

АВТОМАТИЗИРОВАННАЯ СИСТЕМА БРОНИРОВАНИЯ УСЛУГ AUTOMATED RESERVATION SYSTEM

М. А. Авсецина¹, М. М. Ибрахим², И. В. Лефанова¹
M. Ausetyna¹, M. Ibrakhim², I. Lefanova¹

¹Белорусский государственный университет, МГЭИ им. А. Д. Сахарова БГУ,
г. Минск, Республика Беларусь

²Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь
milana666666@gmail.com

¹International Sakharov Environmental Institute of Belarusian State University,
Minsk, Republic of Belarus

²Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus

Рассматривается разработка системы автоматического бронирования услуг. Обоснована необходимость автоматизации процессов, а также важность возможности забронировать услугу, поставлена задача и обзор существующих аналогов, имеется вывод о текущей ситуации на рынке и принципы построения клиент-серверных систем.

The development automatic reservation system of services is considered. The necessity of automation of processes is justified, as well as the importance of the ability to reserve a service, the task and overview of existing analogues are posed, there is a conclusion about the current situation on the market and the principles of client-server systems building.

Ключевые слова: сервер, клиент, автоматизация, typescript, c#, .net core, angular.

Keywords: server, client, automation, typescript, c #, .net core, angular.

<https://doi.org/10.46646/SAKH-2020-2-281-284>

Объектом исследования и разработки является приложение для упрощения бронирования услуг, сокращение времени и затрат на данный процесс.

Целью проекта является разработка программного комплекса, который представляет собой сервер, обрабатывающий все данные компании и предоставляющий возможные временные интервалы для услуги, и клиент, который наглядно визуализирует данный процесс и предоставляет возможность администрирования.

При разработке и внедрении приложения использовался такой стек технологий как C#, .Net, Asp.Net Core, SqlServer, Angular, TypeScript.

Требования для разработки:

- автоматизация процесса высчитывания свободных промежутков времени;
- учетывание всех предусмотренных факторов, влияющих на возможность оказания услуги;
- легкий доступ к системе бронирования для клиентов компании;
- надежное сокрытие данных компании, предоставляемое только для сотрудников;
- слабая связанность серверной и клиентских частей, возможность быстро заменять клиентское приложение, не внося логику в сервер;
- возможность разместить приложения на любой операционной системе;
- красивый и интуитивно понятный пользовательский интерфейс.

Следует отметить, что данная разработка будет актуальна как для небольших компаний, так и для крупных предприятий, так как корректное распределение времени работы персонала является необходимым условием для успешной, продуктивной и прибыльной деятельности компании, а своевременная и актуальная информация о всех бронированиях позволит легко подводить статистику, анализировать успехи и быстро принимать необходимые решения.

Минимальными требованиями к реализации программного продукта являются:

- сервер, обрабатывающий данные, должен иметь клиентскую часть с интуитивно понятным интерфейсом в сети интернет;
- представители компании должны прикладывать минимальные усилия для поддержания функционирования продукта, а именно, заносить только те данные, которые невозможно высчитать или получить другими способами, например, время работы сотрудников;
- продукт должен грамотно высчитывать и предоставлять все возможные промежуточные интервалы, учитывая время работы компании, сотрудников, наличие ресурсов, наличия других бронирований, при этом способствовать устранению всех промежутков между занятыми интервалами;

- гибкая архитектура, позволяющая в будущем работать и с другими клиентскими приложениями, к примеру, мобильными приложениями на платформе Android и iOS;
- обеспечение кроссплатформенности приложения, возможность размещать его на всех доступных операционных системах, а именно, Windows, Linux, MacOS;
- обеспечить гарантию сохранения данных;
- закрытый доступ к данным компании через систему авторизации по логину и паролю;
- иметь функционал по заблаговременному оповещению клиентов о приближающейся дате бронирования;
- возможность любого человека, будь то сотрудник компании или клиент, иметь представление о занятости компании в удобном визуальном виде, а именно, о предстоящих и прошедших бронированиях услуг.

На данный момент имеется несколько подходов к построению клиент серверных систем, а именно двухзвенная и трехзвенная архитектуры. Что касается двухзвенной архитектуры она подходит только для небольших приложений с небольшой загрузкой и количеством запросов, так как масштабирование затруднено. На рисунке 1 можно наглядно увидеть двухфазную архитектуру клиент-серверной системы.



Рисунок 1 – Двухфазная архитектура клиент-серверной системы

Трехфазная архитектура пришла на смену двухфазной, были решены предыдущие проблемы. Логика была вынесена в отдельный уровень, который находится на сервере. Клиенту теперь не обязательно работать напрямую с базой данных, можно абстрагироваться от нее. Появилась возможность горизонтального масштабирования. На рисунке 2 можно наглядно увидеть трехфазную архитектуру клиент-серверной системы.

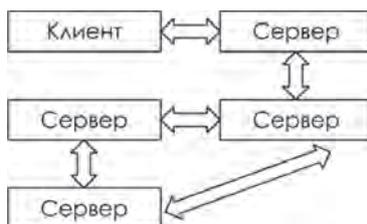


Рисунок 2 – Трехфазная архитектура клиент-серверной системы

В данном проекте реализована архитектура “клиент-сервер”. Все вычислительные нагрузки возложены на сервер, клиент же по сути является заказчиком услуг, которому предоставляются все необходимые данные. В дипломном проекте сервер состоит, собственно, из серверного слоя и слоя доступа к данным. Клиент представляет один слой – слой представления.

Сервисный слой реализовывает логику по приему запросов, валидации данных и выдаче ответов в необходимом формате. При этом серверный слой абстрагирован от всей логики, что упрощает обслуживание, тестирование и позволяет без труда использовать его повторно, то есть, проще говоря, он становится переносимым. Самым основным компонентом слоя является API (Application Programming Interface). По сути, это интерфейс программирования приложений. Для его реализации в проекте используются контроллеры, которые располагаются в папке Controllers. Для взаимодействия с методами контроллеров используются HTTP методы: GET; POST; PUT; DELETE.

В проекте насчитывается 8 контроллеров. Кратко опишем их обязанности и приведем пример кода одного из них:

- AccountController. Это промежуточная ступень для таких действий как регистрация и выдача токена авторизации;
- AppointmentController. Данный контроллер обрабатывает такие запросы, как получение всех бронирований за определенную дату, получение бронирования по уникальному номеру, удаление, сохранение и обновление записей;
- AuthenticatedController. Вспомогательный контроллер, единственной задачей которого является закрытие доступа к ресурсам приложения для неавторизованных пользователей. То есть, если необходимо что-либо скрыть от клиентов, к примеру, возможность добавлять сотрудников в компанию, необходимо лишь произвести наследование от данного класса;
- CompanyController. Данный контроллер отвечает за все запросы к компании, а именно, получение данных компании, получение времени работы, обновление данных;
- EmployeeController. Контроллер для работы с сотрудниками компании. Включает такие методы, как получение всех сотрудников, получение единственного сотрудника по уникальному номеру, проверка на возможность

удаления сотрудника, удаление сотрудника, сохранение нового сотрудника, обновление данных у существующих, получение рабочего времени сотрудника за определенный период;

– ResourceController – это контроллер, ответственный за запросы клиента к ресурсам компании. Позволяет выдать все ресурсы, проверить возможность удаления ресурса, удалить его и сохранить новый;

– ServiceController. Самый объемный контроллер проекта. В его обязанности входит выдача всех возможных услуг, предоставляемых компанией, выдача только наименований услуг, выдача услуг по уникальному номеру бронирования, в котором данные услуги непосредственно участвуют, проверка на возможность удаления и удаление услуги, а также сохранение новой и обновление существующей соответственно;

– TimeController. Данный контроллер необходим лишь для обработки запроса по получению всех свободных временных диапазонов для бронирования.

Слой доступа к данным необходим для непосредственного обращения к базе данных через специальный класс Repository. Этот класс реализовывает такие методы как сохранение сущности, сохранение коллекции объектов, удаление сущности, удаление коллекции объектов, получение данных.

Также в данном слое в папке Domain находятся модели. Это сущности, представленные в виде классов, которые хранятся в базе данных приложения. В нашем случае имеются следующие модели:

– ApplicationUser; Модель, описывающая зарегистрированного пользователя, то есть администратора. Помимо данных о самом пользователе, хранит всю информацию о компании;

– Appointment. Сущность бронирования. Хранит в себе информацию о клиенте, дате встречи, цене и предоставляемой услуге;

– Company. Данная модель помимо данных компании, хранит в себе её время работы, сотрудников, предоставляемых услуг и администраторов;

– Employee – модель сотрудника компании;

– EmployeeWorkingTime – модель, подробно описывающая время работы сотрудника.

– Resource – модель, описывающая ресурсы компании, используемые в услугах, также хранит в себе оставшееся количество самого себя;

– Service предоставляет данные об услуге компании, её название, цену, описание, цвет услуги (цвет, которым закрашивается календарь на веб-странице), информацию о сотрудниках, которые оказывают её, необходимых ресурсах, частях услуг (услуга может быть поделена на части), бронированиях этой услуги.

Слой представления – эта та часть приложения, которую видит пользователь. К данному слою имеются особые требования, такие как удобство использования, дизайн и возможность взаимодействия с пользователем.

При попадании на сайт, перед клиентом отображается календарь за текущую неделю, при необходимости с помощью кнопок в верхнем правом углу имеется возможность перехода на другие даты и быстрого возврата на текущий день. Календарь показан на рисунке 3. Для более удобного использования были обрезаны участки с неиспользованным временем. К примеру, если у компании самое раннее открытие в один из дней в 9 утра, а самое позднее закрытие в 8 вечера, то будет отображен промежуток с 9 утра до 8 вечера. Все нерабочие часы окрашиваются в серый фон. Таким образом наглядно видно рабочее время.

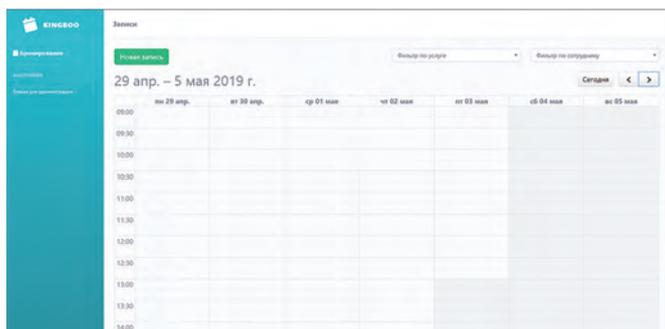


Рисунок 3 – Демонстрация календаря

Возможностей у администратора больше. Это обусловлено тем, что для автоматизации процесса необходимо задать начальные данные. Ниже опишем все возможности системы.

Сначала стоит отметить, что все данные возможности скрыты от посетителей, для получения доступа к ним необходима авторизация. Страница с авторизацией показана на рисунке 5. Попасть на нее можно, нажав на ссылку “Только для администрации” в левой части страницы.

После успешной авторизации в левой части страницы появятся 4 настройки:

- компания;
- ресурсы;
- сотрудники;
- услуги.

На каждой из страниц сверху идёт небольшое пояснение, чтобы администратор быстрее разобрался и приступил к настройке.

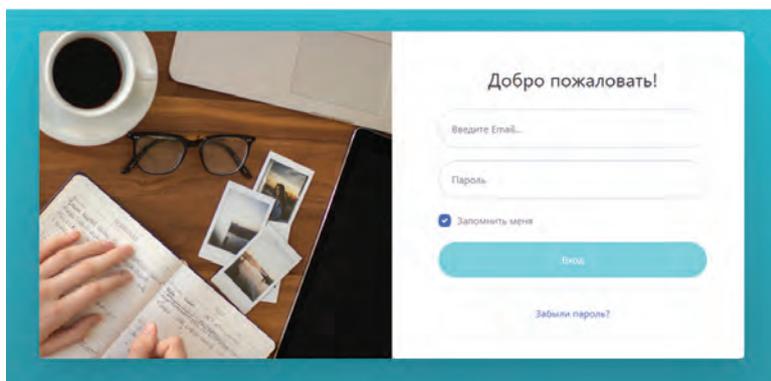


Рисунок 5 – Страница авторизации

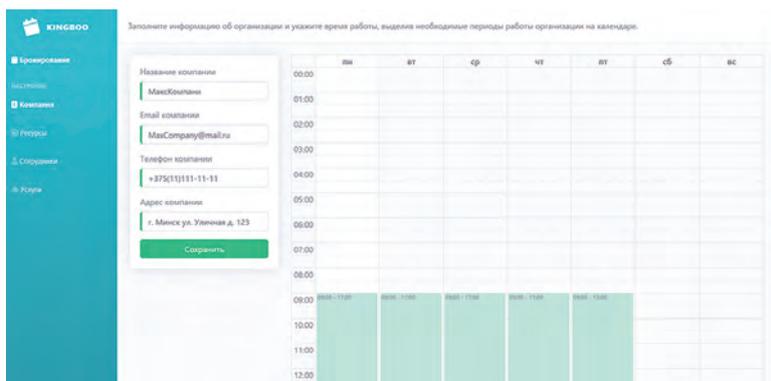


Рисунок 6 – Страница настроек компании

Разработанная автоматизированная система бронирования услуг предназначена для организаций сферы услуг, стремящихся автоматизировать работу с клиентами и сократить расходы на заработную плату менеджеров или работников колл-центра. В рамках системы потенциальные клиенты при обращении на сайт компании могут забронировать любую свободную дату, корректировка которой осуществляется исходя из времени работы компании, наличия подходящих сотрудников, наличия ресурсов и других осуществленных бронирований. Данные нововведения позволяют отказать от услуг операторов и экономят время клиентов. Вся информация и брони наглядно отображаются на календаре, который имеется у каждой компании, использующей систему.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ В ВЫСШЕМ ТЕХНИЧЕСКОМ ОБРАЗОВАНИИ INFORMATION SYSTEMS AND TECHNOLOGIES IN HIGHER TECHNICAL EDUCATION

Г. В. Бельская, Н. Г. Малькевич
H. Belskaya, N. Malkevich

*Белорусский национальный технический университет,
г. Минск, Республика Беларусь
gbelskaja@mail.ru
Belarusian National Technical University, Minsk, Republic of Belarus*

Рассмотрены возможности внедрения информационных технологий в учреждениях высшего технического образования. Подчеркнуты преимущества использования информационных технологий для подготовки современных инженерных кадров в условиях цифровой экономики. Обоснована необходимость всестороннего развития электронного образования в связи с требованиями времени. Отмечены некоторые проблемы электронного образования.

Possibilities of information technologies implementation in training higher technical education establishments are considered. Advantages of information technologies using for training of modern engineers in digital economics