ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

|  |
| --- |
| Институт компьютерных наук и технологий  Высшая школа программной инженерии |
|  |

**ИНДИВИДУАЛЬНЫЙ ПЛАН (ЗАДАНИЕ И ГРАФИК)**

**ПРОВЕДЕНИЯ ПРАКТИКИ**

|  |
| --- |
| Шмерлинг Максим Александрович |

|  |
| --- |
| Направление подготовки (код/наименование) 09.03.04 «Программная инженерия» |
| Профиль (код/наименование) 09.03.04\_01 «Технология разработки и сопровождения качественного программного продукта» |
| Вид практики: научно-исследовательская работа |
| Тип практики: распределенная |
| Место прохождения практики ФГАОУ ВО «СПбПУ», ИКНТ, ВШПИ, СПб, ул.Политехническая, 29 |

|  |
| --- |
| Руководитель практической подготовки от ФГАОУ ВО «СПбПУ»: |

|  |  |
| --- | --- |
| Котлярова Лина Павловна, старший преподаватель ВШПИ ИКНТ | Петров Александр Владимирович, старший преподаватель ВШПИ ИКНТ |

*(Ф.И.О., уч.степень, должность)*

|  |
| --- |
| Руководитель практической подготовки от профильной организации: - |

**Рабочий график проведения практики**

Сроки практики: с02.02.23 по 14.04.23

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Этапы (периоды) практики** | **Вид работ** | **Сроки прохождения этапа (периода) практики** |
| 1 | Организационный этап | Установочная лекция для разъяснения целей, задач, содержания и порядка прохождения практики, инструктаж по технике безопасности, выдача сопроводительных документов по практике | 02.02 |
| 2 | Основной  этап | Разработка мобильного приложения для ведения студентом списка дел на базе расписания университета | 03.02-13.04 |
| 3 | Заключительный этап | Защита отчета по практике | 14.04 |



Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Шмерлинг М. А. /

Руководитель практической подготовки

от ФГАОУ ВО «СПбПУ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Котлярова Л. П. /

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»**

|  |
| --- |
| Институт компьютерных наук и технологий  Высшая школа программной инженерии |
|  |

**Отчет о прохождении научно-исследовательской работы**

|  |
| --- |
| Шмерлинг Максим Александрович |

|  |
| --- |
| 4 курс, 3530904/90102 |

|  |
| --- |
| 09.03.04 «Программная инженерия» |

|  |
| --- |
| **Место прохождения практики:** ФГАОУ ВО «СПбПУ», ИКНТ, ВШПИ, |

|  |
| --- |
| СПб, ул.Политехническая, 29 |

|  |
| --- |
| **Сроки практики:** 02.02.23-14.04.23 |

|  |
| --- |
| **Руководитель практической подготовки от ФГАОУ ВО «СПбПУ»:** |

|  |  |
| --- | --- |
| Котлярова Лина Павловна, старший преподаватель ВШПИ ИКНТ | Петров Александр Владимирович, старший преподаватель ВШПИ ИКНТ |

|  |
| --- |
| **Руководитель практической подготовки от профильной организации:** - |

|  |
| --- |
|  |

|  |
| --- |
| **Оценка:** зачтено |

|  |
| --- |
| Руководитель практической подготовки  от ФГАОУ ВО «СПбПУ»: / Котлярова Л. П. / |

|  |
| --- |
| Руководитель практической подготовки  от профильной организации: - |



|  |
| --- |
| Обучающийся: / Шмерлинг М. А. / |

|  |
| --- |
| Дата: 14.04.23 |

Оглавление

[Постановка задачи 4](#_Toc131790905)

[Обоснование актуальности работы 5](#_Toc131790906)

[Обзор существующих решений 7](#_Toc131790907)

[Обоснование выбора технологий и средств разработки 13](#_Toc131790908)

[Описание планируемой архитектуры 15](#_Toc131790909)

[Список литературы 18](#_Toc131790910)

# **Постановка задачи**

В рамках ВКР планируется реализовать мобильное приложение для ведения студентом списка дел на базе расписания Санкт-Петербургского политехнического университета. В приложении студент, выбрав свою академическую группу, сможет просмотреть расписание занятий, а самое главное начать вести список дел. Каждую задачу можно будет связать с соответствующей ей учебной дисциплиной.

Программа будет реализована с помощью фреймворка, а информация о расписании получена с помощью ресурса ruz.spbstu.ru, который имеет открытое API [1].

Программный продукт планируется разработать для мобильной операционной системы Android [2]. Для тестирования будет использоваться эмулятор, предоставляемый Google.

Для решения поставленной задачи необходимо выполнить следующее:

* изучить существующие решения;
* обосновать и выбрать технологии и инструменты разработки;
* спроектировать, разработать и проверить мобильное приложение.

# **Обоснование актуальности работы**

Тема "Разработка мобильного приложения для ведения студентом списка дел на базе расписания университета" является актуальной в настоящее время, поскольку современный учебный процесс в университетах требует от студентов высокой организованности и управления временем. Кроме того, студенты могут столкнуться с трудностями в планировании своего учебного графика и выполнении заданий в свободное время.

Мобильное приложение поможет студентам лучше организовать свое время и улучшить производительность. Предоставит студентам возможность создания списка задач, которые они должны выполнить в течение учебы, а также интегрировать расписание занятий в список дел. Оно поможет учащимся быть более ответственными и дисциплинированными в учебном процессе, что может положительно сказаться на их успеваемости. Приложение даст студентам возможность получать напоминания о дедлайнах и важных событиях, что снизит риск пропуска важных занятий и дел.

Планирование учебы очень важно для каждого студента, оно несет в себе практическую пользу:

* вы ничего не забываете: записывая и структурируя план задач, вы сохраняете самую необходимую информацию;
* планируя свой день, вы учитесь грамотно распределять время на выполнение заданий;
* вы используете активный, а не реактивный способ действия: когда вы вовремя берётесь за выполнение задачи, у вас появляется возможность сконцентрироваться, сгруппировать похожие задания и быстро с ними справиться;
* планирование учит правильно оценивать задания и результаты их выполнения;
* если вас отвлекли во время выполнения заданий, вы всегда знаете, на какой задаче вы остановились.

В целом разработка мобильных приложений стала очень актуальной в наше время по нескольким причинам:

* рост использования мобильных устройств. С каждым годом все больше людей используют мобильные устройства для своих повседневных задач. Согласно исследованиям, на данный момент количество смартфонов в мире превышает 3 миллиарда, а к 2025 году это число может достигнуть 5,5 миллиарда. Это означает, что мобильные устройства стали неотъемлемой частью нашей жизни, и многие люди проводят в них большую часть своего времени;
* быстрое распространение мобильного интернета. С развитием мобильных устройств распространение мобильного интернета ускорило свои темпы. Сегодня практически везде есть доступ к интернету, даже в отдаленных уголках мира. Это означает, что люди могут использовать приложения в любом месте и в любое время;
* удобство использования. Мобильные устройства очень удобны в использовании, они легко помещаются в кармане или сумке, и всегда под рукой. Приложения, разработанные специально для мобильных устройств, имеют интуитивно понятный интерфейс и адаптированы к размерам экрана. Это делает их удобными для использования, что привлекает пользователей.

# **Обзор существующих решений**

Для введения списка дел на базе расписания Санкт-Петербургского политехнического университета студенты могут воспользоваться следующими мобильными приложениями:

* Расписание СПбПУ;
* Google Keep [3];
* Todoist [4];
* Trello [5];

Рассмотрим пять основных показателей, выбранных для сравнения существующих решений: доступность, интеграция, уведомления, хранимые типы данных и планирование.

Первый показатель – доступность, является одним из самых важных. Он позволяет оценить, насколько приложение доступно для использования и какие возможности можно получить бесплатно. Когда разрабатывается приложение, важно учитывать финансовые возможности пользователей, не все готовы платить за приложения.

Второй показатель – интеграция, позволяет оценить на сколько можно расширить функциональность приложения за счет возможности интеграции с другими сервисами и приложениями. Кроме того, эта возможность может быть полезна для сохранения и обмена данными между различными приложениями.

Третий показатель – уведомления, позволяет оценить на сколько гибко можно будет настроить уведомления о предстоящих занятиях, чтобы не пропустить важную встречу или задачу. Важно, чтобы приложение позволяло настроить уведомления на определенное время или за несколько минут до начала занятия, чтобы пользователь мог подготовиться к встрече или задаче.

Четвертый показатель – хранимые типы данных, определяет, какие типы данных могут быть сохранены в приложении. Хорошее приложение по ведению списка дел должно иметь возможность сохранять различные типы данных помимо текста, так как студенты часто имеют дело с файлами разного формата.

Пятый показатель – планирование, определяет на сколько пользователю удобно планировать свои задачи во времени. Присутствует ли календарь в приложении, где отмечены задачи. Или есть возможность посмотреть на временной график с соответствующей информацией о запланированных делах.

Расписание СПбПУ — это приложение, разработанное для студентов и преподавателей Санкт-Петербургского политехнического университета. Оно позволяет пользователям просматривать расписание занятий, экзаменов и других событий. Приложение доступно бесплатно для загрузки из Google Play Store.

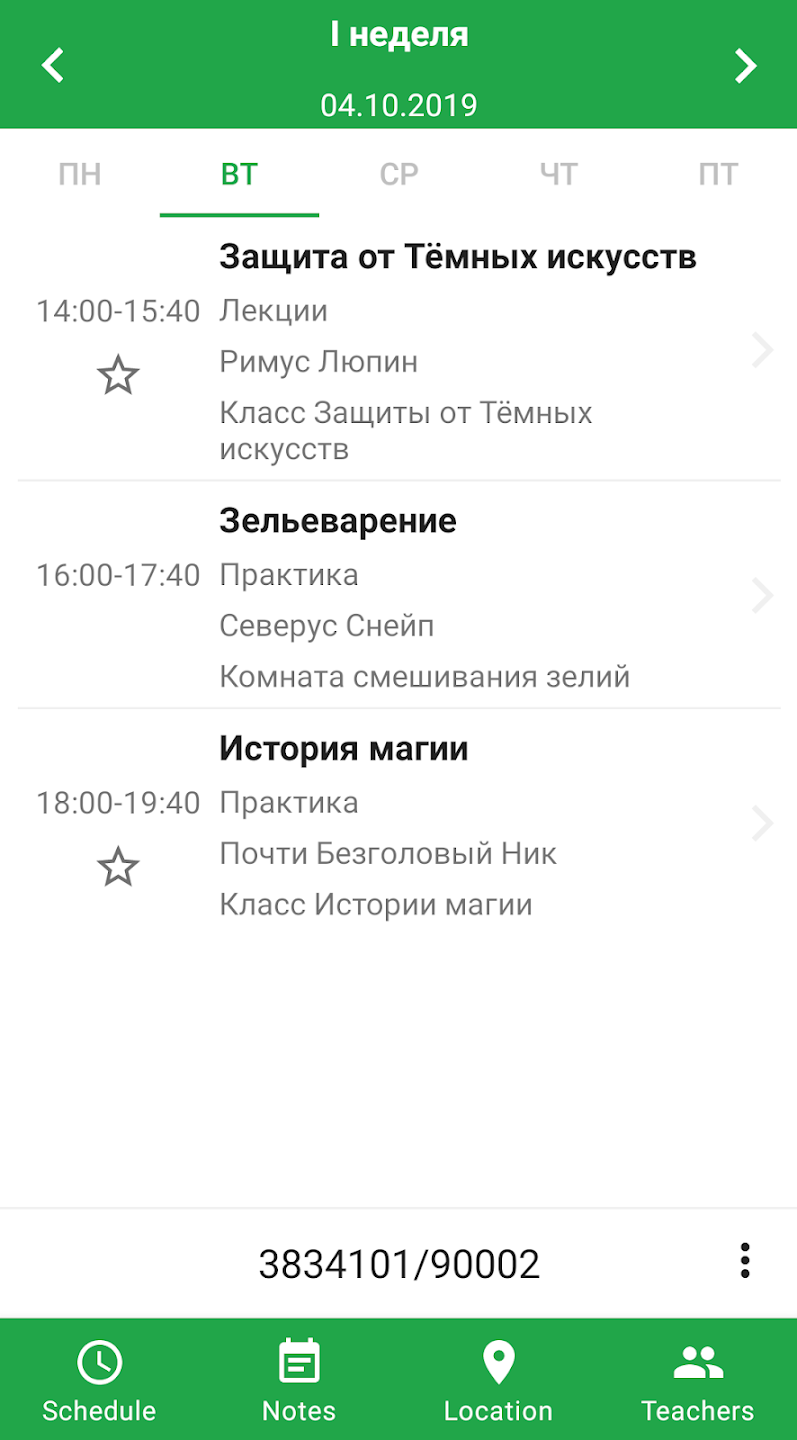


Рис. 1. Расписание СПбПУ

Оценка приложения по выбранным показателям:

+ доступность: доступно бесплатно для загрузки из Google Play Store;

- интеграция: отсутствует;

- уведомления: отсутствует;

-/+ хранимые типы данных: позволяет сохранять только текстовые данные;

-/+ планирование: можем просматривать расписание только по дням. Задачу можно связать с дисциплиной, но не с датой.

В результате имеем приложение, в котором можно просматривать расписание занятий и оставлять заметки, но вести полноценное планирование задач нельзя, для этого не хватает функционала. Главным преимуществом приложения является постоянное получение актуальной информации о расписание, благодаря открытому API ресурса ruz.spbstu.ru.

Google Keep — это бесплатное приложение для заметок и задач, которое можно использовать для планирования дел по учебе. Оно доступно на платформах Android, iOS и веб-браузерах, и позволяет создавать и организовывать заметки и задачи, прикреплять к ним файлы.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 2. Google Keep

Оценка приложения по выбранным показателям:

+ доступность: бесплатно на различных платформах;

+ интеграция: с другими сервисами Google;

+ уведомления: позволяет настраивать;

-/+ хранимые типы данных: позволяет сохранять текст, изображения, видео, аудио файлы;

- планирование: отсутствует.

Приложение удовлетворяет большинству важных для нас требований. Но все-таки оно не будет содержать актуального расписания для студентов, каждому пользователю придется самостоятельно обновлять списки с дисциплинами. Студент в Google Keep не сможет спланировать свое время, так как отсутствует визуальное представление задач и дисциплин в виде календаря или графика. Также к задаче мы не сможем приложить файл, а большинство учебных материалов распространяется в pdf или docx форматах.

Todoist — это приложение для управления задачами, которое помогает организовывать работу и повышать производительность. Оно доступно на различных платформах, включая веб-версию, мобильные приложения для iOS и Android, а также настольные приложения для Windows, MacOS и Linux.

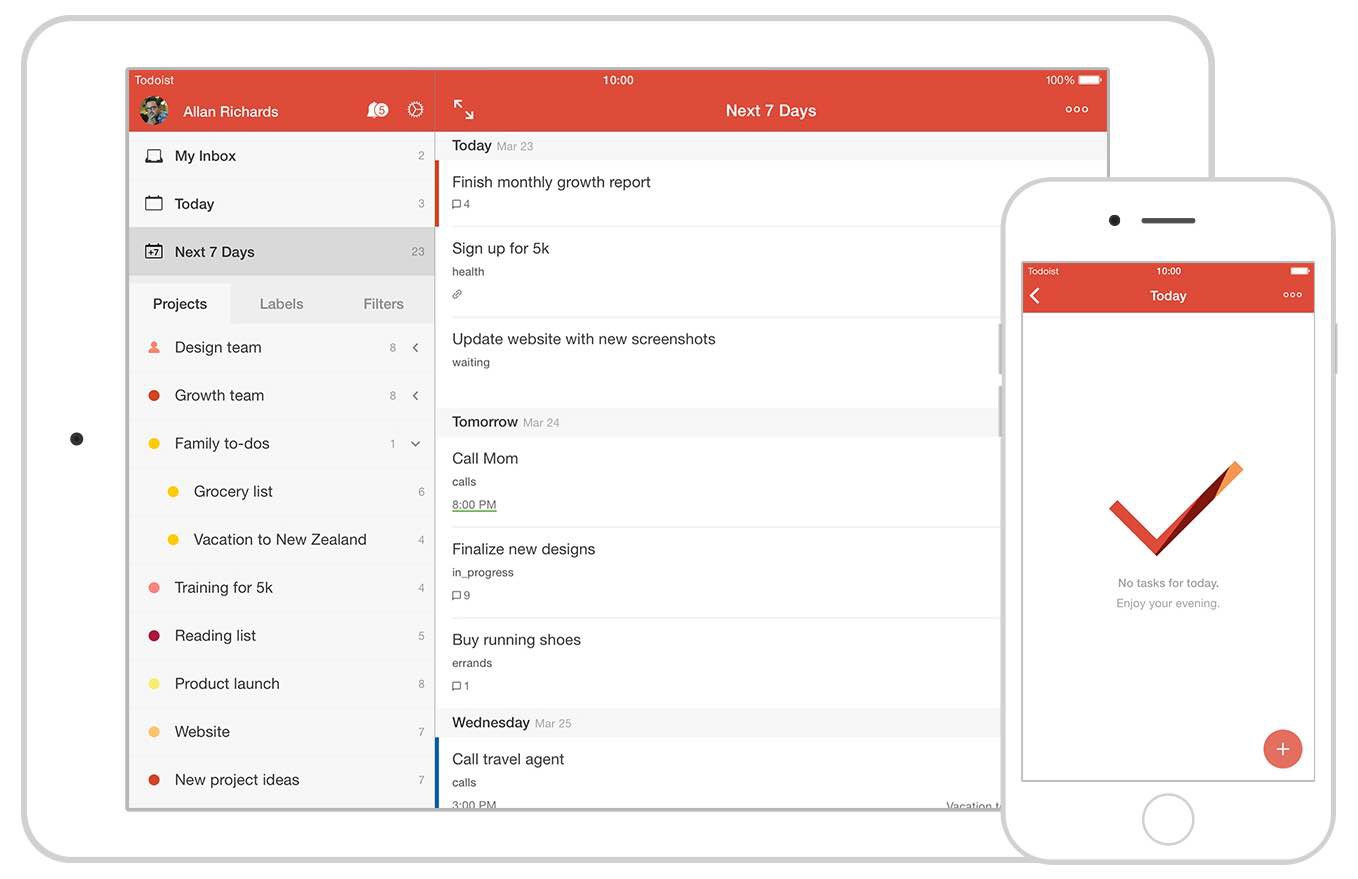


Рис. 3. Todoist

Оценка приложения по выбранным показателям:

+/- доступность: приложение доступно для использования бесплатно, но с ограничениями. Пользователи могут использовать основные функции и имеют ограниченное количество проектов и задач. Однако, доступна подписка на Todoist Premium, которая предлагает больше возможностей и интеграций, а также безлимитное количество проектов и задач;

+ интеграция: интегрируется с множеством сервисов и приложений;

+ уведомления: позволяет гибко настроить уведомления о задачах и событиях в расписании;

+ хранимые типы данных: позволяет прикреплять файлы, изображения и видео к задачам или событиям в расписании;

- планирование: отсутствует.

Главной отличительной особенностью приложение является функционал по отслеживанию продуктивности. Но не хватает важной для нас возможности – планирования. Также студенты придется вручную заносить расписание занятий в приложение и поддерживать его актуальность.

Trello — это онлайн-инструмент для управления проектами и задачами. Он представляет собой виртуальную доску, которая позволяет пользователям создавать списки задач, карточки и метки для управления проектами и задачами. Trello использует методологию Канбан, позволяющую пользователю просматривать статус проекта в режиме реального времени, перетаскивая задачи между списками, добавляя комментарии и прикрепляя файлы. Инструмент доступен в веб-версии, а также имеет приложения для iOS и Android, что позволяет использовать его на разных устройствах и в разных местах. Trello имеет бесплатную версию, а также платную подписку с дополнительными функциями и возможностями.



Рис. 4. Trello

Оценка приложения по выбранным показателям:

+/- доступность: приложение доступно для использования бесплатно, с возможностью использования большинства функций. Однако, существуют платные версии Trello Business Class и Trello Enterprise, которые предлагают дополнительные функции;

+ интеграция: интегрируется с множеством сервисов и приложений, таких как Google Drive, Slack, Dropbox;

+ уведомления: позволяет настроить уведомления на различных уровнях, таких как уведомления о новых комментариях, о перемещении задач в другие колонки или доски, о завершении задачи и многие другие. Пользователи могут настроить уведомления на свой вкус и получать их через электронную почту, настольные приложения или мобильные устройства;

+ хранимые типы данных: позволяет прикреплять файлы, изображения и видео к задачам, карточкам и комментариям;

-/+ планирование: режим просмотра рабочего пространства в виде календаря доступен только с подпиской.

Trello — это отличное приложение для управления задачами и проектами. Оно предоставляет пользователю гибкую и интуитивно понятную систему организации и управления задачами в виде досок, списков и карточек. Однако, если вам нужно просто вести список дел на основе расписания университета, то Trello может показаться избыточным в использовании. Кроме того, планирование в приложение доступно только с платной подпиской, что может не подходить для студентов с ограниченным бюджетом.

Мое решение будет поддерживать хранение любого формата данных. Я

считаю, что это важный функционал, потому что студенту часто приходится

сохранять файлы специфичного формата. Так как приложение будет

включать в себя расписание занятий, то задачу можно будет не

только запланировать на конкретную дату и время, но и связать с учебной

дисциплиной. Студент сможет просмотреть обзорное представление его задач на день, неделю или месяц, что поможет в планирование своего времени. Также приложение будет интегрированно с расписанием университета, поэтому информация о занятиях будет обновляться автоматически. И, конечно, если мое творение увидит свет, оно будет совершенно бесплатным.

# **Обоснование выбора технологий и средств разработки**

Для реализации мобильного приложения будет использоваться фреймворк Flutter [6]. Это комплект средств разработки и фреймворк с открытым исходным кодом для создания мобильных приложений под Android и iOS, веб-приложений, а также настольных приложений под Windows, macOS и Linux с использованием языка программирования Dart [7], разработанный и развиваемый корпорацией Google.

Данная технология была выбрана, поскольку обладает следующими преимуществами:

* скорость разработки: функция «горячей перезагрузки» позволяет мгновенно отслеживать последствия изменения кода на аппаратных платформах, в симуляторах и эмуляторах;
* высокая производительность: программная часть приложений создается с использованием машинного кода и не нуждается в каком-либо промежуточном представлении или интерпретации;
* использование виджетов: поставляется с обширным набором готовых виджетов, которые обеспечивают наличие разнообразных функций в создаваемых приложениях. Разработчик может создать из виджетов пользовательский интерфейс, который будет одинаково корректно отображаться на различных платформах, при различном разрешении и размере экрана. Комбинированием и вложением виджетов разработчик может создавать многофункциональные приложения;
* высокая масштабируемость: возможность использования одного и того же программного кода для всех необходимых платформ обеспечивает высокую продуктивность с точки зрения разработки и технической поддержки создаваемых решений;
* мощное сообщество: развитие и совершенствование фреймворка обеспечивает сообщество опытных разработчиков.

Android Studio [8] - это среда разработки для создания мобильных приложений под операционную систему Android. Я выбрал Android Studio для реализации мобильного приложения в рамках выпускной квалификационной работы, так как она является официальной IDE для разработки приложений под Android, и имеет все необходимые инструменты и функции для работы с фреймворком Flutter. Ее большое сообщество разработчиков и постоянно обновляется, что обеспечивает поддержку новых технологий и инструментов. Кроме того, Android Studio имеет множество плагинов и расширений, которые улучшают ее функциональность и упрощают процесс разработки.

В качестве системы управления версиями мой выбор пал на технологию Git [9], а именно GitHub [10] — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Его преимущество перед конкурентами заключается в следующем:

* бесплатное обслуживание;
* очень быстрый поиск в структуре репозиториев;
* большое сообщество и легко найти помощь;
* он предлагает практические инструменты для сотрудничества и хорошую интеграцию с Git;
* легко интегрируется с другими сторонними сервисами.

# **Описание планируемой архитектуры**

В выпускной квалификационной работе архитектура моего приложения будет соответствовать концепции Clean Architecture (чистая архитектура) [11]. Чистая архитектура — это способ организации кода, который способствует строгому разделению ответственности между интерфейсом, логикой и данными. Приложение разбивается на независимые функциональные компоненты, которые взаимодействуют друг с другом определённым способом, при этом между ними передаются только те ресурсы, которые необходимы для выполнения поставленной задачи. В рамках данной архитектуры обычно выделяют три основных слоя:

* presentation: слой представления содержит все компоненты, которые связаны с UI, и помогает организовать взаимодействие со слоем бизнес-логики и работу с данными;
* domain: слой бизнес-логики, в нем содержится, вся логика работы приложения. Именно к этому слою обращается слой представления для выполнения запросов и получения данных;
* data: в данном слое содержится всё, что связано с хранением данных и управлением ими.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рис. 4. Архитектура приложения

Расшифровка обозначений на схеме:

* internet resource – «интернет ресурс», обозначает источник информации о расписание;
* request – «запрос», показывает, что клиентская часть будет отправлять запросы на сервер;
* response – «ответ», показывает, что сервер будет отвечать на запросы клиента;
* mobile app – «мобильное приложение», обозначает клиентскую часть, в данном случае разрабатываемое мной приложение на Flutter;
* presentation – «презентация», обозначает слой представления;
* screen – «экран», обозначает экран в приложение;
* widgets – «виджет», понятие из Flutter, которое обозначает визуальные компоненты, из которых состоит графический интерфейс;
* utils – «утилиты», обозначает вспомогательные функции необходимые для построения интерфейса;
* controller.method – «контроллер.метод», показывает, что слой представления будет вызывать методы контроллера для изменения состояния приложения;
* new state – «новое состояние», показывает, что слой бизнес-логики будет возвращать новое состояние в ответ на вызов метода контроллера;
* domain – «домен», обозначает слой бизнес-логики;
* models – «модели», обозначает структуры данных, которыми будет оперировать приложение;
* screen controller – «контроллер экрана», обозначает контроллер конкретного экрана содержащий бизнес-логику;
* data – «данные», в одном случае показывает, что слои бизнес-логики и данных будут обмениваться информацией. А в другом обозначает непосредственно сам слой данных;
* api – аббревиатура, расшифровывается, как Application Programming Interface («программный интерфейс приложения»). Обозначает сущность, с помощью которой мобильное приложение будет взаимодействовать с сервером;
* data source – «ресурс данных», обозначает набор сущностей, необходимых для работы с данными;
* data manager – «менеджер данных», является сущностью объединяющей в себе работу с локальными и удаленными данными;
* remote data source – «удаленный ресурс данных», обозначает сущность для работы с удаленными данными;
* local data source – «локальный ресурс данных», обозначает сущность для работы с локальными данными;
* services – «сервисы», обозначает набор сервисов, которые предоставляют определенную функциональность для работы с данными;
* firebase – сущность для взаимодействия с сервисом Firebase от Google, с помощью которого будет собираться аналитика о работе приложения;
* shared preferences – сущность для взаимодействия с данными в Shared Preferences (постоянное хранилище устройства).

Для управления состоянием приложения я буду использовать Riverpod — это библиотека, созданная для Flutter и основанная на концепции provider. С помощью нее можно предоставлять данные или функциональность в виде зависимостей для других объектов.

Для хранения данных я выбрал Shared preferences — это удобный и простой в использовании механизм для хранения данных в виде пар ключ-значение на устройстве пользователя. Поскольку информация о расписание будет поступать в формате JSON, то такой механизм идеально подходит.

Также я буду использовать Firebase для сбора аналитики о работе приложения, чтобы мониторить и анализировать ошибки, которые происходят, и определять их причины.

Для сетевого взаимодействия мой выбор пал на Dio — это пакет для работы с сетевыми запросами. Он предоставляет множество опций для настройки запросов, таких как установка таймаутов, установка заголовков запросов, отправка запросов с параметрами, обработка ошибок и многие другие. Dio также поддерживает множество типов ответов, включая JSON, XML и бинарные данные. Поэтому с помощью него из мобильного приложения будет выполняться HTTPS-запрос [12] к интернет ресурсу https://ruz.spbstu.ru/, который является официальным источником информации о расписании занятий Санкт-Петербургского политехнического университета и имеет открытое API. Форматы отправляемых запросов:

* GET: https://ruz.spbstu.ru/api/v1/ruz/faculties - возвращает список факультетов в формате JSON;
* GET: https://ruz.spbstu.ru/api/v1/ruz/faculties/{ faculty\_id}/groups – возвращает список групп факультета в формате JSON;
* GET: https://ruz.spbstu.ru/api/v1/ruz/scheduler/{group\_id} – возвращает расписание группы на текущую неделю в формате JSON;
* GET: https://ruz.spbstu.ru/api/v1/ruz/scheduler/{group\_id}?date={date} – возвращает расписание группы за неделю, в которую попала дата, в формате JSON;

# **Список литературы**

1. Документация по API - Политех. [Электронный ресурс]. – URL: <https://conf.spbstu.ru/rus/doc/current/api/index.html> (дата обращения 01.04.2022).
2. Android. [Электронный ресурс]. – URL: <https://developer.android.com/guide> (дата обращения 01.04.2023).
3. Google Keep. [Электронный ресурс]. – URL: <https://www.google.com/keep/> (дата обращения 01.04.2022).
4. Todoist. [Электронный ресурс]. – URL: <https://todoist.com/ru> (дата обращения 01.04.2022).
5. Trello. [Электронный ресурс]. – URL: [https://trello.com/](https://trello.com/%20) (дата обращения 01.04.2022).
6. Flutter Documentation. [Электронный ресурс]. – URL: <https://docs.flutter.dev/> (дата обращения 01.04.2022).
7. Dart documentation. [Электронный ресурс]. – URL: <https://dart.dev/guides> (дата обращения 01.04.2022).
8. Android Studio. [Электронный ресурс]. – URL: <https://developer.android.com/studio> (дата обращения 01.04.2022).
9. Git-Book. [Электронный ресурс]. – URL: <https://git-scm.com/book/ru/v2> (дата обращения 01.04.2022).
10. GitHub. [Электронный ресурс]. – URL: <https://docs.github.com/ru> (дата обращения 01.04.2022).
11. Робер С. Мартин. Чистая архитектура: искусство разработки программного обеспечения. Издательство «Питер». Санкт-Петербург, 2018 г.
12. HTTP. [Электронный ресурс]. – URL: <https://httpwg.org/specs/> (дата обращения 01.04.2022).