3.Эскаизный проект

3.1. Общие положения

3.1.1 Полное наименование системы и ее условное обозначение

Полное наименование системы: Мебельный салон Вованчика.

Краткое наименование системы: МСВ.

3.1.2 Назначение системы

Разрабатываемая система предназначена для упрощения работы мебельного салона, выполненных по заказу.

3.1.3 Цели создания системы

Основными целями создания “Мебельный салон” являются:

- быстрый поиск нужных товаров;

- доступность для любого клиента;

- Список типов товара. все будет представлено на одной странице.

Для достижения поставленных целей система должна решать следующие задачи:

- наличие сформированной база данных различных товаров;

- добавление нового товара в список;

- редактирование товара в списке;

- удаление товара в списке;

На данном этапе разберем, то как будем вести себя система при выполнение выделенных модулей. Для этого мы будем использовать диаграммы состояний в нотации UML.

На рисунке

3.1 представлена общая диаграмма состояний. При запуске программы, пользователь будет находиться в том, состоянии, когда ему необходимо авторизоваться, чтобы продолжить работу с системой и осуществлять деятельность согласно выделенным вариантам использования. 

Рисунок 3.1. - Общая диаграмма состояний системы

На рисунке 3.2

показано детальное описание работы пользователя с системой. Можно просматривать список всех товаров.

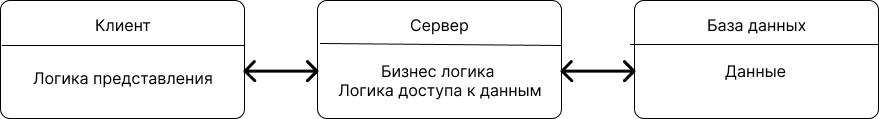
Схема клиент - серверной архитектуры представлена на рисунке 3.6.

Рисунок 3.6. - Клиент - серверная архитектура

Это одна из самых распространенных архитектур. Система делится на уровни, каждый из которых взаимодействует лишь с двумя соседними. Поэтому запросы к БД, которая обычно располагается в самом конце цепочки взаимодействия, проходят последовательно сквозь каждый «слой». Архитектура не подразумевает какое-то обязательное количество уровней — их может быть три, четыре, пять и больше.

Чаще всего используют трехзвенные системы: с уровнем представления (клиентом), уровнем логики и уровнем данных.

Такие архитектуры более разумно распределяют модули обработки данных, которые в этом случае выполняются на одном или нескольких отдельных серверах. Эти программные модули выполняют функции сервера для интерфейсов с пользователями и клиента - для серверов баз данных. Кроме того, различные серверы приложений могут взаимодействовать между собой для более точного разделения системы на функциональные блоки, выполняющие определенные роли. Схема модульной архитектуры представлена на рисунке 3.7.:



Нашу систему можно разделить на отдельные модули, представленные на рисунке 3.8.:

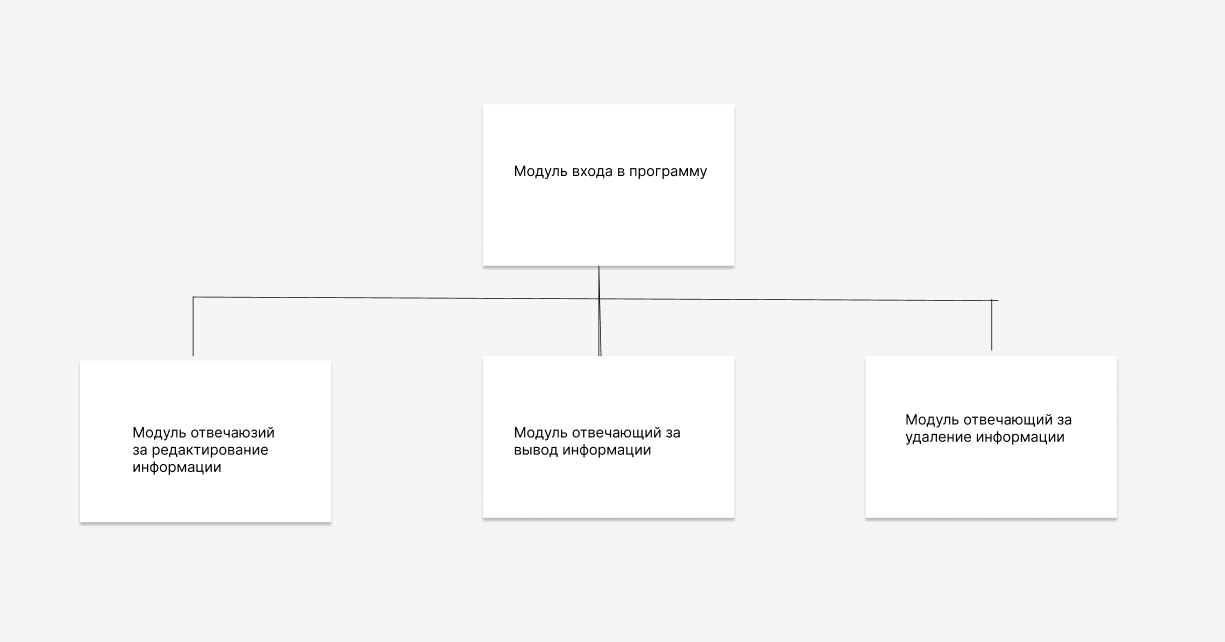


Рисунок 3.8. - Модули разрабатываемой системы

1. Модуль для входа программу. Благодаря нему, менеджер может зайти и пользоваться, введя свои данные.

2.Модуль получение и вывода информации. С помощью этого модуля пользователь может получать данные с БД, а так-же видеть информацию в таблице.

3.Модуль отвечающий за удаление информации. С помощью этого модуля пользователь может удалять нужную ему информацию из БД.

На рисунке 3.9. показана модель работы с системой, спроектированной по архитектуре клиент сервер.

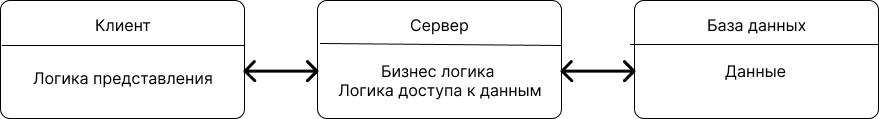


Рисунок 3.9. - Модель работы с системой, спроектированной по архитектуре клиент сервер

3.3.2. Решения по режимам функционирования, диагностированию работы программы

Программа должна поддерживать работу:

- основной режим, в котором все подсистемы выполняют свои основные функции;

В основном режиме функционирования “МСВ” должна обеспечивать:

- выполнение функций – сбор, обработка, загрузка и хранение данных.

В профилактическом режиме “МСВ” должна обеспечивать возможность проведения следующих работ:

- техническое обслуживание программы и БД;

Средняя доступность”МСВ” должна составлять не менее 99%

Среднее время между сбоями — это среднее время, за которое компонент или модуль может выполнять свои функции без перерыва.

Измеряется от начала работы до момента следующего сбоя.

Среднее временя работы без сбоев должно составлять не менее 1500 часов.