

Система управления тестированием TestY

Святослав Алексеевич Латохин, st091713@student.spbu.ru,

Ирина Дмитриевна Шеремет, st087985@student.spbu.ru,

Даниэль Шехаде, st091675@student.spbu.ru,

Глеб Павлович Каргин, st098191@student.spbu.ru

Аннотация

В статье представлена новая система управления тестированием с открытым исходным кодом TestY, позволяющая описывать тестовые сценарии, проекты и тестовые планы, координировать деятельность тестировщиков и фиксировать прогресс тестирования. Описана общая архитектура системы, модель данных, пользовательский интерфейс.

Введение

Для обеспечения качества программных продуктов необходимо заниматься их тестированием. Для этого используют большое количество ручных и автоматизированных тестов. В крупных компаниях может идти речь о тысячах тестовых сценариев для одного проекта. Очевидно, что в таком случае передавать информацию устно или письменно о том, что нужно протестировать, как тестировать и в какие сроки, неудобно.

В процессе тестирования программного обеспечения тестировщики оперируют такими понятиями, как тест-кейсы, сьюты, тест-планы и тесты. Тесты служат для отображения результатов тестирования, тест-планы включают в себя наборы тест-кейсов из сьютов, а сьюты, в свою очередь, объединяют тест-кейсы. С целью эффективного управления данным процессом широко применяются системы управления тестированием, обеспечивающие структурирование и систематизацию всего комплекса действий, связанных с тестированием программных продуктов.

В связи с недостатками аналогичных систем возникла необходимость создания новой системы управления тестированием, которая в первую очередь соответствовала бы требованиям крупных компаний и при этом имела открытый исходный код.

Аналоги

Рассмотрим наиболее популярные существующие системы управления тестированием:

1. TestRail [5] — популярный инструмент для управления тестированием, используемый такими компаниями, как Amazon, Microsoft, Cisco, Intel и др. Приложение предлагает широкий набор функциональности, включая дашборд с информацией по проектам, статистику по каждому из них, тест-кейсы, отчеты, тестовые прогоны и майлстоуны с результатами тестов. Однако система была создана в 2004 году, в связи с чем имеет много устаревших решений, дублирующие сущности, а также затрудненный просмотр тестов при большом количестве экземпляров тест-кейсов.
2. Test IT [4] — российское веб-приложение для тестирования программных продуктов. Оно пользуется популярностью среди отечественных компаний, таких как ВТБ, Магнит и Почта России. Приложение включает в себя разделение на проекты, страницу тестов с тест-кейсами, страницу автотестов с документацией API, тест-планы, конфигурации и дашборд. Стоит отметить большое количество графиков, которые помогают визуализировать результаты. Однако есть и недостатки: невозможность добавлять свои статусы результатов теста и отсутствие атрибутов для тест-результатов.
3. Kiwi TCMS [3] — система с открытым исходным кодом, очень простая, подходящая только для небольших проектов и компаний. В системе нет разделения на проекты и нет тест-сьют, что говорит о невозможности объединять тест-кейсы по тестированию схожей функциональности. Таким образом, создание новой полнофункциональной системы управления тестированием является актуальной задачей, особенно в свете тенденций перехода российских компаний на отечественное программное обеспечение.

Архитектура системы

Клиентская часть системы реализована при помощи библиотеки React и набора компонентов Material UI [2], с использованием языка программирования TypeScript, серверная — на языке программирования Python с использованием фреймворка Django. Связь клиента с сервером реализована при помощи библиотеки Axios [1] — это HTTP-клиент для отправки запросов.

Приложение спроектировано с использованием паттерна MVC (Model-View-Controller). Основная клиентская часть приложения является отдельной компонентой, которая связана с сервером при помощи REST API. Серверная часть подключена к базе данных, используя технологию ORM (Object-Relational Mapping). Плагины являются отдельными компонентами с собственным фронтендом, связь с базой данных также реализована при помощи ORM.

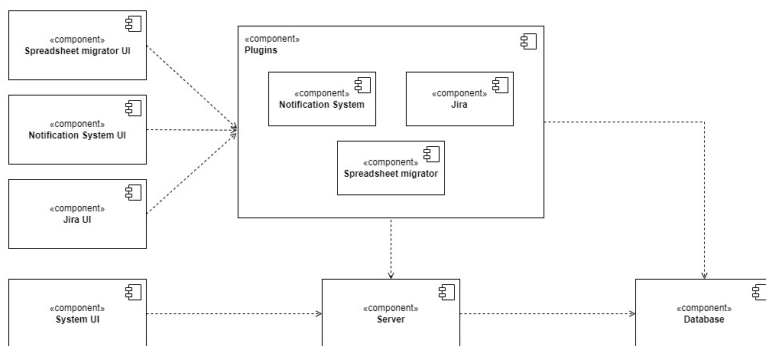


Рис.1: Диаграмма компонентов

Особенности реализации

Разработка серверной части велась на стороне компании, инициировавшей проект.

Фронтенд

Были написаны компоненты для отображения интерфейса пользователю и сервисы, из которых отправляются запросы и принимаются ответы от сервера.

Локализация

Локализация является процессом адаптации приложения под определенную региональную или языковую культуру. Она включает в себя перевод текста, изменение формата даты и времени, настройку валюты и множество других аспектов. Локализация крайне значима, так как способствует увеличению доступности и удобства использования приложения для пользователей из разных стран и культур.

Темная и светлая тема

Темная тема представляет собой вариант дизайна, в котором для фоновых цветов и элементов интерфейса используются темные оттенки, а текст и другие элементы подчеркиваются светлым цветом. Такой дизайн позволяет уменьшить нагрузку на глаза пользователей в темных условиях. В настоящее время многие приложения поддерживают темную тему в качестве опции настройки интерфейса.

Плагины

Система TestY спроектирована расширяемой, с использованием плагиновой системы Django. Плагины и встраиваются в серверную часть системы динамически. Плагины были реализованы на языке программирования Python с использованием фреймворка Django, исходя из технологий серверной части. UI для плагинов написан на TypeScript с использованием библиотеки React. Так как Django производит рендеринг пользовательского интерфейса на стороне сервера, код транслирован в нативный JavaScript.

Jira

Jira изначально являлась сервисом для отслеживания задач и багов, сейчас же — популярный инструмент для управления проектами. При внедрении интеграции с Jira становится возможным ссылаться на задачи и просматривать интересующую информацию о них без непосредственного перехода на сторонний продукт.

Мигратор для Excel-таблиц

Многие отделы тестирования, особенно в небольших компаниях, для контроля качества используют таблицы, сохраняя в них полезную информацию для проведения тестирования. Было необходимо предоставить возможность пользователям перенести данные из Excel таблиц в систему.

Система уведомлений

Система уведомлений позволяет подписаться на все изменения, связанные с сущностью. Например, пользователь может подписаться на изменения отдельного тест-плана или полностью на весь проект. Уведомления могут приходить на почту и/или на Telegram.

Заключение

В ходе выполнения работы получены следующие результаты:

1. сделан обзор существующих решений
2. спроектирован UI и UX дизайн
3. реализована и протестирована клиентская часть приложения
4. добавлены локализация и темная тема
5. реализованы плагины
 - система уведомлений

- мигратор для таблиц
- интеграция с Jira

Список литературы

- [1] Axios. Promise based HTTP client for the browser and node.js. — <https://axios-http.com/>. — [Дата обращения 2023-04-26].
- [2] MaterialUI. The React UI library. — <https://mui.com/>. — [Дата обращения 2023-04-26].
- [3] TCMS Kiwi. — <https://kiwitcms.org/>. — [Дата обращения 2023-04-26].
- [4] TestIT. — <https://testit.software/>. — [Дата обращения 2023-04-26].
- [5] TestRail. — <https://www.gurock.com/testrail/>. — [Дата обращения 2023-04-26].