(данные таблицы), которые являются основой для построения вектора глобальных приоритетов с учетом веса каждой из составляющих стабильности. Удельные веса составили: финансовая составляющая стабильности – 0,48, экономическая – 0,27, организационная-0,04, инвестиционная-0,15 и инновационная-0,07. Индекс согласованности для построенной матрицы расчета глобального вектора приоритетов составил 8%, отношение согласованности – 7%, что говорит об отсутствии нарушения численной и транзитивной согласованности суждений.

Таким образом, на основании применения методики, разработанной автором, мы пришли к выводу, что из трех анализируемых предприятий целесообразно в качестве участника кластера выбрать ООО "ПСК «ЕвроДом", на втором месте - ОАО "Строймонтаж". При этом ООО "Стройкомплект" является аутсайдером в данном рейтинге. В анализе могут участвовать неограниченное количество предприятий, для которых в свободном доступе есть бухгалтерская отчетность. Метод оценки стабильности с применением процедуры анализа иерархий, на взгляд автора, довольно универсален и легко применим на практике. Данный метод дает еще один гибкий инструмент по определению наиболее предпочтительных предпринимательских субъектов для участия в кластере.

ЛИТЕРАТУРА

1. Баулина, О. А., Клюшин, В. В. Концептуальные основы кластерного развития региона [Электронный ресурс] / О. А. Баулина, В. В. Клюшин; М-во образования и науки Рос. Федерации, Волгогр. гос. архит.-строит. ун-т. — Электронные текстовые и графические данные (4,3 Мбайт). — Волгоград : ВолгГАСУ, 2015. — Научное электронное издание. — Систем. требования: PC 486 DX-33; Microsoft Windows XP; Internet Explorer 6.0; Adobe Reader 6.0. — Официальный сайт Волгоградского государственного архитектурно- строительного университета. Режим доступа: http://www.vgasu.ru/publishing/on-line/ — Загл. с титул. экрана.

2. Саати, Т., Кернс, К. Аналитическое планирование. – М.: Радио и связь, 1991. – 224 с.

3. Saaty, Thomas L. and Vargas, G. (1982). The Logic of Priorities. – Boston: Kluwer-Nijhoff.

РАЗРАБОТКА WEB-ОРИЕНТИРОВАННОГО ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ПРОЕКТАМИ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИЙ REST И SPA

А.С. Бенц, В.С. Шахов
(г. Томск, Томский политехнический университет)

DEVELOPMENT OF WEB-BASED PROJECT MANAGEMENT SYSTEM USING REST AND SPA

A.S. Bents, V.S. Shakhov
(Tomsk, Tomsk Polytechnic University)

This article gives information about the implementation of web-based project management system using ReactJS and Flux architecture with RESTful back-end.

Keywords: project management system, REST, flux, ReactJS, single page application

Введение. Система управления проектами – это программный продукт, позволяющий осуществлять контроль за процессом выполнения проекта, отдельных его частей, получать систематизированную информацию о его составных частях. Проект

как объект состоит из нескольких взаимодействующих компонент, основными из которых являются:

- Пользователи – лица, принимающее участие в работе над проектом;
- Задания – отдельные работы, которые нужно завершить для выполнения проекта.

У каждой из компонент имеется определенный набор параметров, например, задания могут находиться на разных этапах выполнения, иметь различные сроки выполнения и т.д.

Подобные системы помогают планировать и организовывать различные этапы работы над проектами, а также осуществлять контроль их выполнения. В частности, их можно внедрить в учебный процесс с целью упрощения взаимодействия научного руководителя и студента.

В данной работе серверная часть приложения реализована в виде RESTful сервиса с использованием СУБД MySQL, а клиентская часть в виде одностраничного приложения (Single Page Application) с использованием библиотеки ReactJS и архитектуры Flux.

База данных. В ходе работы над приложением была разработана база данных, содержащая следующие сущности:

- User – для хранения данных о пользователях;
- Project – для хранения информации о проектах;
- Project member – для хранения информации об участниках проекта;
- Task – для хранения информации о заданиях в проекте;
- Tasklist – для хранения данных о списках заданий, в которые можно группировать задания.

RESTful сервис. Серверная часть приложения реализована используя архитектуру REST. В данной архитектуре клиент совершает HTTP-запрос на определенный адрес, называемый конечной точкой (endpoint) и сервер, в зависимости от используемого метода, возвращает данные или создает новую запись[3]. Например, GET /users/ вернет список всех пользователей, GET /users/1 вернет информацию о пользователе с id равным 1, POST /users/ создаст нового пользователя. При этом данные, возвращаемые сервером, представляются в виде JSON или XML, и на клиентскую часть возлагается отображение этих данных.

В системе поддерживаются следующие конечные точки:

- GET /project/:id – получение информации о проекте с указанным id;
- POST /project – создание нового проекта;
- GET /tasks/:id – получение информации о заданиях проекта;
- POST /tasks/:id – создание нового задания для проекта;
- GET /members/:id – добавление нового участника в проект;
- POST /members/:id – добавление нового участника в проект;
- GET /settings/:id – получение настроек проекта;
- POST /settings/:id – изменение настроек проекта.

Клиентская часть. Клиентская часть реализована в виде SPA web-приложения с использованием ReactJS и архитектуры Flux. ReactJS – JavaScript библиотека для построения пользовательских интерфейсов [1]. Принцип работы библиотеки состоит в композиции интерфейса в виде отдельных компонентов (Components), которые могут содержать другие компоненты. При этом ReactJS следит за изменениями и изменяет состояние компонентов.

Flux – архитектура для создания клиентской части web-приложений[2]. Данная архитектура состоит из трех частей – Dispatcher, Stores, View. Также одной из главных особенностей flux является однонаправленный поток данный (unidirectional data flow). Когда пользователь взаимодействует с представлением (View), представление

распространяет действие через центральный диспетчер (Dispatcher), который в свою очередь передает данные о действии соответствующим хранилищам (Store), которые содержат данные приложения и бизнес-логику. Далее хранилища обновляют данные в соответствии с действием, которое было активировано и сообщают о том, что данные изменились, чтобы различные представления были обновлены. Данный процесс представлен на рис. 1:

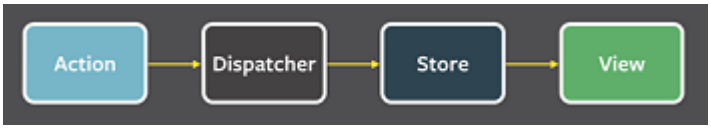

Рисунок 1 - Поток данных в архитектуре Flux

Результаты. Разработанная система имеет вид, представленный на рис. 2:

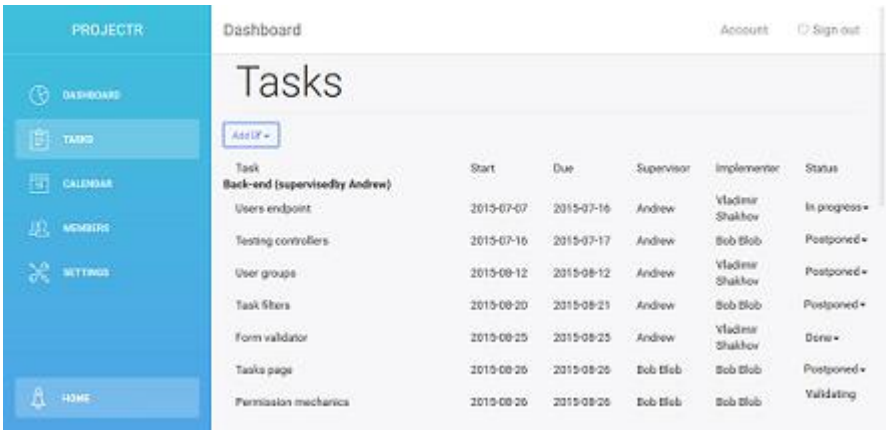

Рисунок 2 - Внешний вид разработанной системы

Интерфейс системы позволяет пользователю создавать профили и входить в систему. Используя созданный профиль, пользователь может создавать проекты, добавлять в него участников, и становиться участником других проектов. В рамках проекта присутствует возможность добавления заданий и назначение исполнителей и руководителей. Руководители могут следить за выполнением заданий и, по ходу их выполнения, менять их статус. Также в разработанной системе присутствует возможность удобного просмотра информации о заданиях в виде календаря.

Заключение. Результатом выполнения проекта является разработка веб-ориентированной системы управления проектам в виде RESTful-сервиса и SPA-приложения, реализованного с использованием библиотеки ReactJS и архитектуры Flux.

ЛИТЕРАТУРА

1. Документация ReactJs [электронный ресурс]: http://facebook.github.io/react/, режим доступа – свободный;

2. Архитектура Flux [электронный ресурс]: https://facebook.github.io/flux/, режим доступа – свободный;

3. L. Richardson, S. Ruby. RESTful Web service // O'Reilly Media. – 2000. – С. 407-416

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СТРАТЕГИЙ ВНЕДРЕНИЯ КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ ПРОЕКТАМИ