

ОРГАНИЗАЦИЯ УЧЕТА ХРАНЕНИЯ И ИСПОЛЬЗОВАНИЯ ХИМИЧЕСКИХ ВЕЩЕСТВ НА ПРЕДПРИЯТИИ

ORGANIZATION OF THE STORAGE AND USE OF CHEMICALS IN AN ENTERPRISE

A. A. Акантинова, В. А. Иванюкович
A. Akantinova, U. Ivaniukovich

Белорусский государственный университет, МГЭИ им. А. Д. Сахарова БГУ
г. Минск, Республика Беларусь
u.ivaniukovich@gmail.com
Belarusian State University, ISEI BSU
Minsk, Republic of Belarus

Описана структура и функциональные возможности мобильного приложения, предназначенного для учета и заказа химических веществ, используемых на предприятии. Приложение позволяет в режиме реального времени использовать базу данных, созданную на специализированном сайте.

The structure and functionality of a mobile application designed for accounting and ordering of chemicals used in an enterprise is described. The application allows real-time use of the database created on a specialized site.

Ключевые слова: учет химических веществ, свойства химических веществ, режим реального времени, химическое предприятие, мобильное приложение, технологии программирования, сайт.

Keywords: accounting of chemicals, properties of chemicals, on-line, chemical enterprise, mobile application, programming technologies, website.

<https://doi.org/10.46646/SAKH-2021-2-395-398>

Во всех развитых странах химическая промышленность составляет существенную часть внутреннего валового продукта. Большие объемы химического сырья и выходной химической продукции вынуждают разрабатывать новые технологии их учета. Сегодня для этого применяются различные информационные технологии специального или общего назначения, такие как интегрированные системы управления производством на платформе 1С, электронные таблицы и т.п. Как правило, они не позволяют организовать ведение учета химических веществ непосредственно в ходе технологического процесса. Возможный путь оптимизации учета – использование мобильных технологий для организации взаимодействия со стационарными программными платформами и хранилищами данных.

В настоящее время специалисты многих химических компаний, университетов и исследовательских центров имеют возможность использовать специализированный web-сайт для учета химических веществ, созданный по заказу NASA и Гонконгского политехнического университета. Сайт предоставляет пользователям специальную платформу, которая может быть адаптирована под потребности промышленных предприятий, научно-исследовательских центров, университетских лабораторий и иных организаций, для которых актуален учет химических веществ.

Пользователи данного сайта неоднократно сообщали о том, что процесс учета достаточно проблематичен, так как многие компании, использующие данный web-сайт, имеют крупные склады и большие производственные мощности. У работников, отвечающих за складские запасы и следящих за использованием ресурсов, нет возможности вносить и изменять данные в режиме on-line. Это приводит к замедлению процесса учета, что влечет за собой удорожание продукции и создает множество проблем, связанных с неактуальностью данных. Для решения этой проблемы предлагается кроссплатформенное мобильное приложение, которое поможет решить данную проблему. Основные требования к разрабатываемому приложению следующие:

- поддержка наиболее популярных мобильных операционных систем iOS, Android и Windows Phone;
- аутентификация пользователя в системе;
- возможность редактирования данных;
- возможность добавления данных;
- доступ к необходимой информации о веществе (место хранения, количество, свойства и т.п.).

Создание кроссплатформенного приложения обусловлено необходимостью поддержки нескольких операционных систем. Реализация таких приложений связана с решением ряда проблем. Принципиальные различия в подходах построения графического интерфейса делают необходимым подстраивать приложение под требования к интерфейсу платформы. Различия в API в программных интерфейсах и реализациях функционалов также требуют различные подходы.

Характерной особенностью кроссплатформенной разработки является создание единой логики приложения с корректно отображающимся интерфейсом и функционированием на разных платформах.

Основная часть программного продукта создана на языке C# в среде разработки Visual Studio. Для реализации требований кроссплатформенности мобильного приложения использован фреймворк Xamarin.Forms. Xamarin основан на Mono, open-source реализации платформы .NET. Фреймворк позволяет создавать приложения с помощью инструментов корпоративного уровня, что позволяет проводить оптимизацию производительности приложения, а также исследование их в среде выполнения для быстрого поиска ошибок. Тестирование разработки проводится на реальных устройствах в тестовом облаке Xamarin. Система включает в себя собственный компилятор C#, среду выполнения, а также основные .NET-библиотеки. Благодаря этому достигается запуск программ, написанных на языке C#, в операционных системах, отличных от Windows, в том числе и в мобильных Android, iOS и Windows Phone, обеспечивая при этом корректность логики программы и визуализации интерфейсов. Использование фреймворка Xamarin.Forms дает ряд преимуществ. В процессе разработки создается единый код для всех платформ. Xamarin предоставляет прямой доступ к собственным API каждой платформы. При создании приложений может быть использована платформа .NET и язык программирования C#. Xamarin Forms поддерживает все упомянутые мобильные платформы. В фреймворке визуальный интерфейс состоит из страниц, которые занимают все пространство экрана, что очень удобно для разработчиков.

Пользовательские интерфейсы Xamarin.Forms отображаются с использованием собственных элементов управления целевой платформы, что позволяет приложениям Xamarin.Forms сохранять соответствующий внешний вид для каждой платформы. Пользовательские средства визуализации позволяют разработчикам настроить внешний вид интерфейсов и поведение элементов управления Xamarin.Forms на каждую платформу, соответствующую кодам самой платформы.

Для создания интерфейсов приложения используется язык разметки XAML, основанный на расширенном языке форматирования xml, который позволяет создавать объекты декларативно. Созданные в XAML объекты по своей природе совместимы с web-технологиями.

Общая архитектура приложения спроектирована с использованием шаблона MVVM (Model-View-ViewModel). Основное преимущество данного подхода – разграничение логики приложения и логики визуальной части. Каждый модуль отвечает только за свою конкретную функцию. Благодаря такому разграничению код становится более гибким и простым в поддержке. Шаблон MVVM состоит из трех компонентов: модели (Model), модели представления (ViewModel) и представления (View). Модель хранит данные и не связана с бизнес-логикой приложения. Приложение содержит отформатированные данные, соответствующие модели, и обеспечивает доступ к виджетам, отображающим информацию на экране. ViewModel обеспечивает связь модели с представлением, организуя логику поведения представления в соответствии с работой модели.

Первый этап разработки включал создание страницы аутентификации пользователя и отображение существующих данных о химических веществах в папках.

Важной задачей при разработке подобных приложений является ограничение доступа к использованию программного продукта. В нашем случае каждый пользователь имеет свой собственный логин и пароль. Аутентификация проводится через отправку запроса на сервер. Для взаимодействия приложения с сайтом (сервером) используется библиотека .NET Refit, которая позволяет применять протокол REST и формат Json для создания простого интерфейса с понятным набором входных и выходных параметров. В запросе отправляется e-mail, password, а также булевский флаг returnSecureToken, который будет указывать, необходимо ли серверу вернуть токен (электронный ключ для доступа к чему-либо). Токен будет необходим в дальнейшем для получения данных с сервера. Модуль аутентификации содержит три поля для ввода (User Name, Account и Password) и кнопку Login, касание к которой будет запускать процесс аутентификации.

Непосредственно реализация логики данной страницы представлена в классе LoginPageModel. Основным в данном классе является метод CheckLogin, который, в свою очередь, вызывает метод LoginAsync.

Метод CheckLogin:

```
private async Task CheckLogin()
{
    IsAccountValid = !string.IsNullOrEmpty(Account);
    IsUserNameValid = !string.IsNullOrEmpty(UserName);
    IsPasswordValid = !string.IsNullOrEmpty>Password);

    if (IsAccountValid && IsUserNameValid && IsPasswordValid)
    {
        ErrorMessage = «»;
        if (networkInfo.GetNetworkState())
        {
            var authSuccess = await LoginAsync();
            if (authSuccess)
            {

```

```

var mainPage = FreshPageModelResolver.ResolvePageModel<StartPageModel>();
var container = new FreshNavigationContainer(mainPage,
    App.NavigationContainerName.MainContainer);
container.BarBackgroundColor = Color.FromHex(«#262831»);
container.BarTextColor = Color.FromHex(«#e2d2b6»);

CoreMethods.SwitchOutRootNavigation(App.NavigationContainerName.MainContainer);
}
else
{
    ErrorMessage = «Authorization error»;
}
else
{
    await CoreMethods.DisplayAlert(«Network error», «Network connection is not available», «OK»);
}
else
{
    ErrorMessage = «Empty field!»;
}

```

В этом методе происходит проверка корректности введенных данных и контроль состояния сети (есть ли интернет-соединение). В случае отрицательного результата проверки выводится информация об ошибке. Если все условия выполнены, вызывается метод `LoginAsync` и при успешном его завершении происходит переход на основную страницу.

Метод `LoginAsync`, используя сервис `WebClientService`, отправляет запрос на сервер. Если ответ получен, происходит сохранение токена, в противном случае отображается ошибка:

```

private async Task<bool> LoginAsync()
{
    bool success = false;
    try
    {
        Visibility = true;
        var response = await webClientService.AuthenticateAsync(Account, Password);
        success = response != null;
        if (success)
        {
            AppContext.Token = response.IdToken;
            SaveApiToken(response);
            SaveApi(response);
        }
        catch (NetworkException)
        {
            await CoreMethods.DisplayAlert(«Network error», «Network connection is not available», «OK»);
        }
        catch (OtherExceptions)
        {
            await CoreMethods.DisplayAlert(«Authorization error», «Please check entered data and then try again.», «OK»);
        }
        finally
        {
            Visibility = false;
        }
        return success;
    }
}

```

Здесь же реализована логика, позволяющая запоминать состояние пользователя (авторизован ли он), для того, чтобы в следующий раз при запуске приложения не нужно было снова вводить персональные данные.

Чтобы узнать, где именно находится конкретное вещество, при помощи страницы-проводника организуется поиск папок. Папки создаются в соответствии с требованиями пользователей и могут соответствовать, например, организационной структуре предприятия. В приложении отображена текущая иерархия папок с их содержимым. Благодаря набору папок обеспечивается быстрый поиск материалов на складе.

Визуальное представление страницы-проводника реализовано в классе SearchBrowsePageModel.xaml.cs. Данные подгружаются через GET запрос, с помощью интерфейсов IFoldersApi для получения папок и IMaterialsApi для получения сведений о материалах (химических веществах). Получение данных происходит в момент инициализации данной страницы – в методе InitAsync.

В методе InitAsync происходит отправка запроса для получения папок и запроса для получения сведений о материалах (химических веществах). Модель ответа FoldersResponse имеет поля Name, Id, ParentId.

После получения данных происходит выборка папок первого поколения (начальное состояние страницы-проводника) с помощью запросов LINQ (Language-Integrated Query). В качестве источника данных могут выступать объекты различных типов, но независимо от типа источника LINQ работает одинаково. На этой же странице организован поиск веществ с предоставлением всей интересующей пользователя информации с использованием фильтров, сортировки по нескольким критериям, статуса (всего 3 статуса: active – в данный момент вещество используется, located – имеется на складе, pending – ожидание поступления на склад). При изменении значения поля “Status” автоматически происходит выборка материалов, статус которых соответствует выбранному, и обновление их свойств.

Со страницы материалов есть возможность перейти к редактированию конкретного химического вещества и внести необходимые данные.

Для отображения месторасположения конкретного химического вещества на складе или в структурном подразделении предприятия подключаются карты Google Map. Создание этого блока связано с тем, что часто различные подразделения предприятия территориально разделены.

Взаимодействие созданного прототипа мобильного приложения с сайтом учета химических веществ, используемых на производстве, обеспечивает реализацию основных требований к программному продукту, в том числе ограничение доступа к хранящимся данным, создание пользователями удобной файловой системы для быстрого доступа к необходимым данным, подключение карт Google Map для отображения местоположения конкретного химического вещества, редактирование и сортировку данных.

КВАНТОВО-ХИМИЧЕСКИЙ РАСЧЕТ ФЛУДАРАБИН ФОСФАТА С ЭМОКСИПИНОВОЙ СОЛЬЮ (F-PE) QUANTUM-CHEMICAL CALCULATION OF FLUDARABINE-P WITH EMOXIPINE SALT (F-PE)

***М. А. Атрошко, С. Альбасри
M. A. Atroshko, S. Albasri***

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

Belarusian State University, ISEI BSU, Minsk, Republic of Belarus

atroshkomikhail@gmail.com

yuridragonv@gmail.com

В работе приведены данные полуэмпирических и теоретических расчетов молекул в среде растворителя, их спектр поглощения и оптимизированная структура с значением полной энергии системы.

The paper presents the data of semi-empirical and theoretical calculations of molecules in the medium of the solvent, their absorption spectrum and the optimized structure with the value of the total energy of the system.

Ключевые слова: PM6, DFT, спектр.

Keywords: PM6, DFT, spectra.

<https://doi.org/10.46646/SAKH-2021-2-398-401>

Предварительное квантово-химическое моделирование молекулы

Для расчетов использован персональный компьютер с процессором intel core i7 (2.21 GHz CPU) с установленной операционной системой Ubuntu 18.04. При вычислениях стартовой геометрии молекулы с азометиновым основанием выбран метод молекулярной механики (ММ⁺) программного пакета HyperChem 08. Выбор метода ММ⁺ обоснован тем, что он разработан для органических молекул, учитывает потенциальные поля, формируемые всеми атомами рассчитываемой системы, и позволяет гибко модифицировать параметры расчета в зависимости от конкретной задачи. Стартовую геометрию молекулы дополнительно оптимизировали в среде растворителя воды (water) полуэмпирическим методом PM6 программного пакета Gaussian 16 до достижения глобального минимума