

2. Постановление Совета Министров Республики Беларусь от 17.03.2016 №205 «Об утверждении Государственной программы «Охрана окружающей среды и устойчивое использование природных ресурсов» на 2016–2020 годы», https://kodeksy-by.com/norm_akt/source/205-17.03.2016, [Электронный ресурс].

3. Об утверждении Государственной программы развития Национальной системы мониторинга окружающей среды в Республике Беларусь на 2006–2010 годы: Указ Президента Республики Беларусь, 18.04.2006, №251 // Нац. реестр правовых актов Республики Беларусь. – 2006. – №69. – 1/7482.

4. Национальная система мониторинга окружающей среды Республики Беларусь: результаты наблюдений. 2011, ежегодник / под общей редакцией С.И. Кузьмина, И.В. Комоско. – Минск, «БелНИЦ «Экология». – 2012. – 320 с., ил. 318.

5. Об утверждении Инструкции об обмене экологической информацией в Национальной системе мониторинга окружающей среды в Республике Беларусь. Постановление Министерства природных ресурсов и охраны окружающей среды Республики Беларусь, 28 дек. 2004 г., № 43: В редакции постановления от 30.05.2008 г., № 56 // ИПС «Эталон» [Электронный ресурс]. / Нац. центр правовой информ. Республики Беларусь. – Минск, 2012.

6. Сайт ГИАЦ Национальной системы мониторинга окружающей среды в Республике Беларусь (<http://www.nsmos.by/>), [Электронный ресурс].

ОБРАБОТКА И ХРАНЕНИЕ ДАННЫХ ДЛЯ СИСТЕМЫ АВТОМАТИЗАЦИИ БРОНИРОВАНИЯ DATA PROCESSING AND STORAGE FOR THE BOOKING AUTOMATION SYSTEM

М. А. Ибрахим¹, М. М. Ибрахим²

M. A. Ibrakhim, M. M. Ibrakhim

*¹International Sakharov Environmental Institute of Belarusian State University
Minsk, the Republic of Belarus
E-mail: milana6666666@gmail.com*

*²Belarusian State University of Informatics and Radioelectronics
Minsk, the Republic of Belarus
E-mail: maksim.ibr.17@gmail.com*

Рассматривается разработка системы обработки и хранения данных при бронировании. Обоснована необходимость автоматизации процессов, надежного хранения данных, экономии человеческих и бумажных ресурсов.

The development of a system for processing and storing data when booking is considered. The necessity of automation of processes, reliable storage of data, saving of human and paper resources is substantiated.

Ключевые слова: АВТОМАТИЗАЦИЯ, .NET CORE, ENTITY FRAMEWORK.

Keywords: AUTOMATION, .NET CORE, ENTITY FRAMEWORK.

<https://doi.org/10.46646/SAKH-2021-2-408-411>

Объектом исследования и разработки является надежное хранилище, способное к обработке бронирования услуг, подходящего для любого рода компаний, в том числе связанных непосредственно с экологией, сокращение времени и затрат на данный процесс.

Целью проекта является разработка программного комплекса, который представляет собой слой обработки и хранения данных, предоставляющий возможные временные интервалы для услуги.

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

Требования для разработки:

- надежное хранилище, не позволяющие потерять данные;
- автоматизация процессов для высчитывания свободных промежутков времени, необходимого для услуги с учетом всех предусмотренных факторов;
- надежное сокрытие данных компании, предоставляемое только для сотрудников;
- слабая связанность данных;
- возможность разместить приложения на любой операционной системе;

Для любой эффективно работающей системы необходимо понимание что же такое архитектура. Итак, архитектура – это организация системы, которая реализована некоторыми компонентами/слоями, их отношением между собой и внешней средой.

Основные типы компонентов архитектуры:

- Серверный слой

Данный 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

- Слой доступа к данным

Данный слой 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.

- Слой представления

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.

В данном случае нас интересует именно Слой доступа к данным, о нем подробнее:

Данный слой необходим для непосредственного обращения к базе данных.

Он реализовывает такие методы как сохранение сущности, сохранение коллекции объектов, удаление сущности, удаление коллекции объектов, получение данных. Для упрощения разработки можно использовать обертку `Entity Framework Core (EF)`, что позволяет нам к тому же расширить функциональность и добавлять свои действия при необходимости. Эта обертка позволяет работать приложению именно с моделями, а не с базой напрямую.

Чтобы база данных была сгенерирована, необходимо описать, собственно, все 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, информацию о сотрудниках, которые оказывают её, необходимых ресурсах, частях услуг (услуга может быть поделена на части), бронированиях этой услуги.

Имеются и другие модели, но они несут лишь вспомогательные функции, например, для обеспечения возможности организации связи “многие ко многим”.

Система моделирования позволяет EF в дальнейшем, при создании миграции в приложении, автоматически переписать модели в язык структурированных запросов или SQL, который уже непосредственно применяется для создания, модификации и управления данными в реляционной базе данных.

Вся работа с базой при использовании такого подхода может быть автоматизирована благодаря сервисам, в которых и заключена вся логика работы приложения, по сути являющимися связующим звеном между серверным слоем и слоем доступа к данным.

Так как каждый сервис работает со своими моделями, к тому же в асинхронном режиме, это позволяет системе выдерживать колоссальные нагрузки и не “падать” в критический момент.

Важно подчеркнуть, что не только асинхронность является залогом успеха, не менее важную роль играет правильная расстановка самих моделей. Приведем пример на Схеме 1:


```

public string Email { get; set; }
public string Phone { get; set; }
public string Address { get; set; }
public virtual ICollection<WorkingTime> WorkingTime { get; set; }
public virtual ICollection<ApplicationUser> Users { get; set; }
public virtual ICollection<Employee> Employees { get; set; }
public virtual ICollection<Service> Services { get; set; }
}

```

Остальные модели реализованы аналогично.

В папке Services же находятся все сервисы, в которых и заключена вся логика работы приложения. Данные являются связующим звеном между серверным слоем и слоем доступа к данным. В данной работе представлены следующие сервисы:

- AccountService. Необходим для работы с администраторами;
- CompanyService. Отвечает за обработку данных компании;
- EmployeeService. Данный сервис выполняет все действия, связанные с сотрудниками компании;
- ProfileService. Данный сервис необходим для предоставления некоторых данных в теле токена авторизации;
- ReservationService. Данный сервис работает непосредственно с бронированиями;
- ResourceService. Сервис, отвечающий за ресурсы;
- ServiceService. Сервис, в котором заключена вся логика по работе с услугами;
- TimeService. Основной сервис, который предоставляет все свободные временные зоны для бронирований.

Данный подход к обработке и хранению данных может быть предназначен для любой экологической организации, которая располагает некими услугами, для более эффективной работы в своей сфере, а также для надежности хранения ценной информации. К тому же позволяет экономить человеческие усилия, которые можно направить в другое русло, и, само собой, бумажные и другие ресурсы.

ЛИТЕРАТУРА

1. Архитектура межсервисного взаимодействия, <https://secretmag.ru/business/methods/7-servisov-dlya-onlajn-zapisi.htm>, [Электронный ресурс].
2. Модель взаимодействия клиент-сервера, <https://ru.wikipedia.org/wiki/>, [Электронный ресурс].
3. Семь сервисов для онлайн записи, <https://secretmag.ru/business/methods/7-servisov-dlya-onlajn-zapisi.htm>, [Электронный ресурс].
4. .Net Framework, <https://www.guru99.com/net-framework.html>, [Электронный ресурс].
5. .Net Core, <https://metanit.com/sharp/aspnet5/>, [Электронный ресурс].

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ ХРАНЕНИЯ ИНФОРМАЦИИ О ДИПЛОМНЫХ, АСПИРАНТСКИХ И МАГИСТЕРСКИХ РАБОТАХ DEVELOPMENT OF A WEB-APPLICATION FOR STORING INFORMATION ABOUT GRADUATE, POST-GRADUATE AND MASTER'S WORKS

А. Л. Карпей, Д. А. Кормужанин
A. L. Karpei, D. A. Kormuzhanin

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

*E-mail: kar_an@tut.by
Belarusian State University, ISEI BSU
Minsk, Republic of Belarus*

С использованием языка программирования Python с фреймворком Django разработано web-приложение, хранящее информацию о дипломных работах студентов, аспирантских и магистерских диссертациях. Web-приложение связано с базой данных SQLite и предоставляет возможности добавления, изменения и удаления информации из базы данных. Для создания приложения были использованы среда разработки PyCharm и программа для просмотра информации о базе данных DB Browser for SQLite.

Using the Python programming language with the Django framework, a web-application has been developed that stores information about students' theses, postgraduate and master's theses. The web application is linked to a SQLite database and provides the ability to add, modify, and remove information from the database. To develop the