3.Эскизный 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.

- быстрый поиск необходимых товаров;

- удобства;

3.1.3 Цели создания системы

Разрабатываемая система предназначена для учета заказов в мебельном салоне

Система для учёта заказов, товаров, и клиентах позволит быстро просматривать необходимые данные.

Менеджер сможет с легкостью заносить информацию об товаре, клиентах и заказов, свободно редактировать и удалять, а также обеспечен удобный поиск по определенным параметрам.

Основными целями создания “Мебельный салон” являются:

- 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 будет вести себя программа 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 программы

На рисунке 3.2 показано детальное описание работы пользователя с системой. Можно просматривать список всех товаров, редактировать, добавлять, удалять и переходить в другие вкладки.

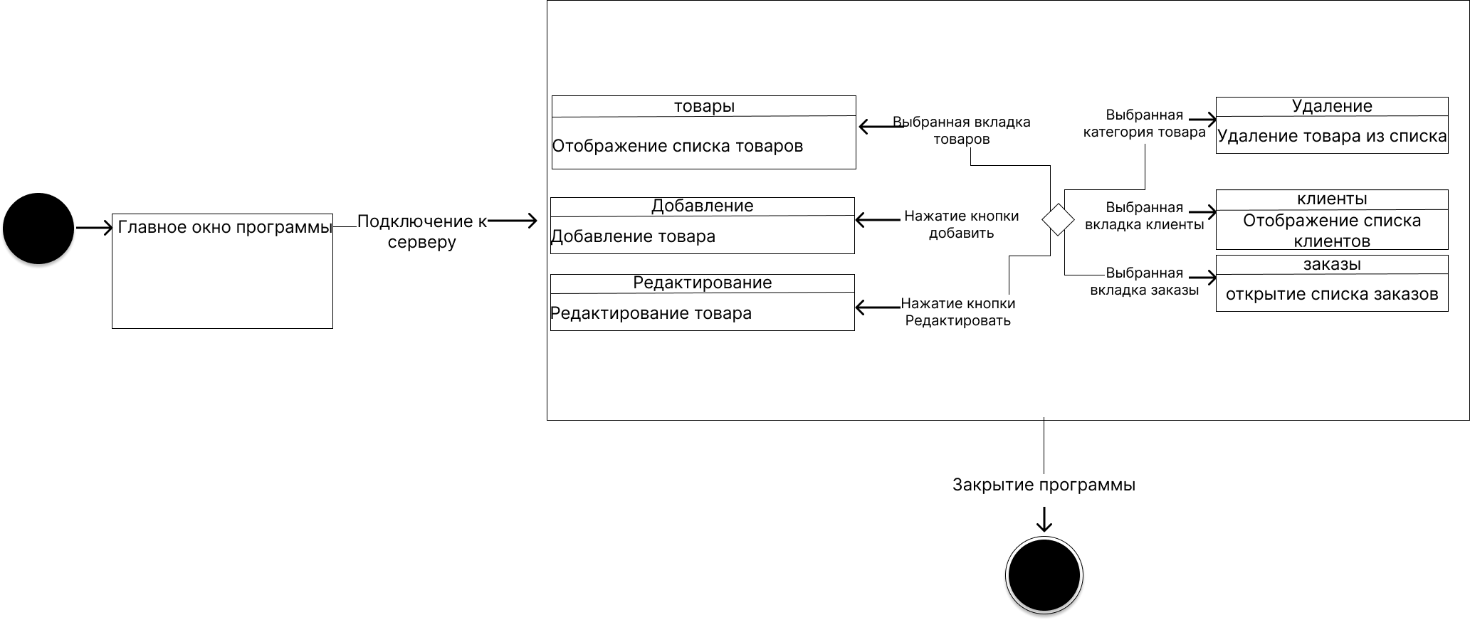


Рисунок 3.2 – Диаграмма состояния Просмотр списка заказов

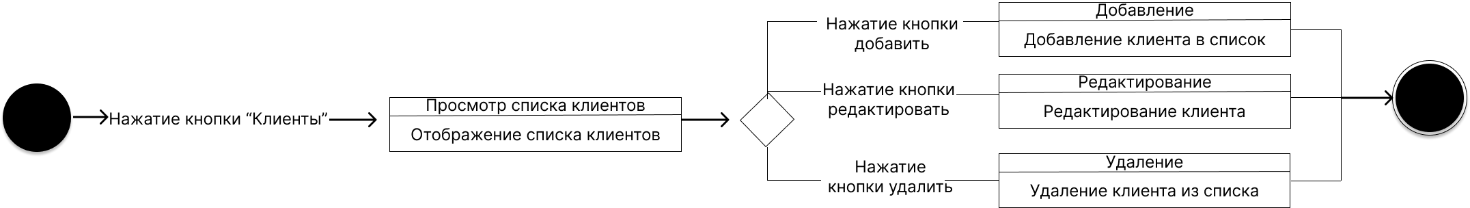
На рисунке 3.3, изображена декомпозиция состояния просмотра списка клиентов. После того как менеджер открыл программу, он может выбрать вкладку клиенты которая ему необходима. Далее он может добавлять, редактировать клиентов или же удалять.

Рисунок 3.3 – Диаграмма состояний "Просмотр списка заказов"

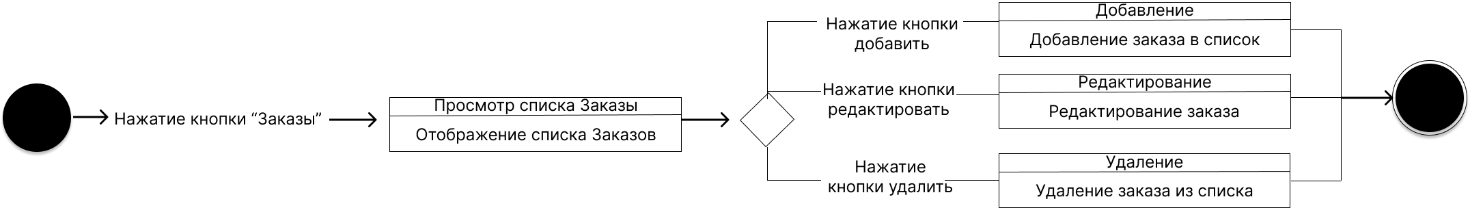
На рисунке 3.4, изображена декомпозиция состояния просмотра списка заказов. После того как менеджер открыл программу, он может выбрать вкладку клиенты которая ему необходима. Далее он может добавлять, редактировать заказы или же удалять.

Рисунок 3.4 – Диаграмма состояний "Просмотр списка клиентов"

На рисунке 3.5, изображена декомпозиция состояния просмотра списка товара. После того как менеджер открыл программу, он может выбрать вкладку клиенты которая ему необходима. Далее он может добавлять, редактировать товар или же удалять.

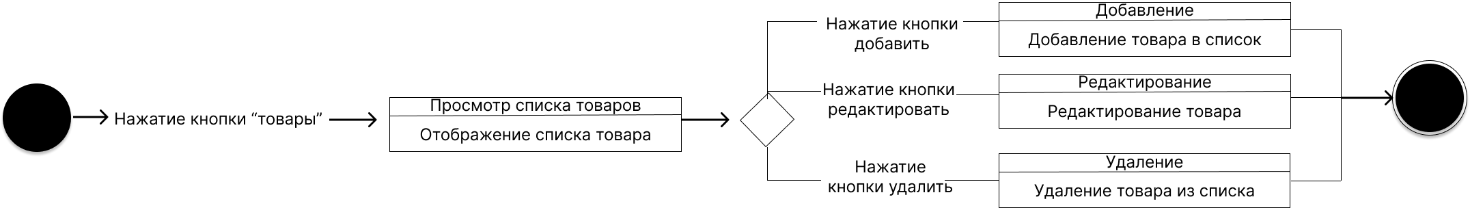


Рисунок 3.5 – Диаграмма состояний "Просмотр списка клиентов"

3.3 Основные технические решения

3.3.1 На данном этапе необходимо описать примерную архитектуру разрабатываемой системы.

Для того, чтобы сделать программу, мы будем использовать клиент-серверную архитектуру. Чтобы убедиться, что она нам точно подходит рассмотрим ее подробнее. В клиент серверной архитектуре имеется три звена:

- Представление данных — на стороне клиента.

- Прикладной компонент — на выделенном сервере приложений, здесь происходит вся бизнес-логика.

- Управление ресурсами — сервер БД, который и представляет запрашиваемые данные.

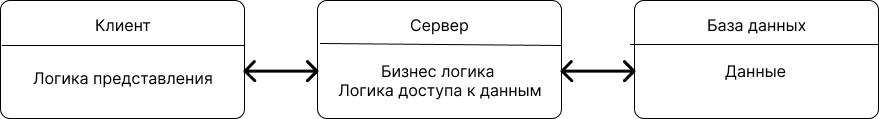
Схема клиент - серверной архитектуры представлена на рисунке 3.6.

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.:



Рисунок 3.7. – Многоуровневая архитектура

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

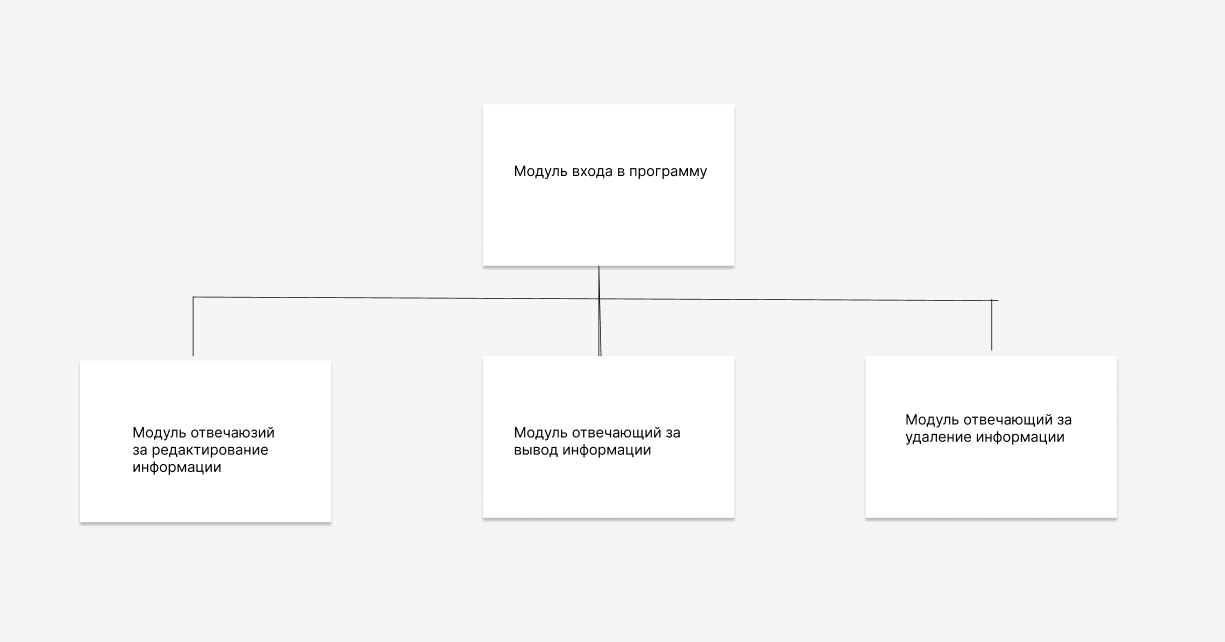


Рисунок 3.8. - Модули разрабатываемой системы

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.

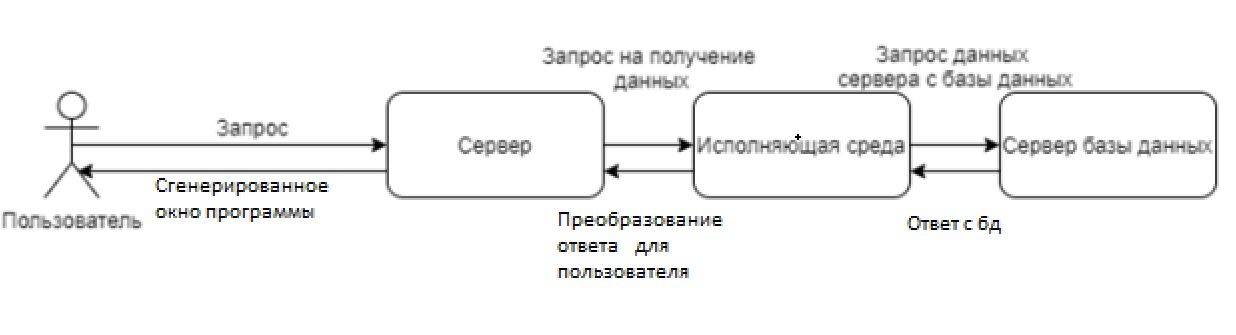


Рисунок 3.9. - Модель работы с системой, спроектированной по архитектуре клиент сервер

3.3.2. Решения по режимам функционирования, диагностированию работы системы

Система должна поддерживать работу в двух режимах:

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

- профилактический режим, все подсистемы не выполняют своих функций.

В основном режиме функционирования “МСВ” должна обеспечивать:

-работу пользователей в режиме 24 часов 7 дней в неделю;

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

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

-модернизация серверной части ИС;

- техническое обслуживание страниц-сайта 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.

3.3.3. Решения по численности, квалификации и функциям персонала АС, режимам его работы, порядку взаимодействия

В состав персонала, необходимого для обеспечения эксплуатации «МСВ» в рамках соответствующих подразделений Заказчика, необходимо выделение следующих ответственных лиц, представленных в таблице 3.1.:

Таблица 3.1. - Состав персонала, необходимого для обеспечения эксплуатации «МСВ»

|  |  |  |
| --- | --- | --- |
| Должность | Количество | Подсистема |
| Менеджер | 1 | Получение списка всех товаров, клиентов, заказов и получение информации |

Данные лицо должно выполнять следующие функциональные обязанности:

- Менеджер

- работа со всеми функциями

3.3.4. Сведения об обеспечении заданных в техническом задании (ТЗ) потребительских характеристик системы (подсистем), определяющих ее качество

Обеспечение заявленных требований будет обеспечено следующим образом:

-Требования к надежности обеспечиваются использованием отказоустойчивых решений, реализованных в составе Microsoft SQL Server.

-Требования к безопасности обеспечиваются использованием (возможностью использования) защищенных протоколов и средствами обеспечения безопасности, реализованными в Microsoft SQL Server.

-Требования к эргономике и технической эстетике обеспечиваются использованием JavaFx и CSS для создания графического интерфейса.

3.3.5. Состав функций, комплексов задач, реализуемых системой (подсистемой)

Выделенные варианты сценариев использования системы и соответствующие им диаграммы последовательностей приведены в разделе 2.4.2 «Требования к функциям, выполняемым ПО».

3.3.6. Решения по комплексу технических средств, его размещению на объекте

Для функционирования системы требуется наличие сервера и компьютера, на котором будет производится работа с программой. Перед вводом системы в действие необходимо:

-Интернет-соединение.

-ПК.

-JVM

- Эксплуатация системы.

3.3.7. Решения по составу информации, объему, способам ее организации, видам машинных носителей, входным и выходным документам и сообщениям, последовательности обработки информации и другим компонентам

Таблицы проектируемой базы данных, связи между ними, а также логическая схема база данных описана в разделе 2.4.3 «Требования к видам обеспечения».

Индекс это - структура данных, которая помогает MSSQL быстрее обнаружить отдельные записи в файле и сократить время выполнения запросов пользователей. Они повышают производительность СУБД таким образом, что если создать индекс по первичному ключу, а затем искать строку с данными, используя значения первичного ключа, то SQL-сервер сначала найдет значение индекса, а затем использует его для быстрого нахождения строки с данными. Без индекса было бы выполнено полное сканирование всех строк таблицы, что значительно бы потратило ресурсы. Задавая первичный ключ, автоматически устанавливается кластеризованный индекс.

Установим индексирование для полей в наших таблицах.

* Products – таблица содержит информацию о продуктах
* Clients - таблица содержит информацию о клиентах
* Orders - таблица содержит информацию о заказах

Таблица 3.2 – Таблица products

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| Id\_product | Integer | 11 | Идентификатор продукта | Первичный ключ |
| Name | Varchar | 40 | Название продукта | Not null |
| category | Integer | 10 | категория | Not null |
| material | Varchar | 40 | Материал | Not null |
| quanitity | Varchar | 40 | Кол-во товара | Not null |

Таблица 3.3. - Таблица orders в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| Id\_order | Integer | 11 | Идентификатор заказа | Первичный ключ |
| Id\_client | Integer | 10 | Внешний ключ клиента | Not null |
| Id\_products | Integer | 10 | Внешний ключ продукта | Not null |
| Date | Datetime |  | Дата заказа | Not null |

Таблица 3.4. - Таблица clients в базе данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип | Размер | Комментарий | Ограничение |
| Id\_clients | Integer | 11 | Идентификатор  пользователя | Первичный ключ |
| F\_name | Varchar | 10 | имя | Not null |
| L\_name | Varchar | 10 | Фамилия | Not null |
| S\_name | Integer | 11 | отчество | Not null |

3.3.8. Решения по составу программных средств, языкам деятельности, алгоритмам процедур и операций и методам их реализации

При разработке ИС использованы следующие программные компоненты:

- JavaFx интерфейс

- Хранилище данных создано на базе MSSQL

В качестве ОС клиентских устройств может использоваться:

- Microsoft Windows 10

Для реализации клиентской части могут использоваться:

- JavaFx;

- CSS;

Клиентское приложение должно быть разработано с применением объектно-ориентированной методологии, используя следующие паттерны проектирования:

- MVC;

- Одиночка;

- Итератор.

3.3.9. Эскиз пользовательского интерфейса МВС

Эскизы пользовательского интерфейса представлены в разделе 2.4.4.4 «Эскизы основных окон системы».