Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Тверской государственный технический университет»

(ФГБОУВПО «ТвГТУ»)

Кафедра "Программное обеспечение".

Курсовая работа по дисциплине

"Унифицированный язык UML".

ТЕМА:

Информатизация мониторинга ремонта ЖД вагонов в депо.

Выполнил:

Иванов Р.В

Студент второго курса

Факультета информационных технологий, группы ПИН 17.06,

Проверил:

Котлинский С.В.

Тверь

2019 г.

Оглавление

Оглавление

[Постановка задачи 2](#_Toc8553753)

[Цель 2](#_Toc8553754)

[Реализация 2](#_Toc8553755)

[Диаграмма вариантов использования 3](#_Toc8553756)

# Постановка задачи

Построить модель, проектируемой информационной системы на основе унифицированного языка моделирования UML (с использованием Rational Rose) с уровнем сложности, согласованным с руководителем работы.

# Цель

Создать систему для информатизация мониторинга ремонта ЖД вагонов в депо

# Реализация

**Общая картина ремонта:**

Клиент решает сдать свой вагон на ремонт в депо. Депо дает ответ клиенту, возьмется он ли ремонтировать вагон или нет. Если депо берется за ремонт, происходят следующие действия: вагон отправляется на мойку и разборку. Разборка в свою очередь происходит в два этапа: вагон разбираю на кузов, который в последствии отправится на ремонт и детали. Детали, разобранного вагона, сначала подвергаются очистки, затем разбираются на более мелкие детали, далее эти детали подвергаются контроль качества. Детали, прошедшие контроль качества отправляются на склад «Годные детали», а детали которые не прошли контроль качества на склад «Детали подлежащие ремонту» и «Негодные детали». Далее мы ремонтируем детали со склада «Детали подлежащие ремонту». Следующим шагом мы начинаем комплектование на сборку уже непосредственно деталей для самого вагон из сборочных единиц с которыми мы занимались ранее. Собранные детали мы тестируем, красим и отправляем на склад «Склад сборочных единиц». Затем мы приступаем к общий сборке вагона(кузов + детали).Собрав вагон мы испытываем его и красим, после чего отправляем клиенту.

# Диаграмма вариантов использования

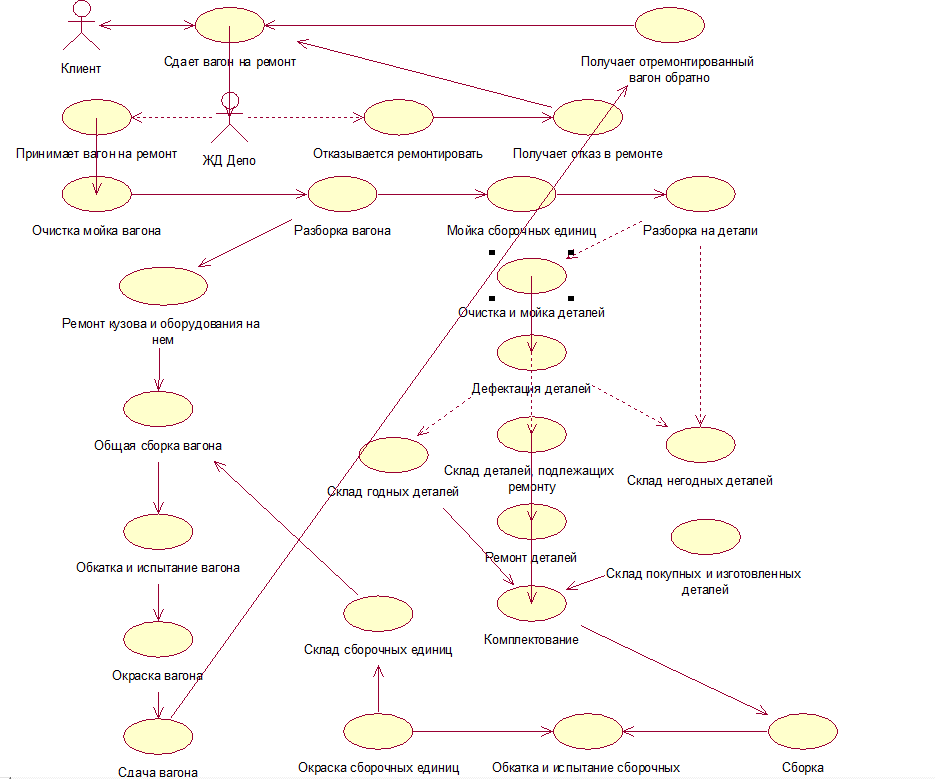


Рис. 1

**Диаграмма вариантов использования** (Use case diagram) – это графическое представление всех или части действующих лиц, вариантов использования и их взаимодействий в системе.   
  
В каждой системе есть главная диаграмма вариантов использования, которая отображает границы системы (действующие лица) и основное функциональное поведение системы (варианты использования). Другие диаграммы использования могут создаваться при необходимости.

**Прецеденты** – это механизм упрощения этапа формулировки требований для всех заинтересованных лиц. По существу, это рассказы об использовании системы в процессе решения поставленных задач.   
  
Основная идея состоит в исследовании и формулировке функциональных требований путем написания историй «из жизни системы». Эти истории помогают сформулировать различные задачи и представляют собой сценарии использования системы. На первый взгляд, описать прецеденты несложно, хотя зачастую трудно определить, что требуется от системы и описать это на нужном уровне детализации.  
  
Сила механизма прецедентов состоит в возможности масштабировать уровень сложности и формальности описания в зависимости от реальных потребностей.

**Действующее лицо** (Actor) – это сущность, обладающая поведением. Это роль, которую пользователь играет по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ. К числу действующих лиц могут относиться и сама рассматриваемая система, если она вызывает службы других систем.

На диаграмме вариантов использования курсовой работы (рис. 1) в качестве актёров выбраны следующие сущности: клиент, жд депо.

Клиент – лицо, заказавшее починку вагона.

ЖД Депо – лицо, непосредственно занимающееся починкой вагона.

# Диаграмма последовательности

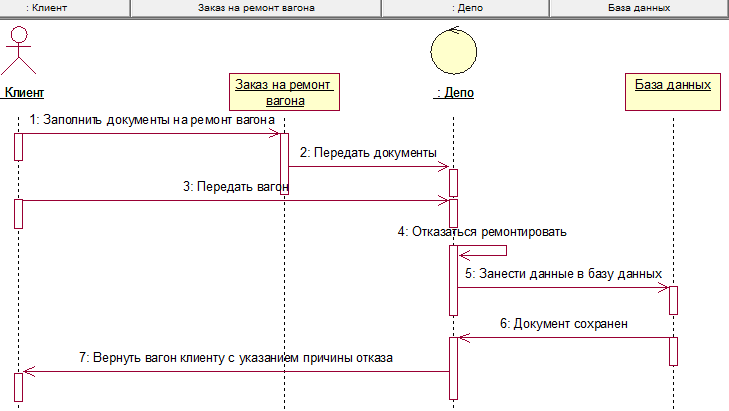
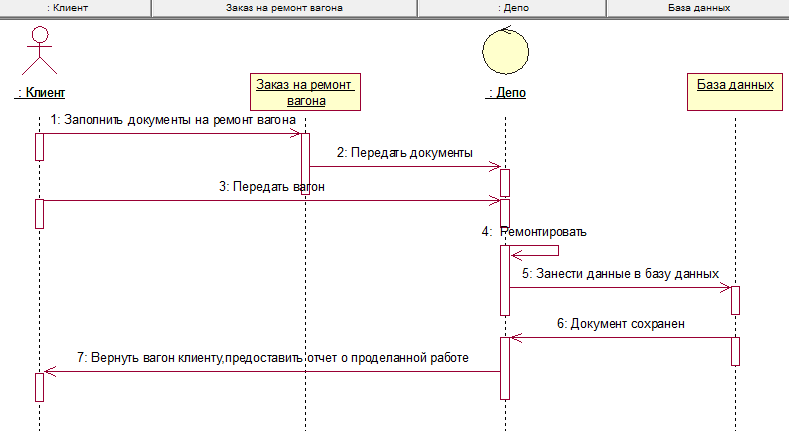
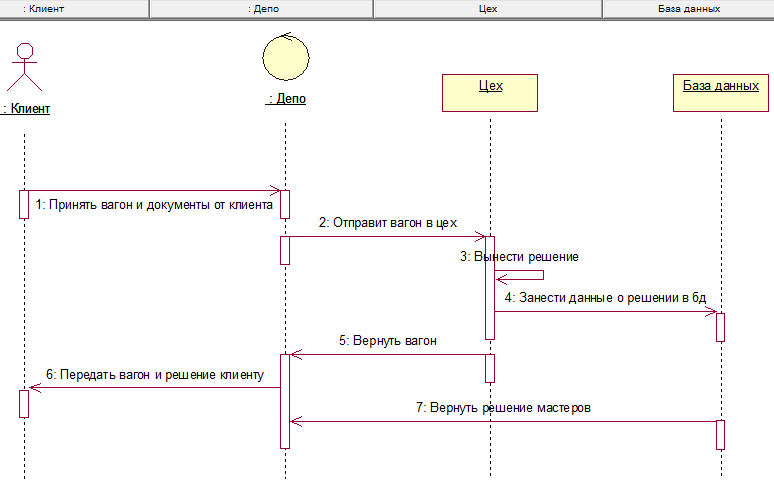


Рис. 2

 Рис.3

Рис.4

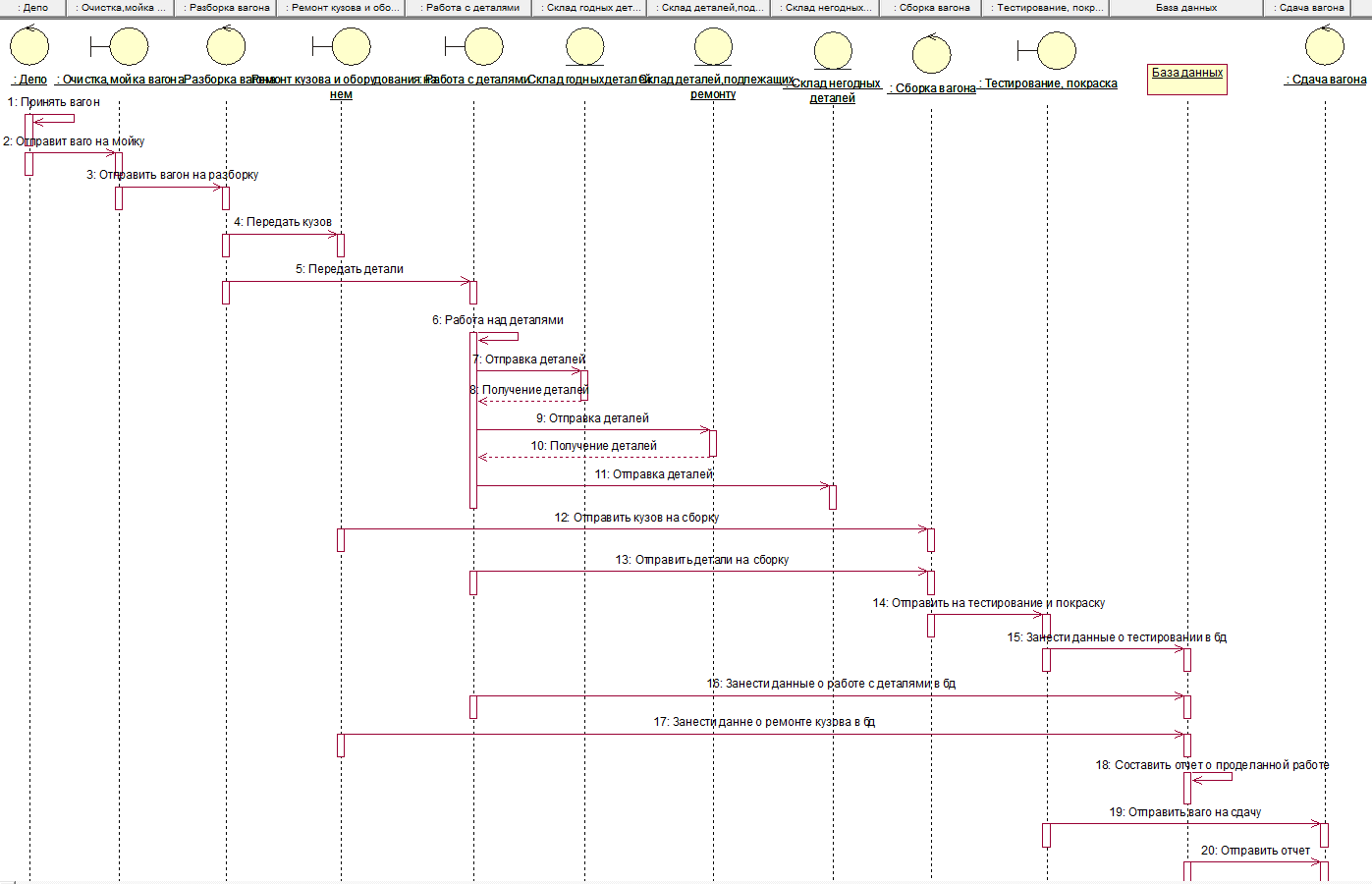


Рис.5

**Диаграмма последовательности** (sequence diagram) — [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов).

Альтернативный поток (клиент) рис. 2 и основной поток (клиент) рис.3 практически одинаковы, за исключением нескольких этапов.

1.Клиент заполняет документа на ремонт.

2.Документы передаются в депо

3.Вагон передается в депо

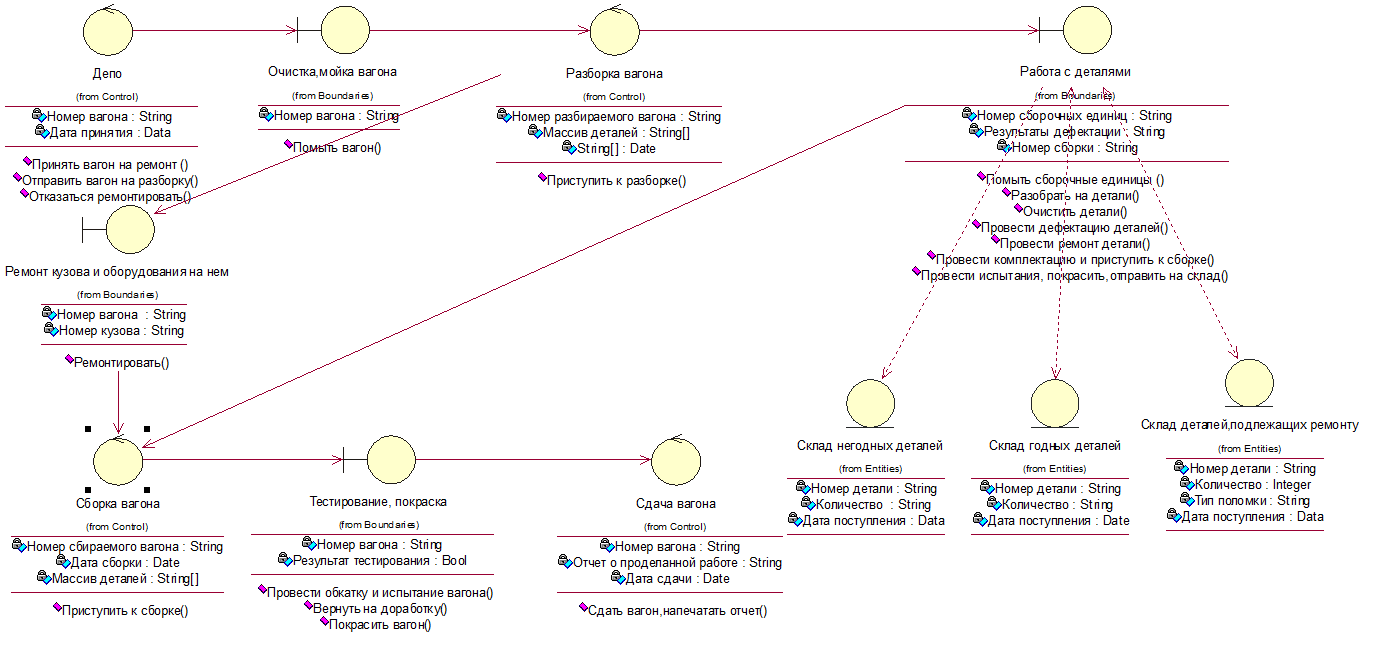
4.Депо выносит вердикт: отказ в ремонте (рис. 2) или же ремонтирует (рис.3)

5.Депо заносит данные в базу данных.

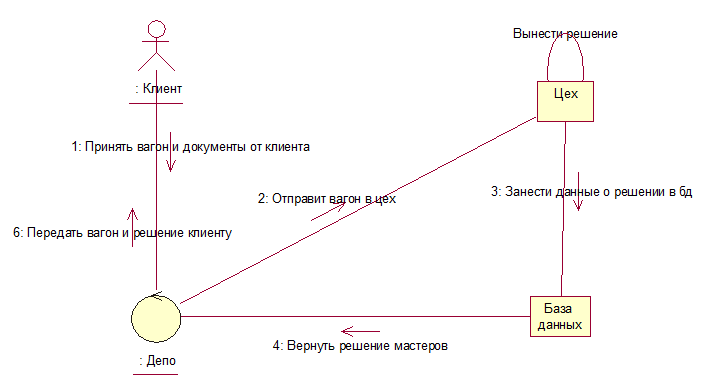
6.База данных возвращает документ в депо

7.Депо возвращает вагон и отчет либо о проделанной работе (рис. 2), либо о причине отказа в ремонте (рис.3).

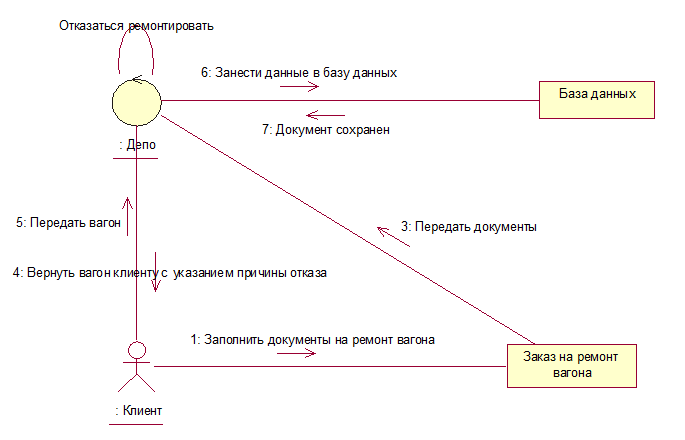
дк



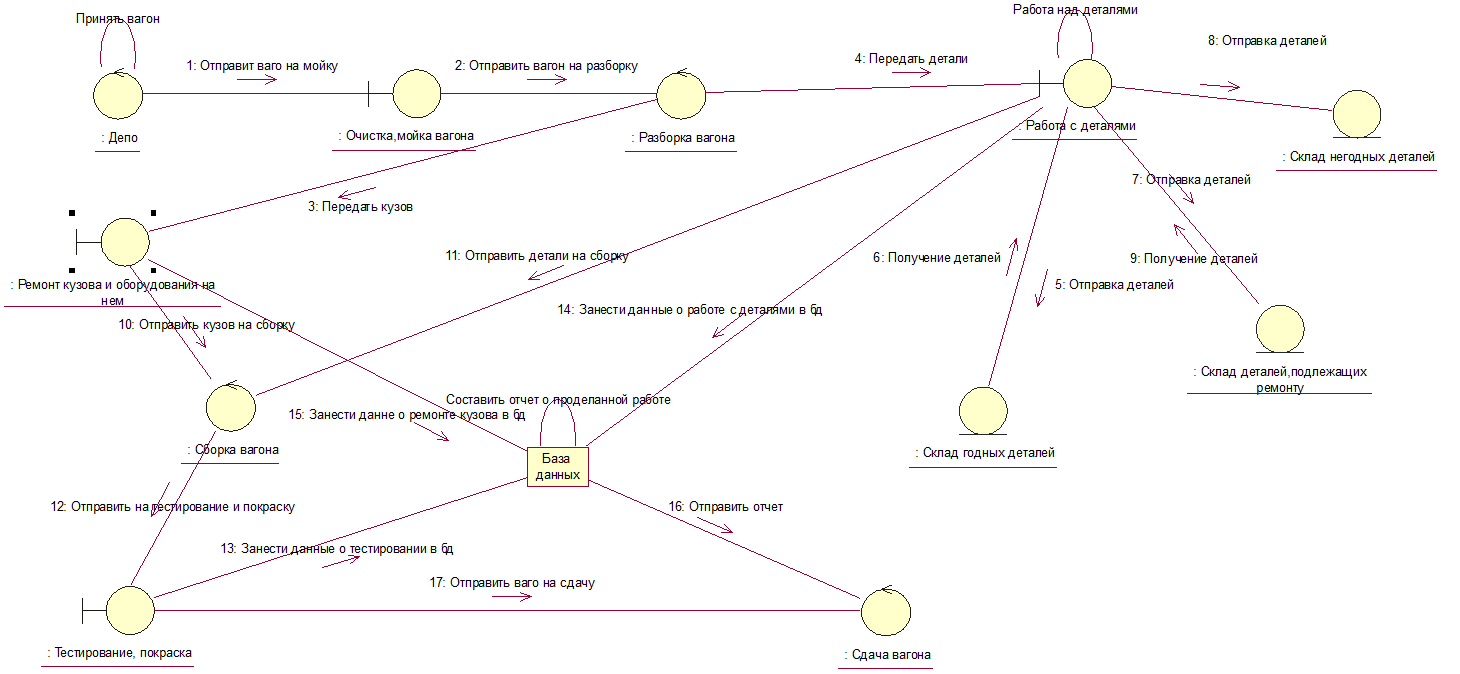
Ап депо



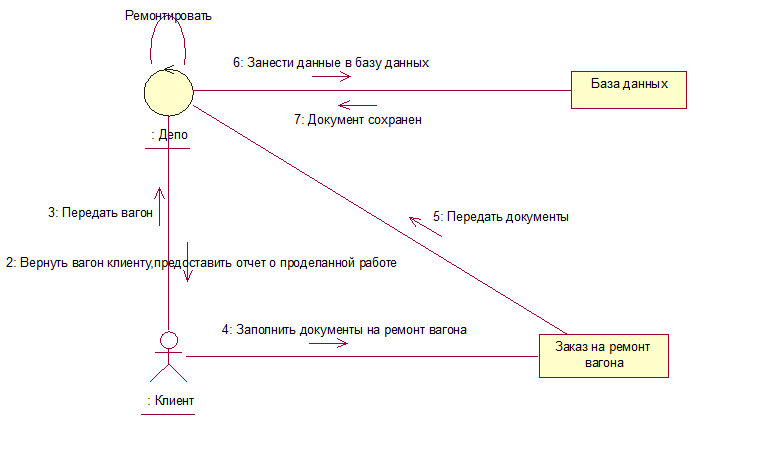
Ап к



Оп депо



Оп к



Тест покраска

