

КОМАНДНОЕ ПРОЕКТИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Аннотация. В статье представлено описание веб-приложения для командного проектирования пользовательских интерфейсов.

Ключевые слова: команда, проектирование, веб-сервис, веб-приложение, интерфейс, пользователь.

Введение

Основная цель разработки веб-приложения – сокращения времени на командное проектирование простых интерфейсов.

Есть множество аналогов таких как Google Forms, различные визуальные редакторы, Microsoft Visio, BPMN, Justinmind и другие, которые могут помочь в проектировании простых форм, но есть несколько особенностей разрабатываемого веб-приложения, которое оно в себе сочетает:

- Веб-приложения позволяет в режиме реального времени вносить одновременно нескольким пользователям изменения.
- Создавать XAML-подобные формы.
- Сохранять XAML-подобные формы для дальнейшей работы в Visual Studio.

Для выполнение поставленной цели были выделены следующие задачи:

- Хранение необходимых данных о командах пользователей и их результатах работы.
- Динамическое изменение содержимого рабочего поля, на котором пользователи вносят изменения, а также в итоге все изменения.
- Система пользователей, которая позволит объединяться в команды с собственными правилами для работы над общими проектами.
- Возможность сохранения созданных форм в XAML-форм для возможности дальнейшей работы при реализации продукта.

Проблематика

Основная проблема при командном проектировании простых интерфейсов - это затрата времени. Данная проблема вызывается из-за невозможности согласовать время и место для совместного процесса проектирования, а также отсутствию соответствующих ресурсов для ускорения процесса проектирования. Для решения основной проблемы было выбрана разработка веб-приложения.

Реализация

Техническая часть веб-приложения состоит из следующих основных компонентов общей системы, которые соответствуют задачам создания веб-приложения:

- Веб-сервис по хранению и изменению данных.
- Веб-сервис по отрисовке и сохранению изменений форм.
- Система пользователей.
- Конструктор XAML-форм.

Для реализации всей технической части будут применены следующие технологии:

- Ruby on Rails - это фреймворк, написанный на языке программирования Ruby, реализует архитектурный шаблон Model-View-Controller для веб-приложений, а также обеспечивает их интеграцию с веб-сервером и сервером баз данных.
- RethinkDB - это open source база данных для real-time приложений. Она располагает встроенной системой уведомления об изменениях, которая непрерывно транслирует обновления для вашего приложения. Вместо постоянного запрашивания новых данных, позволяет базе данных самой отправлять вам последние изменения.
- AngularJS – JavaScript-фреймворк с открытым исходным кодом. Необходим для разработки одностраничного веб-приложения. Расширяет

возможности управления веб-приложением на основе MVC-шаблона, а также упрощение тестирования и разработки.

- jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML.
- jQuery UI — это библиотека на основе jQuery, реализующая большое множество плагинов, среди которых плагины организующие различное поведение, несколько видов виджетов и анимационные эффекты.

Описывая архитектуру веб-приложения стоит упомянуть самые основные её части, которые представляют собой:

- App – папка, которая содержит множество файлов для работы веб-приложения:
 - Assets – папка, в которой хранятся изображения, файлы стиля (css) и файлы javascript.
 - Controllers – контролеры страниц. В них определяются основные образы объектов и определяются все необходимые методы.
 - Models – модели данных из БД. Хранят структуру, которая используется для управления различными данными БД.
 - Views – страницы веб-приложения.
- Config – папка с набором файлов, которые отвечают за настройку веб-приложения, включая подключения различных элементов типа js, css, scss и другие.
- Gemfile – текстовый файл, в котором хранится набор подключаемых gems, специальных библиотек для Ruby on Rails.

Веб-сервис по хранению и изменению данных представляет собой набор алгоритмов для работы с ДСУБД (документо-ориентированная система управления базами данных) RethinkDB, в которой хранятся иерархические документы, в которых содержится вся необходимая информация для обеспечения возможности хранения таких данных как проекты, формы

проектов с их содержимым в виде различных элементов и наборов их свойств, пользователей системы, группы пользователей и роли пользователей, а также хранится информация о самих элементах (см. Рисунок 1). Доступ и выгрузка данных осуществляется с помощью фреймворка Ruby on Rails.

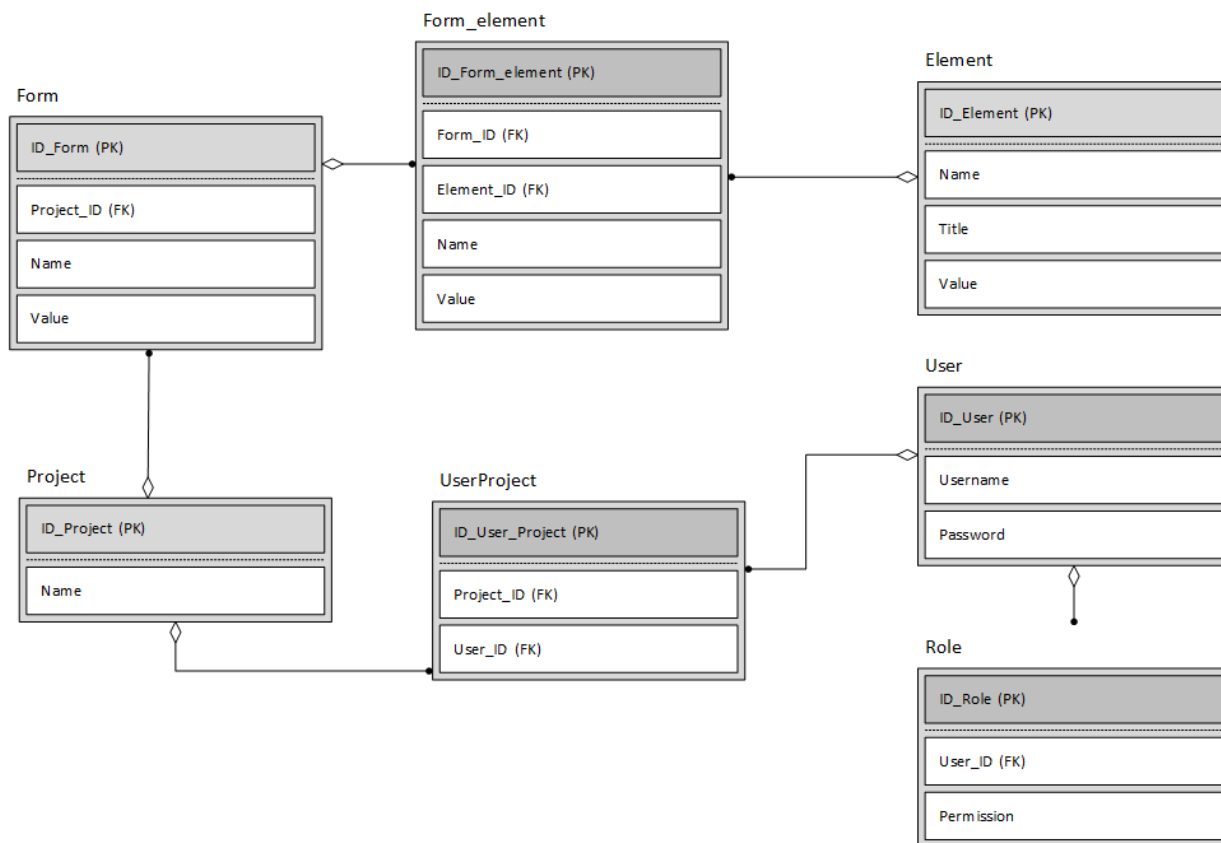


Рис. 1. Схема базы данных.

Веб-сервис по отрисовке и сохранению изменений форм содержит обработчик этих наборов документов, который в режиме реального времени с секундными интервалами отрисовывает содержимое рабочего поля, на котором некоторый пользователь работает с формой. Таким образом, пользователь всегда видит внесенные изменения другими пользователями. Данный сервис основан на связке RethinkDB и AngularJS, которая позволяет совершать постоянное обращение к базе данных, чтобы затем отрисовывать содержимое рабочего поля, а также библиотеки JQuery и JQuery для добавления различных изменений. А также технологии REST, чтобы дать одновременный доступ к отрисованному содержимому рабочего поля.

Система пользователей представляет собой набор функций, которые обращаясь к данным, хранящимся в базе данных позволяют осуществлять определенные действия (см. Рисунок 2).

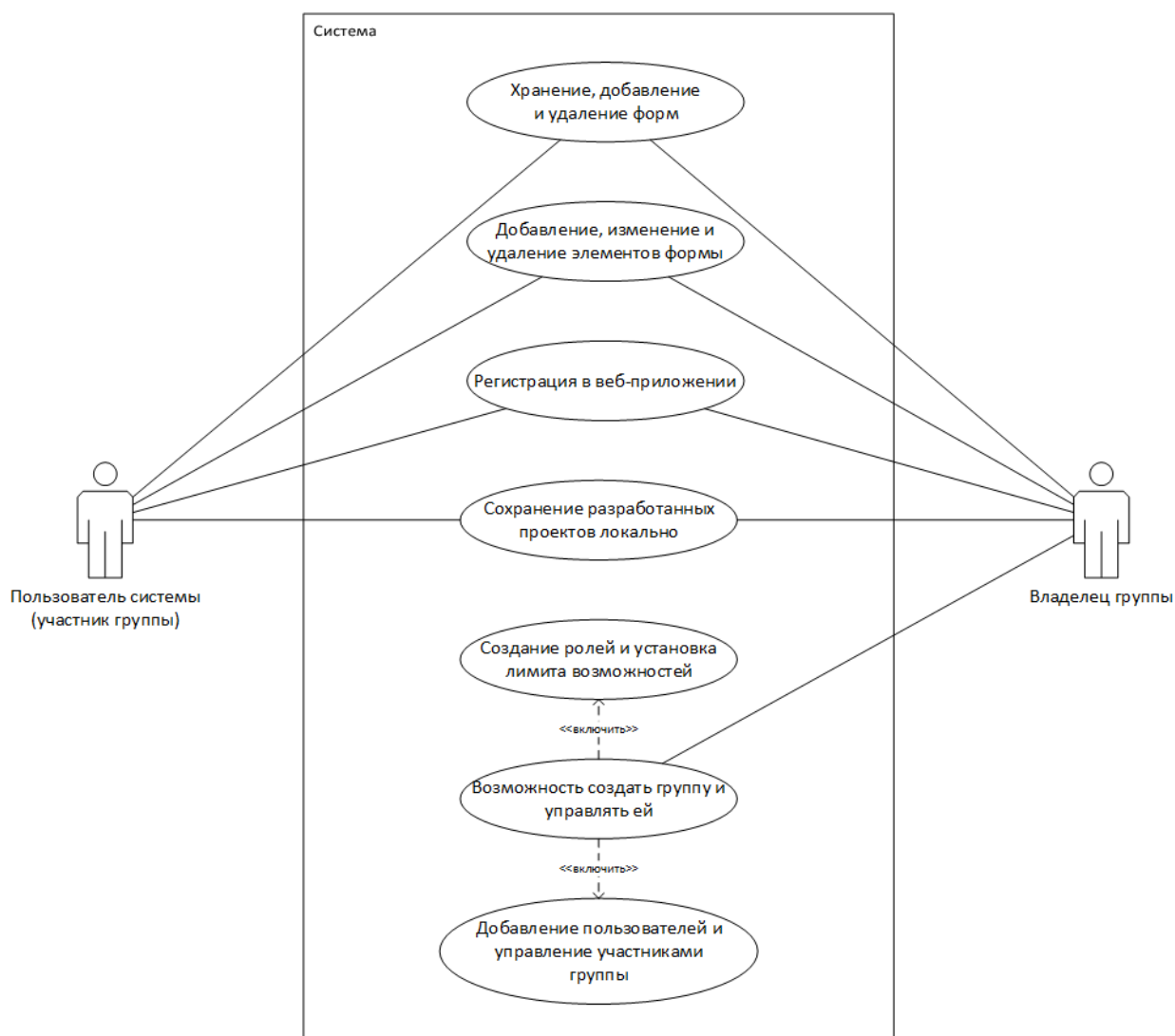


Рис. 2. Диаграмма прецедентов.

Конструктор XAML-форм позволяет обработать формы проекта и выгрузить их в формате, обрабатываемым Visual Studio для дальнейшей работы.

Таким образом, все эти компоненты образуют техническую или Back-end часть веб-приложения, предназначенную для обеспечения основных операций обработки командной разработки простых интерфейсов. Также был разработан Front-end часть для взаимодействия с веб-приложением.

Описание веб-приложения

Веб-приложение для командной разработки простых интерфейсов представляет собой веб-редактор с меню, где расположены основные возможности работы с веб-приложением, изображение пользователя с возможностью посмотреть свои данные и изменить их, удалить учетную запись и выйти из системы, рабочим полем, вкладка набора элементов (текстовое поле, подпись и другие) и вкладка свойства выбранного объекта, расположенные справа от рабочего поля и подвал веб-редактора, где содержится информация об изменениях формы, а также можно посмотреть комментарии, оставленные пользователями.

Перед тем как начать работу в веб-приложении пользователю предлагается войти под своей учетной записью или зарегистрироваться, если таковая отсутствует. При регистрации указывается обязательная информация для заполнения, такая как: имя, фамилия, e-mail, пароль и проверочное повторение пароля. Пароль должен быть как минимум из шести символов. Также реализована возможность восстановления пароля, при активации которой на электронную почту отправляется письмо с указанием нового пароля. После удачной регистрации или уже обладание учетной записью в системе, пользователь может получить доступ к веб-редактору.

При входе в систему пользователь для начала попадает в меню выбора группы, в которой уже состоит или предлагается создать новую, в особенности, если пользователь не состоит ни в одной.

В главном меню можно получить доступ к основным функциям, таким как: управление формами, управление пользователями группы, управление группами, просмотр информации о группе, просмотр информации о себе и выход из системы. Отображение и доступ к части этих функций определяется на основе определенных владельцем группы.

На рабочем поле будет отрисовываться формы, которые размещены каждая в своей вкладке, которыми можно манипулировать, включая удаление вкладки с содержимым. Справа находится вкладки элементов, которыми

можно расположить на форме, кликнув по ним или переместив на форму с помощью технологии Drag&Drop, а также вкладка свойства, где расположены свойства выделенного объекта.

Заключение

В заключение можно сказать, что данное веб-приложение, при использовании описанных технологий, позволит сократить время на командную разработку простых интерфейсов, при этом будет выделяться среди аналогов благодаря особенностям, вложенным в изначальную концепцию разработки.

СПИСОК ЛИТЕРАТУРЫ

1. JQuery Official Site [Электронный ресурс]: <http://jquery.com/>, Docs, Option Reference
2. JQuery UI Official Site [Электронный ресурс]: <http://jqueryui.com/>, Docs
3. AngularJS Official Site [Электронный ресурс]: <http://angularjs.org/>, Docs
4. Ruby On Rails Russian Official Site [Электронный ресурс]: <http://rubyonrails.ru/>, Docs
5. Making a Rails API with Grape [Электронный ресурс]: <http://www.thegreatcodeadventure.com/>