

УДК 004.42

**РЕАЛИЗАЦИЯ ВЕБ-СИСТЕМЫ ДЛЯ ПРОВЕДЕНИЯ АДАПТИВНОГО ТЕСТИРОВАНИЯ****С.С. ВОЙТЕХОВИЧ***(Представлено: канд. физ.-мат. наук, доц. О.В. ГОЛУБЕВА)*

*Рассматривается один из вариантов реализации системы адаптивного тестирования в качестве веб-ресурса. Анализируются основные проблемы, возникающие при реализации подобной системы как веб-ресурса. Как простейший вариант реализации системы адаптивного тестирования разработано приложение, которое выполняет все поставленные цели.*

Адаптивное тестирование – разновидность тестирования, при которой порядок предъявления заданий (или трудность заданий) зависит от ответов испытуемого на предыдущие задания [1]. По сути, адаптивное тестирование является развитием идей простого тестирования и исправляет некоторые ее недостатки, такие как продолжительность тестирования при большом количестве заданий и быстрая потеря актуальности тестовых заданий при многократном использовании. Алгоритмы адаптивного тестирования, при правильной реализации могут сократить количество предоставляемых заданий из тестовой базы без потери точности оценивания. Платой за подобную эффективность стала сложность алгоритмов адаптивного тестирования и невозможность их использования без применения компьютера. Более того, усложняется процесс построения базы тестовых заданий, так как необходимо вводить классификацию сложности для заданий. Также адаптивное тестирование становится неэффективным при малом количестве заданий. Таким образом, создание адаптивного теста для одноразовой проверки знаний является не окупаемой по времени задачей.

Из сказанного выше следует, что система адаптивного тестирования должна обеспечивать много-разовое использование заданий и тестов, а также задавать критерии сложности для заданий. Более того, эффективность системы можно увеличить, если дать возможность одновременного проведения тестирования группой испытуемых. Следовательно, реализация системы адаптивного тестирования как настольного приложения без клиент-серверной архитектуры является задачей, не имеющей смысла, учитывая уровень развития современных сетевых технологий.

Следовательно, при выборе способа реализации серверного приложения встает выбор – реализовать настольное клиент-серверное приложение, либо реализовать веб-приложение. Каждый из подходов имеет свои достоинства и недостатки. В частности, настольное клиент-серверное приложение позволяет более жестко контролировать условия проведения тестирования, но требует установку приложений системы как на серверную, так и клиентскую часть. Это требует дополнительных усилий от конечного пользователя и усложняет портирование приложения на разные платформы. Касательно веб-приложения, можно наблюдать противоположную ситуацию – клиентский интерфейс приложения можно запустить на любом устройстве, которое имеет веб-браузер, а установка требуется только на серверной части системы. При этом сложно контролировать условие проведения теста посредством веб-интерфейса. Более того, в веб-интерфейсе сложнее реализовать некоторые интерактивные элементы, которые в интерфейсе настольного приложения реализуются достаточно легко. Но отсутствие необходимости полного портирования приложения на другие платформы (в том числе и мобильные) перекрывает многие недостатки веб-архитектуры. Данный факт стал решающим при выборе веб-архитектуры для разрабатываемой системы.

Для разработки системы было решено использовать язык программирования Java в связке с фреймворком Spring. Данная связка повсеместно используется для построения корпоративных веб-систем и хорошо себя зарекомендовала. В качестве системы управления базой данных (СУБД) была выбрана MySQL. Это бесплатная СУБД с достаточным функционалом для решения поставленной задачи. Более того, она достаточно проста в управлении, что облегчает процесс установки серверной части приложения. В качестве сервера веб-приложений была использована Apache Tomcat как наиболее популярный вариант. Более того, Apache Tomcat имеет версии практически для всех современных операционных систем.

Разрабатываемую систему было решено разделить на две части: модуль организации тестов и модуль тестирования. В первую часть включены все средства взаимодействия составителя тестов и испытуемых, в том числе модули регистрации составителей тестов и испытуемых, модуль организации курсов для группирования тестов и модуль доступа к тестам. Вторая часть системы включает модуль создание тестовых заданий и модуль их отображения. Разработанный модуль создания тестовых заданий поддерживает создание следующих типов заданий:

- одиночный ответ;
- множественный ответ;
- точный ответ.

Малое количество типов тестовых заданий компенсируется расширяемой архитектурой модуля создания заданий. Текст вопросов было решено хранить в HTML формате, что позволило менять форматирование текста вопроса. Для поддержки подобной возможности была использована библиотека `nicEdit`, написанная на JavaScript. Данная библиотека позволила добавить в веб-интерфейс полнофункциональный текстовый редактор (рис. 1).

### Текст вопроса



Рисунок 1. – Текстовый редактор из библиотеки `nicEdit`

При создании теста есть возможность задать перечень настроек, характеризующих работу адаптивного алгоритма. В разработанной системе реализуется алгоритм процентного соотношения, в рамках которого анализируется некоторое количество последних ответов испытуемого. В зависимости от процента правильных ответов принимается решение: повышать сложность заданий, понижать или оставлять неизменной. Значения, задающие количество анализируемых ответов, граничные значения для изменения уровня, а также ограничение времени на тест, задается на форме создания теста (рис. 2).

Рисунок 2. – Форма создания нового адаптивного теста

Полученная в итоге система не является самым оптимальным решением для проведения адаптивного тестирования, но она допускает расширение. Работа по дальнейшему улучшению может быть направлена на увеличение количества доступных адаптивных алгоритмов, внедрение методов машинного обучения для регулирования сложности вопросов и увеличение количества типов тестовых заданий. Но как простейший вариант реализации системы адаптивного тестирования разработанное приложение выполняет все поставленные цели.

### ЛИТЕРАТУРА

1. Словарь терминов [Электронный ресурс]. – Режим доступа <http://www.profcareer.ru/lib/dicty.php>. – Дата доступа: 25.09.2016.