

Е. В. Рябцева

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ВЕДЕНИЯ ПРОЕКТОВ С ИСПОЛЬЗОВАНИЕМ МЕТОДОЛОГИИ SCRUM

Статья посвящена вопросам организации совместной работы над проектами в командах разработчиков. Рассмотрена методология Scrum, позволяющая реализовать участие заказчика в проекте и определяющая гибкость при реализации проекта. Рассматриваются технологии и функциональность работы данного приложения, приведен пример работы веб-приложения.

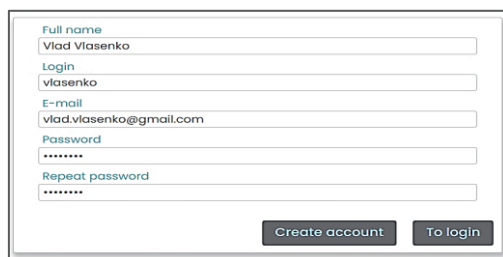
В настоящее время огромной проблемой для заказчиков и исполнителей проектов является организация их совместной работы. Зачастую меняются сроки сдачи проектов, в команде присутствует «нездоровая» конкуренция, не выполняются в срок задания, вынужденно увеличивается финансирование проекта и пр. Заказчики также далеко не всегда остаются довольными итоговым результатом и качеством созданного проекта.

Решение данной проблемы было предложено в 2009 году Джефом Сазерлендом [1] и Кеном Швабером. Они ввели в обиход абсолютно новую методологию. Позже они задокументировали данный метод. Создатели методологии Scrum предлагали создавать IT-проекты дешевле и быстрее, чем это было возможно. Важными составляющими характеристиками методологии Scrum [2] является простота и приспособляемость реализации к различным условиям, а также участие самого заказчика в процессе создания проекта.

Scrum-команда состоит из владельца продукта, команды разработки и Scrum-мастера. Scrum-команды могут самостоятельно выбрать самый лучший способ выполнения работы, а не ждать каких-либо указаний. Такие команды обладают всеми необходимыми навыками выполнения работы. Они не являются зависимыми от кого бы то ни было извне. Методология Scrum была создана для улучшения гибкости, получения лучшей продуктивности и достижения необходимой креативности в работе команды разработчиков.

Разработанное веб-приложение и было создано с целью возможности создания проектов и добавления необходимого количества участников разработки. При создании приложения были использованы следующие технологии: система управления БД PostgreSQL послужила для хранения задач, спринтов и беклога проекта, проектов и пользователей; отображение Prisma помогло соединить сервер приложения с СУБД; с помощью TypeScript стало возможным предотвратить возможные ошибки еще в процессе написания программного кода; и, наконец, Node.js использовался при создании сетевого приложения. Дальнейшее применение Vue.js помогло просматривать изменения в состояниях компонентов и обновлять documentobjectmodel (DOM), использование Sass послужило для создания вложенных CSS правил и функций.

Для того, чтобы начать работу с данным приложением, пользователю необходимо сначала зарегистрироваться и войти в свой профиль (рисунок 1).



Full name
Vlad Viasenko

Login
viasenko

E-mail
vlad.viasenko@gmail.com

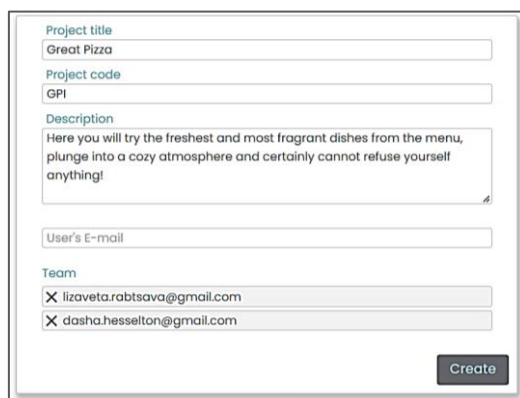
Password

Repeat password

Create account To login

Рисунок 1 – Форма для регистрации в приложении

Затем данный пользователь может создать свой проект или его могут добавить в уже существующий проект (рисунок 2).

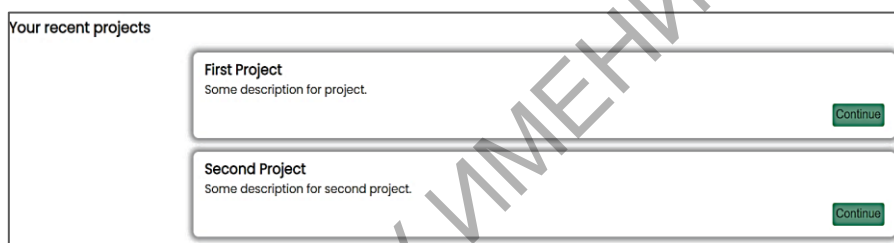


The form contains the following fields and content:

- Project title:** Great Pizza
- Project code:** GPI
- Description:** Here you will try the freshest and most fragrant dishes from the menu, plunge into a cozy atmosphere and certainly cannot refuse yourself anything!
- User's E-mail:** (empty)
- Team:**
 - X lizaveta.rabtsava@gmail.com
 - X dasha.hesselton@gmail.com
- Create** button

Рисунок 2 – Форма для создания/редактирования проекта

Каждый пользователь может просматривать проекты, участником которых он является (рисунок 3).

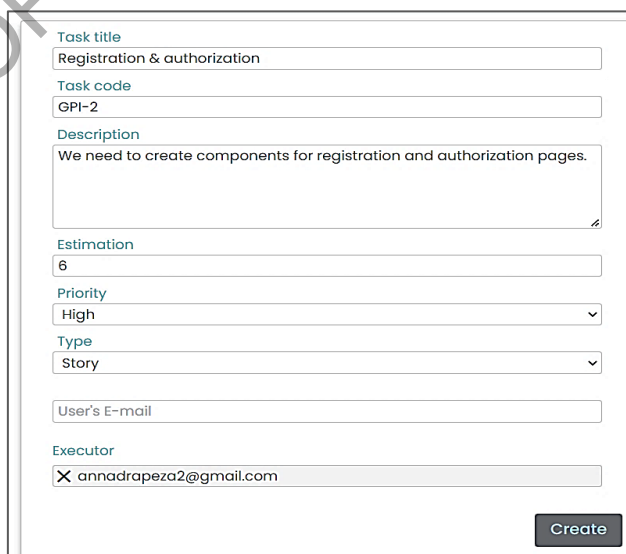


The list is titled "Your recent projects" and contains two items:

- First Project**
Some description for project. [Continue](#)
- Second Project**
Some description for second project. [Continue](#)

Рисунок 3 – Список проектов

Каждый участник команды может создавать новые задания, далее оставлять их в беклоге или добавлять в текущий спринт (рисунок 4). При создании задания важно указать правильное описание того, что нужно выполнить или создать.



The form contains the following fields and content:

- Task title:** Registration & authorization
- Task code:** GPI-2
- Description:** We need to create components for registration and authorization pages.
- Estimation:** 6
- Priority:** High
- Type:** Story
- User's E-mail:** (empty)
- Executor:** X annadrapeza2@gmail.com
- Create** button

Рисунок 4 – Форма для создания / редактирования задания для проекта

Для каждого задания может быть назначен конкретный исполнитель, также предусмотрена возможность оставлять комментарии, устанавливать очередность и приоритет выполнения и задавать текущий статус. Для управления выполнением задач у каждого спринта есть доска, которая разделена на 4 статуса (рисунок 5).

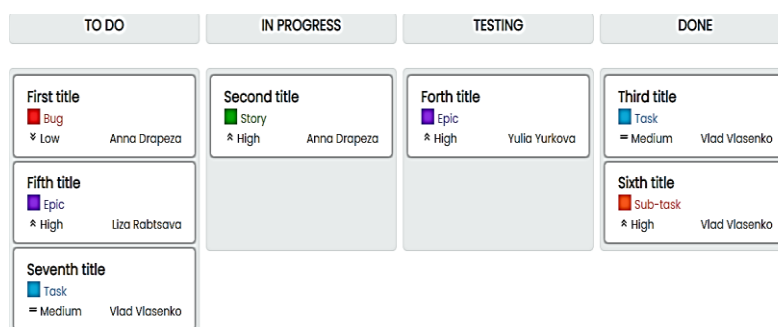


Рисунок 5 – Доска с задачами для проекта

Задания можно перемещать между статусами. Такая функциональность была выполнена с помощью событий «drag» и «drop» DOM.

Кликнув на одно из заданий, пользователь попадает на страницу с подробной информацией об этой задаче.

Second title

■ Story ⚡ High

Description

Ut enim ad minima veniam, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, ut et voluptates repudiandae sint et molestiae non recusandae? Quis autem vel eum iure reprehenderit, qui blanditis praesentium voluptatum deleniti atque corrupti, quos dolores et quas molestias excepturi sint, sunt in culpa qui officia deserunt mollit anim id est laborum!

Details

Status
IN PROGRESS

Executor
Anna Drapeza

Author
Anna Drapeza

Sprint
First sprint

Estimation
7

Рисунок 6 – Информация о задаче

Литература

- 1 Сазерленд, Д. Scrum. Революционный метод управления проектами / Д. Сазерленд. – Москва : Манн, Иванов и Фербер, 2016. – 186 с.
- 2 Кон, М. Scrum : гибкая разработка ПО / М. Кон. – Москва : ООО «Вильямс», 2017. – 576 с.

УДК 005

Д. В. Самусенко

ЭТАПЫ РАЗРАБОТКИ САЙТА-ПОРТФОЛИО НА ПРИМЕРЕ РЕАЛИЗАЦИИ САЙТА ДИЗАЙНЕРА

В статье описываются этапы разработки сайта-портфолио. Анализируются преимущества и недостатки инструментальных средств достижения цели. На примере