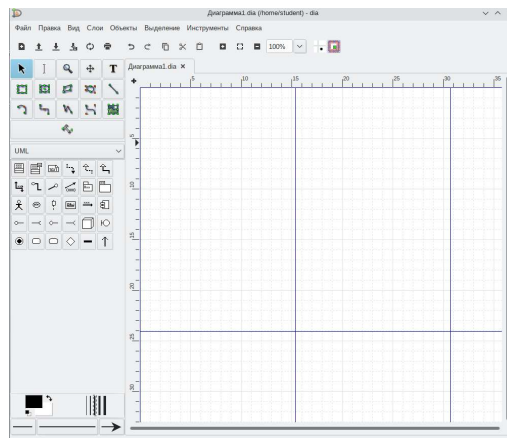


Выполнение Модуль 1.

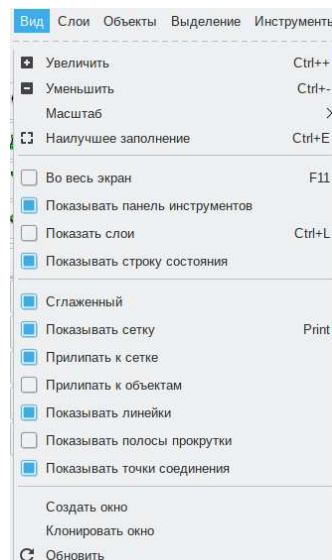
1. Запускаем ярлык DIA:




2. Выбираем инструмент UML:

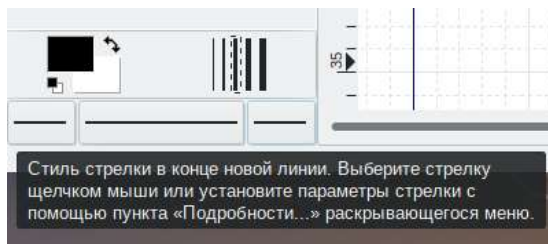



Перед началом построения диаграмм, в панели инструментов выбрать пункт «Вид», выставить в активное положение «Прилипание к сетке», «Показывать точки соединения», чтоб выравнивать элементы диаграммы.

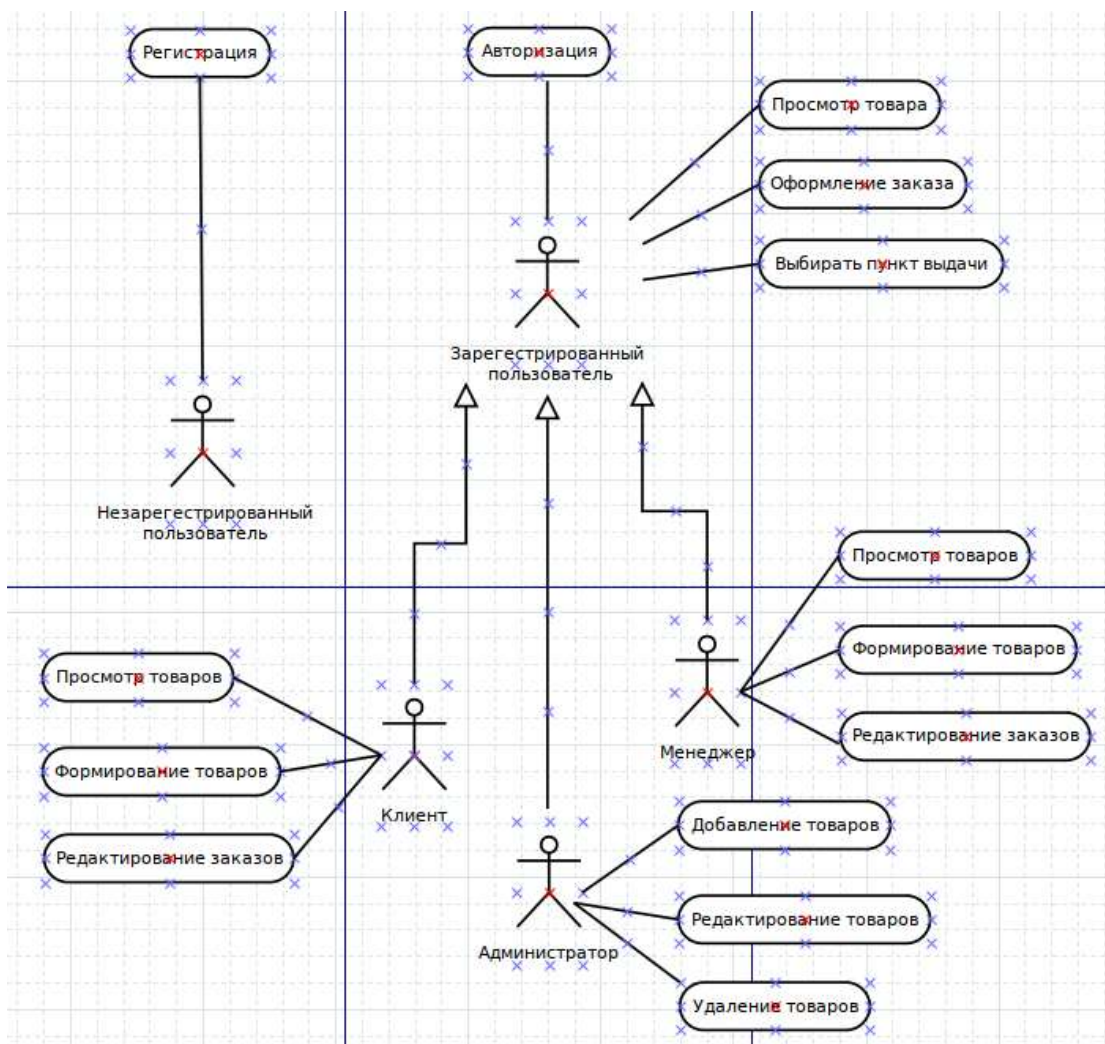


3. Определяем требования к информационной системе на основе анализа описания предметной области и разрабатываем Use-Case диаграмму.

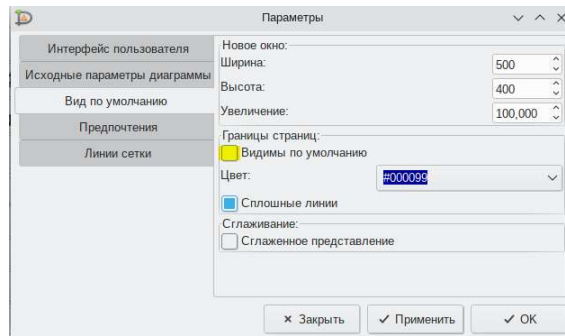
Используем инструмент «Линия» для соединения объектов в диаграмме . В нижнем левом углу программы изменяем стиль линий на прямую, без стрелочек.



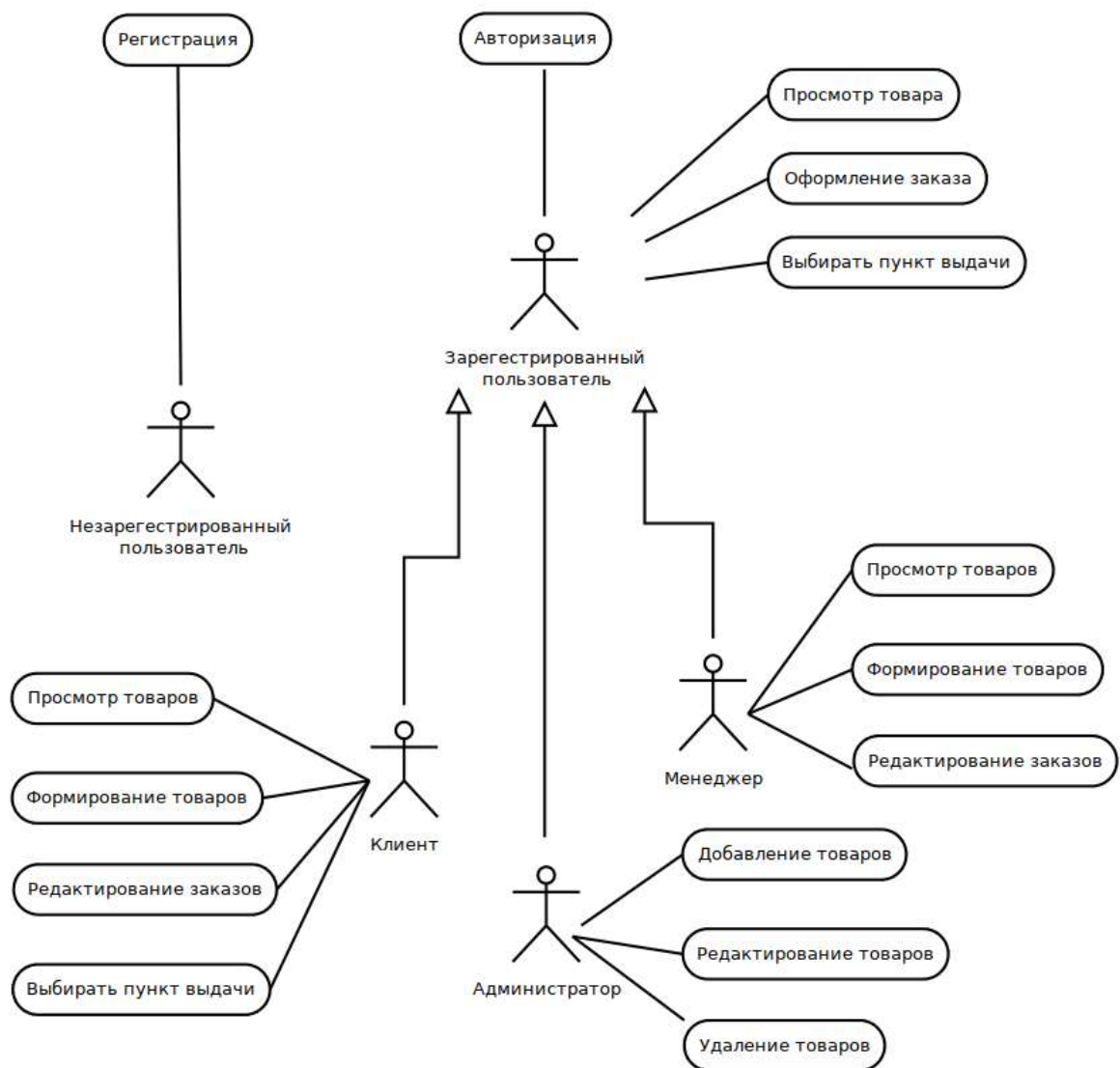
Для текстового сопровождения создаваемых объектов, необходимо сперва выбрать режим ввода текста , после чего выбрать объект и внести необходимые комментарии. Черновой вариант Use-Case диаграммы представлен ниже



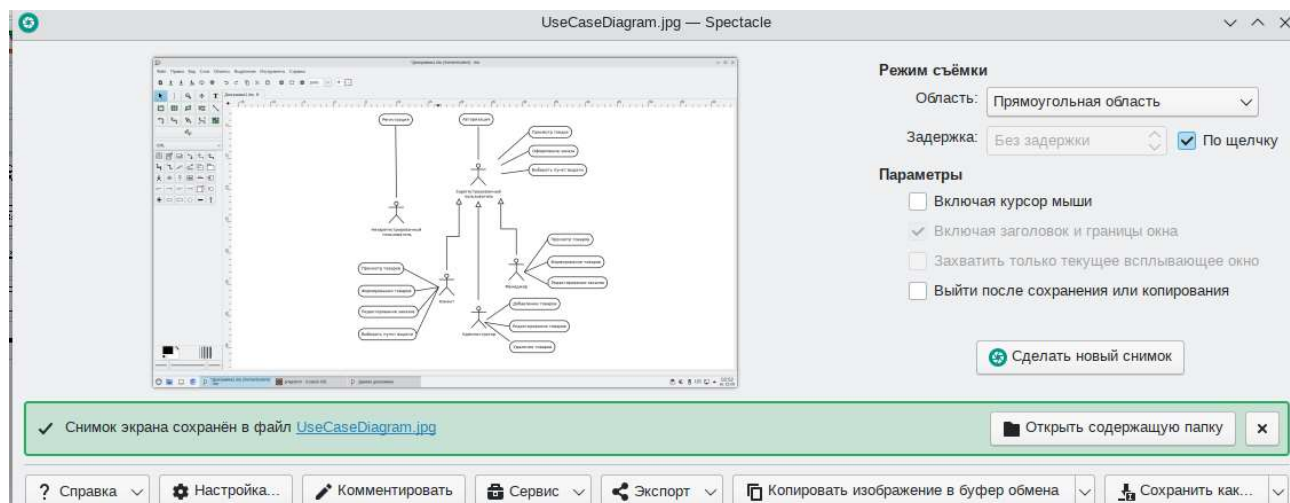
Отключить видимость границ листа: Файл>Параметры> Вид по умолчанию.



Получим упрощенную Use Case диаграмму



Для быстрого сохранения диаграммы воспользуемся встроенным скринером ALT Linux. Предварительно настроить режим съемки на выделение прямоугольной области по клику мышки.



Сохраняем полученный скрин диаграммы под названием **UseCaseDiagram.jpg** .
Также сохраняем проект в **DIA** под названием **Diagram.dia** .

3. Описываем действующие лица бизнес-процессов.

Пользователи информационной системы:

- актер «**неавторизованный пользователь**» может просмотреть товары, сформировать заказ и выбрать удобный для него пункт выдачи.
- актер «**клиент**» может просматривать товары, формировать, редактировать заказы и выбрать удобный для него пункт выдачи.
- актер «**менеджер**» может просматривать товары, формировать, редактировать заказы.
- актер «**администратор**» добавлять/редактировать/удалять товары.

Данные о пользователях ИС занести в файл **README-Template_rus.md**

ЗАДАНИЕ 1.

Самостоятельно создать диаграмму деятельности (**Activity**) и диаграмму последовательностей (**Sequence**) для данной задачи. Сохранить соответственно полученные диаграммы под названием

ActivityDiagram.jpg и **SequenceDiagram.jpg**.

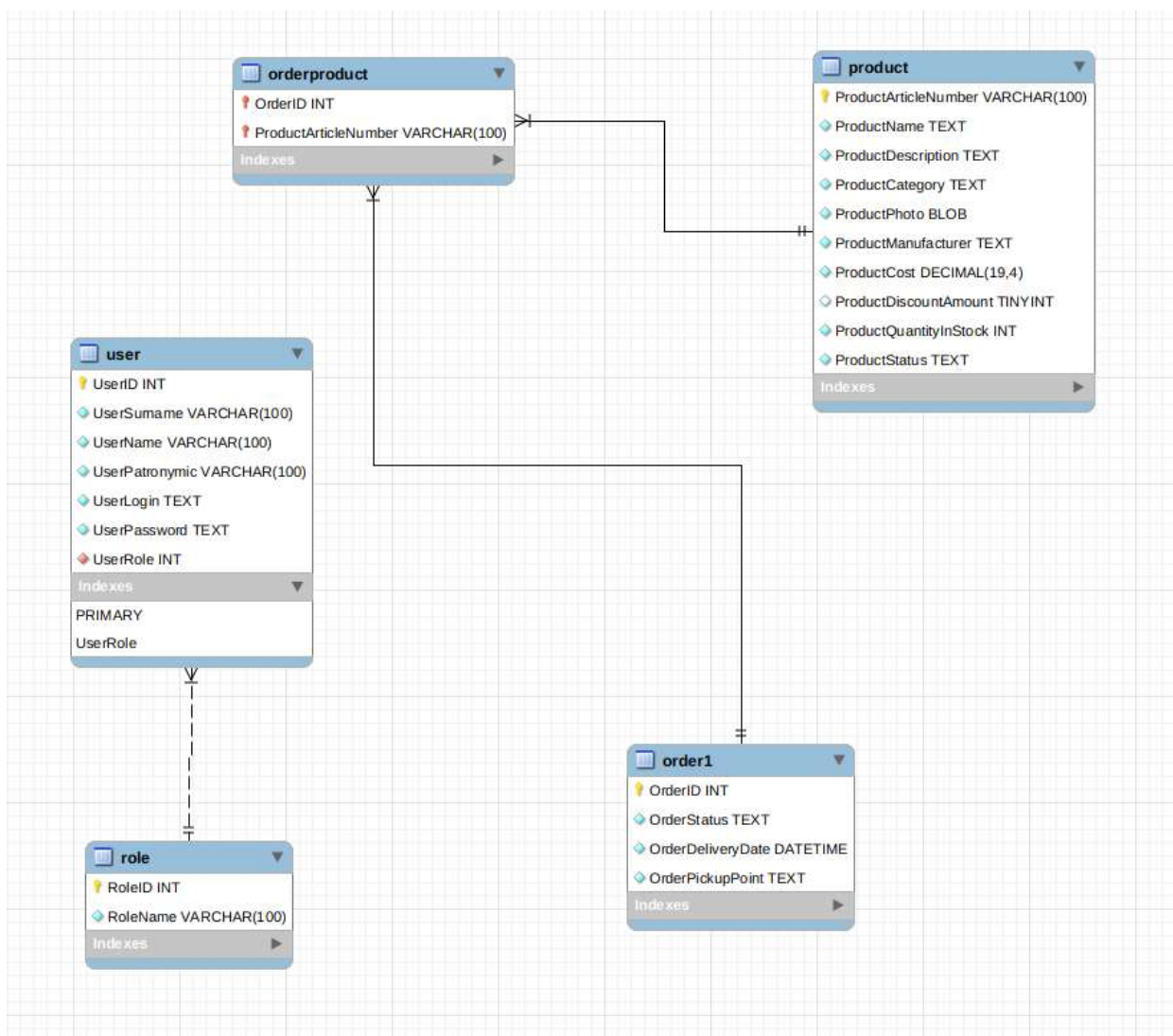
Выполнение Модуль 2

- спроектировать на их основе диаграммы сущность-связь ERD,
- Создать словарь данных.

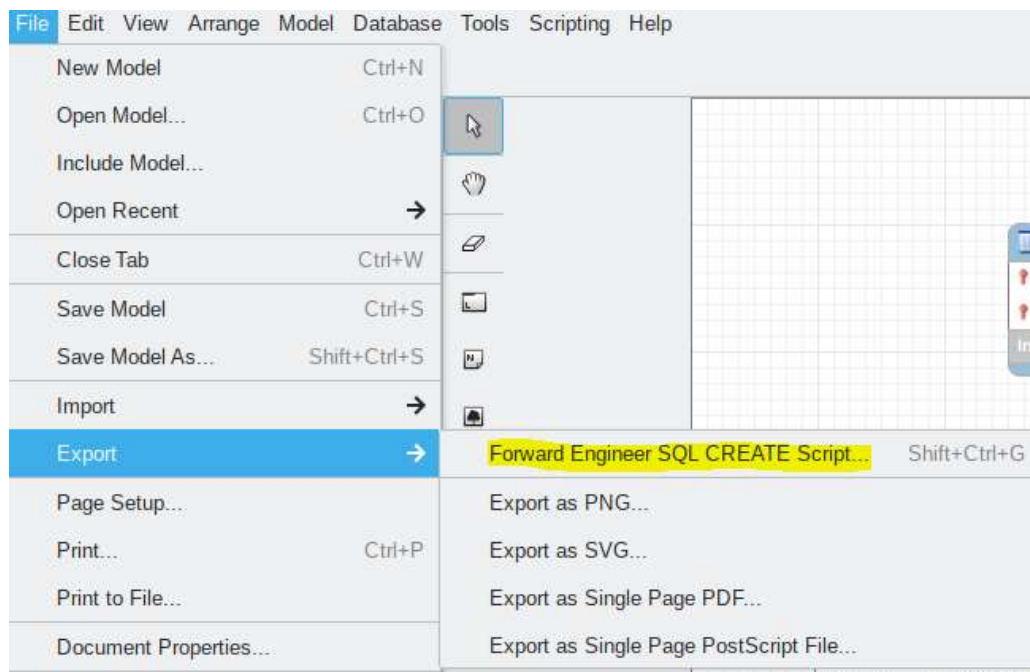
Для реализации бизнес-процесса продажи товаров участвуют следующие объекты:

- пользователь.
- продукция.
- место выдачи продукции.

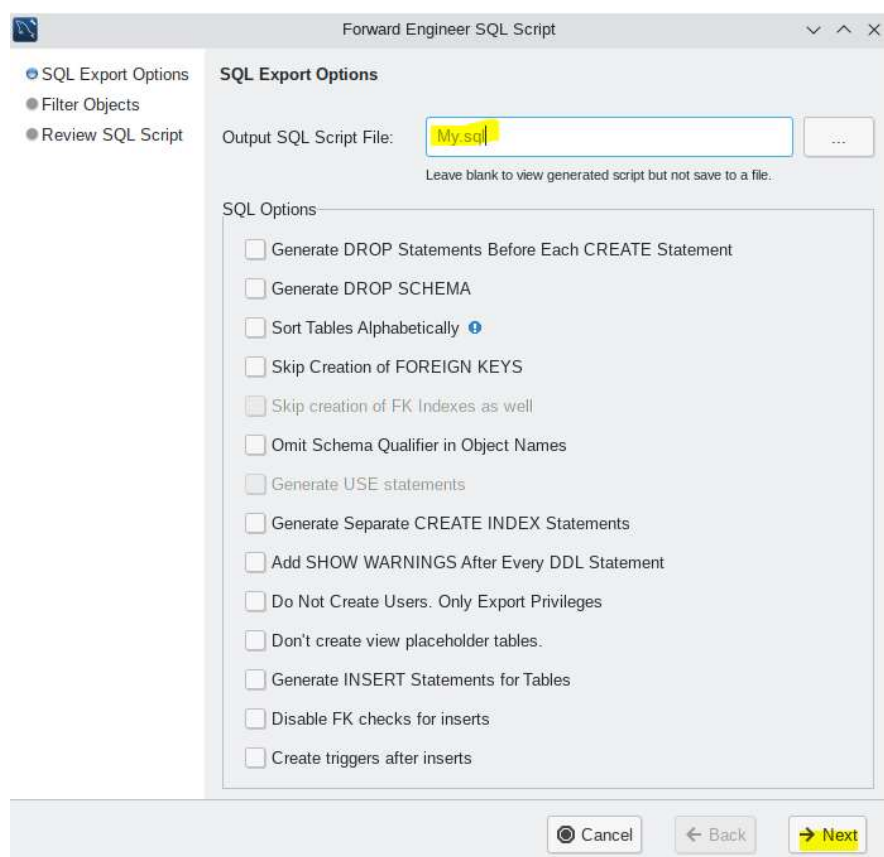
Спроектируем на основе входных данных диаграмму сущность-связь ERD в Mysql Workbench.



Дополнительно создать sql-скрипт создания схемы баз данных



Сохранить файл под названием **My.sql**



- Создать словарь данных на основе ERD диаграммы.

Воспользуемся шаблоном и заполним его согласно **My.sql**

Data Dictionary

Шаблон

Role

KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
PK	Role	int	Y	Auto increment
	RoleName	varchar(100)	Y	

User

KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
PK	UserID	int	Y	Auto increment
	UserSurname	varchar(100)	Y	
	UserName	varchar(100)	Y	
	UserPatronymic	varchar(100)	Y	
	UserLogin	text	Y	
	UserPassword	text	Y	
FK	UserRole	int	Y	foreign key (UserRole) references Role(RoleID)

Product

KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
PK	ProductArticleNumber	nvarchar(100)	Y	
	ProductName	text	Y	
	ProductDescription	text	Y	
	ProductCategory	text	Y	
	ProductPhoto	blob	Y	
	ProductManufacturer	text	Y	
	ProductCost	decimal(19,4)	Y	
	ProductDiscountAmount	tinyint	N	
	ProductQuantityInStock	int	Y	
	ProductStatus	text	Y	

Order1

KEY	FIELD NAME	DATA TYPE / FIELD SIZE	REQUIRED?	NOTES
PK	OrderID	int	Y	Auto increment foreign key (OrderID) references Order1(OrderID)

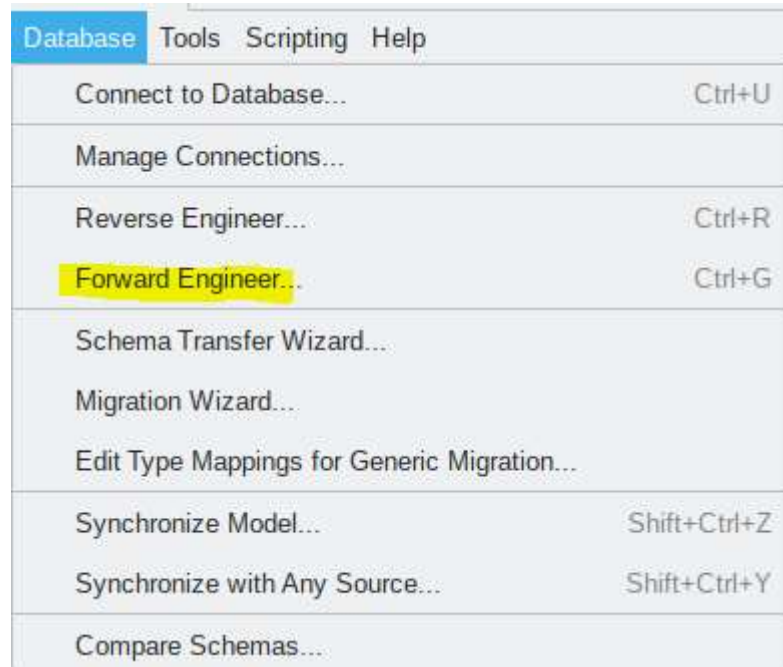
				foreign key (ProductArticleNumber) references Product(ProductArticleNumber)
	ProductArticleNumber	nvarchar(100)	Y	
	OrderDeliveryDate	datetime	Y	
	OrderPickupPoint	text	Y	

Сохранить словарь данных под названием **DataDictionary.docx**

Выполнение Модуль 3

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

На основе созданной ER-диаграммы с помощью Forward Engineer создаем базу данных в выбранной СУБД.



Импортируем данные в базу данных.

ВАЖНО!

По завершению выполнения **сессии 1** необходимо загрузить в папку **Session_1** следующие файлы:

UseCaseDiagram.jpg,

ActivityDiagram.jpg,

SequenceDiagram.jpg,

Diagram.dia,

My.sql,

DataDictionary.docx,

README-Template_rus.md в **GOGS**.

Создать архив **Session_1.zip**, загрузить его в **GOGS**.

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.

САРТСНА - 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.

<div><div></div><div>Фото</div></div>	<div>Наименование товара</div> <div>Описание товара</div> <div>Производитель:</div>	<div>Наличие на складе</div>
---------------------------------------	---	------------------------------

Обратите внимание, должна выводиться 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: 300X200 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	08:00-08:30 08:30-09:00 09:00-09:30 09:30-10:00 11:30-12:00 12:00-12:30

Working Times	12:30-13:00
08:00-18:00	13:00-13:30
Consultation Time	13:30-14:00
30	14:00-14:30
	14:30-15:00
	15:40-16:10
	16:10-16:40
	17:30-18:00

Модуль 7: Тестирование программных решений

Тестирование

Модульные тесты

Реализуйте 10 unit-тестов на основе технологии TDD для библиотеки, функционал которой описан ранее. Важно, чтобы тестовые данные предусматривали различные ситуации. Например, недостаточное время в промежутках между ранее созданными консультациями, либо в начале рабочего дня, либо в конце рабочего дня; различная длительность консультация и т.д.

Тестовая документация

Для выполнения процедуры тестирования добавления товара администратором Вам нужно описать пять сценариев. Добавление может быть выполнено успешно, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон `testing-template.docx`.