|  |
| --- |
| УДК |
| ГУРКОВ АЛЕКСЕЙ СЕРГЕЕВИЧ |
| Разработка мобильного приложения «Расписание занятий» |
|  |
| Квалификационная работа на степень «БАКАЛАВР» по направлению 09.03.04 Южный федеральный университет, ИКТИБ, кафедра МОП ЭВМ – 2017 г., с. |

**АННОТАЦИЯ**

Работа посвящена разработке мобильного приложения для получения информации о расписание занятий университета. Целью данной работы является изучение платформы Android, её среды разработки, проектирование архитектуры мобильного приложения, создание мобильного приложения и его тестирование, для быстрого и удобного доступа к информации о расписании занятий.

Актуальность работы заключается в том, что на данный момент мобильные приложения, предоставляющие доступ к расписанию занятий, не имеют доступа к Api сервера с расписанием занятий ИТА ЮФУ, в результате не могут предоставить запрашиваемую информацию. Таким образом разработанное приложение акцентирует своё внимание на работе не с сервером, а внутри приложения с сохраненными копиями ранее полученных данных, для контроля актуальности данных, приложение с определенным интервалом сверяет версии данных находящихся на сервер.

Особое внимание было уделено архитектуре Model-View-Presenter, языку программирования Java, паттерну Adapter, механизму взаимодействия с сервером Rest и системе контроля версий Git.

ВВЕДЕНИЕ

В настоящее время мобильные технологии прочно вошли в нашу повседневную жизнь. Мы используем смартфоны и планшеты как для развлечений, так и для решения повседневных задач. Общение, развлечение, досуг и работа – буквально все можно делать с помощью предустановленных программ на смартфоне. И не удивительно, доступные интернет, легкость в передачи информации и файлов, дешевизна – все это делает программы удобным и комфортным инструментом.

Разработанный программный продукт, предоставляет пользователю быстрый доступ к информации о расписание занятий учебного заведения.

Актуальностью темы является то, что расписание занятий — это неотъемлемая часть обучения студента в учебном заведении. Расписание занятий время от времени подвергается изменения, предметы переносят на другое время, в другие аудитории или того хуже отменяются или добавляются. Поэтому вовремя полученные актуальные данные о расписании занятий, упрощают жизнь студента. Получение расписания занятий в печатном виде, это уже вчерашний день, поэтому на данный момент актуальной является разработка решений, предоставляющих мгновенный доступ к данным о расписание занятий, с удобной навигацией и ярким дизайном таких как мобильные и веб приложения.

Целью разработки приложения является, предоставление пользователю данных о расписании занятий, кроме того, что на весь семестр, так и в отсортированном удобном для чтения виде, на сегодняшний и завтрашний. С функциями поиска расписания проведения занятий определенным преподавателем, выводом расписания занятий для требуемой пользователем даты, уведомлением пользователя о времени, оставшемся до начала следующей пары и информированием о номере текущей на данный момент недели. Организовать доступность данных как при имеющемся подключении к сети интернет, так и использование ранее полученных данных, хранящихся в памяти приложения.

Для достижения поставленной цели были решены следующие задачи:

1. Анализ аналогов, предоставляющих доступ к информации о расписании занятий университета;
2. Изучение стиля архитектуры взаимодействия с сервером, Rest;
3. Проектирование интерфейса приложения используя концепцию, Material Design;
4. Проектирование архитектуры мобильного приложение с использованием архитектуры Model-View-Presenter [1];
5. Тестирование мобильного приложения;
6. Создание технико-экономического обоснования разработки проекта.

Для реализации приложения, была использована среда разработки Android Studio версии 2.3.2, условно-бесплатный эмулятор операционной системы Android – Genymotion, язык программирования Java, язык разметки xml, для разметки внешнего вида элементов, для обмена данными использовался формат JSON, текстовый редактор Notepad++.

Новизна разработки заключается в том, что существующие на данный момент аналоги имеют более ограниченный функционал упраздняя такие функции, как вывод уже отсортированного расписания, а именно для сегодняшнего дня, завтрашнего дня, для удобства чтения его пользователем, так же многие пренебрегают расчетом времени до начала пары и отображением номера учебной недели.

1. Анализ постановки задания

Целью дипломного проекта является задача, разработать мобильное приложение для получения данных о расписании занятий университета.

Приложение должно удовлетворять следующим требованиям:

1. Иметь интерфейс взаимосвязи с клиентом;
2. Формировать данные для отправки запроса к Api сервера;
3. Получать данные от сервера;
4. Обрабатывать полученные данные;
5. Сохранять локальную копию полученных данных (расписания);
6. Выводить отсортированное расписание для всего семестра;
7. Для данной даты обучения;
8. Для следующего дня занятий;
9. Должно иметь, поиск расписания занятий преподавателя по ФИО;
10. Искать расписание занятий для требуемой даты;
11. Отображать номер и тип текущей недели;
12. Расчет времени до начала следующей пары.

В результате анализа задачи, были разработаны такие модули как:

1. Модуль взаимодействия с сервером;
2. Модуль вывода формы с результатом;
3. Модуль вывода диалога;
4. Модуль формирования запроса для получения расписания занятий на весь семестр;
5. Модуль вывода основной информации на экран приложения;
6. Модуль отображения и взаимодействия с Navigation View;
7. Модуль формирования запроса для нужной даты;
8. Модуль формирования запроса для нужного преподавателя;
9. Модуль отображения номера недели;
10. Модуль расчета времени до начала пары;
11. Модуль сохранения полученных от сервера данных;
12. Модуль проверки актуальности данных в мобильном приложении.

Модуль взаимосвязи с сервером предназначен для формирования и отправки запроса на сервер с последующим ожиданием ответа.

Модуль вывода формы с результатом предназначен для вывода полученных данных сервера, либо обработанных приложением.

Модуль вывода диалога предназначен для отображения диалога, с выводом информации, после взаимодействия пользователя с определенными функциями приложения.

Модуль формирования запроса для получения расписания занятий на весь семестр, выполняет функцию формирования запроса с указанием полученных от пользователя данных, таких как номер группы, номер семестра.

Модуль вывода основной информации на экран приложения, под основные информации понимает вывод на экран расписания занятий на весь семестр, на сегодняшний день и на следующий день обучения.

Модуль отображения и взаимодействия с Navigation View, предназначен для быстрого взаимодействия с функциями, представленными в его меню, такими как: изменить группу, запросить расписание для нужной даты, запросить расписание интересующего преподавателя, вывести номер текущей недели, вывести время до начала следующей пары.

Модуль формирования запроса для нужной даты, предназначен для создания корректного запроса для сервера с указанием в виде параметра требуемой даты отображения информации.

Модуль формирования запроса для нужного преподавателя предназначен для создания корректного запроса для сервера с указанием в виде параметра ФИО преподавателя, для которого прислать расписание.

Модуль отображения номера недели предназначен для вывода на экран приложения номера текущей недели.

Модуль расчета времени до начала пары предназначен для расчета времени начала следующей пары.

Модуль сохранения полученных от сервера данных, должен обеспечивать сохранность данных в приложении на длительный срок.

Модуль проверки актуальности данных в мобильном приложении, предназначен для проверки актуальности хранимых в приложении данных о расписании, с целью обновлять их только при необходимости, без лишней траты мобильного трафика.

* 1. Обзор существующих аналогов

В качестве аналога были рассмотрены бесплатно распространяемые приложения из сервиса google play - Расписание ИТА ЮФУ, Расписание ИТА ЮФУ Ogetto Web, Расписание Радика (ИТА ЮФУ) [2].

Данные приложения обладают схожим функционалом, но на данный момент не предоставляют пользователю запрашиваемые данные, в связи с отсутствием доступа к Api на сервере ИТА ЮФУ [3].

В качестве критериев были взяты такие параметры как: удобный интерфейс, поиск по преподавателю, Расписание по календарю, история поиска, расписание на сегодня, расписание на завтра, расписание на весь семестр, расписание на выбранную дату, сохранение расписания, расчет времени для начала следующей пары.

В таблице 1 предоставлена сравнительная характеристика аналогов.

Таблица 1 - Обоснование выбора аналога для сравнения

|  |  |  |  |
| --- | --- | --- | --- |
| Параметры, характеризующие действующие аналоги и разработку | «Расписание ИТА ЮФУ» | «Расписание ИТА ЮФУ Ogetto Web» | «Расписание Радика (ИТА ЮФУ)» |
| Удобный интерфейс | + | + | - |
| Поиск по преподавателю | + | + | - |
| Расписание по календарю | + | - | + |
| История поиска | - | + | - |
| Расписание на сегодня | - | + | - |
| Расписание на весь семестр | + | + | + |
| Сохранение расписания | - | + | - |
| Расчет времени для начала следующей пары | - | + | - |

* 1. Требования к разрабатываемому ПО

1.2.1 Требование к взаимодействию между модулями

Взаимодействие между мобильным приложением и сервером осуществляется при помощи глобальной сети Интернет.

1.2.2 Требования к защите информации от несанкционированного доступа

Компоненты подсистемы защиты от НСД должны обеспечивать:

* 1. Идентификацию пользователя;
  2. Безопасность персональных данных при их обработке;
  3. Свобода поиска, получения, передачи, производства и распространения информации любым законным способом;
  4. Достоверность информации и своевременность ее предоставления;
  5. Неприкосновенность частной жизни, недопустимость сбора, хранения, использования и распространения информации о частной жизни лица без его согласия.

1.2.3 Требования по сохранности информации при авариях

Система для обеспечения безопасности хранения данных пользователей должна предусматривать возможность регулярного формирования резервных копий, с целью обеспечения их сохранности в случае аварийной ситуации.

1.2.4 Требования к входной информации

Система должна быть способна принимать входную информацию, вводимую в мобильное устройство.

Входная информация для сервера со стороны клиентского мобильного приложения будет посылаться при помощи HTTP-запроса, такой запрос называется REST-запросом.

Основными источниками входной информации для системы являются:

1. Текстовое поле ввода;
2. Данные выбранные с UI controls.

1.2.5 Требования к выходной информации

Выходная информация системы, оформляется в выдаче ответа сервера и выдача результата пользователю на экран.

Требованием к выходной информации сервера является текстовый формат обмена данными JSON.

1.2.6 Требования к языку взаимодействия с пользователями системы

Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно использовать русский и английский языки.

1.2.7 Требования к применению языков программирования

В разработке клиентского обильного приложения могут быть использованы следующие языки программирования:

1. Java;
2. C# Xamarin.

1.2.8 Требования к средствам описания предметной области

Для исследования предметной области используются диаграммы UML. Диаграммы позволяют описать поведение системы (для анализа) или показать детали архитектуры (для проектирования). С помощью диаграмм можно разработать детальный план создаваемой системы, содержащий не только ее концептуальные элементы, такие, как системные функции, но и конкретные особенности, например, классы, написанные на специальных языках программирования, схемы баз данных и программные компоненты многократного использования.

1.2.9 Требования к общесистемным программным средствам

Клиентское приложение должно функционировать под управлением ОС Android версии не ниже 4.4 KitKat.

1.2.10 Требования к вычислительным сетям

Для функционирования и взаимодействия с сервером расписания занятий, необходимо мобильное подключение, либо иные источники передачи и получения данных в сети интернет.

* 1. Обзор инструментальных средств

Для разработки программного продукта были рассмотрены следующие программные средства:

* 1. Среда разработки Android Studio;
  2. Эмулятор Genymotion;
  3. JSON Editor Online;
  4. Редактор Notepad++;
  5. Система контроля версий Git.;
  6. Среда разработки Eclipse;
  7. Сервис для хостинга проектов Github;
  8. Сервис для хостинга проектов Bitbucket.

1.3.1 Среда разработки Android Studio

Android Studio – это интегрированная среда разработки для работы с платформой Android. Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux. Среда разработки находилась в свободном доступе начиная с версии 0.1, опубликованной в мае 2013, а затем перешла в стадию бета-тестирования, начиная с версии 0.8, которая была выпущена в июне 2014 года. Первая стабильная версия 1.0 была выпущена в декабре 2014 года, тогда же прекратилась поддержка плагина Android Development Tools (ADT) для Eclipse [4].

1.3.2 Эмулятор Genymotion

В состав IDE Android Studio входит встроенный эмулятор, но он имеет существенный недостаток – очень большую ресурсоемкость. При его запуске на слабых компьютерах процесс разработки становиться очень сложным и медлительным. По этой причине разумнее использовать другой эмулятор, ничем не уступающий в характеристиках встроенному – эмулятор Genymotion[5]. Данный эмулятор не требует большого количества ресурсов и полноценно работает на слабых компьютерах. Данный эмулятор распространяется условно бесплатно. Он так же предоставляет большой выбор устройств для запуска на них приложений, возможен так же выбор моделей устройств и версий ОС Android.

1.3.3 JSON Editor Online

JSON Editor Online – это веб-инструмент для просмотра, редактирования данных в формате JSON. Данный инструмент показывает с одной стороны ваш JSON запрос в виде строки, а с другой – в виде структурированного дерева редактируемый код. Поддерживаемые браузеры: хром, Firefox, Safari, Opera, Internet Explorer 8+ [6].

1.3.4 Редактор Notepad++

Текстовый редактор, предназначенный для программистов и всех тех, кого не устраивает скромная функциональность входящего в состав Windows Блокнота.

Основными особенностями программы является:

* 1. Подсветка текста и возможность сворачивания блоков, согласно синтаксису языка программирования;
  2. Поддержка большого количества языков (C, C++, Java, XML, HTML, PHP, Java Script, ASCII, VB/VBS, SQL, CSS, Pascal, Perl, Python, Lua, TCL, Assembler);
  3. WYSIWYG (печатаешь и получаешь то, что видишь на экране);
  4. Настраиваемый пользователем режим подсветки синтаксиса;
  5. Авто-завершение набираемого слова;
  6. Одновременная работа с множеством документов;
  7. Одновременный просмотр нескольких документов;
  8. Поддержка регулярных выражений Поиска/Замены;
  9. Полная поддержка перетягивания фрагментов текста;
  10. Динамическое изменение окон просмотра;
  11. Автоматическое определение состояния файла;
  12. Увеличение и уменьшение;
  13. Заметки;
  14. Выделение скобок при редактировании текста;
  15. Запись макроса и его выполнение [7].

1.3.5 Система контроля версий Git

Система управления – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое [8]. Немаловажным является тот факт, что в среде разработки Android Studio имеется плагин для работы с Git, что упрощает процесс просмотра и отправки на сервер изменений, сделанных в процессе разработки программного продукта.

1.3.6 Среда разработи Eclipse

Eclipse это бесплатная платформа написанная на языке программирования Java, для увеличения эффективности разработки программного обеспечения. В отличие от аналогов в ней используется платформо-зависимая библиотека - Standard Widget Toolkit, для создания элементов пользовательского интерфейса. Исходный код данной среды разработки открытый, что дает разработчикам свободу действий для улучшения данной IDE. Eclipse не ограничен инструментами для разработки только на Java языке программирования и может служить инструментом для создания программных продуктов на разных языка. Немало важным является то, что плагины для среды разработки, создаются в самой этой среде разработке[9].

1.3.7 Сервис для хостинга проектов Github

GitHub – условно-бесплатный веб-сервис для хранения IT проектов, работающий с системой контроля версий Git. Данный сервис дает возможность бесплатного пользования сервисом для проектов с открытым исходным кодом. Для тех, кто хочет иметь приватное хранилище, имеется несколько различных платных тарифных планов [10].

Возможности:

1. Имеется возможность отслеживать чужие проекты, комментировать и править их;
2. Так же есть возможность объединять свои репозитории с отображением вклада каждого участника;
3. Есть система отслеживания ошибок, имеется подсветка синтаксиса для большинства языков программирования;
4. Есть возможность не только копирования кода проекта через Git, но и скачивания в виде архивов.

1.3.8 Сервис для хостинга проектов Bitbucket

Bitbucket - веб-сервис аналогичный сервису GitHub, с таким же функционалом, но имеющий бесплатный приватный репозиторий для команд меньше пяти человек. В некоторых моментах данный сервис превосходит Github, например, поддержка Mercurial, имеется универсальная панель быстрой навигации [11].

1.3.9 Выбор инструментальных средств для разработки

В качестве ПО для разработки были выбраны:

1. Среда разработки Android Studio;
2. Эмулятор Genymotion;
3. JSON Editor Online;
4. Редактор Notepad++;
5. Система контроля версий Git.;
6. Сервис для хостинга проектов Github.

В сравнении с популярным аналогом eclipse, Android Studio быстрее, более приятный интерфейс, меньше багов. Eclipse хоть имеет функционал как у Android Studio, но для этого нужно установить огромное множество плагинов.

Поэтому не раздумывая в качестве среды разработки была выбрана Android Studio.

В сравнение со стандартным эмулятором, входящим в состав Android Studio, операционной системы Android, Genimotion более быстрый и менее ресурсозатратный, поэтому разумнее использовать его.

В виду распространенности и более удобного интерфейса в качестве веб-сервиса для хранения проекта, был выбран сервис Github.

1. Анализ и разработка сценариев использования
   1. Объектно-ориентированный анализ

В ходе проведения объектно-ориентированного анализы были выявлены два основных пользователя – пользователь, не предоставивший данные о группе обучения и пользователь предоставивший данные.



Рисунок 1 – Выявленные пользователи системы.

Таблица 2 – Выявленные пользователи системы.

|  |  |
| --- | --- |
| *Пользователь* | *Краткое описание* |
| Заинтересованный пользователь | Если еще не предоставив свои данные, такие как интересующая группа, преподаватель, то должен предоставить эти данные, иначе может приступить к использованию функционала приложения. |
| Пользователь не предоставивший свои данные | Пользователь не предоставивший свои данные, такие как интересующая группа, должен предоставить их. Только после этого предоставляется полный функционал системы. |
| Зарегистрированный пользователь | Имеет возможность просмотра расписания занятий, для нужной группы, для нужной даты, определенного преподавателя, так же будет отображаться номер недели и рассчитано время до начала ближайшей пары. |

В результате выявления основных пользователей системы, были выявлены варианты взаимодействия данных пользователей с системой.

Таблица 3 – Выявление вариантов взаимодействия пользователей с системой.

|  |  |  |
| --- | --- | --- |
| *Пользователь* | *Наименование* | *Формулировка* |
| Пользователь не предоставивший свои данные | Предоставление данных | Этот вариант взаимодействия позволяет пользователю, который еще не ввел свои данные, предоставить их системе. |
| Пользователь предоставивший свои данные | Просмотр расписания занятий, для нужной группы, ИТА ЮФУ | Этот вариант взаимодействия позволяет пользователю, предоставившему свои данные, просматривать расписание, для своей группы, ИТА ЮФУ. |
| Пользователь предоставивший свои данные | Просмотр расписания занятий, для нужной даты | Этот вариант взаимодействия позволяет пользователю, предоставившему свои данные, просматривать расписание занятий для требуемой даты. |
| Пользователь предоставивший свои данные | Просмотр расписания занятий, определенного преподавателя | Этот вариант взаимодействия позволяет пользователю, предоставившему свои данные, просматривать расписание занятий определенного преподавателя. |
| Пользователь предоставивший свои данные | Отображение номера недели | Этот вариант взаимодействия позволяет пользователю, предоставившему свои данные, всегда знать номер учебной недели. |
| Пользователь предоставивший свои данные | Расчет времени до начала пары | Этот вариант взаимодействия позволят пользователю, предоставившему свои данные, видеть оставшееся время до начала ближайшей пары. |

Все варианты использования показаны на рисунке 2.



Рисунок 2 – Диаграмма прецедентов системы.

* 1. Структуризация вариантов использования

Анализ вариантов использования выявил следующие взаимосвязи.

1. Вариант использования «предоставление данных» включает в себя действия по заполнению формы данными, отправку данных на сервер, получение ответа от сервера и выдачу ответа об результате. Поэтому нужно добавить следующие варианты использования: «Форма ввода данных», «Отправка данных на сервер», «Получение ответа сервера», «Ответ об результате». Данные варианты использования связаны отношением включения с вариантом использования «Предоставление данных». Рассмотренные варианты использования показаны на рисунке 3.



Рисунок 3 – Связь включение для реализации процедуры ввода данных.

2. Вариант использования «Просмотр расписания занятий» включает в себя действие по выбору формы с выводом расписания занятий на неделю, далее будет отображена форма с расписанием занятий на текущую неделю. Варианты использования связаны отношением включения с вариантом использования «Просмотр расписаний занятий». Рассмотренные варианты использования показаны на рисунке 4.



Рисунок 4 – Связь включение для реализации процедуры просмотра расписания занятий.

3. Вариант использования «Просмотр расписания занятий для нужной даты» включает в себя действия по заполнению полей формы выбора даты, отправку данных на сервер, получение ответа от сервера и выдачу результатов поиска. Поэтому нужно добавить следующие варианты использования: «Форма выбора даты», «Отправка данных на сервер», «Получение ответа от сервера», «Выдача результата». Варианты использования связаны отношением включения с вариантом использования «Просмотр расписания занятий для нужной даты». Рассмотренные варианты использования показаны на рисунке 5.



Рисунок 5 – Связь включение для реализации процедуры просмотра расписания занятий для нужной даты.

4. Вариант использования «Просмотр расписания занятий, определенного преподавателя» включает в себя действия по заполнению полей формы поиска преподавателя, отправку данных на сервер, получение ответа от сервера и выдачу результатов поиска. Поэтому нужно добавить следующие варианты использования: «Форма поиска преподавателя», «Отправка данных на сервер», «Ответ сервера», «Выдача результата». Варианты использования связаны отношением включения с вариантом использования «Поиск людей». Рассмотренные варианты использования показаны на рисунке 6.



Рисунок 6 – Связь включение для реализации процедуры поиска преподавателя.

5. Вариант использования «Отображение номера текущей недели» включает в себя действия по отображению формы с номером текущей недели, отправку данных на сервер, получение ответа от сервера и выдачу результатов. Поэтому нужно добавить следующие варианты использования: «Форма с номером текущей недели», «Отправка данных на сервер», «Ответ сервера», «Выдача результат». Варианты использования связаны отношением включения с вариантом использования «Отображение номера недели». Рассмотренные варианты использования показаны на рисунке 7.



Рисунок 7 – Связь включение для реализации процедуры отображение номера текущей недели.

6. Вариант использования «Расчет времени до начала пары» включает в себя действия по поиску ближайшей пары, отправку данных на сервер, получение ответа от сервера и выдачу результата. Поэтому нужно добавить следующие варианты использования: «Отправка данных на сервер», «Ответ сервера», «Вывод результата». Варианты использования связаны отношением включения с вариантом использования «Расчет времени до начала пары». Рассмотренные варианты использования показаны на рисунке 8.



Рисунок 8 – Связь включение для реализации процедуры расчет времени до начала пары.

7. Варианты использования «Вывод результата», «Ответ об результате» выполняют однотипные действия по выдаче результата выполнения операции. Поэтому можно ввести вариант использования «Результат операции», который будет связываться с вышеописанными вариантами использования отношением расширения. Рассмотренные варианты использования показаны на рисунке 9.



Рисунок 8 – Связь расширение для реализации процедуры выдачи результата операции.

Результирующая диаграмма вариантов использования показана на рисунке 9.



Рисунок 9 – Модифицированная диаграмма вариантов использования мобильного приложения.

* 1. Реестр вариантов использования

Результаты анализа, проделанного выше «Структуризация вариантов использования» представлены в результирующем списке вариантов использования, показанном в таблице 4.

Таблица 4 – Реестр вариантов использования.

| Код | Основной актор | Наименование | | Краткое описание |
| --- | --- | --- | --- | --- |
| H1 | Пользователь не предоставивший данные | Предоставление данных | | Позволяет пользователю, не предоставившему данные, предоставить их. |
| H2 | Пользователь не предоставивший данные | Заполнение формы данными | | Пользователь может заполнить форму ввода данных. |
| Н3 | Пользователь не предоставивший данные и пользователь предоставивший данные | Отправка данных на сервер | | Пользователь формирует запрос к серверу и отправляет его для обработки на сервере. |
| Н4 | Пользователь не предоставивший данные и пользователь предоставивший данные | Получение ответа от сервера | | После обработки пользовательского запроса, сервер направляет ответ. |
| Н5 | Пользователь не предоставивший данные и пользователь предоставивший данные | Результат операции | | Происходит отображение успеха выполнения определенной операции. |
| С1 | Пользователь предоставивший данные | Просмотр расписания занятий ИТА ЮФУ | | Пользователь может запросить показать расписание занятий ИТА ЮФУ. |
| С2 | Пользователь предоставивший данные | Форма с выводом расписания занятий на неделю | | Пользователю выводится расписание занятий ИТА ЮФУ. |
| С3 | Пользователь предоставивший данные | Просмотр расписания занятий для нужной даты | | Пользователь может просматривать расписание занятий для нужной даты. |
| С4 | Пользователь предоставивший данные | Форма выбора даты | | Пользователь может выбрать интересующую его дату, для которой показать расписание занятий. |
| С5 | Пользователь предоставивший данные | Просмотр расписания занятий определенного преподавателя | | Пользователь может выбрать интересующую его преподавателя, для которой показать расписание занятий. |
| С6 | Пользователь предоставивший данные | Форма поиска преподавателя | | Пользователь вводит данные интересующего его преподавателя. |
| С7 | Пользователь предоставивший данные | | Отображение номера недели | Выводит пользователю номер текущей недели. |
| С8 | Пользователь предоставивший данные | | Расчет времени до начала ближайшей пары | Выводит пользователю время до начала ближайшей пары. |
| С9 | Пользователь предоставивший данные | | Форма отображения времени до начала пары | Отображает время начала ближайшей пары. |

* 1. Спецификации вариантов использования

В качестве средства спецификаций вариантов использования была использована диаграмма деятельности UML.

Диаграмма деятельности позволяет определить поведение с помощью последовательного исполнения поведений более низкого уровня. Исполнение следующего действия может начинаться в результате наступления одного из следующих событий:

* завершение исполнения предыдущего действия (потоки работ);
* появление необходимых данных (потоки данных);
* наступление определенного события (потоки управления).
  + 1. Вариант использования «Предоставление данных»

На рисунке 10 показана спецификация прецедента «Предоставление данных» при помощи диаграммы деятельности.



Рисунок 10 – Диаграмма деятельности для варианта использования «Предоставление данных».

* + 1. Вариант использования «Просмотр расписания занятий»

На рисунке 11 показана спецификация прецедента «Просмотр расписания занятий» при помощи диаграммы деятельности.



Рисунок 11 – Диаграмма деятельности для варианта использования «Просмотр расписания занятий».

* + 1. Вариант использования «Просмотр расписания занятий для нужной даты»



Рисунок 12 – Диаграмма деятельности для варианта использования «Просмотр расписания занятий для нужной даты».

На рисунке 12 показана спецификация прецедента «Просмотр расписания занятий для нужной даты» при помощи диаграммы деятельности.

* + 1. Вариант использования «Просмотр расписания занятий определенного преподавателя»



Рисунок 13 – Диаграмма деятельности для варианта использования «Просмотр расписания занятий определенного преподавателя».

На рисунке 13 показана спецификация прецедента «Просмотр расписания занятий определенного преподавателя» при помощи диаграммы деятельности.

* + 1. Вариант использования «Отображение номера недели»



Рисунок 14 – Диаграмма деятельности для варианта использования «Отображение номера недели».

На рисунке 14 показана спецификация прецедента «Отображение номера недели» при помощи диаграммы деятельности.

* + 1. Вариант использования «Расчет времени до начала пары»



Рисунок 15 – Диаграмма деятельности для варианта использования «Расчет времени до начала пары».

На рисунке 15 показана спецификация прецедента «Расчет времени до начала пары» при помощи диаграммы деятельности.

1. Проектирование приложения

Данное приложение имеет один экран на котором содержатся три фрагмента – «Расписание занятий на сегодня», «Расписание занятий на завтра», «Расписание занятий на семестр», так же есть Action Bar, меню навигации и несколько диалоговых окон.

* 1. Экран «Главная страница»

На данном экране расположены все фрагменты приложения. Тут обрисовывается так называемый tab layout – фрагмент, состоящий из компонентов библиотеки support v7 от компании Google, таких как RecycleView и CardView [12]. Так же из этой же библиотеки используется Toolbar вместо устаревшего на данный день стандартного Action Bar и Navigation View для быстрого доступа к некотором функциям приложения, рассмотренным дальше.

Взаимодействие с сервером происходит в следующем порядке:

1. пользователь выбирает действие, которое требует обращение к серверу;
2. приложение обрабатывает данное действие, в большинстве случаев требуя от пользователя определенные параметры;
3. в фоновом режиме, приложение составляет запрос к серверу;
4. приложение отправляет пакет данных на сервер;
5. ожидается ответ от сервера;
6. при успешном выполнение запроса, сервер вернет данные, относительно параметров запроса в формате JSON;
7. в случае недоступности сервера или неправильного запроса, пользователю будет сообщено об ошибке.
   1. Авторизация

При первом запуске приложения, оно потребует указать интересующую группу, для которой и будет получено расписание занятий. Пользователю будет выведено диалоговое окно, в котором и необходимо указать группу обучения, поле валидируемое – поэтому если пользователь ошибся при вводе, то приложение сообщит ему об этом сразу. Далее формируется запрос Get к Api сервера, в котором в качестве параметров выступает номер группы и номер семестра. Отправив запрос, Api сервера сформирует ответ, либо же выдаст ошибку, если был неверный запрос. В ответе придёт расписание на весь семестр в формате JSON и создастся локальная копия полученных данных, для возможности в дальнейшем работать без подключения к сети интернет. В последствие номер группы можно изменить.

Далее мобильное приложение уведомит пользователя, что группа изменена на введённую им. После чего пользователь сможет начать пользоваться полным функционалом системы.

* 1. Фрагмент расписания на семестр

Пользователь авторизованный в мобильном приложении, получает данные о расписании занятий ИТА ЮФУ на целый семестр – это и является первым рассматриваемым фрагментом на главном экране. При активном данном фрагменте, приложение проверяет подключение к сети интернет, если оно есть, то просматривает, актуальны ли эти данные по сравнению с сервером и в случае различий, в фоновом режиме формируется запрос Get к Api сервера, с параметрами – номер группы, номер семестра, отправляет запрос и в случае успешного выполнения операции, обновляет данные для всего семестра и перезаписывает локальную копию. Если же по каким-то причинам доступ к серверу ограничен или отсутствует подключение к сети интернет, то используются локальные данные.

В результате на экране мобильного приложения отображается расписание занятий для всего семестра, с указанием времени проведения занятия, названия предмета, номера аудитории, ФИО преподавателя.

* 1. Фрагмент расписания на данный день

Переключивших на вторую вкладку, приложение проверяет подключение к сети интернет, если оно есть, то просматривает, актуальны ли эти данные по сравнению с сервером и в случае различий, в фоновом режиме формируется запрос Get к Api сервера, с параметрами – номер группы, номер семестра, четность недели, сегодняшний день недели и отправляет запрос и в случае успешного выполнения операции, обновляет данные для этого дня. Если же по каким-то причинам доступ к серверу ограничен или отсутствует подключение к сети интернет, то используются локальные данные. Затем, так же в фоновом режиме, приложение формирует запрос Get к Api сервера с параметрами, номер группы, номер семестра и отправляет запрос для обновления локальной копии на весь семестр. Результатом операции будет вывод фрагмента с расписанием занятий на данный учебный день, с указанием времени проведения занятия, названия предмета, номера аудитории, ФИО преподавателя.

* 1. Фрагмент расписания на следующий день

В результате выбора третьей вкладки, приложение проверяет подключение к сети интернет, если оно есть, то просматривает, актуальны ли эти данные по сравнению с сервером и в случае различий, в фоновом режиме формируется запрос Get к Api сервера, с параметрами – номер группы, номер семестра, четность недели, номер следующего дня недели и отправляет запрос и в случае успешного выполнения операции, обновляет данные для этого дня. Если же по каким-то причинам доступ к серверу ограничен или отсутствует подключение к сети интернет, то используются локальные данные. Затем, так же в фоновом режиме, приложение формирует запрос Get к Api сервера с параметрами, номер группы, номер семестра и отправляет запрос для обновления локальной копии на весь семестр.

По окончанию операции, будет выведено расписание занятий на следующий учебный день, с указанием времени проведения занятия, названия предмета, номера аудитории, ФИО преподавателя.

* 1. Фрагмент меню навигации приложения

В приложении так же имеется меню навигации для доступа к функциям содержащемся в его меню, пользователю необходимо выбрать один из пунктов, приложение обработает нажатие по данному пункту меню и вызовет функцию, предназначенную для работы с данным пунктом.

В меню навигации доступны такие функции как:

Изменить группу;

1. Запросить расписание для нужной даты;
2. Запросить расписание интересующего преподавателя;
3. Вывести номер текущей недели;
4. Вывести время до начала следующей пары.

Так же в шапке меню отображается, выбранная пользователем группа для получения расписания занятий.

* 1. Пункт меню, изменить группу

При взаимодействии с данным пунктом меню, приложение выдаст пользователю диалоговое окно с просьбой ввести интересующую его группу, поле ввода, валидируется, поэтому при неправильном вводе номера группы, приложение выдаст сообщение об ошибке и предложит заново произвести выше описанные действия. Далее приложение, проверит возможность доступа к сети интернет и в случае его отсутствия, сообщит пользователю о невозможности выполнения данной операции, в следствие отсутствия подключения к сети интернет. Если же подключение есть, то приложение в фоновом режиме составит запрос Get запрос к Api сервера с параметрами – номер группы, номер семестра относительно даты выставленной на телефоне и отправит запрос серверу. В случае корректности данных и доступности сервера, он отправит в ответ данные в формате JSON. Приложение в свою очередь обработает данные и сохранит их локальную копию. Если запрошенная группа не существует в базе сервера, то в ответ будет отправлено сообщение об ошибке. Возможна ситуация, когда недоступен сервер, в этом случае приложение будет ждать от сервера ответ до тайма аута и в случае неполучения ответа от сервера сообщит об этом пользователю.

* 1. Пункт меню, расписание для нужной даты

При взаимодействии с данным пунктом меню, приложение выдаст пользователю диалоговое окно с просьбой ввести интересующую его дату, поле ввода, валидируется, поэтому при неправильном вводе даты, приложение выдаст сообщение об ошибке и предложит заново произвести выше описанные действия. Далее приложение, проверит возможность доступа к сети интернет и в случае его отсутствия, сообщит пользователю о невозможности выполнения данной операции, в следствие отсутствия подключения к сети интернет. Если же подключение есть, то приложение рассчитает для данной даты номер семестра, номер недели и номер дня и в фоновом режиме составит запрос Get запрос к Api сервера с параметрами – номер группы, номер семестра, четность недели и номер дня, отправит запрос серверу. В случае корректности данных и доступности сервера, он отправит в ответ данные в формате JSON. Приложение в свою очередь обработает данные и выведет их на экран мобильного приложения, с указанием времени проведения занятия, названия предмета, номера аудитории, ФИО преподавателя. Если запрошенная группа не существует в базе сервера, то в ответ будет отправлено сообщение об ошибке. Возможна ситуация, когда недоступен сервер, в этом случае приложение будет ждать от сервера ответ до тайма аута и в случае неполучения ответа от сервера сообщит об этом пользователю.

* 1. Пункт меню, расписание преподавателя

При взаимодействии с данным пунктом меню, приложение выдаст пользователю диалоговое окно с просьбой ввести ФИО интересующего его преподавателя, поле ввода, валидируется, поэтому при неправильном вводе, приложение выдаст сообщение об ошибке и предложит заново произвести выше описанные действия. Далее приложение, проверит возможность доступа к сети интернет и в случае его отсутствия, сообщит пользователю о невозможности выполнения данной операции, в следствие отсутствия подключения к сети интернет. Если же подключение есть, то приложение рассчитает для данной даты номер семестра и в фоновом режиме составит запрос Get запрос к Api сервера с параметрами – ФИО преподавателя, номер семестра и отправит запрос серверу. В случае корректности данных и доступности сервера, он отправит в ответ данные в формате JSON. Приложение в свою очередь обработает данные и выведет их на экран мобильного приложения, с указанием времени проведения занятия, названия предмета, номера аудитории, ФИО преподавателя. Если в базе данных нету совпадения по ФИО указанному в запросе, то в ответ будет отправлено сообщение об ошибке. Возможна ситуация, когда недоступен сервер, в этом случае приложение будет ждать от сервера ответ до тайма аута и в случае неполучения ответа от сервера сообщит об этом пользователю.

* 1. Пункт меню, номер текущей недели

При нажатие по данному пункту, приложение вызывает стандартную функцию Java, GetWeek(), и получает в ответ номер текущей недели по счету, относительно 1 января, далее приложение от полученного результата отнимает 5 – число недель, после которого начинается новый семестр в начале календарного года и выводит пользователю в виде диалога результат операции.

* 1. Пункт меню, время до начала пары

При нажатие по данному пункту, приложение вызывает стандартную функцию Java, GetTime() и получает в ответ название дня и время на данный момент, далее приложение в локальной копии находит какие пары в этот день проходят и в какое время, находится пара, которая начинает после времени полученного на данный момент, если таковая отсутствует для данного дня, то просматривается следующий день, для избежание зацикливания, поиск пары останавливается, при возврате ко дню и времени, которое актуально на данный момент. Если такое занятие было найдено, то приложение при помощи диалогового окна выведет данные, с указанием времени проведения занятия, названия предмета, номера аудитории, ФИО преподавателя.

1. Программная реализация приложения

При разработке приложения была использована концепция Material Design, шаблон проектирования Model View Presenter и такие библиотеки как:

1. Android support library;
2. Retrofit;
3. Dagger 2.
   1. Иерархия классов

Приложение разрабатывалось, используя шаблон проектирования Model-View-Presenter, более подробно описанный далее. В результате получаем следующую архитектуру:

1. Model – уровень в котором, реализуется активация какого либо действия;
2. View – сам фрагмент, вызывает методы Presenter, когда с ней взаимодействуют;
3. Presenter – прокладка между моделью и представлением.

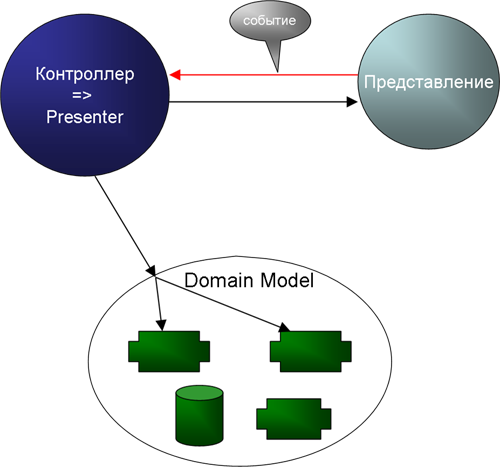


Рисунок 16 – графически вид шаблона Model View Presenter.

* + 1. Model

Классами уровня модел являются:

1. HalfDTO – в данном классе описывается структура данных, для передачи при помощи уровня презентера на уровень вью, чтобы отобразить расписание занятий на весь семестр, имеется конструктор и гетеры с сеттерами для необходимых полей;
2. TodayDTO – класс в котором описывается структура для передачи данных презентеру, который обработает их и передаст на уровень вью в удобного для него формате, чтобы вывести их на фрагмент расписания занятий на сегодняшний день;
3. TomorrowDTO – в данном классе описывается структура данных, для передачи их в дальнейшем через презентер уровню вью;
4. JsonDTO – класс в котором описана структура для работы с сохраненными или полученными из сети интернет данными в формате JSON.
   * 1. View

Классами уровня вью являются:

1. Half – в данный класс передаются данные, которые будут отображаться на первом фрагменте, главного экрана мобильного приложения, в нем вызываются функции HalfListAdapter, для наполнения активити данными;
2. Today – класс получающий данные которые будут отображены на втором фрагменте, используя функции TodayListAdapter-а;
3. Tomorrow – в данном классе реализуется логика отображения данных для отображения следующего дня расписания занятий на третьем фрагменте главного активити.
   * 1. Presenter

Классами уровня презентер являются:

1. TabsFragmentAdapter – абстрактный класс, от которого наследуются адаптеры обрабатывающие данные для вывода расписания занятий на семестр, сегодняшнюю дату и следующий день;
2. HalfListAdapter – класс который реализует функции для обработки данных о расписании занятий на семестр и передачей этих данных уровню вью;
3. TodayListAdapter – класс который реализует функции для обработки данных о расписании занятий на сегодняшний день и передачей этих данных уровню вью;
4. TomorrowListAdapter – класс который реализует функции для обработки данных о расписании занятий на завтрашней день и передачей этих данных уровню вью;
5. JsonAdapter – класс реализующий функции для обработки данных из локальной копии, в случае их актуальности или отсутствия возможности получения от сервера ввиду отсутствия подключения к сети интернет или недоступности сервера или при доступе к серверу, если данные не актуальны с последующей передачей их фрагментам.
   1. Пример взаимодействия уровней

Для примера на рисунке 17 показано взаимодействие класса Half, HalfDTO, HalfListAdapter, JsonDTO и JsonAdapter. Остальные модули реализованые используя паттерн Model View Presenter взаимодействуют по такой же схеме, поэтому описывать их все не имеет смысла.

Рисунок 17 – пример взаимодействия классов

Как видно на UML диаграмме, при вызове метода OnStart главного класса приложения – MainActivity, создается объект абстрактного класса AbstractTabFragment в котором инициализируются три фрагмента главного экрана, в данном случае рассматривается один из них – Half, которой предназначен для вывода расписания на весь семестр. Далее вызывается его наследник, класс Half, уровня View, как было описано выше про алгоритм шаблона Model View Presenter, при взаимодействии с View, оно вызывает методы уровня Presenter, который в диаграмме UML обозначен как HalfListAdapter. Класс HalfListAdapter является ключевым звеном для взаимодействия между классом уровня View – Half и классом уровня Presenter – HalfListAdapter. То есть после создания объекта класса Half, в нем инициализируется фрагмент для вывода расписания на семестр и виджет RecycleView из библиотеки SupportLibrary от компании Google. Затем вызывается метод, для наполнения вышеописанного виджета – AddMockHalfListData, в котором и создаются объекты класса HalfListAdapter. Класс уровня Presenter инициализирует виджет CardView, являющийся частью RecycleView и TextView для вывода времени занятия, названия предмета, аудитории и ФИО преподавателя. Для заполнения вышеописанных полей вызывается метод уровня Model – HalfDTO в котором описывается используемая структура, для их заполнения. Так же при вызове функции AddMockHalfListData класса Half уровня View создается объекта класса JsonAdapter уровня Presenter, в свою очередь данный метод организует данные, которыми и будут заполнены CardView.

* 1. Material Design

Material Design – концепция построения логики приложения, его внешнего вида, сервисов и функций, для получения максимально легкого и интуитивно понятного восприятия пользователями программного продукта.

Четыре основных принципа данной концепции:

1. Тактильные поверхности. В данной концепции интерфейс приложения состоит из так называемой «цифровой бумаги». Элементы расположены на разном уровне и отбрасывают тень, друг на друга. Реалистичное освещение показывает разделение между элементами помогая пользователям лучше понимать предназначение интерфейса и принцип взаимодействия с ним.
2. Полиграфический дизайн. Если есть бумага, то как же обойтись без чернил для неё? В данной концепции контент отображается на цифровой бумаге в журнальном или плакатном виде.
3. Осмысленная анимация. Тут придерживаются реальности, что объект не может появится из неоткуда или пропасть в некуда. При помощи анимации в Material Design пользователю даются подсказки о том, как работать с интерфейсом приложения.
4. Адаптивный дизайн. Данный принцип предусматривает корректность отображения остальных трех принципов на разных устройствах и размерах экрана.
   1. MVP

Model View Presenter – шаблон проектирования, наследованный от не менее известного паттерна Model View Controller. Данный паттерн убирает зависимость логики приложения от его представления, то есть от внешнего вида приложения не зависит его поведение. Данный паттерн хоть и не является архитектурным шаблоном, но буду даже ответственным только за представление, все равно улучшает её. Стандартного подхода к написанию приложения с использованием паттерна Model View Presenter нету, но обычно используют три составляющих, это Model (модель), View (представление), Presenter (предъявитель).

1. Model – объект, который реализует активацию какого либо действия.
2. View – обычно является активити или фрагментом, в зависимости от приложения. Все что делает эта часть, это вызывает методы презентера, когда с ней происходит взаимодействие.
3. Presenter – ответственен за взаимодействие между моделью и представлением, посредством извлечения данных из модели и передает их в удобном формате View. В отличии от MVC он так же решает, что произойдет, когда вы взаимодействуете с представлением.
   1. Android support library

В приложении использовалась данная библиотека, так как на данный процент пользователей, использующих версию ОС Android 4.4 Kitkat – 73,9 %, поэтому целесообразно поддерживать данную версию, а библиотека support library позволяет использовать нововведения новых версий ОС Android на старых версиях. Например, обеспечивает поддержку:

1. Material Design – концепция взаимодействия с пользователем, новых версий ОС Android;
2. Navigation View – панель навигации;
3. Floating Labels – плавающий ярлык;
4. Floating Action Button – плавающая кнопка;
5. Snackbar – всплывающее уведомление с возможностью взаимодействия с ним;
6. Tabs - вкладки;
7. Motion and Scroll framework – управление жестами и скролом.
   1. Retrofit

Retrofit является REST-клиентом для безопасной работы в Android и Java. Библиотекой удобно пользоваться для запроса к различным веб-сервисам с командами GET, POST, PUT, DELETE. Может работать в асинхронном режиме, что избавляет от лишнего кода. Сразу имеет при себе набор популярных конвертеров, таких как:

* 1. Gson;
  2. Jackson;
  3. Moshi;
  4. Protobuf;
  5. Wire;
  6. Simple XML;
  7. Scalars.

Так же используя фабрику конвертера, можно создать собственный алгоритм конвертирования.

* 1. Dagger 2

Dagger 2 – один из самых популярных Java/Android фреймворков, предназначенных для внедрения зависимостей. При использовании этой техники упрощается тестирование и пере использование классов. Он генерирует за разработчика большую часть инициализирующего кода. При использовании Dagger 2 точка входа в наше приложение может быть написана всего в несколько строк кода, независимо от того, сколько классов у нас есть и сколько в них присутствует зависимостей. Одна и та же зависимость может быть использована во множестве классов или когда по каким-то причинам нет возможности аннотировать класс.

* 1. Элементы пользовательского интерфейса

Разработанное мобильное приложение содержит следующие схемы для вывода информации:

1. Activity\_main – главный экран приложения на нем и располагаются все далее описанные фрагменты;
2. Fragment\_half – часть таба, фрагмент для вывода расписания занятий на весь семестр, в нем создается RecycleView, который содержит в себе все CardView;
3. Fragment\_today – часть таба, фрагмент для вывода расписания на сегодняшний день, в нем создается RecycleView, который содержит в себе все CardView;
4. Fragment\_tommorow – часть таба, фрагмент для вывода информации о занятиях на завтрашний день, в нем создается RecycleView, который содержит в себе все CardView;
5. Half\_item – часть фрагмента fragment half, тут создается CardView и поля TextView для вывода в них времени занятия, названия предмета, аудитории и ФИО преподавателя;
6. Today\_item - часть фрагмента fragment today, тут создается CardView и поля TextView для вывода в них времени занятия, названия предмета, аудитории и ФИО преподавателя;
7. Tomorrow\_item - часть фрагмента fragment tomorrow, тут создается CardView и поля TextView для вывода в них времени занятия, названия предмета, аудитории и ФИО преподавателя;
8. Navigation\_header – схема для вывода меню навигации;
9. Toolbar – бар для вывода меню.

В описанные выше элементы пользовательского интерфейса проиллюстрированы в приложении 2.