

Список использованных источников

1. Как работает Интернет-радио <https://radio-tochka.com>
2. Пети, В. Микрокомпьютеры Raspberry Pi. Практическое руководство/ В.А. Петин – БХВ-Петербург, 2017.-240с

ВЕБ-ПРИЛОЖЕНИЕ СОЗДАНИЯ И ПРОВЕДЕНИЯ ОПРОСОВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ REACT, REDUX, ASP.NET WEBAPI 2

Ермошин М.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Болтак С.В. – ассистент

В современном мире многим компаниям для того, чтобы оставаться конкурентоспособными необходимо собирать отзывы своих клиентов и быстро реагировать на их изменения. Для этого есть много способов. К ним относятся такие как эксперимент, наблюдение, изучение различных документов. Но один из лучших способов сбора этой информации – проведение опроса.

Опрос – это чаще всего небольшой перечень вопросов, способных дать необходимую информацию. Для улучшения качества результатов вопросы могут разбиваться по темам и вводиться вопросы-фильтры для отсеивания анкет, заполненных ненадлежащим образом.

Вот некоторые преимущества онлайн-опросов:

- экономия ресурсов (денег, времени и трудозатрат);
- широта охвата (преодоление границ и расстояний, доступ к различным социальным группам и сообществам);
- оставляют возможность выбора удобного времени и места участия, могут быть завершены в любое удобное для респондента время;
- релевантность (самостоятельность) коммуникации, т. е. более низкий уровень влияния интервьюера на респондента, возможность давать более развёрнутые ответы;
- широта охвата предметных полей (возможность изучать деликатные и закрытые для публичного обсуждения темы);
- возможность автоматической проверки анкет.

Однако пока что в онлайн-опросах так же присутствуют некоторые недостатки, которые являются критическими для некоторых методологий проведения опроса. Главный недостаток онлайн-опросов связан с проблемой обеспечения репрезентативности выборки. Во-первых, это отсутствие основы выборки. Эта проблема может быть успешно решена в исследованиях организаций с широкими сетевыми базами, а также при построении выборки по результатам оффлайн опроса. Во-вторых, проблема охвата, то есть неспособность используемой выборочной процедуры охватить реальную генеральную совокупность (то есть задать известную ненулевую вероятность попадания в выборку для каждой единицы совокупности). И, в-третьих, это не-ответы либо отказы от участия. Обычно успешно удаётся решить первые две [1].

В разрабатываемом приложении учтено большинство из преимуществ онлайн-опросов. Процедура создания опроса проста и интуитивно понятна. Создать опрос можно на любую интересующую создателя опроса тему. Сразу же после создания опрос публикуется, и респонденты-пользователи сайта смогут проходить опрос. Для приглашения человека поучаствовать в опросе необходимо любым способом (*email*, почта, звонок, факс и т. д.) переслать ему ссылку на опрос.

Стек технологий для приложения был выбран неслучайно. Для клиентской стороны приложения были выбраны такие библиотеки как *React* и *Redux*. Они легковесные, легко дополняемые другими модулями и просты в использовании. Так же их использование уменьшит количество работы, исполняемой на серверной части, что позволит снизить требования к аппаратному обеспечению.

React используется для генерации пользовательского интерфейса. В основе данной библиотеки лежит компонентный подход, т. е. вся разметка разбивается на много компонентов различного уровня. Каждый компонент отвечает только за одну определённую цель.

Клиентское приложение требует хранения его состояния (идентификатор пользователя, вошедшего в систему, промежуточные данные при создании опроса и т. д.). Так как в *React* нету необходимых компонент для хранения состояния, выбор пал на *Redux*. Эта библиотека предоставляет единственное хранилище состояния приложения, которое и является «единственным источником правды» для всего приложения.

Для серверной части приложения была выбрана технология .Net, а именно её фреймворк *WebApi 2*, который позволяет быстро создавать простой *restful* сервис.

В качестве базы данных выбрана *MSSQL* как входящая в комплект поставки *Microsoft Visual Studio Enterprise*. Для работы с базой данных используется *EntityFramework* с подходом *code first*.

Список использованных источников:

1. Девятко И.Ф., Онлайн исследования и методология социальных наук: новые горизонты, новые (и не столь новые) трудности/ Онлайн-исследования в России 2.0. М.: РИЦ «Северо-Восток», 2010. — С.17-31. — ISBN 978-5-9901939-1-8 https://www.hse.ru/data/2013/01/23/1306499729/Online_Research_in_Russia_2_complete.pdf.

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА PATCHWORK ДЛЯ РЕАЛИЗАЦИИ СИСТЕМЫ ВОДЯНЫХ ЗНАКОВ

Ждан В.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ярмолик В.Н. – д.т.н., профессор

В работе рассматриваются угрозы интеллектуальной собственности, а также способы защиты от них. Особое внимание уделяется системам цифровых водяных знаков, а также использованию стеганографического алгоритма *patchwork* в рамках его применения для реализации системы водяных знаков.

Для программного обеспечения и других объектов интеллектуальной собственности актуальными являются следующие угрозы: несанкционированное использование, обратное проектирование и несанкционированная модификация.

Одно из направлений для ликвидации данных угроз – системы цифровых водяных знаков. Данное направление использует различные методы: графическая стеганография, текстовая стеганография, обфускация и другие.

Стеганография — это наука, которая занимается вопросами скрытой передачи информации путём сохранения в тайне самого факта передачи. Открытый текст, где скрыта зашифрованная стеганографическим алгоритмом информация, называется контейнером. Графическая стеганография использует в качестве контейнера графические файлы, например растровые изображения [1].

Алгоритм *Patchwork* является одним из базовых алгоритмов графической стеганографии. В его основе лежит статистический подход. Суть этого алгоритма состоит в следующем. При помощи криптостойкого генератора псевдослучайных чисел, используя заранее известный секретный ключ, выбираются два пикселя изображения. Затем значение яркости одного из них увеличивается или уменьшается на некоторое значение в зависимости от типа встраиваемой информации. Значение яркости другого – уменьшается или увеличивается на ту же величину соответственно. Процесс повторяется около 10000 раз. Значение приращения варьируется от 1 до 5 [2].

Пусть значения яркостей выбираемых на каждом шаге пикселей a_i и b_i , а величина приращения δ . Тогда сумма разностей значений пикселей после n преобразований:

$$S_n = \sum_{i=1}^n [(a_i \pm \delta) - (b_i \mp \delta)] = \pm 2\delta n + \sum_{i=1}^n (a_i - b_i)$$

Математическое ожидание величины $\sum_{i=1}^n (a_i - b_i)$, то есть суммы разности значений пикселей в незаполненном контейнере равно нулю, и его значение в неизменном изображении будет много меньше чем n . Математическое ожидание величины S_n в заполненном контейнере будет равно $\pm 2\delta n$, а вычисленное значение S_n будет иметь большой соответствующий n порядок. В стегодекодере используя ключ проверяется значение S_n . Решение о значении встроенного бита выносится в зависимости от знака S_n . При положительной разности 1, а при отрицательной – 0 [3].

Таким образом, владелец может доказать свои интеллектуальные права, предъявив секретный ключ, который использовался для выборки изменяемых пикселей изображения.