

- внесение занятий, аудиторий, преподавателей из БД;
- взаимодействие с внешней СУБД MySQL, хранящей данные о преподавателях, аудиториях, предметах.

Разработанное web-приложение облегчает составление расписания для студентов, оповещает студентов и преподавателей об изменениях в расписании.

Литература

- 1 Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. – Киев : Эксмо, 2014. – 480 с.
- 2 Современный учебник JavaScript [Электронный ресурс] – Режим доступа : <https://learn.javascript.ru/function-basics>. – Дата доступа : 10.03.2021.
- 3 Веллинг, Л. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – Москва : Вильямс, 2016. – 848 с.

А. С. Жиженский

*Науч. рук. О. М. Дерюжкова,
канд. физ.-мат. наук, доцент*

АРХИТЕКТУРА ANDROID-ПРИЛОЖЕНИЯ

Долгое время разработчики андроид-приложений не использовали какую-либо архитектуру в своих проектах. На конференции Google I/O 2017 компания представила Android Architecture Components – набор библиотек, которые призваны помочь с проектированием приложения и написанием более чистого кода [1].

Основной архитектурный паттерн, который используется в проектах, называется MVVM. MVVM – это шаблон проектирования архитектуры клиентских приложений, предложенный в 2005 году Джоном Госсманом. Он расшифровывается как Model-View-ViewModel. Основная его концепция заключается в отделении логики представления данных от бизнес-логики путём вынесения её в отдельный класс для более чёткого разграничения.

Model – абстрактное имя архитектурного слоя, который содержит логику, связанную с данными приложения. Он содержит классы данных, работы с API, работы с базами данных и т. д. Model подготавливает данные для ViewModel, но не должен знать о нём. Чаще всего представлен паттерном «репозиторий». Репозиторий позволяет абстрагироваться от конкретных подключений к источникам данных, с которыми работает программа, и является промежуточным звеном между классами, непосредственно взаимодействующими с данными и остальной программой. Данный паттерн обеспечивает обращение к данным как к коллекции объектов, что упрощает взаимодействие с ним.

View – это часть пользовательского интерфейса, основное назначение которого – отображение текущего состояния данных, которые видит пользователь. View не должна содержать логику или какие-либо данные. Она должна хранить только ссылку на ViewModel, откуда и получает данные для отображения. На андроиде View представляет собой Activity, Fragment или же Composable-функцию.

ViewModel – это объект, в котором описывается логика поведения View в зависимости от результата работы Model. Он хранит в себе текущее состояние, а также ссылку на одну или несколько моделей с данными, откуда и получает данные. ViewModel не должен ничего знать о View. Класс ViewModel является частью Android Architecture Components и предназначен для хранения и управления данными для View,

а также позволяет избавиться от проблемы, связанной с пересозданием UI-элементов при таких операциях, как поворот экрана. Он гарантирует, что не будет уничтожен на протяжении жизненного цикла приложения.

В совокупности данные архитектурные элементы приложения предоставляют разработчикам широкие возможности и удобство работы в команде, т. к. два человека могут одновременно работать над разными архитектурными слоями, простоту тестирования, а также более гибкий, простой в поддержке и читабельный код за счёт большего разграничения слоёв.

Литература

1 Guide to app architecture | Android Developers [Electronic resource]. – 2021. – Mode of access : <https://developer.android.com/jetpack/guide>. – Date of access : 24.04.2021.

В. И. Крылов

Науч. рук. О. М. Дерюжкова,

канд. физ.-мат. наук, доцент

КАК РАБОТАЕТ ВЕБ-САЙТ?

Каждый человек, который имеет доступ в интернет, может воспользоваться им, чтобы найти интересующую его информацию. Но где же эта информация хранится? Она располагается на специальных компьютерах или же веб-серверах. Зачастую это огромные сооружения, наполненные компьютерами, в которых хранится различная информация.

Так же веб-сервером можно назвать программное обеспечение, которое выполняет функции веб-сервера. Другими словами веб-сервер – это сервер, который принимает HTTP-запросы, чаще всего от браузеров, и выдает HTTP-ответ.

Для того чтобы клиенту, зачастую им является браузер, получить данные с сервера необходимо передать запрос на получение данных с определенным URL-адресом. Данными могут выступать HTML-страницы, изображения, файлы или любые другие данные, необходимые пользователю.

Большая часть всей этой информации находится в виде веб-сайтов, каждый из которых имеет свое уникальное доменное имя (адрес) в интернете.

Веб-сайт – это информация, представленная в виде одной или нескольких веб-страниц, которые имеют логическую связь. Например, интернет-магазин, у которого большое количество страниц, но все они связаны между собой по смыслу. Веб-сайт располагается на веб-сервере и имеет свое уникальное имя (адрес). Для того чтобы отобразить веб-сайт, необходимо воспользоваться специальными программами, которые называются браузерами (Google, Yandex). В зависимости от того, какое имя (адрес) сайта мы введем в поисковую строку, такую информацию он нам отобразит.

Веб-сайт может состоять из нескольких связанных между собой веб-страниц или же из одной страницы. Веб-страница представляет собой текстовый файл с расширением (.html), который содержит информацию в виде HTML-документа [1]. Он и определяет, в каком виде и в каком порядке информация будет отображена в окне браузера. В HTML-коде страницы содержатся только указания на расположение файлов. Если мы захотим подгрузить картинку или аудио, необходимо разместить эти файлы вместе с файлами проекта, к которым HTML-документ будет иметь доступ.

В настоящее время в интернете размещено огромное разнообразие веб-сайтов, и их число только растет. Это могут быть блоги, личные страницы, каналы, содержащие