Для интегрированного контроля уровня усвоения теоретического материала была разработана тестовая база по всему теоретическому материалу, включающая 392 вопроса. На рисунке 1 представлена диаграмма, которая отражает процент содержания вопросов различных типов в тестовой базе.

На основании разработанной и размещенной в СДО *Moodle* тестовой базы был разработан итоговый тест.

Заключение. Таким образом, в результате выполнения настоящей работы: изучена роль тестов в педагогическом процессе; изучены возможности использования элементов дистанционного обучения при обучении студентов дневной формы обучения; рассмотрена возможность системы СДО *Moodle* для создания интерактивных лекций; разработаны и внедрены 14 интерактивные лекции; разработана тестовая база вопросов по 21 темам курса «Термодинамика и статистическая физика», содержащая 392 вопроса.

Литература:

1. Галузо, И.В. Система дистанционного обучения MOODLE в рисунках и схемах / И.В. Галузо. – Витебск: ВГУ имени П.М. Машерова, 2013. – 28 с.
2. Роберт, И.В. Теория и методика информатизации образования / И.В. Роберт. – Москва: БИНОМ. Лаборатория знаний, 2014. – 398 с.

АРХИТЕКТУРА РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ С КЛИЕНТОМ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ

Кухарев А.А.[1], Сидоров А.А.[2],
студенты [1]5 курса и [2]4 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь
Научные руководители – Ермоченко С.А., канд. физ.-мат. наук; Новый В.В.

В настоящее время актуальным вопросом в сфере информационных технологий является разработка гибко масштабируемых приложений. Но что именно означает создание и управление масштабируемым приложением? На примитивном уровне это просто соединение пользователей с удаленными ресурсами через Интернет. Но при этом организуется доступ к ресурсам, которые физически рассредоточены на множестве серверов, что и обеспечивает масштабируемость приложения.

При разработке сложного масштабируемого приложения, время, потраченное на планирование архитектуры web-службы, может помочь в дальнейшем сэкономить время при внесении изменений в требования к приложению.

Целью работы является проектирование архитектуры и разработка клиент-серверного приложения для занятия спортом, таким как Workout. Для такого рода приложений пользователям удобнее пользоваться различными мобильными устройствами, что и определило выбор вида клиентского приложения.

Материал и методы. Материалом исследования является клиент-серверное приложение для занятия спортом. Клиентское приложение разработано под операционную систему Android. Оно взаимодействует с удалённым сервером через сеть Интернет. Структура серверной части приложения основывается на ряде современных технологий: Java 8, Spring Framework 4, jOOQ, реализация JAX-RS API на базе Jersey, PostgreSQL, пул соединений HikariCP. Клиентская часть также использует современные решения для построения надежного и многофункционального приложения: Java 8, DI Framework Dagger 2, RxJava/RxAndroid, Retrofit2, ButterKnife, Google Play Services, Picasso. В качестве методов исследования использовались объектно-ориентированное моделирование и анализ.

Результаты и их обсуждение. Современные архитектуры приложений построены на базе REST-сервисов. В данной работе также было решено создавать приложение на базе REST-сервисов.

REST (Representational state transfer) – это стиль архитектуры программного обеспечения для распределенных систем, таких как World Wide Web, который, как правило, используется для построения веб-служб [1]. Каждая единица информации однозначно определяется глобальным идентификатором, таким как URL. Каждый URL в свою очередь имеет строго заданный формат. Управление информацией сервиса по некоторому URL целиком и полностью основывается на протоколе HTTP с помощью методов: GET (получить), PUT (добавить, заменить), POST (добавить, изменить, удалить), DELETE (удалить).

Общая схема серверной части приложения приведена на рисунке 1. Основой архитектуры является популярный framework Spring MVC. Часть Repository отвечает за взаимодействие с базой данных, которая функционирует под управлением СУБД PostgreSQL. Для решения различных сопутствующих проблем с производительностью приложения и удобством разработки, были использованы специальные библиотеки, реализующие пул соединения и концепцию Object-Relational Mapping.

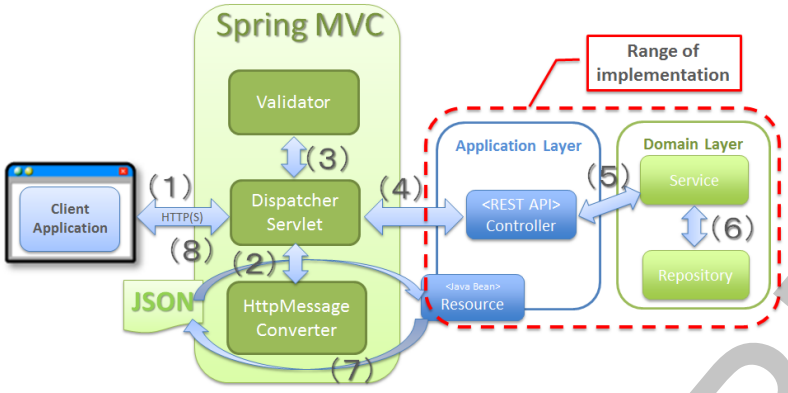

Рисунок 1 – Архитектура серверной части приложения

В ходе разработки распределенной инфраструктуры возникает потребность в параллельной разработке и взаимодействия между разработчиками клиентской и серверной частей. Для удовлетворения данной потребности было решено использовать OpenAPI спецификацию для описания, создания, использования и визуализации веб-сервисов REST. Это позволяет клиентским системам и документации синхронизировать свои обновления с изменениями на сервере.

Принципиальная схема работы клиентского приложения на базе платформы Android приведена на рисунке 2.

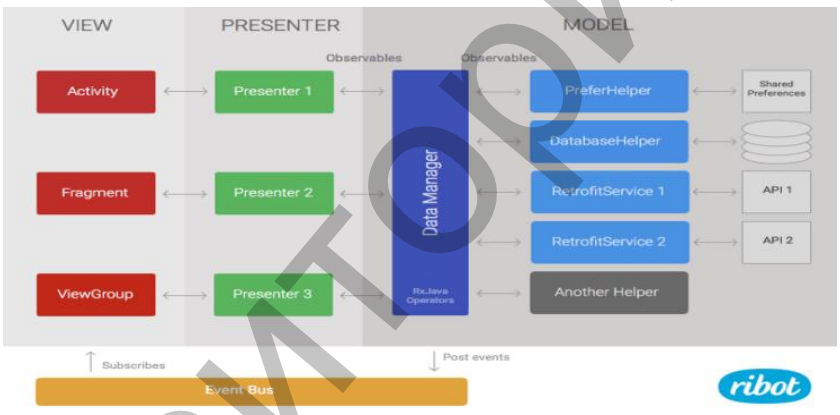

Рисунок 2 – Архитектура клиентской части приложения

Заключение. В результате работы было разработано приложение, которое демонстрирует работу различных современных технологий. Были разработаны модули на стороне клиента:

- модуль с картой площадок для тренировок;
- модуль для общения с сервером;
- модуль для работы со сторонними сервисами (VK API, Google+ API);
- модуль для составления программ тренировок.
- На стороне сервера разработаны:
- модуль для взаимодействия с базой данных;
- модуль для взаимодействия с клиентским приложением;
- промежуточный модуль бизнес-логики.

Разработанная архитектура позволяет гибко расширять функционал приложения и масштабировать его.

Литература:

1. Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000