

В. Н. Копачев, Е. М. Березовская

СОЗДАНИЕ WEB-ПРИЛОЖЕНИЯ ПО СОПРОВОЖДЕНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ SPRING FRAMEWORK

Статья посвящена проектированию и разработке веб-приложения с использованием технологии Spring Framework. Были получены следующие результаты: спроектирована архитектура приложения, на основе AngularJS, AJAX и JSP разработан графический интерфейс, организации уровня доступа к данным и бизнес-логике осуществляется на базе Spring Framework, разработана структура базы данных, взаимодействие с которой осуществляется на основе Hibernate Framework. Приложение позволяет выявлять и классифицировать неполадки в процессе разработки и сопровождения программного обеспечения, отслеживать работу команды и распределять задания.

В настоящее время существует множество компаний занимающихся разработкой программного обеспечения. Каждая из таких компаний имеет различные команды разработчиков, которые занимаются разработкой и тестированием разрабатываемого программного обеспечения. Для упрощения процесса разработки и сопровождения программного обеспечения необходимо автоматизировать эти процессы, что и послужило основной целью разработки данного веб-приложения.

Качество выпускаемых программных средств – это одна из главных задач, которая стоит перед индустрией программных средств. В настоящий момент имеется достаточно много специальных инструментов обеспечения качества, которые в зависимости от назначения делятся на:

- системы управления процессом разработки: системы трекинга задач и системы управления проектами;
- системы отслеживания дефектов и запросов на изменение;
- средства тестирования.

Разработанное веб-приложение предназначено для трекинга разрабатываемого программного обеспечения на протяжении всего жизненного цикла разработки.

Приложение создавалось в среде разработки Eclipse, которая написана на Java, поэтому является платформо-независимым продуктом, за исключением библиотеки SWT, которая разрабатывается для всех распространённых платформ. Библиотека SWT используется вместо стандартной для Java библиотеки Swing [1, 2].

Eclipse содержит плагин для работы с системой параллельных версий (CVS – Concurrent Versions System) для управления исходным кодом. Плагин Team (Команда) соединяется с CVS-сервером, позволяя членам команды разработчиков работать с набором файлов, содержащих исходные тексты, не вмешиваясь в изменения, вносимые другими. Плагины, которые поддерживаются и распространяются Фондом Eclipse, можно найти на Web-сайте Eclipse. Наиболее полный список имеющихся плагинов доступен на странице Eclipse Plug-in Central, которая служит каталогом плагинов.

Веб-приложение реализует технологию «клиент-сервер», т.к. состоит из клиентской и серверной частей.

Клиентская часть реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Для разработки клиентской части приложения использовались такие технологии как: JS, Ajax, AngularJS, CSS и HTML.

Серверная часть получает запрос от клиента, выполняет необходимые операции, после этого формирует веб-страницу и отправляет её клиенту по сети с помощью протокола

HTTP. Для разработки серверной части приложения использовались такие технологии как: Spring MVC, Java, Hibernate, Spring Security, Log4J, Tomcat и Derby.

Используя механизм отображения при выборе контроллера для обработки запроса, определяющих порядок вывода результатов, фреймворк Spring MVC обеспечивает слабую связанность между тем, как будет обрабатываться запрос, и тем, как будут представлены результаты. Это делает Spring более привлекательным, по сравнению со многими другими веб-фреймворками MVC.

Фреймворк Spring Security обеспечивает безопасность с двух сторон. Для ограничения доступа и обеспечения безопасности на уровне запросов в Spring Security используются сервлет-фильтры. А для обеспечения безопасности на уровне вызовов методов с использованием Spring AOP фреймворк Spring Security предоставляет объекты-обертки, гарантирующие авторизацию пользователей.

Hibernate – библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного проецирования (object-relational mapping – ORM). Она представляет собой свободное программное обеспечение с открытым исходным кодом (open source), распространяемое на условиях GNU Lesser General Public License. Данная библиотека предоставляет лёгкий в использовании каркас (фреймворк) для отображения объектно-ориентированной модели данных в традиционные реляционные базы данных.

Разработанная база данных состоит из 10 таблиц с соответствующими связями между ними:

- «Position» – информация проектных позиций.
- «Employee» – информация о сотрудниках.
- «Project» – информация о проекте.
- «Task» – информация о заданиях.
- «Status» – информация о статусе задания/проекта.
- «Role» – роль пользователя.
- «Member» – информация об участниках проекта.
- «Assignment» – информация о назначенных заданиях.
- «Activity» – информация об активности пользователей.
- «Attachment» – информация о прикрепленных файлах.

Для того, чтобы начать работу с данным приложением достаточно зарегистрировать пользователя в системе, определить его. При запуске программы клиента (браузера) на экране появляется окно для заполнения информации о пользователе (логин и пароль) (рисунок 1).

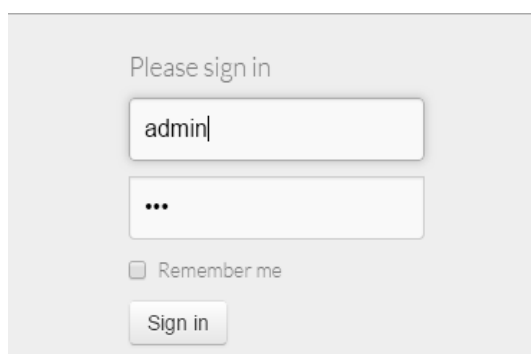


Рисунок 1 – Форма аутентификация пользователей

Если подключение пользователя завершено успешно, то появляется главная страница приложения (рисунок 2).

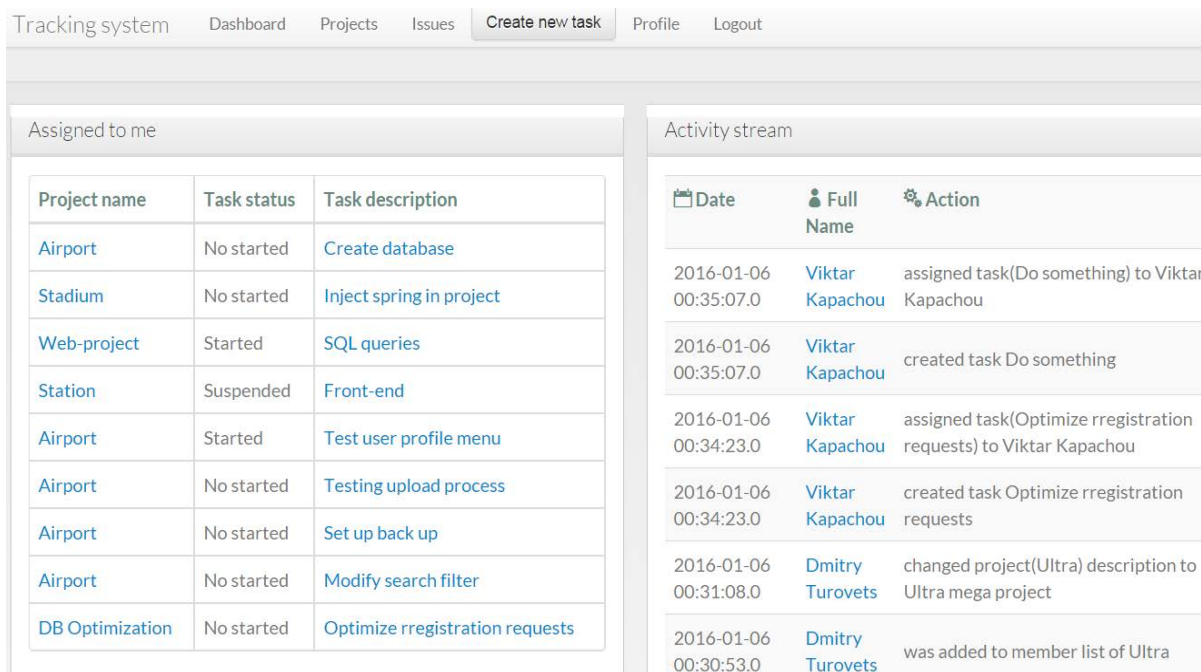


Рисунок 2 – Главное окно приложения

На главной странице приложения можно увидеть элементы навигации, последнюю активность пользователей данного приложения и закрепленные за текущим пользователем задания (рисунок 3).



Рисунок 3 – Главное меню приложения

Далее необходимо создать проект, заполнив всю необходимую для этого информацию. Для того чтобы создать задачу, необходимо выбрать проект, за которым будет закреплена текущая задача, и заполнить ряд предоставленных пользователю полей (тему, тип, приоритет, цель задачи).

Из главного меню приложения пользователь может получить доступ к информации своего профиля (рисунок 4), списку доступных проектов (рисунок 5) и закрепленных за ним задач (рисунок 6).

На панели доступных проектов отображается список доступных проектов текущего пользователя. Если пользователь имеет определенный уровень доступа, то он может производить какие-либо действия над проектами (изменить описание, статус проекта или добавить новый проект).

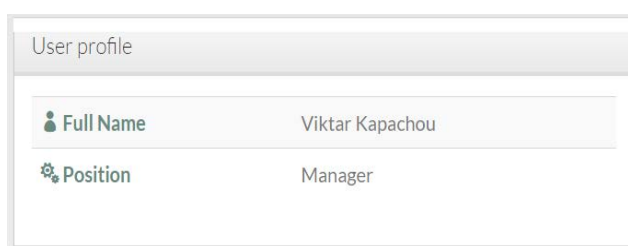


Рисунок 4 – Профиль пользователя

Project list			
Project name	Project description	Status	Team
Airport	Access system for airport	No started	Members
Stadium	Security system	Started	Members
Web-project	Test project	Suspended	Members
Station	Shedule of trains	Finished	Members
DB Optimization	Optimize old version of db	Suspended	Members
Ultra	Ultra mega project	No started	Members

Рисунок 5 – Доступные проекты

Assigned to Viktor Kapachou		
Project name	Task status	Task description
Airport	No started	Create database
Stadium	No started	Inject spring in project
Web-project	Started	SQL queries
Station	Suspended	Front-end
Airport	Started	Test user profile menu
Airport	No started	Testing upload process
Airport	No started	Set up back up
Airport	No started	Modify search filter
DB Optimization	No started	Optimize rregistration requests

1 2 Next

Рисунок 6 – Закрепленные задачи

Задача может быть расширена прикрепленным материалом – документами, изображениями или комментариями. Пользователь может добавлять, удалять и редактировать созданные проекты и задачи, изменять статус, например, из «открыт» в «закрыт», добавлять новых участников проекта и управлять уровнем доступа добавленных пользователей. Также каждый пользователь имеет доступ к личному профилю, в котором указывается личная информация, проектная активность, роли на проектах и закрепленные за пользователем задачи. Любые изменения в приложении, протоколируются в журнал, который отображается всем пользователям, для того чтобы пользователи могли видеть происходящую активность.

Для работы с данным приложением нет необходимости устанавливать какое-либо программное обеспечение, так как оно может осуществляться в любом браузере.

Приложение позволяет выявлять и классифицировать неполадки в процессе разработки и сопровождения программного обеспечения, отслеживать работу команды и распределять задания. Его функционал может быть расширен и усовершенствован в перспективе до более высокого уровня, также возможна реализация взаимодействия данного приложения со сторонними сервисами, с целью дальнейшего повышения информативности, привлекательности и удобства приложения.

Литература

- 1 Блинов, И. Н. Java. Промышленное программирование / И. Н. Блинов, В. С. Романчик: УниверсалПресс, 2007. – 704 с.
- 2 Eckel, В. Thinking in Java / В. Eckel. – New York, 2011. – 1057 p.

УДК 004.7

А. С. Кулешов, М. И. Жадан

РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ ДЛЯ ТЕЛЕФОНОВ ПОД ОС ANDROID

В статье речь идёт о связи сервера с клиентским приложением. Для этих целей создано клиент-серверное приложение для телефонов под операционную систему Android. Приложение представляет собой программу-контейнер с контентом различного вида и состоит из нескольких Activity, которые включают в себя компоненты Fragments. Обмен данными с сервером осуществляется через компонент AQuery. Навигация по приложению осуществляется с помощью NavigationView и TabLayout. Приложение успешно залито на play.google.com.

Для описания связи сервера с клиентским приложением создано приложение для телефонов под операционную систему Android. Приложение представляет собой программу-контейнер с контентом различного вида: обои для рабочего стола и телефонных контактов, рингтоны, игры и другое. В дальнейшем на основе приложения будет создан конструктор, с помощью которого пользователи сами смогут создавать своё собственное приложение (дизайн, наполнение, некоторый выборочный функционал).

Со стороны сервера приложение имеет веб-сайт, на который пользователи могут загружать соответствующий контент в свои аккаунты. Обмен данными между клиентом и сервером основан на работе с json-файлами [1]. В клиентском приложении за работу с json отвечает компонент AQuery. Пример работы с json на стороне клиента:

```
final AQuery aq = new AQuery(this);
final String url =
getString(R.string.url_wallpaper_rating) + "?limit=10";
aq.ajax(url, JSONArray.class, new Ajax-
Callback<JSONArray>() {
    @Override
    public void callback(String url, JSONArray json,
AjaxStatus status) {
        if (json != null) {
            try {
                allWallpaperCategoriesList = JsonParserWallpaper-
Categories.parseJSON(json);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
});
```

Со стороны сервера придёт соответствующий запросу ответ. В нашем случае он будет выглядеть соответствующим образом: