

запрашивает выполнение некоторых действий у сервера, а сервер принимает решения о выполнении или не выполнении запрошенного действия в зависимости от предоставленных клиентом данных.

В данном web-приложении реализованы следующие возможности: регистрация и авторизация пользователей, создание, изменение и удаление проектов, создание и редактирование итераций в рамках проекта. Для каждой из итераций была реализована Scrum-доска, которая содержит следующие статусы: «Необходимо сделать», «В процессе», «Выполнено». Между этими статусами перемещаются карточки с заданиями. Также карточки в рамках итерации можно создавать, редактировать или удалять. Разработан бэклог проекта, журнал оставшейся работы, которую команде необходимо выполнить. Реализована возможность добавлять пользователей на проект с целью совместной разработки.

Клиентская часть приложения разработана с использованием следующих инструментов: фреймворк Angular, Bootstrap и другие. При разработке серверной части приложения были использованы среда Node.js и фреймворк Express.js, используя которые были реализованы REST-сервисы. В качестве базы данных использовалась документно-ориентированная система управления базами данных Mongo DB.

В. В. Сыч

Науч. рук. **Н. Б. Осипенко,**

канд. физ.-мат. наук, доцент

РАЗРАБОТКА КОМПОНЕНТОВ КЛИЕНТСКОЙ ЧАСТИ ПРИЛОЖЕНИЯ «ПОДБОР РЕШЕНИЙ, СООТВЕТСТВУЮЩИХ ПРЕДНАЗНАЧЕНИЮ ЧЕЛОВЕКА»

Данная работа является частью проекта формирования рекомендаций и индивидуального прогноза при «Подборе решений, соответствующих предназначению человека». Описываемое приложение посвящено подбору рекомендаций по выбору профессии и сфер деятельности. Для этого разработанное клиент-серверное приложение предоставляет возможность пользователю прохождения тестов, выполняет обработку полученных данных, а также выдает результаты. Интерфейс приложения создан помощью фреймворка «Bootstrap». В процессе разработки приложения были созданы тесты, предназначенные для выявления индивидуально-типологических различий и особенностей темперамента человека. Тесты представляют из себя инструкцию в виде картинки или текста с набором вариантов ответов.

Описываемая программная система реализована на основе конструкционного шаблона MVC, основанного на четком разделении ответственности за различный функционал приложения. После выбора теста вызывается функция *Start()*, модель получает данные с сервера из БД и отправляет их через контроллер на вью, где они отображаются с помощью директивы *ng-repeat*. После ответа пользователем на вопросы теста вызывается функция *itemSelected(answer)*, которая передает данные о выборе в контроллер. Для сохранения результатов вызывается функция *saveResults()* и контроллер обрабатывает полученные ответы.

Результаты тестирования обрабатываются согласно внутренним алгоритмам и используются для уточнения рекомендаций по выбору профессии. Так, например, из полученных ответов в тесте «Предпочтения школьных предметов» формируется матрица $Disc(i, k)$, $i=1,17$, $k=1,2$. Для обработки данных строится вектор $t(j)$, $j=1,18$ на основе статичной матрицы $School(i, k)$, $i=1,17$, $k=1,18$ со значениями весов для соотношения школьных предметов и сфер деятельности. После этого полученные данные о результатах прохождения теста путем аддитивной свертки с соответствующим весом соотносятся со сферами деятельности.

Бета тестирование разработанного клиент-серверного приложения показало перспективность его дальнейшей разработки путем увеличения количества и четкости формулировок тестов, а также уточнения коэффициентов аддитивной свертки, возможного за счет обратной связи с тестируемыми, состоявшимися в определенных профессиях.