**Сессия 1**

**Модуль 3: Разработка баз данных, объектов баз данных и импорт**

**Восстановление базы данных из скрипта**

Для восстановления таблиц в созданную базу данных воспользуйтесь предоставленным скриптом (ms.sql или my.sql в зависимости от выбранной СУБД). В процессе разработки приложения Вы должны доработать эту базу данных, обеспечив хранение всех данных, которые предоставил Вам заказчик. Для этого вам необходимо будет добавить необходимые сущности, атрибуты и связи.

**Импорт данных**

Заказчик системы предоставил файлы с данными (с пометкой import в ресурсах) для переноса в новую систему. Подготовьте данные файлов для импорта и загрузите в разработанную базу данных.

**Модуль 4: Разработка desktop-приложений**

**Авторизация**

При запуске приложения окно входа – первое, что видит пользователь. На ней пользователю предлагается ввести свой логин и пароль или есть возможность перейти на экран просмотра товаров в роли гостя.

Только после удачной авторизации пользователь получает доступ к остальным модулям системы:

* авторизованный клиент может просмотреть товары;
* менеджер может просматривать товары;
* администратор может добавлять/редактировать/удалять товары.

Реализуйте необходимые интерфейсы для всех пользователей системы. После входа в любую учетную запись должна быть реализована возможность выхода на главный экран – окно входа. При переходе в любую учетную запись в интерфейсе (правый верхний угол) должны отображаться ФИО пользователя.

После первой попытки неуспешной авторизации система выдает сообщение о неуспешной авторизации, а затем помимо ввода логина и пароля просит ввести captcha, состоящую из 4 символов (цифры и буквы латинского алфавита) и графического шума.

CAPTCHA - должна содержать минимум 4 символа (буква или цифра), которые выведены не в одной линии. Символы должны быть либо перечеркнуты либо наложены друг на друга.

После попытки неудачной авторизации с вводом captcha, система блокирует возможность входа на 10 секунд.

**Список товаров**

Очень важно постоянно получать актуальную информацию об остатках товаров на складах, поэтому необходимо реализовать вывод товаров, которые хранятся в базе данных, согласно предоставленному макету. При отсутствии изображения необходимо вывести картинку-заглушку из ресурсов (picture.png) .

Вывод должен осуществляться списком.

**Наименование товара**

Описание товара

Производитель:

Цена:

Наличие на складе

Фото

Обратите внимание, должна выводится вся информация о товаре из базы данных.

Товары, которых нет на складе, должны отображаться на сером фоне.

Пользователь должен иметь возможность искать товары, используя поисковую строку. Поиск должен осуществляться по всем доступным данным, в том числе по нескольким атрибутам одновременно.

Пользователь должен иметь возможность отсортировать товары (по возрастанию и убыванию) по стоимости.

Кроме этого, пользователь должен иметь возможность отфильтровать данные по производителю. Выбор производителя должен быть из выпадающего списка. Первым элементом в выпадающем списке должен быть “Все производители”, при выборе которого настройки фильтра сбрасываются.

Поиск, сортировка и фильтрация должны происходить в реальном времени, без необходимости нажатия кнопки “найти”/ ”отфильтровать” и т.п. Фильтрация и поиск должны применяться совместно. Параметры сортировки, выбранные ранее пользователем, должны сохраняться и во время фильтрации с поиском.

В верхней части окна необходимо показывать количество выведенных данных и общее количество записей в базе. Например, 15 из 37 . В случае, если данные в таблицу выводятся после фильтрации или поиска, количество выведенных данных необходимо обновить исходя из размера выборки.

Реализуйте удаление товара. Товар, который присутствует в заказе, удалить нельзя. Товар, у которого есть дополнительные товары, удаляется из базы данных вместе с информацией о дополнительных товарах, если ни один из товаров (даже дополнительный) не заказан.

Реализуйте переход на окно добавления товара и окно редактирования выбранного товара.

**Добавление/ редактирование товаров**

Необходимо реализовать возможность добавления и редактирования исходных товаров (сырья) для всей продукции компании. Редактировать данные может только администратор.

Необходимо добавить возможность редактирования данных существующего товара, а также добавление нового товара в новом окне - форме для добавления/ редактирования товара. Переходы на данное окно должны быть реализованы из формы списка: для редактирования - при нажатии на конкретный элемент, для добавления - при нажатии кнопки “Добавить товар”.

На форме должны быть предусмотрены следующие поля: наименование, категория (выпадающий список), количество на складе, единица измерения, поставщик, стоимость за единицу, изображение и подробное описание (с возможностью многострочного ввода) . Стоимость товара может включать сотые части, а также не может быть отрицательной. Минимальное количество также не может принимать отрицательные значения.

При открытии формы для редактирования все поля выбранного объекта должны быть подгружены в соответствующие поля из базы данных, а таблица заполнена актуальными значениями.

ID товара при добавлении не отображается, автоматически вычисляется +1 к имеющемуся в БД, при редактировании ID доступно только для чтения.

Пользователь может добавить/ заменить изображение у товара. Изображение, которое загружает пользователь при добавлении или редактировании, должно сохраняться в папку с приложением. Для оптимального объема реализуйте ограничение на размер фото: 300Х200 пикселей. В базе данных необходимо хранить путь к изображению. При замене изображения, старое фото из папки должно быть удалено.

Для того, чтобы администратор случайно не изменял несколько товаров, предусмотрите невозможность открытия более одного окна редактирования.

После редактирования/добавления/удаления товаров данные в окне списка товаров должны быть обновлены.

**Разработка библиотеки**

Разработайте библиотеку (отдельным проектом), которая позволит вернуть список свободных временных интервалов (заданного размера) в графике сотрудника для формирования оптимального графика работы сотрудников.

Необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

Вход:

* список занятых промежутков времени (в двух массивах: startTimes - начало, durations - длительность),
* минимальное необходимое время для работы менеджера (consultationTime),
* рабочий день сотрудника (начало - beginWorkingTime и завершение - endWorkingTime)

Выход:

* список подходящих свободных временных промежутков (в массив строк формата HH:mm-HH:mm)

Требования к именованиям и форматам:

|  |  |  |  |
| --- | --- | --- | --- |
|  | C# | Java | Python |
| Библиотека классов | SF2022User{NN}Lib.dll | SF2022User{NN}Lib.jar | SF2022User{NN}Lib |
| Название класса | Calculations | Calculations | Calculations |
| Название метода | AvailablePeriods() | availablePeriods() | available periods() |
| Входящие обязательные параметры | TimeSpan[] startTimes,  int[] durations,  TimeSpan beginWorkingTime,  TimeSpan endWorkingTime,  int consultationTime | LocalTime[] startTimes,  int[] durations,  LocalTime beginWorkingTime,  LocalTime endWorkingTime,  int consultationTime | time[] start times,  int[] durations,  time begin working time,  time end working time,  int consultation time |
| Возвращаемые параметры | string[] | string[] | string[] |

Пример:

|  |  |
| --- | --- |
| Вход | Выход |
| startTime | duration  10:00 60  11:00 30  15:00 10  15:30 10  16:50 40  Working Times  08:00-18:00  Consultation Time  30 | 08:00-08:30  08:30-09:00  09:00-09:30  09:30-10:00  11:30-12:00  12:00-12:30  12:30-13:00  13:00-13:30  13:30-14:00  14:00-14:30  14:30-15:00  15:40-16:10  16:10-16:40  17:30-18:00 |