

УДК 004.42

Студ. Д.Н. Юрашевич
Науч. рук. ст. преп. И.Г. Сухорукова
(кафедра программной инженерии, БГТУ)

РАЗРАБОТКА СИСТЕМЫ БРОНИРОВАНИЯ УСЛУГ НА СТАНЦИИ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

Цель работы заключается в разработке системы бронирования услуг на станции технического обслуживания (далее СТО). В настоящее время в мире растет количество автомобилей. По данным авторитетного российского аналитического агентства, Беларусь в СНГ занимает первое место по количеству машин на тысячу человек: у нас 355 авто. И как прогнозируют аналитические агентства, количество автомобилей к 2050 году вырастет в 2 раза. С ростом автомобилей растет и количество СТО.

Для экономии времени подавляющее количество СТО используют предварительное бронирование. Существует три основных вида бронирования: с посещением самого СТО, телефонное бронирование и онлайн бронирование. Для многих людей предпочтительнее онлайн бронирование.

В настоящее время нет единого ПО или интернет ресурса, которые используют СТО – у каждого отдельного СТО своя система, но у всех систем похожий функционал. Анализируя системы онлайн бронирования СТО можно прийти к выводу, что функционал онлайн бронирования должен быть следующий:

- регистрация и авторизация пользователей;
- бронирование времени и даты;
- при необходимости добавлять комментарии к бронированию и/или прикреплять фотографии;
- подтверждение бронирования по email;
- просмотр администрацией забронированных дат и времени;
- добавление администрацией работников и рабочего времени.

Разрабатываемая система бронирования услуг реализует вышеперечисленный функционал. Так же данную систему можно использовать для разных СТО т.к. для установки так называемой Landing-page на заглавную страницу нужно всего лишь выполнить скрипт.

Проект состоит из 4 частей:

- DAL(Data access layer) – содержит все модели данных, хранящихся в БД, а также классы, через которые идет взаимодействие с БД;

- BLL(Business Logic Layer) – уровень инкапсулирует всю бизнес-логику, все необходимые вычисления, получает объекты из уровня доступа к данным и передает их на уровень представления, либо, наоборот, получает данные с уровня представления и передает их на уровень данных;

- WebApi – предоставляет доступ к функционалу. Реализован при помощи .Net Framework WebApi;

- Front-end – является клиентским представлением и обеспечивает доступ через UI к WebApi системы. Реализация при помощи фреймворка React.

Зачастую владельцы интернет сервисов с бронированием сталкиваются с такими проблемами как спам и интернет-боты. Они рассылают нежелательный контент, который вредит владельца и/или пользователям сервиса. В целях защиты от спама и интернет-ботов в системе реализованы такие методы как подтверждение бронирования по E-mail и reCAPTCHA. В отличие от традиционных систем распознавания пользователя как человека путём введения определённого набора символов и цифр, система reCAPTCHA предлагает пользователю ввести два слова. Одно из них уже распознано и известно системе, второе слово системе неизвестно и она не способна его распознать программой распознавания текста. Второе слово берется из источника, требующего распознавания (например, книги). Проверка и прохождение «капчи» осуществляется по тому слову, которое известно системе. Неизвестное второе слово вводить не обязательно. Второе слово, введённое пользователем, сохраняется в системе и используется в качестве возможного варианта распознавания. Окончательное распознавание слова производится путём выбора слова, наиболее часто используемого для ввода. Система reCAPTCHA предоставляет пользователям изображения для распознавания и собирает результаты, после чего передает их организаторам оцифровки материалов.

Алгоритм подтверждения по E-mail можно описать следующим образом. Посетитель сайта заполняет форму бронирования и ему присваивается специальный код, который с другими данными сохраняется в базе данных. На указанный им e-mail он получает сообщение – подтвердить бронирование и переходит по указанной ссылке. Она ведет на специальную страницу сайта, где скрипт проверяет код (используя GET-параметр адресной строки) и, если такой код присутствует в БД, активирует данное бронирование. В случае не подтверждения бронирования сервис автоматически перестает учитывать неподтвержденное бронирование, а подтверждение после указанного времени завершается ошибкой.

Так же важной частью разработки сложной системы является тестирование. Разработка через тестирование (англ. test-driven development, TDD) — является передовой техникой разработки программного обеспечения, основывающемся на повторении кратких циклов разработки: изначально пишется тест, который покрывает желаемое изменение, а далее пишется сам код, позволяющий пройти данный тест, и в конце проводится рефакторинг нового кода к соответствующим стандартам.

Причина, по которой целесообразно выбирать .NETWebAPI это то, что Web API представляет иной способ построения приложения ASP.NET несколько отличный от ASP.NET MVC. Web API представляет собой веб-службу, которая может взаимодействовать с различными приложениями. При этом приложение может быть веб-приложением ASP.NET, либо может быть мобильным или обычным десктопным приложением.

Для реализации Front-end выбор пал на React по следующим причинам:

- отлично подходит для командной разработки, строгое соблюдение UI, и шаблона рабочего процесса;
- UI код читабельный и прост в сопровождении;
- разработка UI на основе отдельных компонентов.

В заключение можно отметить, что сейчас открывается много новых СТО и для экономии времени клиентов и удобства бронирования одним из вариантов является онлайн бронирование. К сожалению, многие СТО не могут себе позволить самостоятельно реализовать свой сервис. Данная работа позволит администрации СТО легко построить и запустить свой сервис онлайн бронирования.

ЛИТЕРАТУРА

1. Документация по Reactjs. [Электронный ресурс] - Режим доступа: <https://reactjs.org> (дата обращения 15.12.2017).
2. Документация по .NetFramework[Электронный ресурс] - Режим доступа: <https://docs.microsoft.com/en-us/dotnet/framework> (дата обращения 20.12.2017).
3. Документация по ReCaptcha[Электронный ресурс] - Режим доступа: <https://www.google.com/recaptcha/intro/android.html> (дата обращения 04.01.2018).