобильное устройство по ссылке [m.tt.binarus.ru/stealthapp](http://m.tt.binarus.ru/stealthapp).
2. Запустить установочный файл.

3. Подтвердить разрешение на использование персональных данных GoogleCalendar.
4. Ожидать окончания установки.

Таким образом, были сформированы диаграмма развёртывания и инструкция по установке мобильного приложения для планирования событий, состоящая из 4 действий.

## **ЗАКЛЮЧЕНИЕ**

В ходе проведения работы, в подразделе 1.1, был проведён анализ предметной области на примере ФГБОУ ВО «ТГУ» с точки зрения разработки мобильного приложения для планирования событий.

В подразделе 1.2 был проведен анализ системы публикации расписания учебных занятий ФГБОУ ВО «ТГУ».

В подразделе 1.3 был проведен анализ существующих систем планирования времени.

Анализ данных разделов, выявил необходимость реализации мобильного приложения для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО «ТГУ», для более удобного планирования времени.

По системе классификации требований FURPS+, в разделе 1.4, были выработаны требования к разрабатываемому приложению для планирования событий с интеграцией расписания учебных занятий с модулем уведомления.

На основе требований к разрабатываемому алгоритму, в разделе 2.2, были выработаны 6 функциональных требований и 10 нефункциональных ограничений к технологии мобильного приложения для планирования событий.

На основе выработанных требований к технологии приложения для планирования событий и модулю уведомления, в подразделе 2.3, была разработана архитектура мобильного приложения планирования событий.

На основе разработанной архитектуры мобильного приложения для планирования событий, в подразделе 2.4, были разработаны взаимодействия между элементами приложения посредством диаграммы последовательности.

Таким образом, в подразделе 2.5, был разработан модуль создания списка событий в разрабатываемой системе.

На основании этого, в пункте 2.6, был разработан основной алгоритм модуля уведомления.

Исходя из этого, в подразделе 2.7, было описано обоснование математической модели алгоритма для классификации событий.

На основании этого, в пункте 2.8 были разработаны основные классы и методы разрабатываемого приложения для планирования событий с интеграцией учебных занятий ФГБОУ ВО «ТГУ».

В ходе выбора средств реализации мобильного приложения для планирования событий, в подразделе 3.1, был определен язык программирования и интегрированная среда разработки для реализации мобильного приложения для планирования событий.

В подразделе 3.3 приведена диаграмма развёртывания и инструкция по установке мобильного приложения для планирования событий ФГБОУ ВО «ТГУ».

В ходе работы было разработано мобильное приложения для планирования событий с интеграцией расписания учебных занятий ФГБОУ ВО «ТГУ».



## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

### *Научная и методическая литература*

1. Уокенбах Дж. Microsoft Excel 2010. Библия пользователя. : Пер. с англ. – М.: Вильямс, 2011. –912 с.
2. Вигерс К. Разработка требований к программному обеспечению // К. Вигерс, Д. Битти. -Спб:ВНУ,2014.-736с.
3. Харди Б., Филлипс Б., Стюарт К., Марсикано К. X20 Android. Программирование для профессионалов / 2-е изд. — СПб.: Питер, 2016. — 640 с.
4. А.А. Барсегян, И.И. Холод, М.Д.Тесс, М.С. Куприянов, С.И. Елизаров Анализ данных и процессов - СПб.: БХВ-Петербург, 2009
5. Фаулер М. Рефакторинг. Улучшение существующего кода // М. Фаулер. – СПб : Символ Плюс, 2015. – 415с.
6. Дейтел П., Дейтел Х., Уолд А. Д27 Android для разработчиков / 3-е изд. — СПб.: Питер, 2016. — 512 с
7. Хорстманн К. Java SE

### *Электронные ресурсы*

8. Тольяттинский государственный университет [Электронный ресурс] // Тольяттинский государственный университет: История Тольяттинского государственного университета. – URL: <http://www.tltsu.ru> (1.06.2016)
9. Yandex календарь [Электронный ресурс] – URL: <https://calendar.yandex.ru/> (дата обращения:20.10.2017)
10. Google календарь [Электронный ресурс] – URL: <https://www.google.com/calendar> (дата обращения:20.10.2017)
11. Windows Live календарь [Электронный ресурс] – URL: <https://office.live.com/start/Calendar> (дата обращения:20.10.2017)
- 12.Расширение ICS [Электронный ресурс] – URL: <http://fileext.ru/ics> (дата обращения: 20.10.2017)

13. Masoud Kalali, Developing RESTful Services with JAX-RS 2.0, WebSockets, and JSON / Masoud Kalali – United Kingdom, Birmingham,: Packt Publishing, LTD, 2013. – 107 p.
14. Требования к системе: классификация FURPS+ [ Электронный ресурс] – URL: <https://sysana.wordpress.com/2010/09/16/furps/> (дата обращения: 1.11.2017)
15. Закон РФ от 10.07.1992 N 3266-1 (ред. от 12.11.2012) "Об образовании" [Электронный ресурс] // СПС КонсультантПлюс: Законодательство: Версия Проф. – URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_1888](http://www.consultant.ru/document/cons_doc_LAW_1888) (1.06.2016)
16. IBM Knowledge Center / [Электронный ресурс] – URL: [https://www.ibm.com/support/knowledgecenter/ru/ssw\\_aix\\_61/com.ibm.aix.performance/advantages\\_java.htm](https://www.ibm.com/support/knowledgecenter/ru/ssw_aix_61/com.ibm.aix.performance/advantages_java.htm), Режим доступа: свободный.
17. IDEF0 - методология функционального моделирования / Itstan [Электронный ресурс]. – URL: <http://www.itstan.ru/funk-strukt-analiz/idef0.html/>, Режим доступа: свободный.
18. Tutorials Technology [Электронный ресурс]. – URL: <http://www.gwtproject.org/doc/latest/tutorial/index.html> (дата обращения: 08.03.2017).
19. Extreme Programming: A gentle introduction [Электронный ресурс]. – URL: <http://www.extremeprogramming.org/> (дата обращения: 28.05.2017).
20. Википедия – свободная энциклопедия / [Электронный ресурс] – URL: <https://ru.wikipedia.org/wiki/>, Режим доступа: свободный.

*Ресурсы на иностранном языке*

21. Baesens, B. Beginning Java Programming: The Object-Oriented Approach / B. Baesens, A. Backiel, S. Vanden Broucke. – 1st edition, Wrox, 2015.
22. Bill Burke, RESTful Java with JAX-RS 2.0, Second Edition / Bill Burke - O'Reilly Media, Inc. – 392 p. 1

23. Deitel, H. Java How to Program / H. Deitel, P. Deitel. – 9th edition, Prentice Hall, 2015.
24. Hudson O. Getting started with IntelliJ IDEA // O. Hudson, Birmingham: Packt Publishing, 2013. – 114p.
25. Krochmalski J. IntelliJ IDEA Essentials // J. Krochmalski. – Birmingham: Packt Publishing, 2014.-263p.

## **ПРИЛОЖЕНИЕ А**

**Листинг кода реализации классов**

Прикреплен на диске