

ся 1000–1500 минут, т.е. 16,67–25 часов, что составляет 2-3 рабочих дня специалиста, в то время как при проведении психодиагностики посредством автоматизированной методики психолог затратит время лишь на сбор информации, получив результаты автоматически.

3. Отсутствие необходимости диагностирования респондентов посредством платных электронных аналогов.

4. Получение исследователем результатов диагностики респондентов в наглядной и систематизированной форме.

Отрицательные стороны автоматизации психологического инструментария:

1. Специалисту-психологу необходимо затратить некоторое время на автоматизацию методики (анкеты, теста, опросника и т.п.), количество которого зависит от его уровня компьютерной грамотности.

2. Отсутствие в некоторых как государственных, так и частных учреждениях возможности произвести психодиагностику респондентов с помощью компьютерных технологий: не все учреждения (например, учреждения образования, расположенные в сельской местности) обладают достаточной материально-технической базой.

В заключение отметим, что, несмотря на определенные недостатки, достоинства автоматизации психодиагностического инструментария не вызывают сомнений в целесообразности такой деятельности. Поэтому современному психологу – квалифицированному специалисту в своей сфере деятельности – необходимо не только осуществлять психологическое консультирование, коррекцию, профилактику, но также использовать при сборе и обработке результатов респондентов автоматизированный диагностический инструментарий.

#### **Литература**

1. Практическая психодиагностика. Методики и тесты: учеб. пособие / сост.д.Я. Райгородский. – Самара: БАХРАХ-М, 2001. – 672 с.

## **ОСОБЕННОСТИ РАЗРАБОТКИ ВЕБ-СЕРВИСА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ JAVA**

*И.Н. Вирич, Минский университет управления, инженерно-информационный ф-т, магистрант  
Научный руководитель: С.Г. Шульдова, к.т.н., зав. кафедрой*

Большинство современных веб-сервисов при разработке клиентской части используют следующий стек технологий – язык разметки HTML5, каскадные таблицы стилей CSS3, язык программирования JavaScript.

Для разработки серверной части приложения (бэкэнд) выбор технологий крайне велик. Это PHP, .NET, JAVA, JS, Ruby, Python и др. Для каждой технологии существует множество фреймворков. Все они имеют свои особенности и служат для разных целей.

Использование фреймворков для разработки веб-приложения конкретной предметной области имеет ряд преимуществ:

- возможность чисто описать логику взаимодействия объектов предметной области;
- возможность более гибко изменять логику работы приложения;
- чистый код;
- логика отображения, управления данными и бизнес-логика разделены между собой.

Единственный недостаток – необходимо больше времени для создания проекта.

Java является одним из самых популярных языков для бэкэнд-разработки современных веб-приложений. С Java и основанными на нем фреймворками разработчики могут создавать масштабируемые веб-приложения для широкого круга пользователей [1]. Только под Java-платформу существует более 50 фреймворков. Наиболее популярные – Spring Framework, JSF, Struts, GWT, Play Framework, Grails, Wicket, Tapestry.

Spring Framework (или, коротко, Spring) – это универсальный фреймворк с открытым исходным кодом для Java-платформы. Несмотря на то что Spring Framework не обеспечивал какую-либо конкретную модель программирования, он стал широко распространенным в Java-сообществе. Spring Framework предоставляет большую свободу Java-разработчикам в проектировании, кроме

того, он предоставляет хорошо документированные и легкие в использовании средства решения проблем, возникающих при создании приложений корпоративного масштаба.

Spring Framework может быть рассмотрен как коллекция меньших фреймворков, или фреймворков во фреймворке. Большинство этих фреймворков может работать независимо друг от друга, однако они обеспечивают большую функциональность при совместном их использовании. Эти фреймворки делятся на структурные элементы типовых комплексных приложений [1]:

- фреймворк доступа к данным;
- фреймворк управления транзакциями;
- фреймворк MVC;
- фреймворк аутентификации и авторизации;
- фреймворк работы с сообщениями;
- тестирование.

Таким образом, при разработке веб-сервисов на основе Java-технологий целесообразно использовать фреймворк Spring. Его применение оправданно как для малых, так и для крупных проектов.

#### **Литература**

1. Самые востребованные языки программирования в 2015-м [Электронный ресурс]. – Режим доступа: <http://dev.by/lenta/main/top-15-samyh-vostrebovannyh-yazykov-programmirovaniya-v-2015-m>. – Дата доступа: 22.03.2015.

## **ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ «ЛИЧНЫЙ КАБИНЕТ СТУДЕНТА»**

*К.В. Дементьев, Минский университет управления, инженерно-информационный ф-т, магистрант  
Научный руководитель: С.Г. Шульдова, к.т.н.*

С каждым годом растут продажи мобильных устройств. Пользователи становятся все более и более требовательны к приложениям и сервисам, встроенным в различные устройства. Это говорит о необходимости повышенного внимания к интерфейсам таких мобильных приложений.

Мобильные приложения – это программные продукты, ориентированные и разработанные именно для мобильных устройств, таких как телефоны (смартфоны), планшеты, навигаторы, медиаплееры и др. Они помогают пользователю в решении большинства прикладных задач.

Интерфейсы мобильных приложений играют важнейшую роль в мобильной разработке и проектировании, потому что конечный пользователь в первую очередь ориентируется на то, что он видит своими глазами. Разработчики стараются привлечь потенциальных покупателей своего приложения красочностью и интуитивностью конкретного интерфейса.

Принципы создания привлекательных и удобных интерфейсов основываются на ряде негласных правил, которые разработчик должен соблюдать в процессе создания интерфейса. В первую очередь, это сосредоточенность на том, что по-настоящему необходимо пользователю. Разработчик должен четко понимать, что он делает и для какой конкретной аудитории. Интерфейс должен стать «личным», т.е. надежным, дружелюбным и веселым (по возможности). Во-вторых, дизайн должен увлекать и заставлять забывать о времени. Примеры: «Angry Birds», «Twitter». Интерфейс может быть смелым, но достаточно понятным и привычным. Кроме того, приложение не должно быть расточительно в аппаратных ресурсах устройства и в то же время иметь высокую скорость отклика [1].

При разработке интерфейса для мобильного приложения «Личный кабинет студента» следует учесть такие необходимые факторы, как доступность, надежность, понятность, информативность. Кроме того, в силу ограниченности размера дисплея устройства интерфейс должен отвечать всем функциональным требованиям, заявленным на сайте вуза, но при этом иметь компактность и аккуратность.

Мобильный интерфейс приложения планируется разбить на несколько основных составляющих (окон). Главное окно будет иметь поле для авторизации с последующим автозаполнением для ускорения работы. Далее пользователю будет представлена возможность выбора следующей функциональной возможности, будь то доступ к оценкам или же расписанию на конкретный день. Все возможности приложения должны быть в шаговой доступности пользователя. Все управляющие кнопки должны находиться на расстоянии, не превышающем длину большого пальца.