

Заключение. Независимо от того, насколько большим и информативным будет приложение, необходимо грамотно и ответственно отнестись к его созданию. Ценность работы заключается в реализованной возможности обобщения опыта разработки приложений как видов программного обеспечения и интернет-магазинов как отдельного класса ресурсов, интерес к которым в настоящее время неуклонно растет.

Литература

1 Бэнкс, А. React и Redux : функциональная веб-разработка / А. Бэнкс, Е. Порселло. – Санкт-Петербург : Питер, 2018. – 336 с.

2 Официальный сайт MockAPI [Электронный ресурс]. – Режим доступа: <https://mockapi.io>. – Дата доступа : 25.04.2022.

УДК 316.74:004.946

Д. Н. Дуброва

РАЗРАБОТКА ПРИЛОЖЕНИЯ КОМАНДНОГО УПРАВЛЕНИЯ ДЕЛАМИ НА ЯЗЫКЕ JAVA В ПЛАТФОРМЕ ADOBE EXPERIENCE MANAGER

Статья посвящена описанию созданного прототипа клиент-серверного приложения командного управления делами на языке Java в платформе Adobe Experience Manager. Для базы данных использовалась PostgreSQL. Для клиентской части использовались HTML, JavaScript, jQuery, CSS, LESS. Вся разработка велась в интегрированной среде разработки IntelliJ IDEA. Для упрощения разработки приложения были использованы библиотеки, которые предоставляет АЕМ: Sling, OSGi, JCR. Приведены результаты апробации в коллективной разработке.

С возникновением интернета появилась возможность работать в интернете над какими-то проектами и появилась необходимость управлять проектом и его командой. Приложение командного управления делами (ПКУД) предоставляет возможность лидерам проектов организовать работу над ними в легкодоступной форме. В ПКУД лидеры могут создавать задания, назначать других пользователей на эти задания, управлять всеми заданиями и пользователями. Лидеры проектов могут автоматизировать процессы и решать проблемы, с которыми они столкнулись или могут столкнуться, работая над проектом какими-то другими способами.

Аналогами ПКУД являются такие приложения, как Jira, ClickUp, Binfire, Trello. Однако все эти аналоги имеют некоторые минусы, которые не подходят для небольших команд: требуется ежемесячная оплата каждого пользователя (если не требуется, то только с большими ограничениями для пользователей), сложный интерфейс для обычных пользователей, много бесполезных функций для пользователей, требуется большое количество ресурсов, чтобы поставить к себе на сервер. Разработка описываемого веб-приложения велась с целью устранения этих минусов.

Разработка клиент-серверного веб-приложения командного управления делами выполнена на языке Java в платформе Adobe Experience Manager. Для базы данных использовалась PostgreSQL. Для клиентской части использовались HTML, JavaScript, jQuery, CSS, LESS. Вся разработка велась в интегрированной среде разработки IntelliJ IDEA. Для упрощения разработки приложения были использованы библиотеки, которые предоставляет Adobe Experience Manager (AEM) [1]: Sling [2], OSGi, JCR. Проект был собран с помощью Maven.

Основными задачами разработанного веб-приложения являются: создание удобного и красивого интерфейса; адаптация под мобильные устройства; возможность регистрации пользователей; возможность менять роль пользователя; возможность создавать задания; возможность назначать пользователей на задания; возможность просматривать и сортировать задания; смена статуса задания; удаление заданий.

Для решения поставленных задач реализованы четыре основные части приложения: система пользователей (регистрация, вход, выход, роли); система заданий (создание, удаление, редактирование); система взаимодействия пользователей с заданиями; пользовательский интерфейс с использованием инструментов АЕМ.

Платформа АЕМ создана таким образом, чтобы можно было разделить обязанности разработки и поддержки веб-приложения. Поэтому присутствуют такие роли, как контент-менеджер – человек с этой ролью добавляет и редактирует контент на самой странице. Это позволяет поддерживать приложение на протяжении всего его жизненного цикла. Сам контент создают разработчики, а так как АЕМ позволяет иметь много разных экземпляров одного и того же приложения, то это помогает каждому разработчику иметь свою локальную среду разработки. Когда один программист закончил со своей частью, то он просто отправляет commit со своей веткой, например, на Github, а другой разработчик начинает работать с ней, чтобы в дальнейшем эту ветку объединить с основной. Также на проектах создаются отдельные экземпляры для тестировщиков, для бизнес-аналитиков и для заказчиков. Заказчики, имея свой экземпляр, могут посмотреть на текущий прогресс и оставить отзыв.

Таким образом, платформа АЕМ создана для того, чтобы в ней могло одновременно работать большое количество разработчиков, при этом, никак не мешая друг другу. АЕМ позволяет также контент-менеджерам поддерживать страницы сайта так, как они этого хотят, при этом делая это без какого-либо знания в области программирования, что помогает крупным компаниям поддерживать крупные сайты без каких-либо проблем.

Так как проект собран для развёртывания в АЕМ, то он имеет структуру и реализацию такую, как требует АЕМ. В АЕМ существует понятие компонента. Компонент в АЕМ – это структура данных, которая может содержать в себе html-файл, файл диалога и клиентские библиотеки (css, js и другие файлы для UI). Компонент можно добавить на страницу, и на странице он отобразится со своими стилями и диалоговым окном (если добавлено).

Для реализации поставленных целей нужно для начала спроектировать шаблон, чтобы можно было на его основе создавать страницы. Шаблон определяет исходную структуру страницы. В дальнейшем по шаблону можно строить страницы. При реализации приложения использованы шаблоны, которые предоставляет АЕМ. В них изначально определены header и footer, которые в дальнейшем можно будет использовать в дочерних страницах. После того, как определена изначальная структура для страниц, можно начинать разрабатывать компоненты, которые будут добавлены на страницы, чтобы заполнить их каким-то контентом.

Схема взаимодействия серверной и клиентской частей разработанного приложения командного управления делами показана на рисунке 1. Пользователь делает запрос на Sling Servlet [2] и затем Sling Framework, двигаясь по JCR, отображает страницы и компоненты.

Так как приложение разработано в АЕМ, то его может доработать человек, не обладающий знаниями в сфере разработки, с помощью добавления контента. Также программное обеспечение полностью адаптировано под мобильные устройства, что позволяет пользоваться им с любого устройства. Основные задачи, которые выполняет разработанное веб-приложение: 1) вход и регистрация пользователей; 2) форма создания задания с такими полями как название, короткое описание, тип задания и пользователи, на которых назначается это задание; 3) отображение этих заданий, сортировка заданий; 4) управление ролями в проекте; 5) удаление заданий и несколько вспомогательных компонентов.

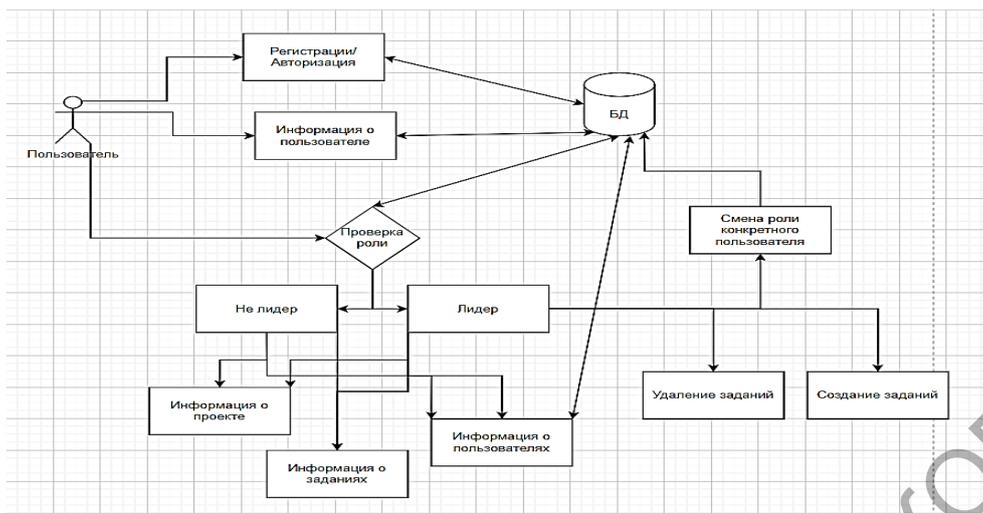


Рисунок 1 – Схема взаимодействия серверной и клиентской частей

Так как приложение разработано в АЕМ, то его может доработать человек, не обладающий знаниями в сфере разработки, с помощью добавления контента. Также программное обеспечение полностью адаптировано под мобильные устройства, что позволяет пользоваться им с любого устройства. Основные задачи, которые выполняет разработанное веб-приложение: 1) вход и регистрация пользователей; 2) форма создания задания с такими полями, как название, короткое описание, тип задания и пользователи, на которых назначается это задание; 3) отображение этих заданий, сортировка заданий; 4) управление ролями в проекте; 5) удаление заданий и несколько вспомогательных компонентов.

Описываемое приложение прошло апробацию в коллективной разработке небольшого веб-приложения, состоящего из трёх разработчиков. Были отмечены положительные моменты от его использования: удобный и понятный интерфейс, полная адаптация под мобильные платформы, идеально подходит для разработки чего-либо в небольшой группе людей. Были высказаны пожелания на улучшение проекта: добавить больше возможностей сортировки заданий, разработать систему историй статусов для заданий.

Результаты работы в виде готового рабочего приложения находятся в сети Интернет по адресу: https://github.com/dimaxdqwerty/project_management_app.

Литература

1 Документация АЕМ [Электронный ресурс]. – Режим доступа : <https://experienceleague.adobe.com/docs/experience-manager-cloud-service.html>. – Дата доступа : 28.04.2022.

2 Документация Sling [Электронный ресурс]. – Режим доступа : <https://sling.apache.org>. – Дата доступа : 28.04.2022.

УДК 004.415.53

А. С. Зайцев

РЕАЛИЗАЦИЯ ТЕСТИРОВАНИЯ РЕШЕНИЯ ЗАДАЧИ ВЕБ-ПРИЛОЖЕНИЕМ LEARN_JS ПО ИЗУЧЕНИЮ JAVASCRIPT

Статья посвящена описанию разработанного в кроссплатформенной среде WebStorm и документоориентированной базе данных MongoDB клиент-серверного