



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра промышленной информатики (КПИ)

ИТОГОВЫЙ ОТЧЕТ
по дисциплине «Проектирование баз данных»

Студент группы ИВБО-08-22 Стецюк В.В.

(подпись)

Старший
преподаватель Серебрянкин В.А.

(подпись)

Отчет представлен «___»_____2024г.

Москва 2024

СОДЕРЖАНИЕ

Практическое задание 1 (IDEF0)	3
Практическое задание 2 (DFD)	6
Практическое задание 3 (UML ДИАГРАММЫ).....	9
Use case диаграмма.....	9
Class диаграмма	10
Sequence диаграмма	11
Collaboration диаграмма.....	12
Практическое задание 4 (Питера Чена).....	14
Практическое задание 5 (IDEF1X).....	15
Логический уровень	16
Физический уровень	17
Практическое задание 6 (Реляционная алгебра)	18
Операция объединения	18
Операция разности	18
Операция выборка	19
Операция естественное соединение	19
Операция соединение по условию.....	20
ЗАКЛЮЧЕНИЕ	21
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	23

Практическое задание 1 (IDEF0)

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ).

На вход подается следующее:

- ТС клиента
- Требования клиента

На управление подается следующее:

- Нормативные документы
- Приказ Минтруда России от 09.12.2020 N 871н
- ГОСТ 18322-2016 Система технического обслуживания и ремонта техники

На механизм подается следующее:

- Мастера
- Оборудование
- Детали

На выход идут:

- Отремонтированный ТС
- Исполненные требования
- ТС не подлежит ремонту

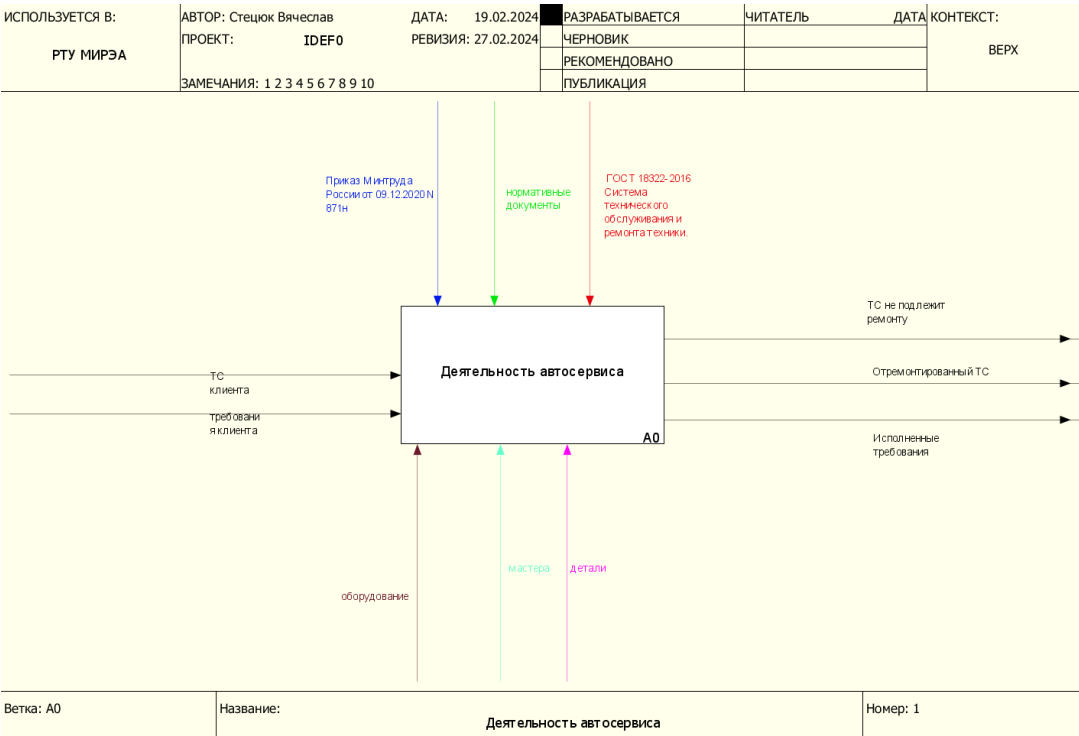


Рисунок 1 – Основной блок

Далее декомпозируем основной блок на 3 новые функции: принятие заказа, проведение работ и выдача ТС клиенту.

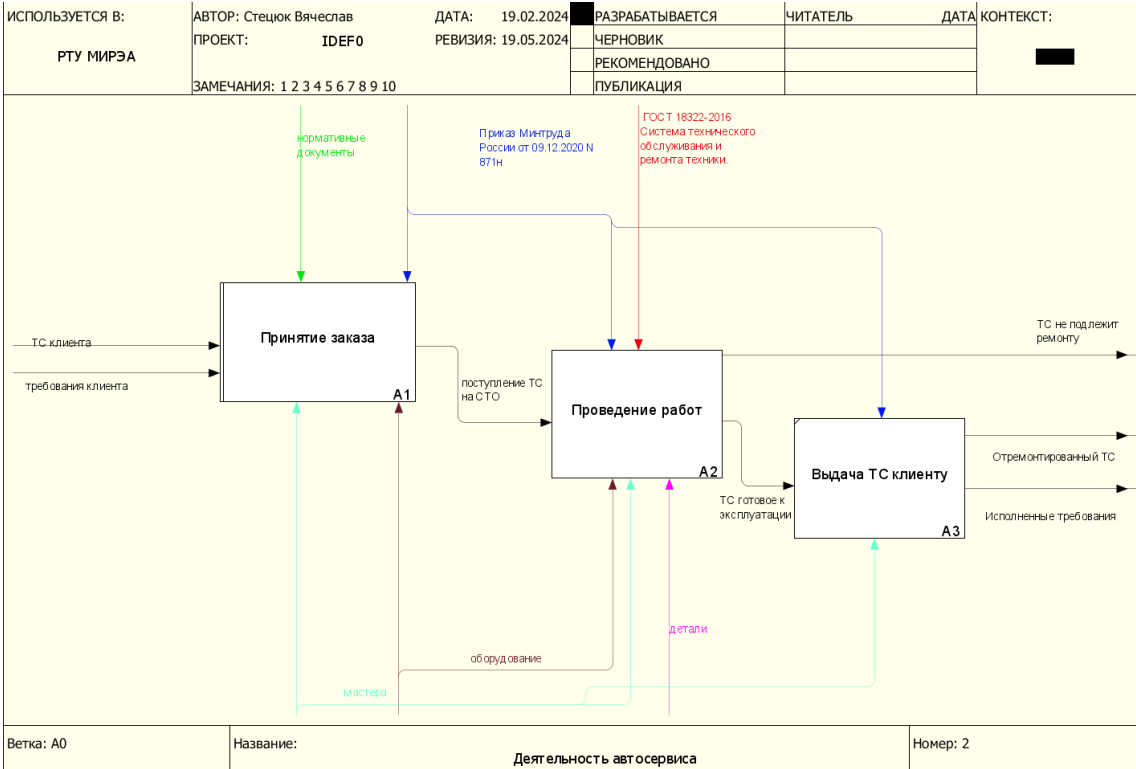


Рисунок 2 – Декомпозиция основного блока

Далее декомпозируем блок принятие заказа на блоки Прием ТС клиента, Общая диагностика и Заказ деталей для ТС и проведение работ на блоки Удаление неисправных деталей, Установка исправных деталей и Подготовка ТС к эксплуатации (рис. 3-4).

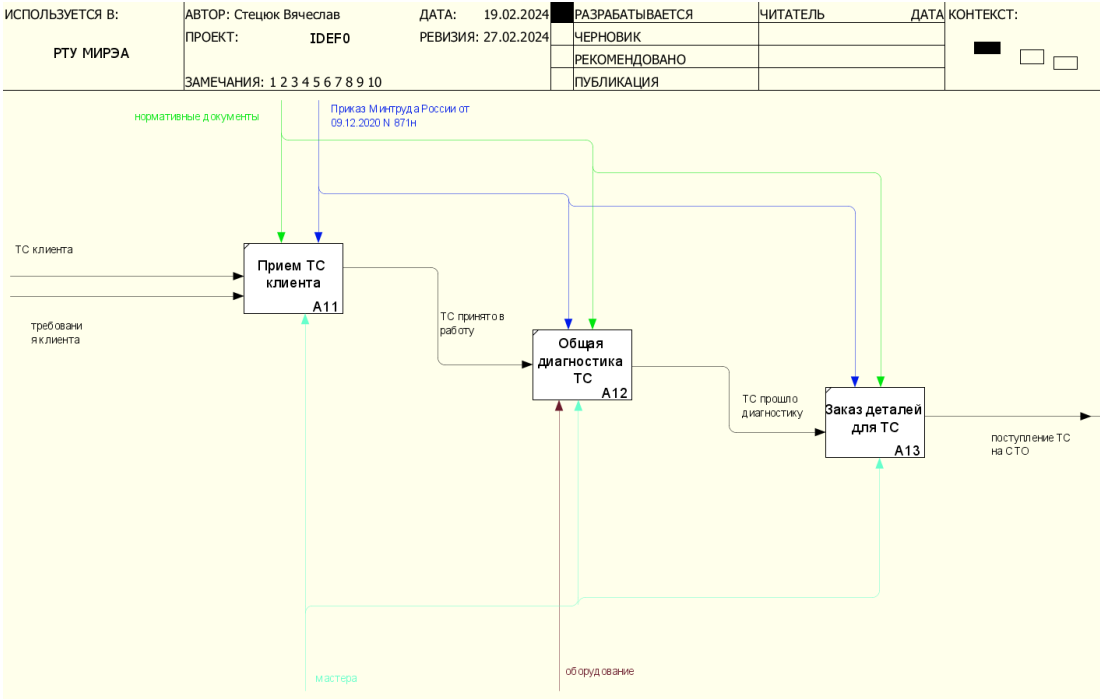


Рисунок 3 – Декомпозиция блока принятие заказа

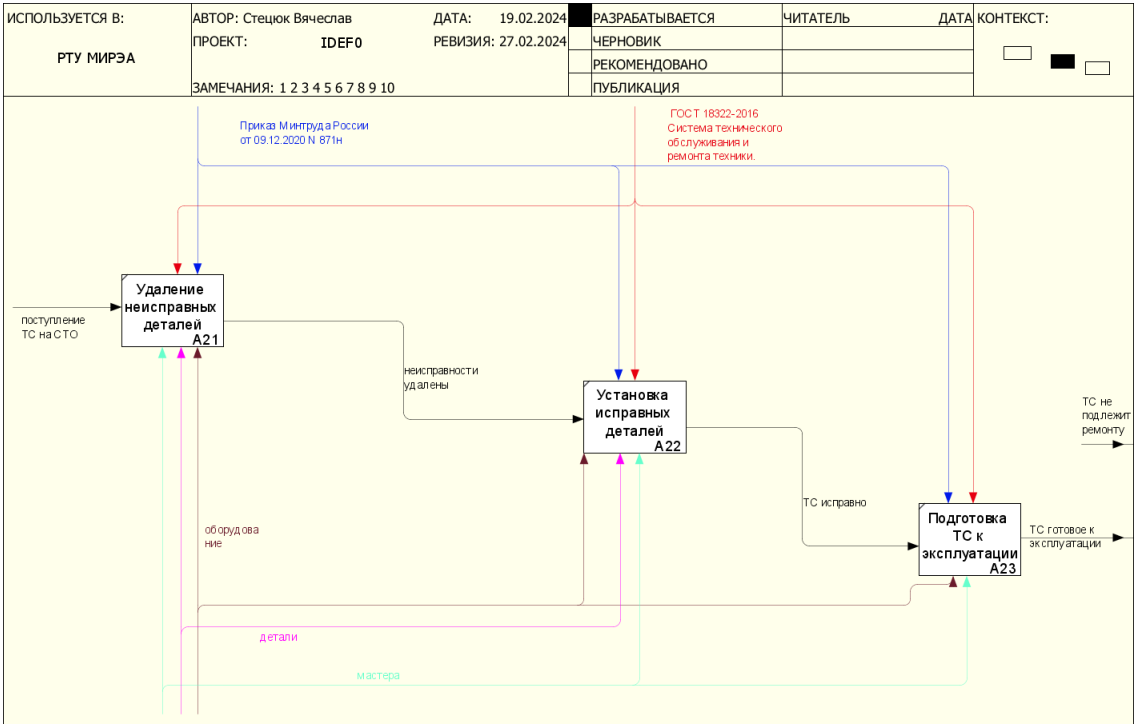


Рисунок 4 – Декомпозиция блока проведение работ

Практическое задание 2 (DFD)

DFD (от англ. data flow diagrams) — это графическое представление потоков данных в информационной системе.

С помощью DFD можно описывать входящие и выходящие потоки данных и хранилища этих данных.

Основные элементы диаграммы DFD:

- 1. Процесс — активность, которая приводит к преобразованию данных.
- 2. Внешние сущности — участники процесса, которые взаимодействуют с данными.
- 3. Хранилище данных — место, где хранится информация для последующего использования системой.
- 4. Потоки данных — маршруты, по которым информация перемещается между внешними сущностями, процессами и хранилищами данных.

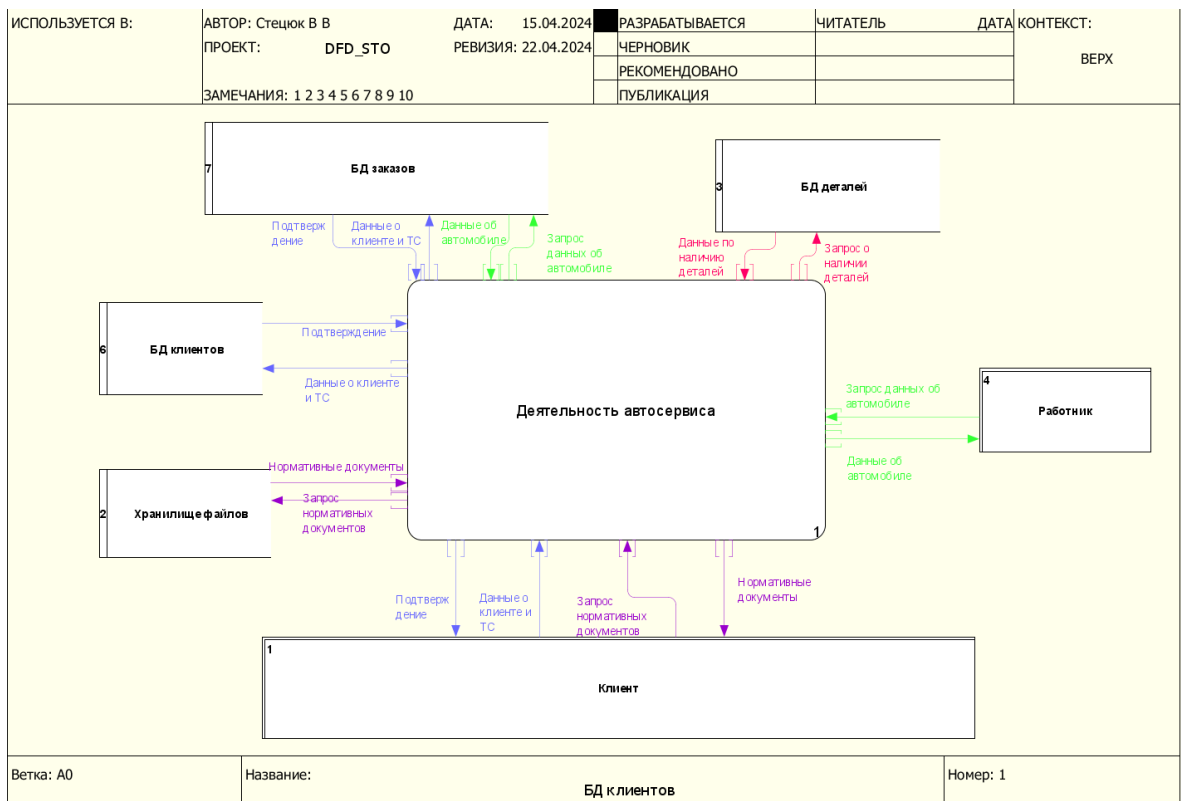


Рисунок 5 – Диаграмма DFD

Блок Деятельность автосервиса декомпозируется на:

- Принятие заказа
- Проведение работ
- Выдача ТС Клиенту

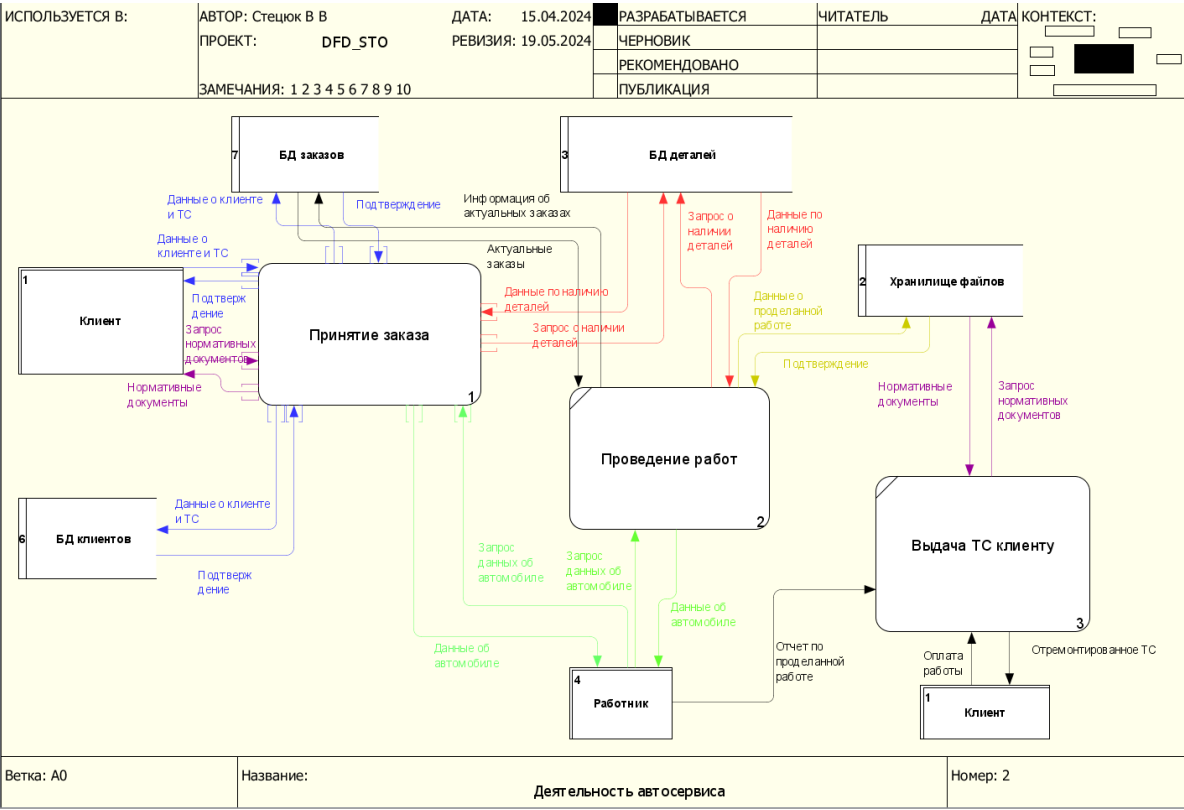


Рисунок 6 -Декомпозиция основного блока

Блок Принятие заказа декомпозируется на:

- Прием ТС клиента
- Общая диагностика ТС
- Заказ деталей для ТС

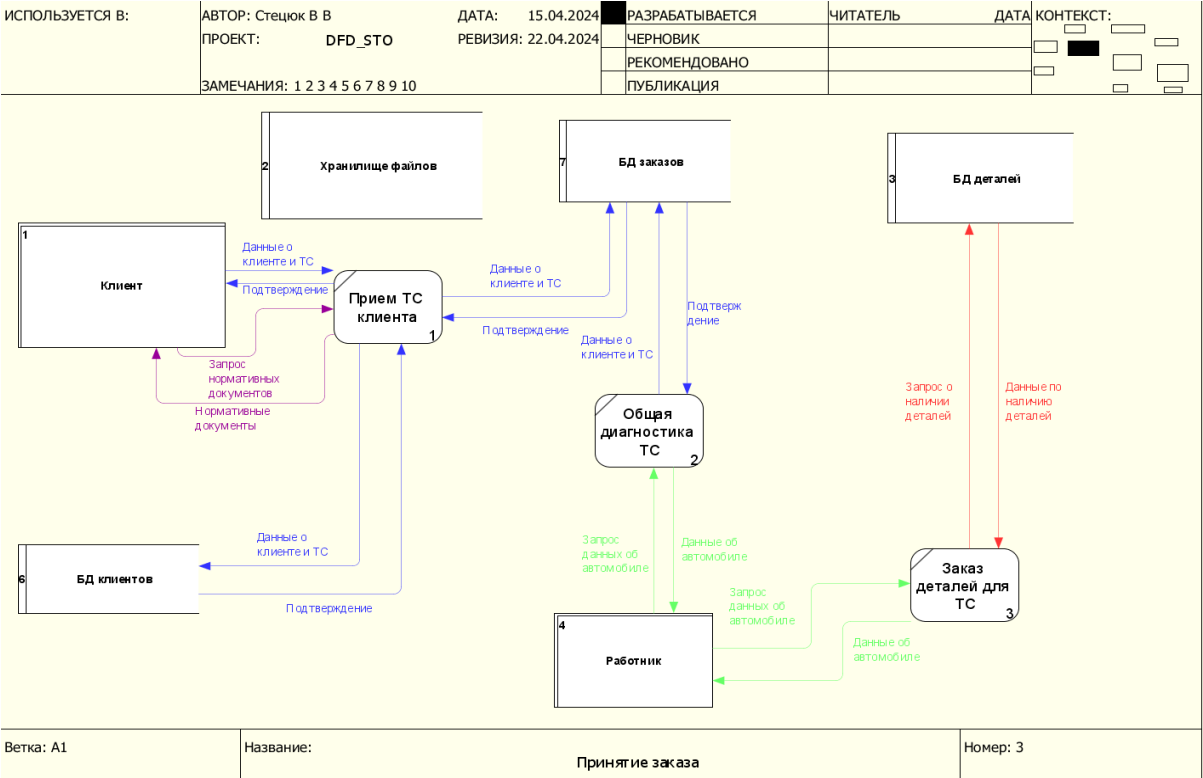


Рисунок 7 – Декомпозиция блока принятие заказа

Практическое задание 3 (UML ДИАГРАММЫ)

Use case диаграмма

Диаграмма вариантов использования (use case diagram) — диаграмма, на которой изображаются отношения между актерами и вариантами использования.

Для отражения модели прецедентов на диаграмме используются:

1. рамки системы (англ. system boundary) — прямоугольник с названием в верхней части и эллипсами (прецедентами) внутри. Часто может быть опущен без потери полезной информации,
2. актёр (англ. actor) — стилизованный человечек, обозначающий набор ролей пользователя (понимается в широком смысле: человек, внешняя сущность, класс, другая система), взаимодействующий с некоторой сущностью (системой, подсистемой, классом). Актёры не могут быть связаны друг с другом (за исключением отношений обобщения/наследования),
3. прецедент — эллипс с надписью, обозначающий выполняемые системой действия (могут включать возможные варианты), приводящие к наблюдаемым актёрами результатам. Имя прецедента связано с непрерывным (атомарным) сценарием — конкретной последовательностью действий, иллюстрирующей поведение. В ходе сценария акторы обмениваются с системой сообщениями.

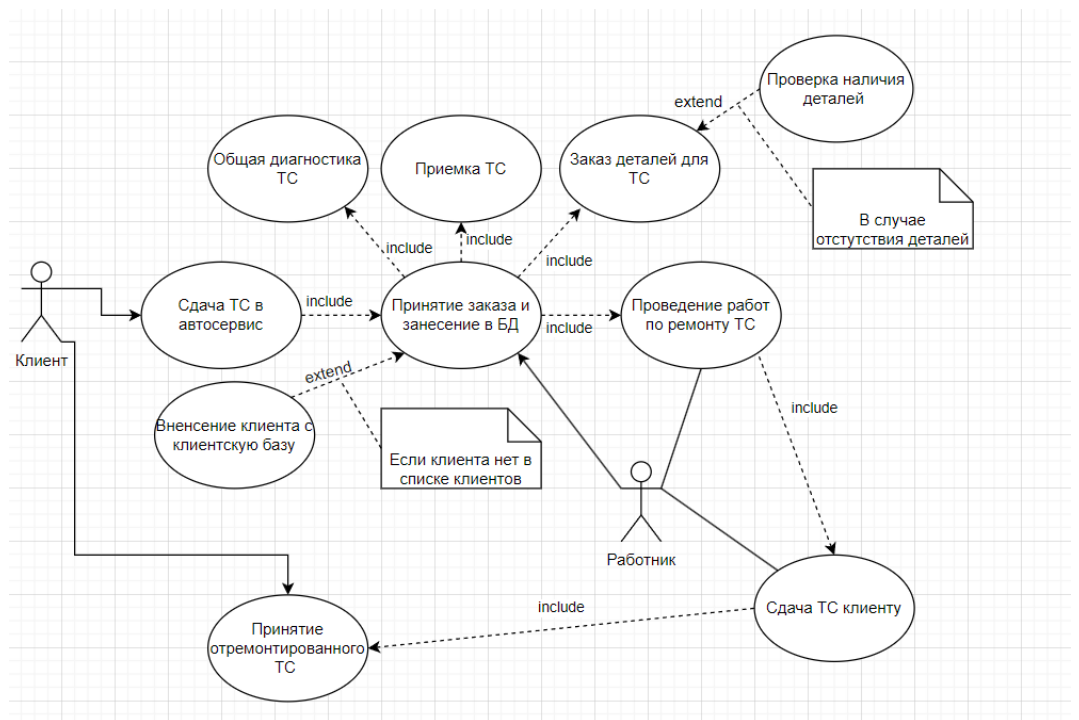


Рисунок 8 – Use case диаграмма

Class диаграмма

Диаграмма классов (англ. class diagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

Класс является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

4. В верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом. Имена классов начинаются с заглавной буквы. Если класс абстрактный — то его имя пишется полужирным курсивом.
5. Посередине располагаются поля (атрибуты) класса. Они выровнены по левому краю и начинаются с маленькой буквы.
6. Нижняя часть содержит методы класса. Они также выровнены по левому краю и пишутся с маленькой буквы.

Язык UML предоставляет механизмы для представления членов

класса, например, атрибутов и методов, а также дополнительной информации о них.

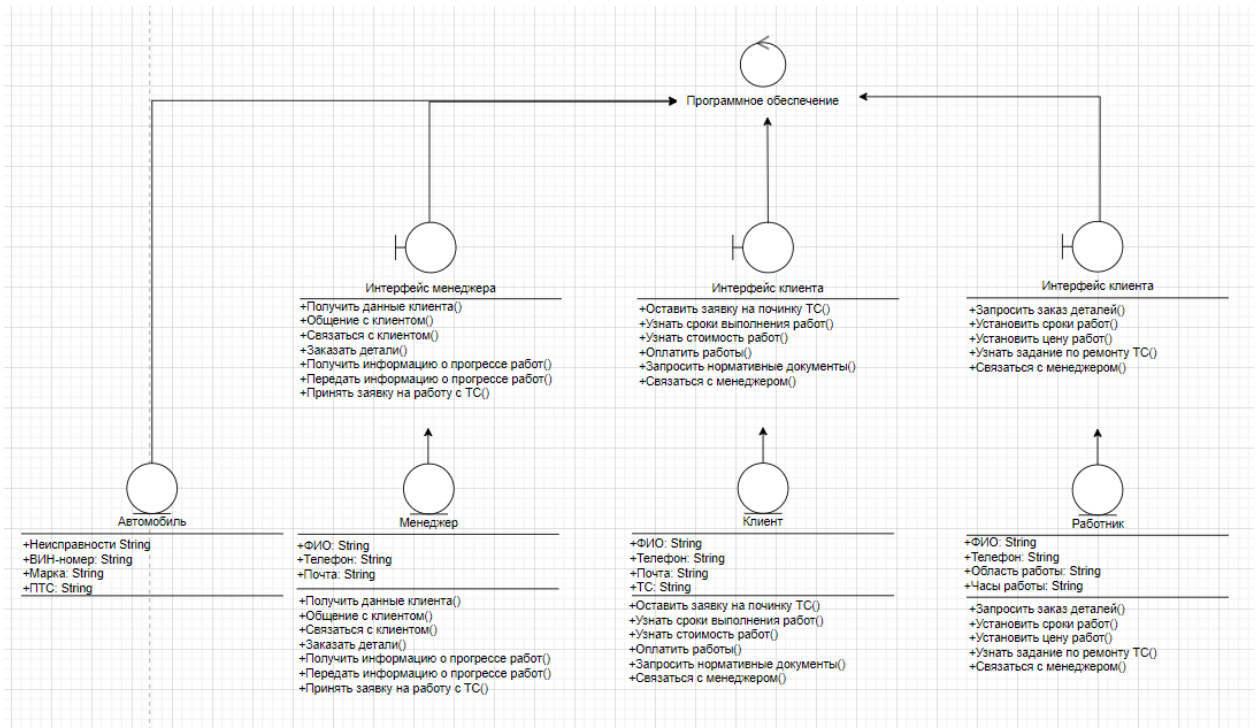


Рисунок 9 – Class диаграмма

Sequence диаграмма

Диаграмма последовательности (англ. sequence diagram) — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие акторов (действующих лиц) информационной системы в рамках прецедента.

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни» (англ. lifeline), отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни» — фокусы контроля, англ. focus of control), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

Виды стрелок:

- Синхронное сообщение
- Ответное сообщение
- Асинхронное сообщение
- Потерянное сообщение
- Найденное сообщение

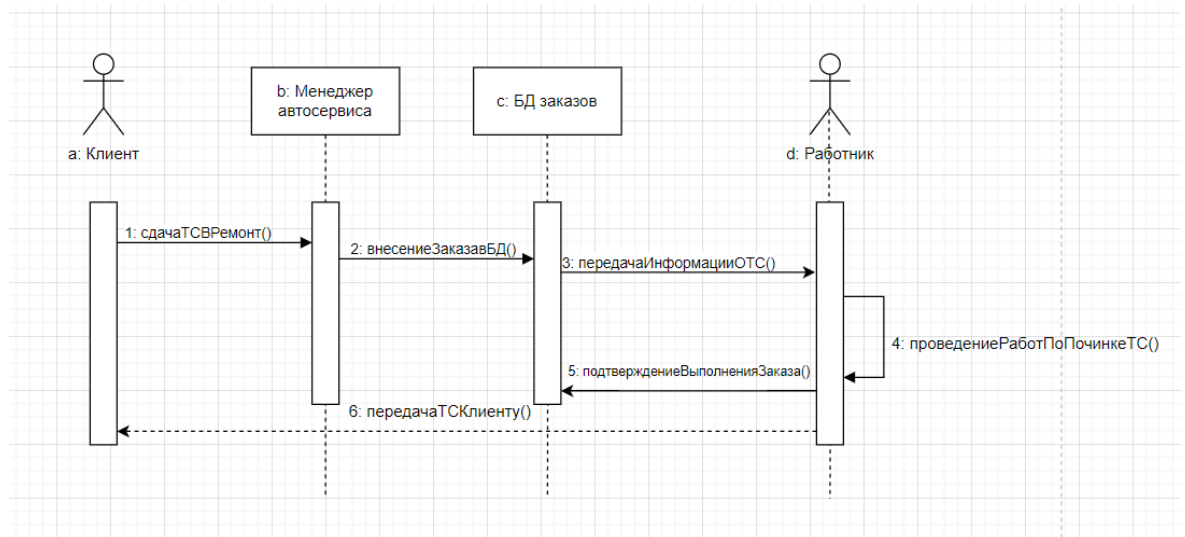


Рисунок 10 – Sequence диаграмма

Collaboration диаграмма

Диаграмма коммуникации (англ. communication diagram, в UML 1.x — диаграмма кооперации, collaboration diagram) — диаграмма, на которой изображаются взаимодействия между частями композитной структуры или ролями кооперации. В отличие от диаграммы последовательности, на диаграмме коммуникации явно указываются отношения между объектами, а время как отдельное измерение не используется (применяются порядковые номера вызовов).

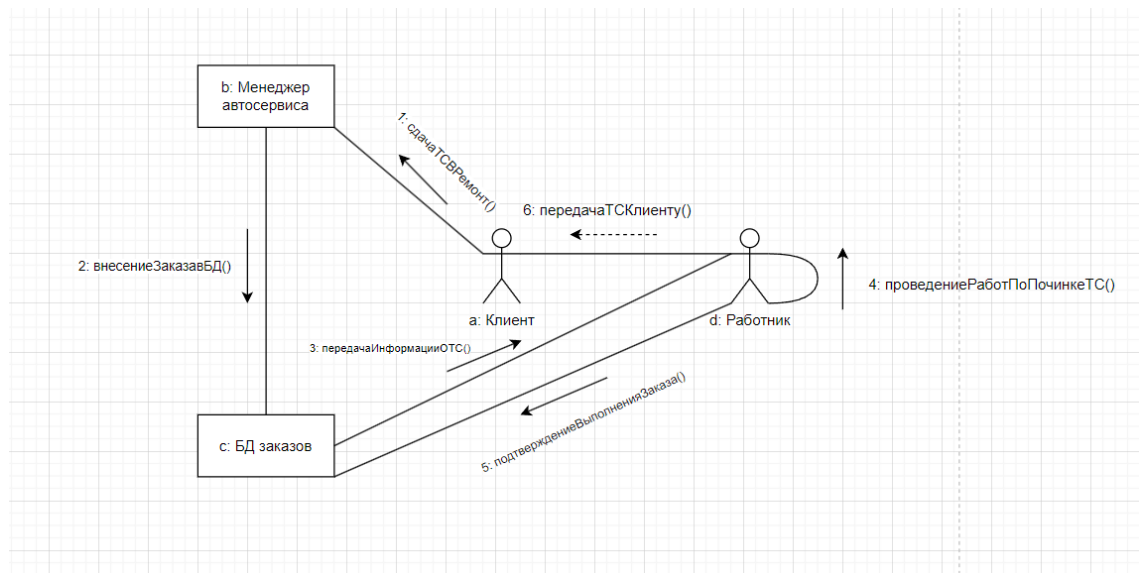


Рисунок 11 – Collaboration диаграмма

Практическое задание 4 (Питера Чена)

ER-модель (от англ. Entity-Relationship model, модель «сущность — связь»)

— модель данных, позволяющая описывать концептуальные схемы предметной области.

Множества сущностей изображаются в виде прямоугольников, множества отношений изображаются в виде ромбов. Если сущность участвует в отношении, они связаны линией. Если отношение не является обязательным, то линия пунктирная. Атрибуты изображаются в виде овалов и связываются линией с одним отношением или с одной сущностью.

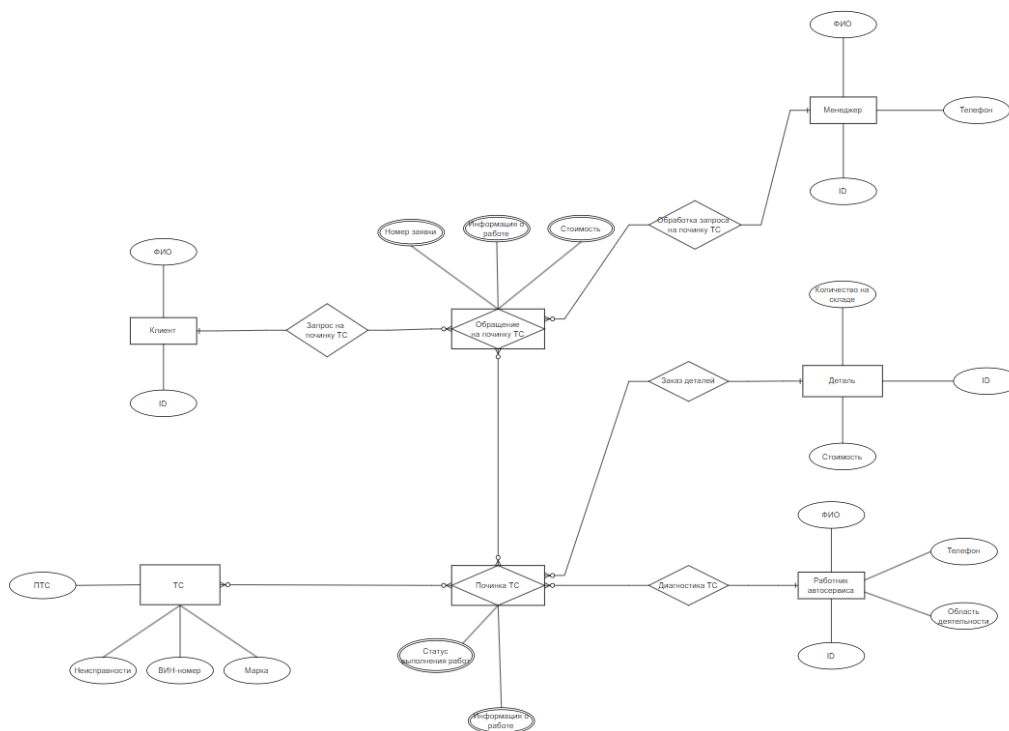


Рисунок 12 – Диаграмма Питера Чена

Практическое задание 5 (IDEF1X)

Integration DEFinition for information modeling (IDEF1X) язык моделирования данных[en] для разработки семантики моделей данных. IDEF1X используется для формирования графических представлений информационных моделей, которые отражают структуру и семантику информации внутри среды или системы.

Составные элементы IDEF1X:

- 1 **Сущности** - Представление класса реальных или абстрактных вещей (людей, предметов, мест, событий, идей, комбинаций вещей и т.д.), которые признаются экземплярами одного и того же класса, поскольку они имеют одни и те же характеристики и могут участвовать в одних и тех же отношениях.
- 2 **Домены** - Именованный набор значений данных (фиксированного размера или, возможно, бесконечных) одного и того же типа данных, на основе которого строится фактическое значение для экземпляра атрибута. Каждый атрибут должен быть определен только в одном базовом домене. Несколько атрибутов могут быть основаны на одном и том же базовом домене.
- 3 **Атрибуты** - Свойство или характеристика, которое является общим для некоторых или всех экземпляров сущности. Атрибут представляет использование домена в контексте сущности.
- 4 **Ключи** - Атрибут или комбинация атрибутов сущности, значения которых однозначно идентифицируют каждый экземпляр сущности. Каждый такой набор представляет собой ключ-кандидат.
- 5 **Первичные ключи** - Ключ-кандидат, выбранный в качестве уникального идентификатора сущности.
- 6 **Внешний ключ** - Атрибут или комбинация атрибутов

дочернего экземпляра сущности или экземпляра категории, значения которых совпадают со значениями в первичном ключе связанного родительского или универсального экземпляра сущности. Внешний ключ можно рассматривать как результат "миграции" первичного ключа родительской или универсальной сущности через определенную связь или отношение категоризации. Атрибуту или комбинации атрибутов во внешнем ключе может быть присвоено имя роли, отражающее его роль в дочерней сущности или сущности категории.

Логический уровень

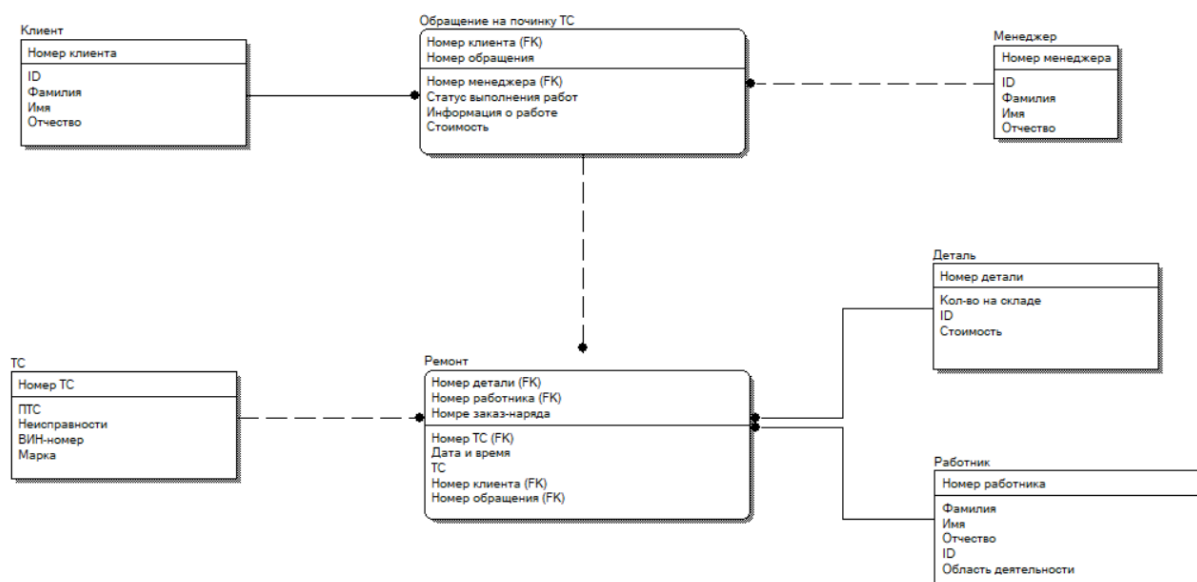


Рисунок 13 – Диаграмма логического уровня IDF1X

Физический уровень

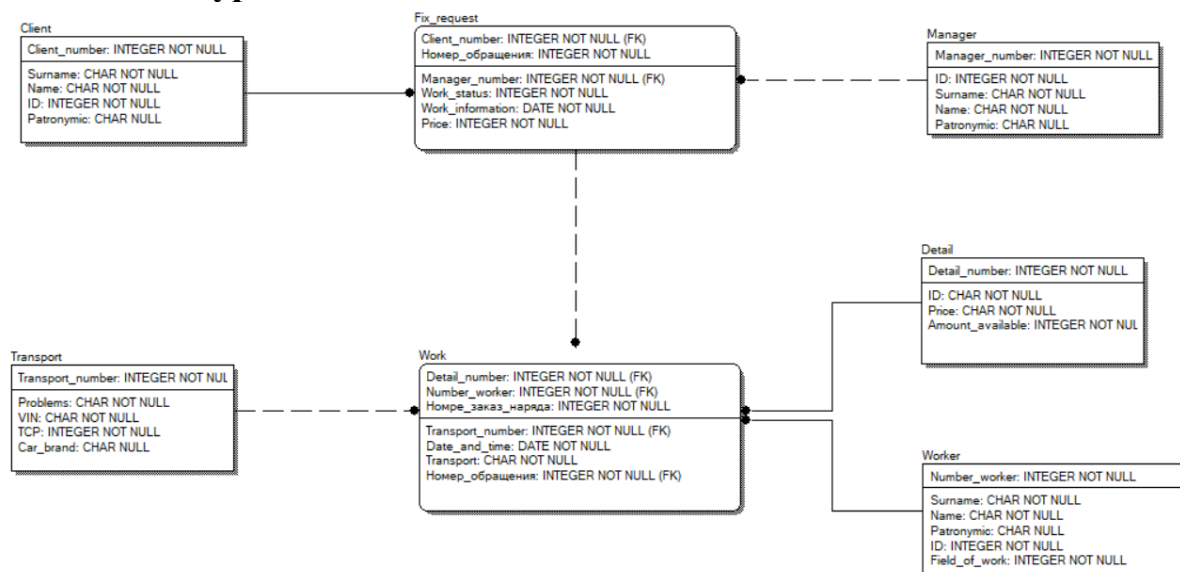


Рисунок 14 – Диаграмма физического уровня IDF1X

Практическое задание 6 (Реляционная алгебра)

Реляционная алгебра — замкнутая система операций над отношениями в реляционной модели данных. Операции реляционной алгебры также называют реляционными операциями.

Операция объединения

Таблица 1 Деталь (R1)

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
3	103	21	7300

Таблица 2 Деталь (R2)

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
4	104	300	500

$$R3 = R1 \cup R2 = \{r \mid r \in R1 \cup r \in R2\}.$$

Таблица 3 объединения таблиц 1 и 2

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
3	103	21	7300
4	104	300	500

Операция разности

Таблица 1 Деталь (R1)

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
3	103	21	7300

Таблица 2 Деталь (R2)

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
4	104	300	500

$$R3 = R1 \setminus R2 = \{r \mid r \in R1 \cap r \notin R2\}$$

Таблица 3 разность таблиц 1 и 2

Detail_number	ID	Amount_available	Price
3	103	21	7300

Операция выборка

Таблица 1 Деталь (R1)

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
3	103	21	7300

X и Y (R where X θ Y), где θ означает любой скалярный оператор сравнения(=, \neq , \leq , \geq), X и Y – атрибуты.
 σ (Amount_available < 45)

Таблица операции 3

Detail_number	ID	Amount_available	Price
2	102	33	15000
3	103	21	7300

Операция естественное соединение

Таблица 1 Деталь (R1)

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
3	103	21	7300

Таблица 2 Ремонт (R2)

Detail_number	Transport_number	Number_worker	Manager_number	Client_number	Data_and_time	Transport
1	707	001	010	100	01.12.24	BMW
2	777	002	020	200	02.12.24	AUDI
3	77	003	030	300	03.12.24	LADA

Естественным соединением (R1 JOIN R2) отношений R1 (X, Y) и R2 (Y, Z)называется отношение R3 с заголовком {X, Y, Z}

Таблица 3 (R1 join R2) (R3)

Detail_number	Transport_number	Number_worker	Manager_number	Client_number	Data_and_time	Transport	ID	Amount_available	Price
1	707	001	010	100	01.12.24	BMW	101	71	5000
2	777	002	020	200	02.12.24	AUDI	102	33	15000
3	77	003	030	300	03.12.24	LADA	103	21	7300

Операция соединение по условию

Таблица 1 Деталь (R1)

Detail_number	ID	Amount_available	Price
1	101	71	5000
2	102	33	15000
3	103	21	7300

Таблица 2 Ремонт (R2)

Detail_number	Transport_number	Number_worker	Manager_number	Client_number	Data_and_time	Transport
1	707	001	010	100	01.12.24	BMW
2	777	002	020	200	02.12.24	AUDI
3	77	003	030	300	03.12.24	LADA

θ – соединением отношений R1 по атрибуту X с R2 по атрибуту Y называется результат вычисления выражения $R3=(R1 \times R2) \text{ WHERE } X \theta Y$, X и Y

– атрибуты

$\sigma(\text{Detail_number} < 2)$

Таблица 3 (R1 соединение R2) (R3)

Detail_number	Transport_number	Number_worker	Manager_number	Client_number	Data_and_time	Transport	ID	Amount_available	Price
1	707	001	010	100	01.12.24	BMW	101	71	5000

7 Практическое задание 7 (Диаграммы состояний)

Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой операции в предыдущем состоянии.

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому. На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фиксируется на результате деятельности.

Компонентами диаграммы деятельности являются:

- состояния действия,
- переходы,
- дорожки,
- объекты.

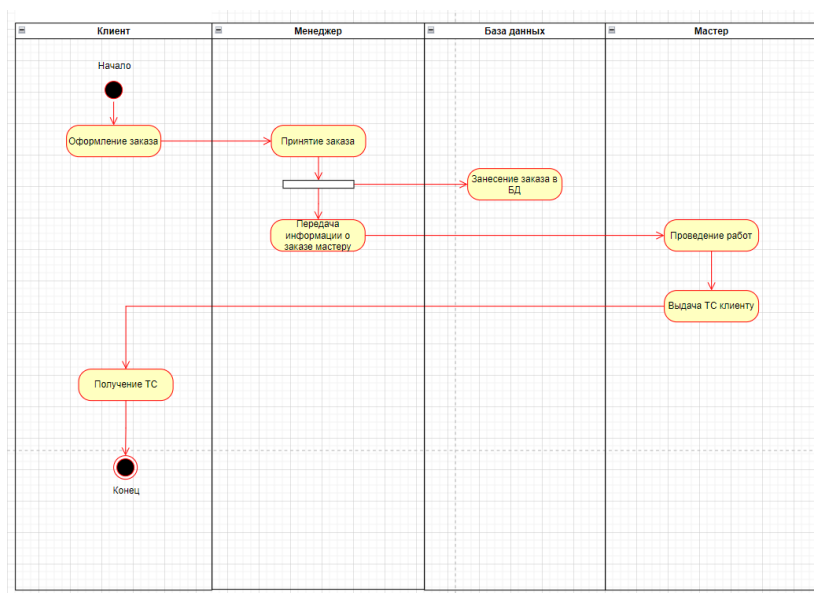


Рисунок 15 – Диаграмма UML – модели системы

ЗАКЛЮЧЕНИЕ

В ходе выполнения практических заданий по различным методам моделирования, от структурного анализа до реляционной алгебры, были изучены и применены основные концепции и инструменты для проектирования информационных систем. Понимание и применение различных методов моделирования позволило лучше представить структуру и функционал будущей системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

7. Курс: Проектирование баз данных — URL: <https://online-edu.mirea.ru/course/view.php?id=6579>