|  |  |
| --- | --- |
|  | ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА  ФАКУЛТЕТ ПО ИЗЧИСЛИТЕЛНА ТЕХНИКА И АВТОМАТИЗАЦИЯ |

**КУРСОВ ПРОЕКТ**

по

Обектно-ориентирано програмиране­ – 2. част

Група 6

**ЗАДАНИЕ** *№ 5*

В А Р Н А

2 0 2 2 год.

## V. Склад с наличности

Да се разработи информационна система, предоставяща услуга склад. Програмата съхранява и обработва данни за складови помещения. Системата позволява множествен достъп.

Системата поддържа два вида потребители администратор и оператори (складов агент) с различни роли за достъп до функционалностите в системата.

Операции за работа с потребители:

* Създаване на складови оператори от администратор;
* Създаване на доставчици;
* Създаване на клиенти;
* Създаване на каса (Парична наличност).

Системата поддържа операции за работа със събития:

* Създаване на номенклатури;
* Работа с фактури o Приемане на стока от доставчик на доставна цена; o Изписване на стока на продажна цена;
* Наблюдение за наличност на стоки в склада;
* Наблюдение за наличност на пари в касата;

Системата поддържа справки по произволен период за:

* Доставки и доставчици;
* Изписване и клиенти;
* Дейност на складовите оператори;
* За наличности в склада;
* Разходи,приходи,печалба.
* Движение на наличността в касата.

Системата поддържа Известия за събития:

* Критичен минимум и липса на стока;
* Критичен минимум и липса на парична наличност;

**Анализ на проблема**

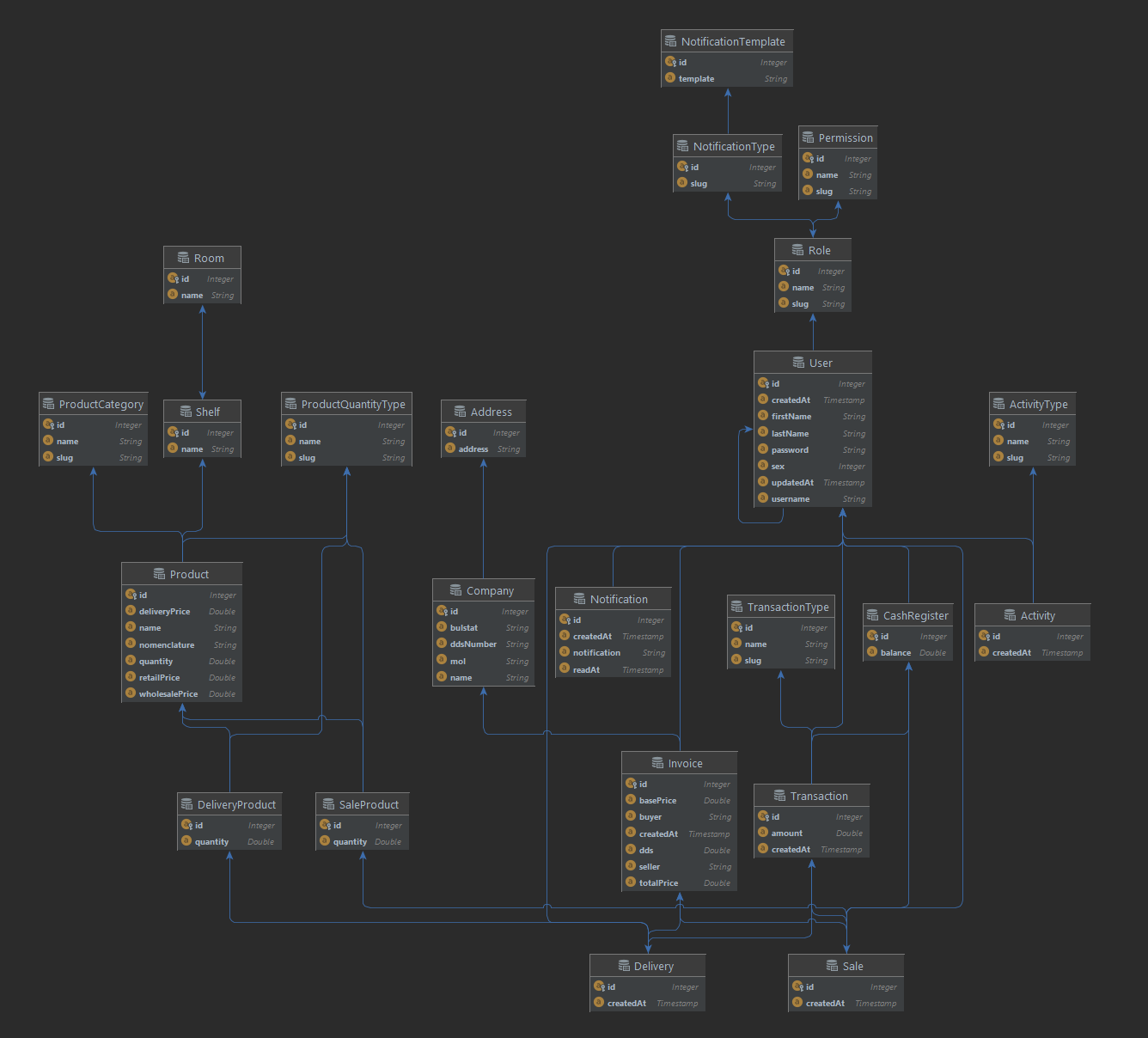
1. **Функционални изисквания:**

Софтуерът трябва да може да управлява и работи с множество потребители, да съхранява и обработва информация за складови помещения, каси и продукти. От него се изисква да поддържа създаване на фактури за доставка и продажба на стока, също и известия за критичен минимум на стоки и пари. На потребителя е необходимо да се предостави възможност за справка на стоки и пари в склада и касата съответно.

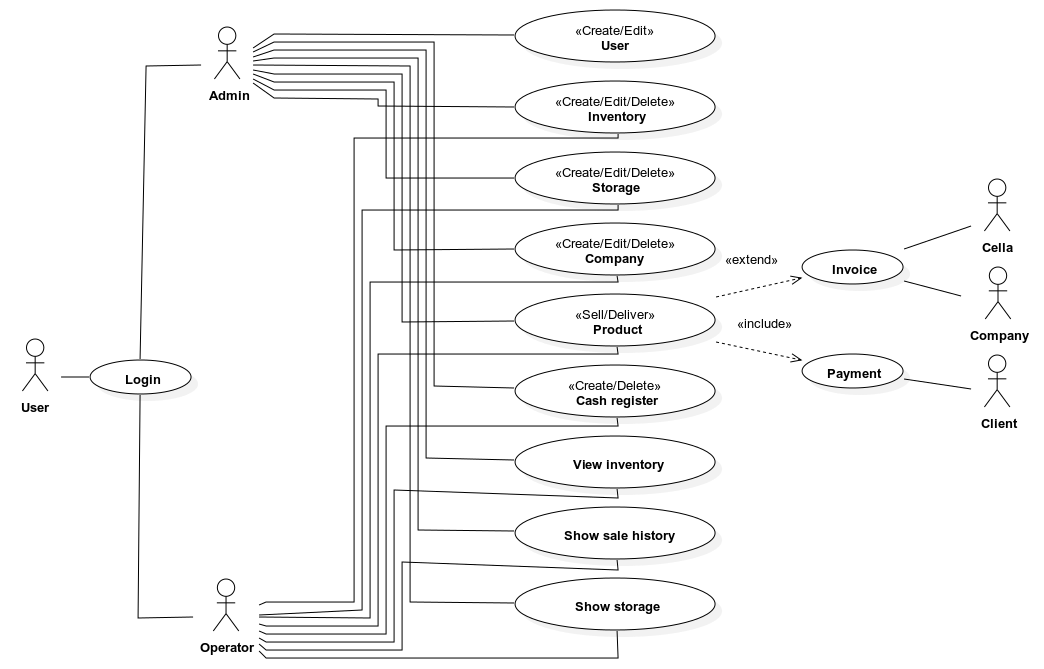
1. **Структура на проекта:** 
   * Контролери (controllers) – Всички класове, които управляват FXML файловете. Те заедно образуват презентационния слой.
   * Модели (entities) – Класовете, които прадставляват таблиците и техните връзки от базата данни.
   * Хранилища (repositories) – Всички класове, които управляват данните в базата данни. Заедно с моделите формират данновия слой.
   * Услуги (services) – Всички класове, които работят с данните от хранилищата.
   * Общи услуги (utils) – Класовете, които се използват от няколко услуги. Services и Utils образуват бизнес логиката.
   * Начална точка (warehouse) – Класовете за стариране на проекта.
   * Активи (assets) – Файлове за стилизация и изображения.
   * Конфигурационни файлове (configurations, META-INF)
   * Изгледи (views) – Всички FXML файлове.

**Проектиране на системата**

1. **Еntity-Relationship диаграма:**

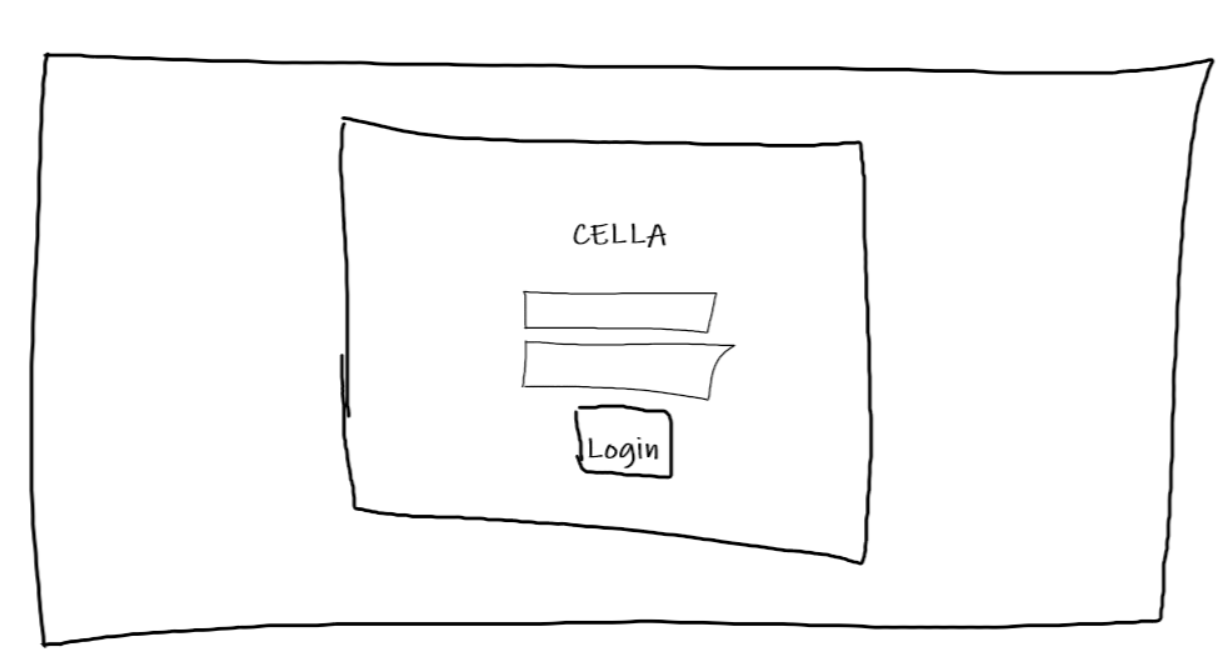


1. **Use case диаграма**

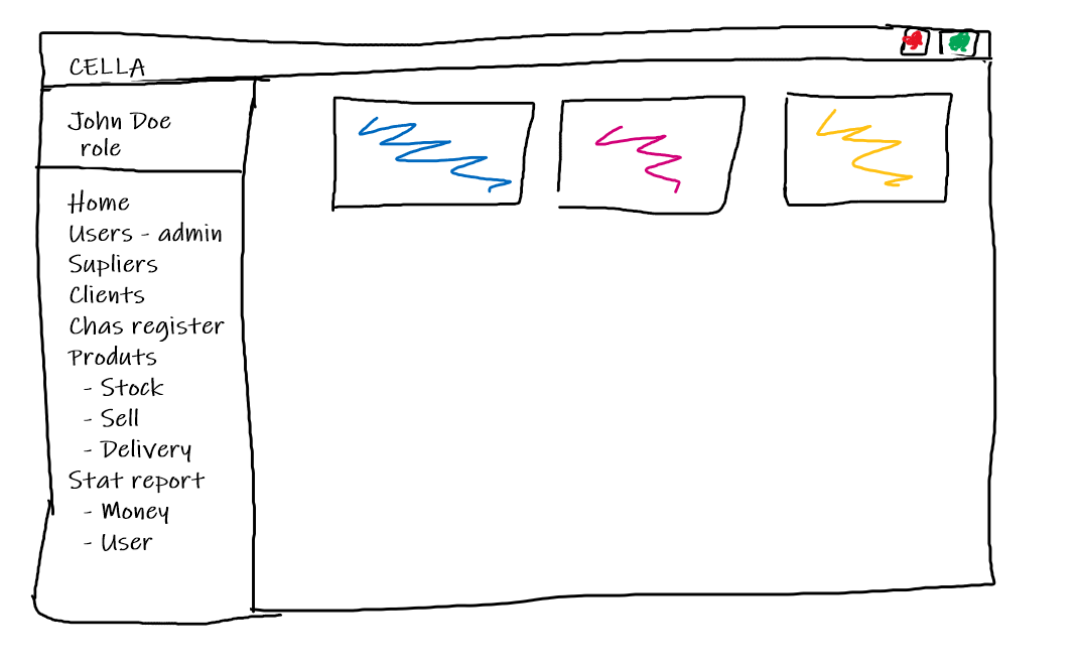


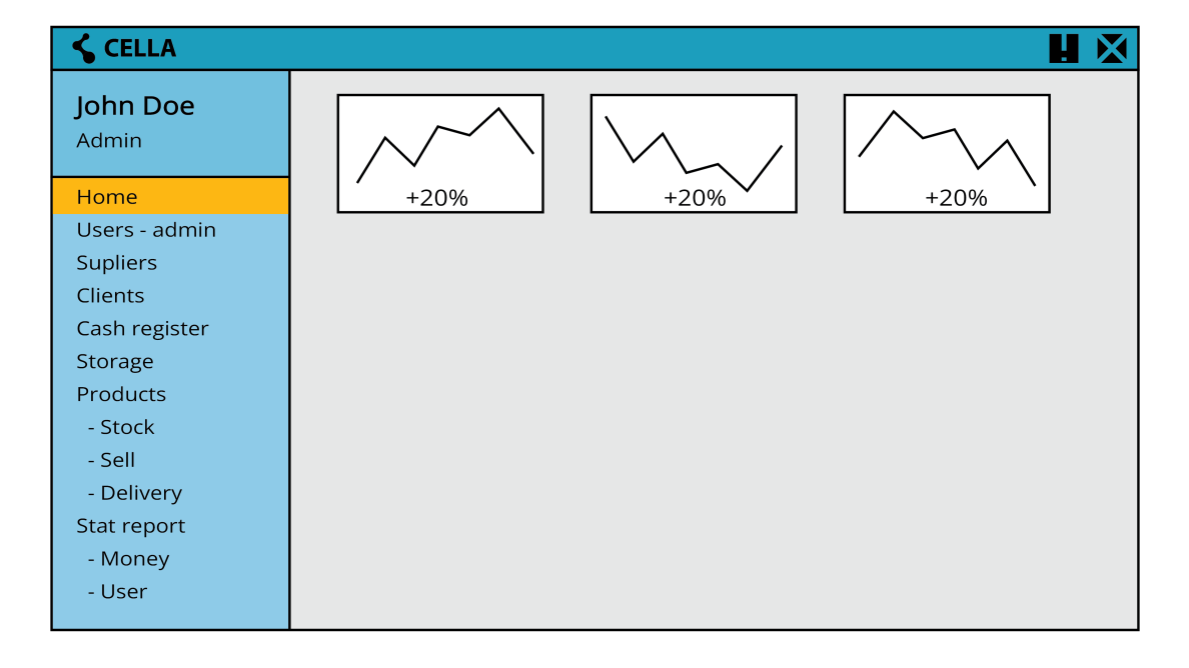
1. **Class диаграма**
2. **Sequence диаграма**
3. **Прототипи**

* Екран за вход

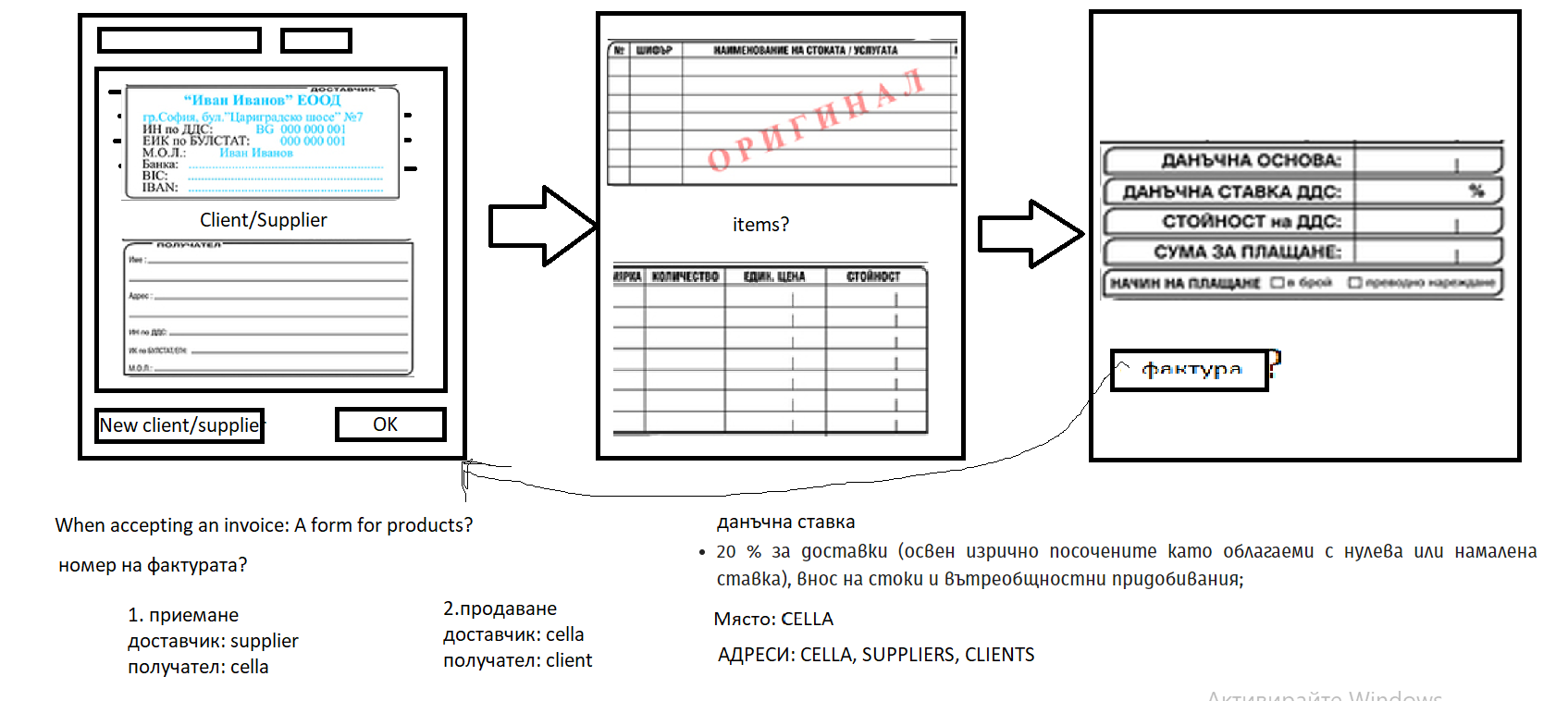


* Главен енран

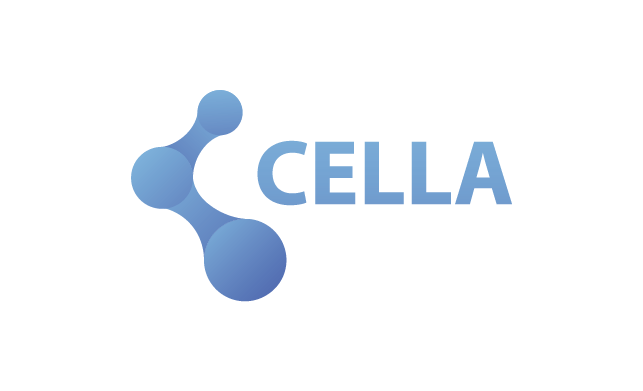




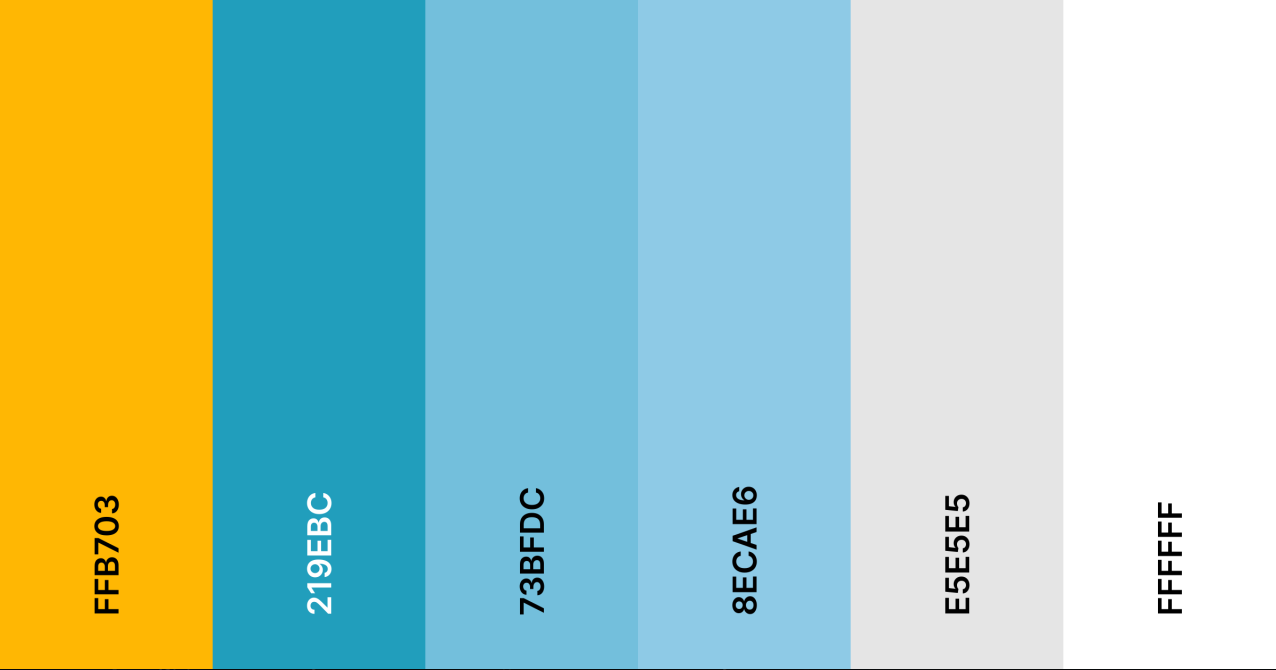
* Съветник (wizard) за доставка и продажба



1. **Лого и палитра**

****

****

****

**Реализация на системата**