**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное учреждение**

**Высшего образования**

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**Факультет компьютерных наук**

**Кафедра технологий обработки и защиты информации**

**Отчет по предмету Архитектура информационных систем**

**1 лабораторная работа**

**Прокат автомобилей**

Грибкова М.А.

Пузатых М.С.

**Воронеж 2018**

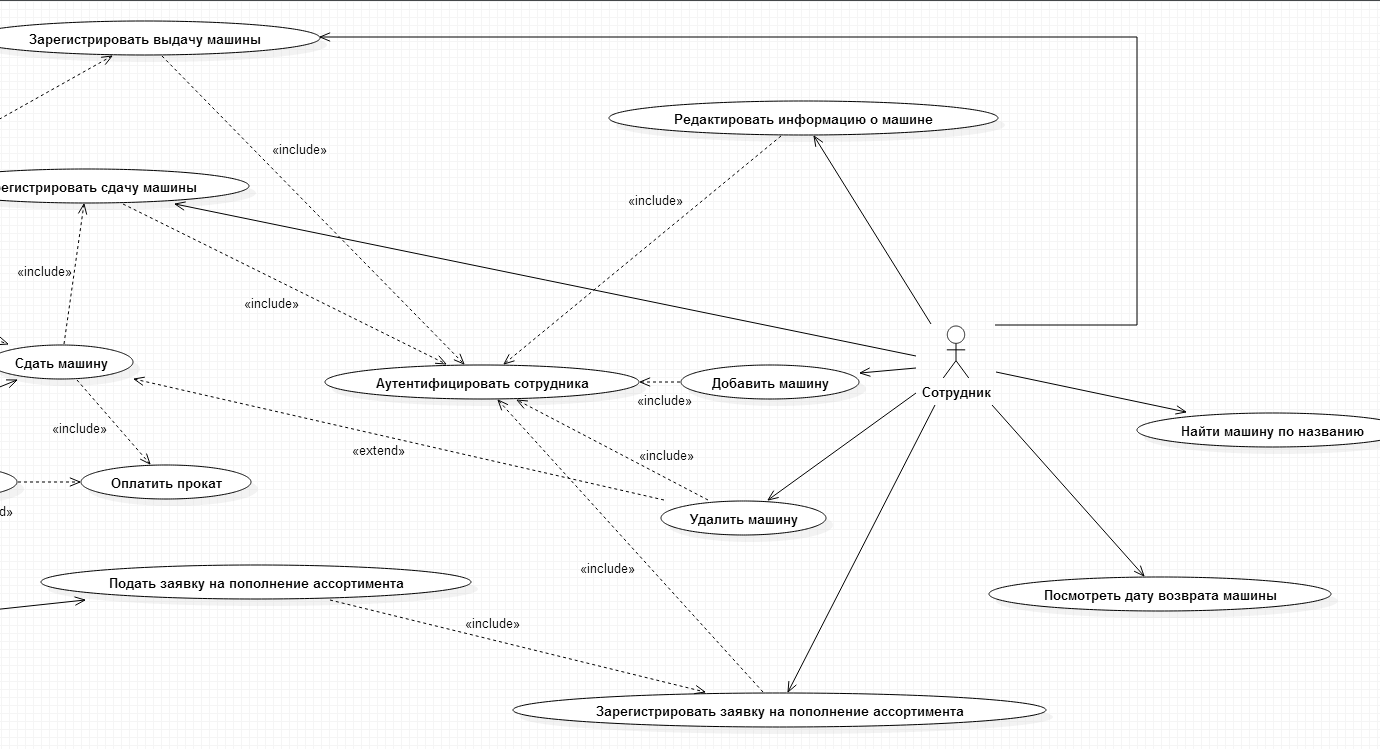
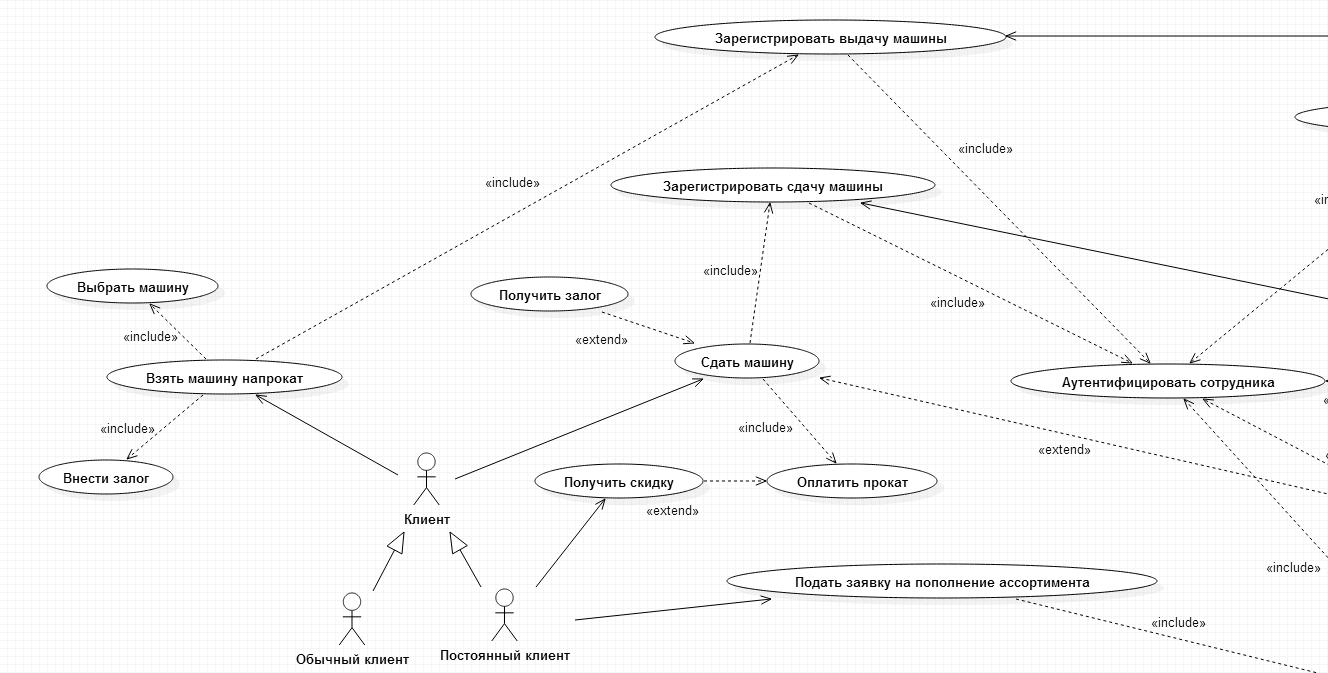
**Описание**

Прокат автомобилей – это компания, которая занимается сдачей автомобилей в прокат. Основными клиентами являются: туристы, бизнесмены, которые ездят в другие города на деловые встречи, люди, которые не могут представить свою повседневную жизнь без автомобиля или же их собственный автомобиль находится в ремонте.

Сотрудник ИС «Прокат автомобилей» должен ознакомить клиента с тем, какие автомобили имеются в наличии, и подобрать подходящий для клиента; также ознакомить со стоимостью, которые предоставляются на автомобили. Прежде чем ознакомить клиента с информацией об автомобилях, которые находятся в системе, сотрудник должен аутентифицироваться в системе, введя логин и пароль. О каждой сдаваемой в прокат машине доступна следующая информация: марка, цвет, время в прокате, срок аренды и имеется ли автомобиль в наличии.

Для клиентов существует разделение на постоянных клиентов и пришедших первый раз. Для клиентов, которые пользовались прокатом автомобилей более чем 5 раз, предусмотрена скидка на прокат, а так же предусмотрена подача заявления на пополнение базы машин автомобилем, которого нет в каталоге.

Когда клиент выбрал подходящую ему машину, сотрудник регистрирует выдачу машины и берет залог. По истечении срока проката, клиент должен сдать машину в прокат. Если машина находится в прежнем состоянии, клиент оплачивает прокат, и ему возвращают залог. Если у машины находят неисправности, то залог не возвращается, а также клиент должен оплатить ремонт машины.

****

**Актеры**   
1. Клиент – любой пользователь системы, который хочет взять машину в прокат.   
• Обычный клиент – пользователь системы, который выбирает машину и оплачивает прокат на общих основаниях, не имея никаких привилегий.   
• Постоянный клиент – пользователь системы, который брал машину в прокат более 5 раз; он имеет право на получение скидки при оплате проката, а также может подать заявку на пополнение ассортимента.

2. Сотрудник – пользователь, который работает с системой: может просматривать информацию о машинах в каталоге, может добавлять, удалять машины из каталога, а также редактировать информацию о машине. Сотрудник регистрирует выдачу машины в прокат клиенту.

**«Взять авто напрокат»**

**Описание:** Вариант использования «Взять авто напрокат» позволяет любому клиенту выбрать авто по каталогу, внести залог и забрать авто на определенный срок.

**Предусловия:** Во время выполнения данного варианта использования должны быть выполнены варианты использования «Аутентифицировать сотрудника», «Выбрать авто», «Внести залог» и «Зарегистрировать выдачу авто».

**Основной поток событий:**

1. Клиент приходит в прокат авто и желает взять машину напрокат.
2. Сотрудник аутентифицируется в системе.
3. Сотрудник открывает каталог машин, имеющихся в наличии в данный момент в системе.
4. Клиент выбирает нужную ему машину по каталогу. Если нужной ему машины нет, выполняется альтернативный поток событий А1.
5. Клиент вносит залог за машину. Если у клиента недостаточно денег, выполняется альтернативный поток событий А2.
6. Сотрудник предоставляет машину клиенту.
7. Клиент сообщает сотруднику срок проката в сутках.
8. Сотрудник запрашивает данные клиента.
9. Клиент сообщает свои данные.
10. Сотрудник проверяет, является ли этот клиент постоянным.
11. Сотрудник рассчитывает стоимость проката за оглашенный период, учитывая скидку, если клиент является постоянным.
12. Сотрудник сообщает клиенту итоговую стоимость проката машины.
13. Клиент соглашается с оглашённой суммой. Если клиента не устраивает сумма проката, выполняется альтернативный поток событий А3.
14. Сотрудник регистрирует сдачу машины в прокат.
15. Вариант использования завершается.

**Альтернативный поток событий А1. Нужная машина отсутствует в каталоге.**

1. Клиент сообщает сотруднику, что нужной ему машины нет в каталоге.
2. Сотрудник проверяет, не сдана ли нужная машина в прокат, и если сдана, то на какой срок.
3. Если машина уже сдана в использование, сотрудник сообщает клиенту дату возврата машины. Если машины нет в системе, выполняется альтернативный поток событий А4.
4. Вариант использования завершается.

**Альтернативный поток событий А2. Недостаточно денег для внесения залога.**

1. Сотрудник сообщает клиенту, что имеющейся суммы недостаточно для внесения залога.
2. Вариант использования завершается.

**Альтернативный поток событий А3. Клиента не устраивает сумма проката.**

1. Сотрудник сообщает клиенту, что выдача машины в случае отказа от оплаты невозможна.
2. Вариант использования завершается.

**Альтернативный поток событий А4. Машины нет в системе.**

1. Сотрудник оповещает клиента об отсутствии машины в системе.
2. Сотрудник проверяет, является ли клиент постоянным (пользовался прокатом 5 и более раз).
3. Если клиент является постоянным, сотрудник регистрирует заявку на пополнение системы данной машиной.
4. Вариант использования завершается.

**«Выбрать машину»**

**Описание:** Вариант использования «Выбрать машину» позволяет любому клиенту выбрать машину по каталогу из имеющихся в системе в данный момент времени.

**Предусловия:** Данный вариант использования выполняется во время выполнения варианта использования «Взять машину напрокат».

**Основной поток событий:**

1. Сотрудник открывает каталог машин, имеющихся в наличии в данный момент в системе.
2. Клиент выбирает нужную ему машину по каталогу. Если нужной ему машины нет, выполняется альтернативный поток событий А1.
3. Вариант использования завершается.

**Альтернативный поток событий А1. Нужная клиенту машина отсутствует в каталоге.**

1. Клиент сообщает сотруднику, что нужной ему машины нет в каталоге.
2. Сотрудник проверяет, не сдана ли нужная машина в прокат, и если сдана, то на какой срок.
3. Если машина уже сдана в использование, сотрудник сообщает клиенту дату возврата машины. Если машины нет в системе, выполняется альтернативный поток событий А2.
4. Вариант использования завершается.

**Альтернативный поток событий А4. Машины нет в системе.**

1. Сотрудник оповещает клиента об отсутствии машины в системе.
2. Сотрудник проверяет, является ли клиент постоянным (пользовался прокатом 5 и более раз).
3. Если клиент является постоянным, сотрудник регистрирует заявку на пополнение системы данной машиной.
4. Вариант использования завершается.

**Постусловие:** после выполнения варианта использования должен быть выполнен ВИ «Внести залог».

**«Внести залог»**

**Описание:** Вариант использования «Внести залог» позволяет любому клиенту взять машину напрокат, внеся за неё в залог необходимую сумму денег.

**Предусловия:** Данный вариант использования выполняется во время выполнения варианта использования «Взять машину напрокат», после выполнения варианта использования «Выбрать машину».

**Основной поток событий:**

1. Клиент вносит залог за машину. Если у клиента недостаточно денег, выполняется альтернативный поток событий А1.
2. Вариант использования завершается.

**Альтернативный поток событий А1. Недостаточно денег для внесения залога.**

1. Сотрудник сообщает клиенту, что имеющейся суммы недостаточно для внесения залога.
2. Вариант использования завершается.

**Постусловие:** после выполнения варианта использования должен быть выполнен ВИ «Зарегистрировать выдачу машины».

**«Зарегистрировать выдачу машины»  
Описание:** Вариант использования «Зарегистрировать выдачу машины» позволяет сотруднику после выдачи машины, получения залога от клиента и уточнения срока сдачи машины зарегистрировать выдачу.

**Предусловия:** Данный вариант использования выполняется после выполнения варианта использования «Аутентифицировать сотрудника».

**Основной поток событий:**

1. Сотрудник аутентифицируется в системе.
2. Сотрудник получает от клиента необходимые сведения о сдаваемой в прокат машине: название машины, срок проката, данные клиента.
3. Сотрудник регистрирует в системе необходимые данные, подтверждающие выдачу машины клиенту.
4. Вариант использования завершается.

**«Сдать машину»**

**Описание:** Вариант использования «Сдать машину» позволяет любому клиенту сдать машину, взятую ранее напрокат, и получить залог.

**Предусловия:** Во время выполнения данного варианта использования должны быть выполнены варианты использования «Аутентифицировать сотрудника», «Оплатить прокат», «Зарегистрировать сдачу машины», а также могут быть выполнены ВИ «Получить залог», «Получить скидку» и «Удалить машину».

**Основной поток событий:**

1. Клиент приходит в автопрокат и желает сдать машину.
2. Сотрудник аутентифицируется в системе.
3. Сотрудник принимает машину от клиента.
4. Сотрудник проверяет, является ли клиент постоянным.
5. Сотрудник рассчитывает и сообщает клиенту стоимость проката.
6. Клиент оплачивает прокат машины, причём сумма учитывает скидку для постоянных клиентов. Если у клиента недостаточно средств, выполняется альтернативный поток событий А1.
7. Сотрудник осматривает машину и заводит её.
8. Если машина не повреждена, сотрудник возвращает клиенту залог. Если машина повреждена, выполняется альтернативный поток событий А2.
9. Сотрудник регистрирует сдачу машины.
10. Сотрудник регистрирует возврат машины системе.
11. Вариант использования завершается.

**Альтернативный поток событий А1. Недостаточно денег для оплаты проката.**

1. Клиент сообщает сотруднику, что у него недостаточно денег для оплаты проката.
2. Клиент просит продлить срок проката на определенное количество дней, чтобы в течение этого времени найти достаточное количество средств.
3. Сотрудник рассчитывает новую сумму проката, учитывая скидку для постоянных клиентов.
4. Сотрудник сообщает клиенту новую сумму.
5. Если клиента устраивает новая сумма, он подтверждает срок продления проката. Если сумма не устраивает, выполняется альтернативный поток событий А2.
6. Сотрудник продлевает срок проката машины на определенное число дней.
7. Клиент забирает машину.
8. Вариант использования завершается.

**Альтернативный поток событий А2. Машина повреждена или клиент не может оплатить прокат.**

1. Сотрудник оставляет залог в кассе.
2. Сотрудник удаляет информацию о машине из системы.
3. Вариант использования завершается.

**«Получить залог»**

**Описание:** Вариант использования «Получить залог» позволяет любому клиенту получить оставленный в кассе проката машин залог при условии, что машина не повреждена.

**Предусловия:** Данный вариант использования выполняется во время выполнения варианта использования «Сдать машину», после выполнения варианта использования «Оплатить прокат».

**Основной поток событий:**

1. Сотрудник принимает машину от клиента.
2. Сотрудник осматривает машину и заводит её.
3. Если машина не повреждена, сотрудник возвращает клиенту залог. Если машина повреждена, выполняется альтернативный поток событий А1.
4. Вариант использования завершается.

**Альтернативный поток событий А1. Машина повреждена.**

1. Сотрудник оставляет залог в кассе.
2. Сотрудник удаляет информацию о машине из системы.
3. Вариант использования завершается.

**«Оплатить прокат»**

**Описание:** Вариант использования «Оплатить прокат» позволяет клиенту оплатить прокат машины, взятой на определенное количество суток.

**Предусловия:** Данный вариант использования выполняется во время выполнения варианта использования «Сдать машину».

**Основной поток событий:**

1. Сотрудник проверяет, является ли клиент постоянным.
2. Сотрудник рассчитывает и сообщает клиенту стоимость проката.
3. Клиент оплачивает прокат машины, причём сумма учитывает скидку для постоянных клиентов. Если у клиента недостаточно средств, выполняется альтернативный поток событий А1.
4. Вариант использования завершается.

**Альтернативный поток событий А1. Недостаточно денег для оплаты проката.**

1. Клиент сообщает сотруднику, что у него недостаточно денег для оплаты проката.
2. Клиент просит продлить срок проката на определенное количество дней, чтобы в течение этого времени найти достаточное количество средств.
3. Сотрудник рассчитывает новую сумму проката, учитывая скидку для постоянных клиентов.
4. Сотрудник сообщает клиенту новую сумму.
5. Если клиента устраивает новая сумма, он подтверждает срок продления проката. Если сумма не устраивает, выполняется альтернативный поток событий А2.
6. Вариант использования завершается.

**Альтернативный поток событий А2. Машина повреждена или клиент не может оплатить прокат.**

1. Сотрудник оставляет залог в кассе.
2. Вариант использования завершается.

**Постусловие:** после выполнения данного варианта использования выполняется «Получить залог».

**«Зарегистрировать сдачу машины»**

**Описание:** Вариант использования «Зарегистрировать сдачу машины» позволяет сотруднику принять машину от клиента.

**Предусловия:** Данный вариант использования выполняется после выполнения варианта использования «Аутентифицировать сотрудника».

**Основной поток событий:**

1. Сотрудник аутентифицируется в системе.
2. Сотрудник регистрирует сдачу машины.
3. Сотрудник регистрирует возврат машины в системе.
4. Вариант использования завершается.

**«Получить скидку»**

**Описание:** Вариант использования «Получить скидку» позволяет постоянному клиенту получить скидку за пользование машиной.

**Предусловия:** вариант использования выполняется во время выполнения ВИ «Оплатить прокат».

**Основной поток событий:**

1. Клиент желает получить скидку на прокат при сдаче машины.
2. Сотрудник принимает машину от клиента.
3. Сотрудник проверяет, является ли клиент постоянным.
4. Если клиент является постоянным, сотрудник рассчитывает сумму за прокат машины с учётом скидки. Иначе выполняется альтернативный поток событий А1.
5. Вариант использования завершается.

**Альтернативный поток событий А1. Клиент не является постоянным**

1. Клиенту отказывают в предоставлении скидки.
2. Вариант использования завершается.

**«Подать заявку на пополнение ассортимента»**

**Описание:** Вариант использования «Подать заявку на пополнение ассортимента» позволяет постоянному клиенту подать заявку на расширение каталога машин.

**Основной поток событий:**

1. Клиент приходит в прокат и просматривает каталог машин.
2. Если нужной клиенту машины нет, он просит сотрудника предоставить ему возможность подать заявку на пополнение каталога.
3. Сотрудник запрашивает у клиента его данные.
4. Сотрудник проверяет, является ли клиент постоянным.
5. Если клиент постоянный, сотрудник предоставляет клиенту такую возможность. Иначе выполняется альтернативный поток событий А1.
6. Клиент оформляет заявку в требуемом виде.
7. Сотрудник принимает заявку клиента.
8. Вариант использования завершается.

**Альтернативный поток событий А1. Клиент не является постоянным**

1. Клиенту отказывают в предоставлении возможности подать заявку.
2. Вариант использования завершается.

**Постусловия:** после выполнения данного ВИ выполняется ВИ «Зарегистрировать заявку на пополнение ассортимента»

**«Зарегистрировать заявку на пополнение ассортимента»**

**Описание:** Вариант использования «Зарегистрировать заявку на пополнение ассортимента» позволяет сотруднику принять от постоянного клиента заявку на расширение каталога машин.

**Предусловия:** ВИ выполняется после выполнения ВИ «Аутентифицировать сотрудника».

**Основной поток событий:**

1. Сотрудник аутентифицируется в системе.
2. Сотрудник проверяет, что заказываемая машина отсутствует в системе, а также не находится в данный момент в пользовании у какого-либо клиента.
3. Сотрудник запрашивает у клиента его данные.
4. Сотрудник проверяет, является ли клиент постоянным.
5. Если клиент постоянный, сотрудник принимает от клиента заявку в требуемом виде. Иначе выполняется альтернативный поток событий А1.
6. Сотрудник регистрирует заявку в системе.
7. Вариант использования завершается.

**Альтернативный поток событий А1. Клиент не является постоянным**

1. Клиенту отказывают в регистрации заявки.
2. Вариант использования завершается.

**«Аутентифицировать сотрудника»**

**Описание:** Вариант использования «Аутентифицировать сотрудника» позволяет сотруднику войти в систему и даёт возможность выполнять какие-либо действия с системой.

**Основной поток событий:**

1. Сотрудник нуждается в выполнении какого-либо действия с системой.
2. Сотрудник запускает систему.
3. Система открывает окно входа.
4. Сотрудник вводит логин.
5. Сотрудник вводит пароль.
6. Сотрудник нажимает на кнопку «Войти». Если во время входа возникают ошибки, выполняется поток ошибок Е1
7. Система аутентифицирует сотрудника по введённым данным и предоставляет ему интерфейс для совершения операций.
8. Вариант использования завершается.

**Поток ошибок Е1. Неправильный логин или пароль.**

1. Система выводит сообщение о неверном логине или пароле.
2. Система предлагает сотруднику заново ввести его учётные данные.
3. Сотрудник заново вводит логин.
4. Сотрудник заново вводит пароль.
5. Сотрудник нажимает на кнопку «Войти». Если во время входа возникают ошибки, выполняется поток ошибок Е1
6. Система аутентифицирует сотрудника по введённым данным и предоставляет ему интерфейс для совершения операций.
7. Вариант использования завершается.

**«Посмотреть дату возврата машин»**

**Описание:** Вариант использования «Посмотреть дату возврата машины» позволяет сотруднику проверить, когда будет возвращена та или иная машина из сданных в прокат.

**Основной поток событий:**

1. Сотрудник аутентифицируется в системе.
2. Сотрудник ищет нужную ему машину.
3. Сотрудник открывает подробное описание машины, в котором указано, когда машина будет возвращена.
4. Вариант использования завершается.

**«Найти машину по названию»**

**Описание:** Вариант использования «Найти машину по названию» позволяет сотруднику найти машину, имеющуюся в системе, по названию (или части названия).

**Основной поток событий:**

1. Сотрудник аутентифицируется в системе.
2. Сотрудник вводит название или часть названия машины в определенное поле в системе.
3. Сотрудник нажимает кнопку «Найти».
4. Система выводит все найденные по данному запросу машины. Если машины с таким названием нет, выполняется альтернативный поток событий А1.
5. Сотрудник просматривает список найденных машин и выбирает нужную.
6. Вариант использования завершается.

**Альтернативный поток событий А1. Машин с таким название нет.**

1. Система сообщает сотруднику, что таких машин нет.
2. Система возвращает сотрудника на стадию ввода названия машины.
3. Сотрудник повторно вводит название или завершает работу с системой.
4. Вариант использования завершается.

**«Удалить машину»**

**Описание:** Вариант использования «Удалить машину» позволяет сотруднику удалить машину из системы, например, если она повреждена.

**Предусловия:** ВИ выполняется после выполнения ВИ «Аутентифицировать сотрудника».

**Основной поток событий:**

1. Сотрудник аутентифицируется в системе.
2. Сотрудник находит нужную ему машину, подлежащую удалению.
3. Сотрудник нажимает кнопку «Удалить».
4. Машина удаляется из системы. Если происходит какой-либо сбой в системе, выполняется поток ошибок Е1.
5. Система оповещает сотрудника об успешном удалении машины.
6. Вариант использования завершается.

**Поток ошибок Е1. Ошибка при удалении машины.**

1. Система выводит сообщение о том, что в процессе удаления машины из системы произошла ошибка.
2. Система предлагает попробовать совершить операцию удаления позже.
3. Система возвращает сотрудника в основное меню приложения.
4. Вариант использования завершается.

**«Добавить машину»**

**Описание:** Вариант использования «Добавить машину» позволяет сотруднику добавить новую машину в систему.

**Предусловия:** ВИ выполняется после выполнения ВИ «Аутентифицировать сотрудника».

**Основной поток событий:**

1. Сотрудник аутентифицируется в системе.
2. Сотрудник выбирает пункт меню системы «Добавить машину».
3. Система открывает окно с полями, которые требуется заполнить информацией о добавляемой машине.
4. Сотрудник вводит необходимую информацию.
5. Сотрудник нажимает кнопку «Добавить».
6. Машина добавляется в систему. Если происходит какой-либо сбой в системе, выполняется поток ошибок Е1.
7. Система оповещает сотрудника об успешном совершении операции.
8. Вариант использования завершается.

**Поток ошибок Е1. Ошибка при добавлении машины.**

1. Система выводит сообщение о том, что в процессе добавления машины произошла ошибка.
2. Система предлагает попробовать совершить операцию позже.
3. Система возвращает сотрудника в основное меню приложения.
4. Вариант использования завершается.

**«Редактировать информацию о машине»**

**Описание:** Вариант использования «Редактировать информацию о машине» позволяет сотруднику изменить информацию о машине в системе.

**Предусловия:** ВИ выполняется после выполнения ВИ «Аутентифицировать сотрудника».

**Основной поток событий:**

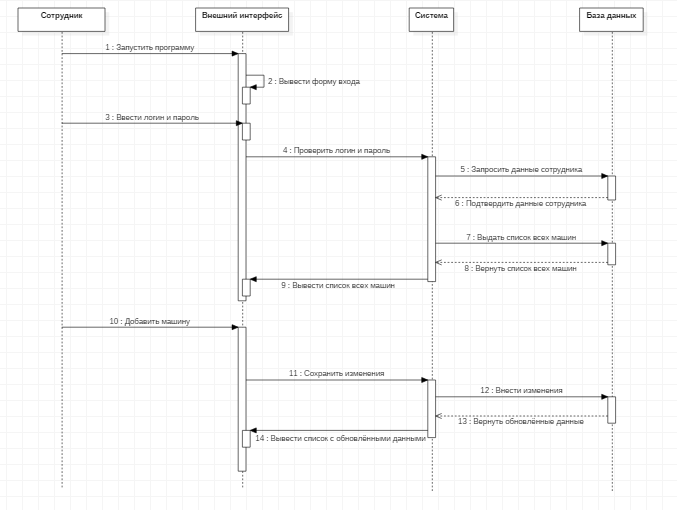
1. Сотрудник аутентифицируется в системе.
2. Сотрудник находит нужную ему машину.
3. Сотрудник нажимает кнопку «Редактировать».
4. Система открывает окно с полями, которые содержат информацию о машине.
5. Сотрудник вносит необходимые изменения.
6. Сотрудник нажимает кнопку «Сохранить изменения».
7. Система сохраняет отредактированные данные. Если происходит какой-либо сбой в системе, выполняется поток ошибок Е1.
8. Система оповещает сотрудника об успешном совершении операции.
9. Вариант использования завершается.

**Поток ошибок Е1. Ошибка при сохранении информации о машине.**

1. Система выводит сообщение о том, что в процессе сохранения изменённой информации о машине произошла ошибка.
2. Система предлагает попробовать совершить операцию позже.
3. Система возвращает сотрудника в основное меню приложения.
4. Вариант использования завершается.

**«Добавить машину»**

**Диаграмма последовательности**



**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника
7. Выдать список всех машин
8. Вернуть список всех машин
9. Вывести список всех машин
10. Добавить машину
11. Сохранить изменения
12. Внести изменения
13. Вернуть обновленные данные
14. Вывести список с обновленными данными

Добавление информации о машине инициируется пользователем системы – сотрудником. Перед выполнением данного варианта использования выполняется ВИ «Аутентифицировать сотрудника»: после запуска программы внешний интерфейс выводит окно входа, в которое сотрудник вводит свои учётные данные – логин и пароль. Внешний интерфейс отправляет для подтверждения введённые данные системе, которая, в свою очередь, совершает запрос к базе данных. База данных возвращает системе данные сотрудника. В случае верно введённых данных система запрашивает у базы данных список всех имеющихся машин. База данных возвращает требуемые данные, а система требует от внешнего интерфейса вывести список машин сотруднику. Сотрудник просматривает список и проверяет имеется ли запрашиваемая машина в списке. При ее отсутствие в списке, Внешний интерфейс открывает форму добавления машины. Сотрудник вводит данные и нажимает кнопку «Сохранить». Внешний интерфейс отправляет системе данные о маштине, а та, в свою очередь, совершает запрос к базе данных о внесении новой машины. База данных возвращает обновленные данные системе, после чего система требует от внешнего интерфейса вывести список с обновленными данными. После вывода на экран обновленных данных вариант использования завершается.

**Диаграмма кооперации**

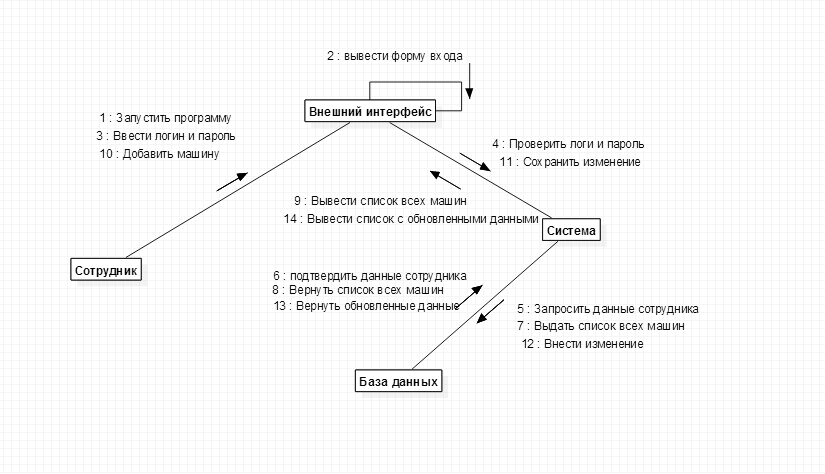
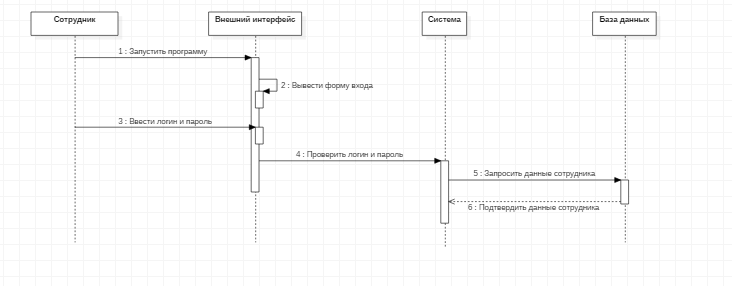


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Аутентификация»**

**Диаграмма последовательности**

****

**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника

**Диаграмма кооперации**

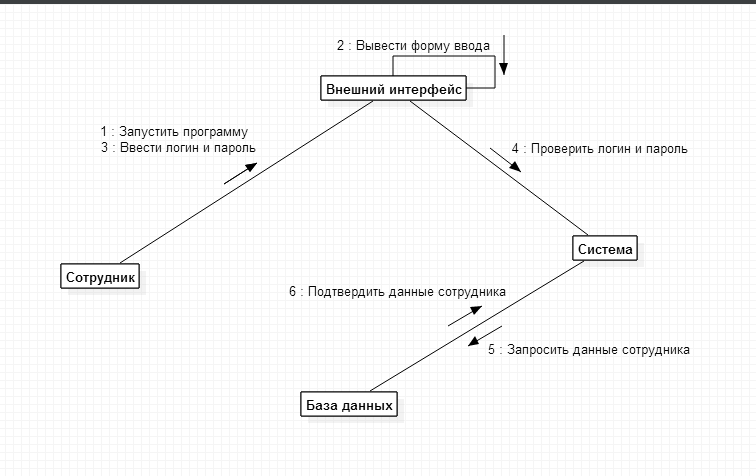
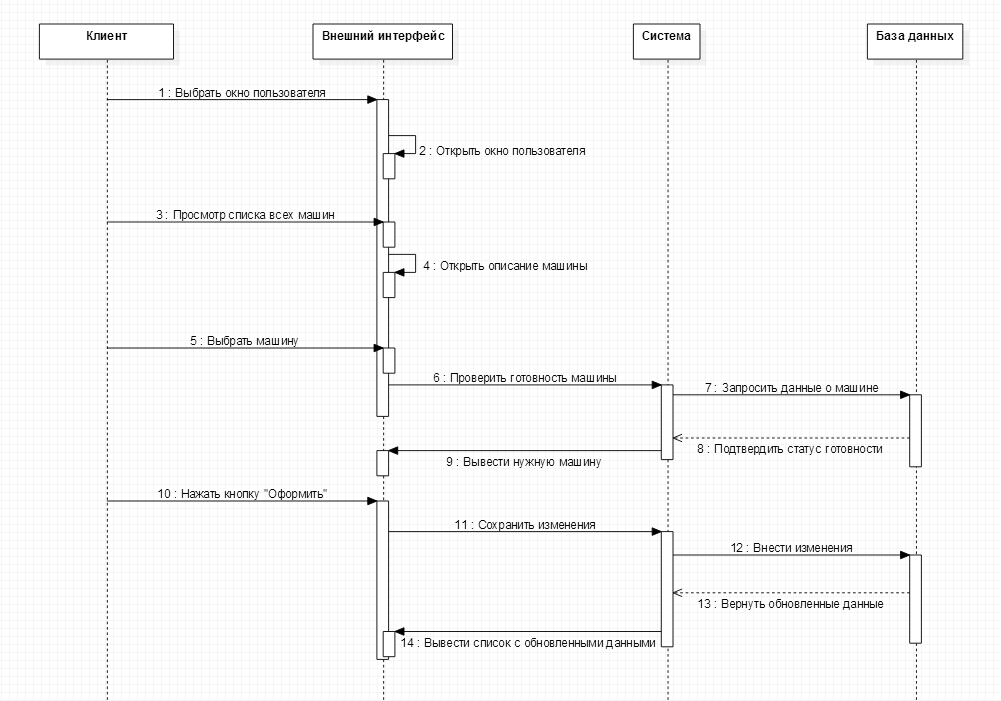
****

Диаграмма кооперации отражает те же объекты и сообщения, что и диаграмма последовательности.

**«Взять машину на прокат»**

**Диаграмма последовательности**

****

**Объекты**

1.Клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Просмотр списка всех машин
4. Открыть описание машины
5. Выбрать машину
6. Проверить готовность машины
7. Запросить данные о машине
8. Подтвердить статус готовности
9. Вывести нужную машину
10. Нажать кнопку «Оформить»
11. Сохранить изменения
12. Внести изменения
13. Вернуть обновленные данные
14. Вывести список с обновленными данными

**Диаграмма кооперации**

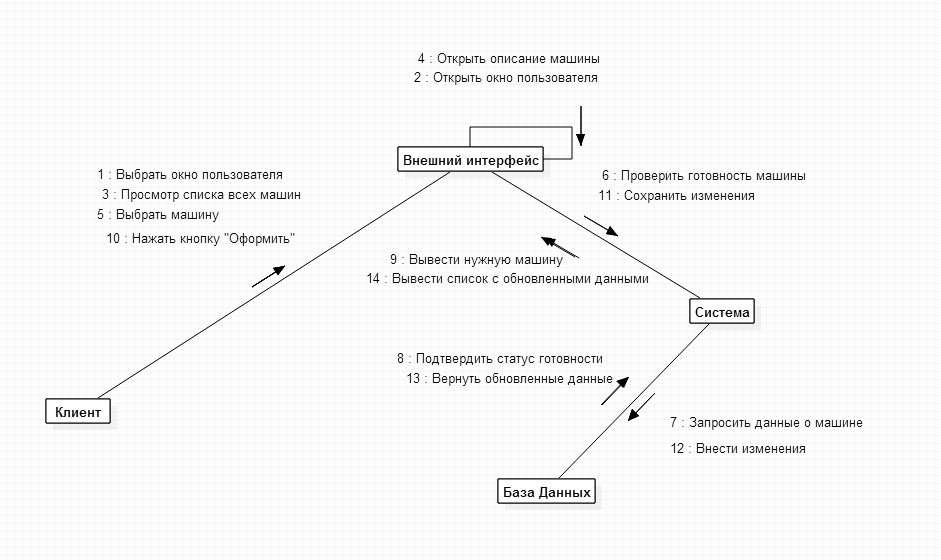
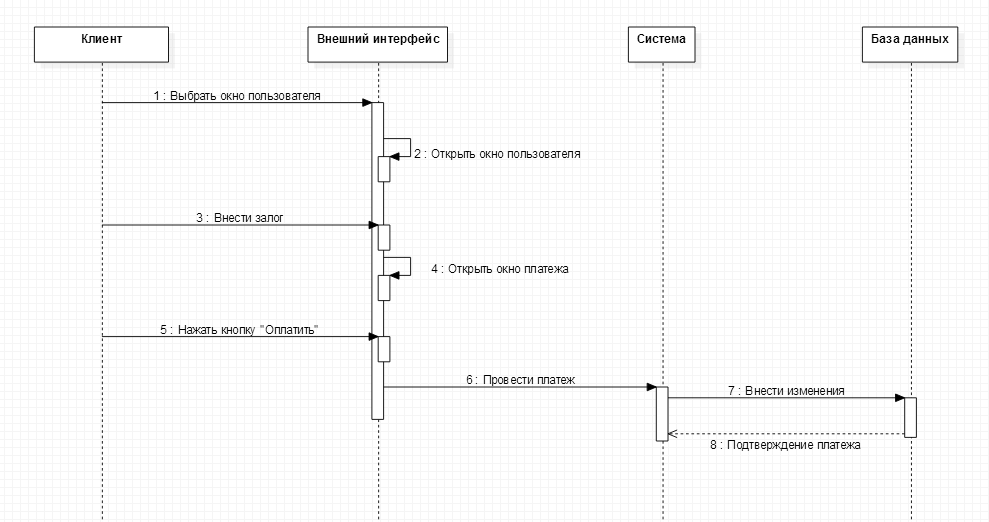


Диаграмма кооперации отражает те же объекты и сообщения, что и диаграмма последовательности.

**«Внести залог»**

**Диаграмма последовательности**

****

**Объекты**

1.Клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Внести залог
4. Открыть окно платежа
5. Нажать кнопку оплатить
6. Проверить платеж
7. Внести изменения
8. Подтверждение платежа

**Диаграмма кооперации**

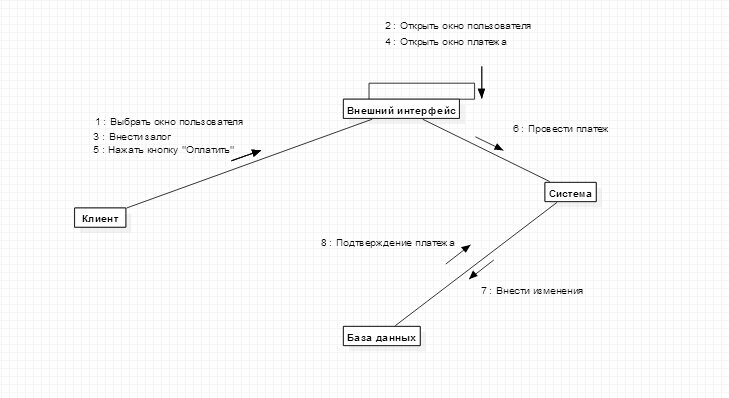
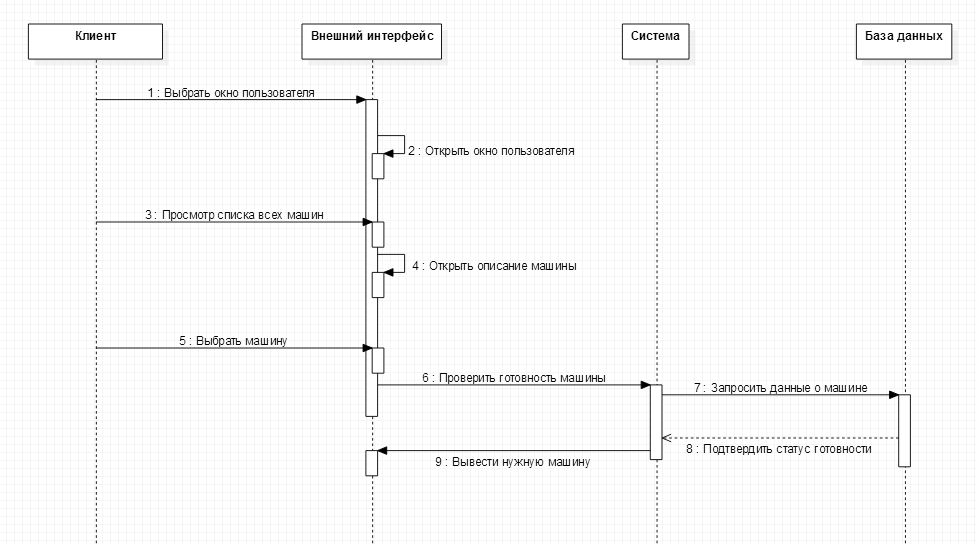
****

Диаграмма кооперации отражает те же объекты и сообщения, что и диаграмма последовательности.

**«Выбрать машину»**

**Диаграмма последовательности**



**Объекты**

1.Клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Просмотр списка всех машин
4. Открыть описание машины
5. Выбрать машину
6. Проверить готовность машины
7. Запросить данные о машине
8. Подтвердить статус готовности
9. Вывести нужную машину

**Диаграмма кооперации**

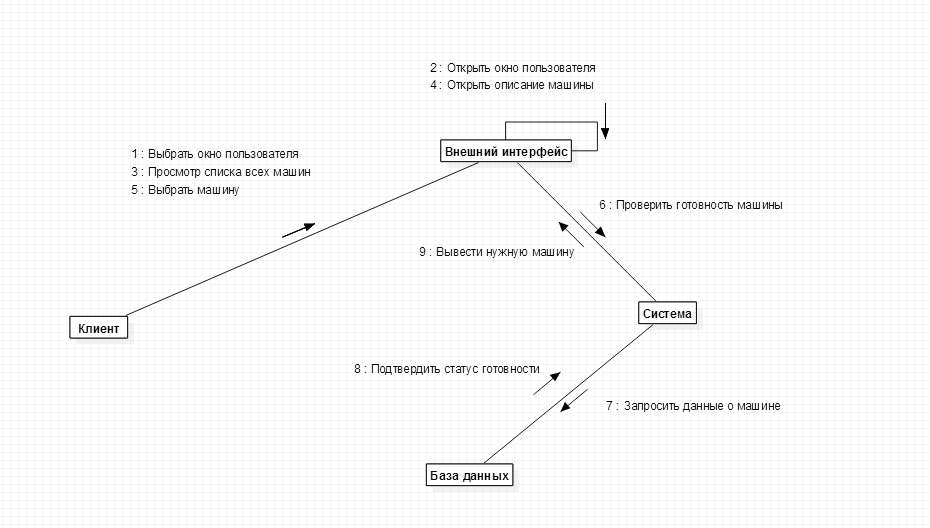
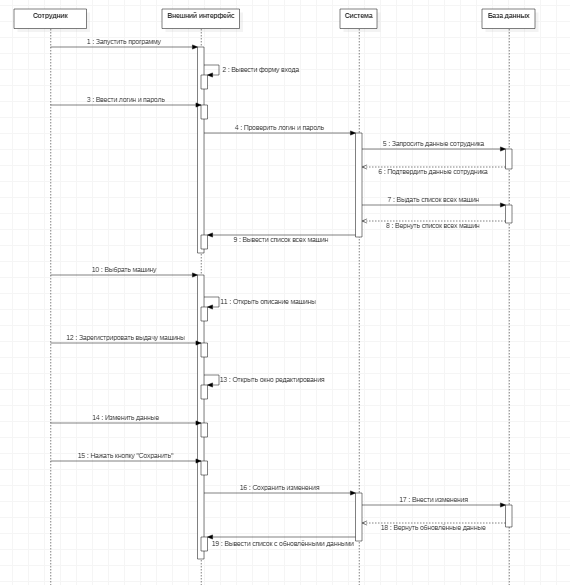


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Зарегистрировать выдачу машины»**

**Диаграмма последовательности**



**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника
7. Выдать список всех машин
8. Вернуть список всех машин
9. Вывести список всех машин
10. Выбрать машину
11. Открыть описание машины
12. Зарегистрировать выдачу машины
13. Открыть окно редактирования
14. Изменить данные
15. Нажать кнопку «Сохранить»
16. Сохранить изменения
17. Внести изменения
18. Вернуть обновленные данные
19. Вывести список с обновленными данными

**Диаграмма кооперации**

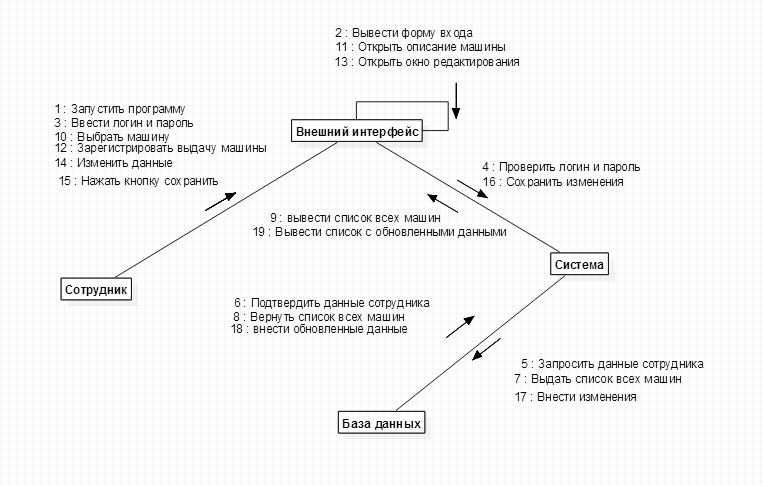
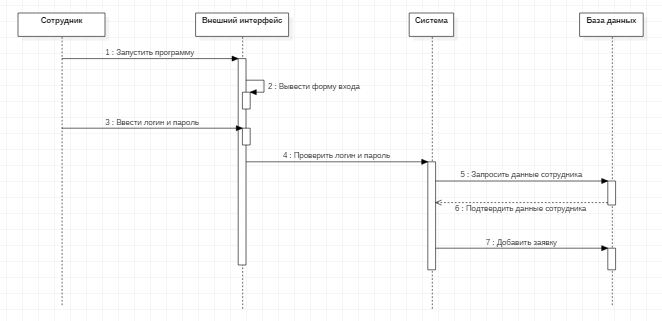


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Зарегистрировать заявку на пополнение ассортимента»**

**Диаграмма последовательности**



**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника
7. Добавить заявку

**Диаграмма кооперации**

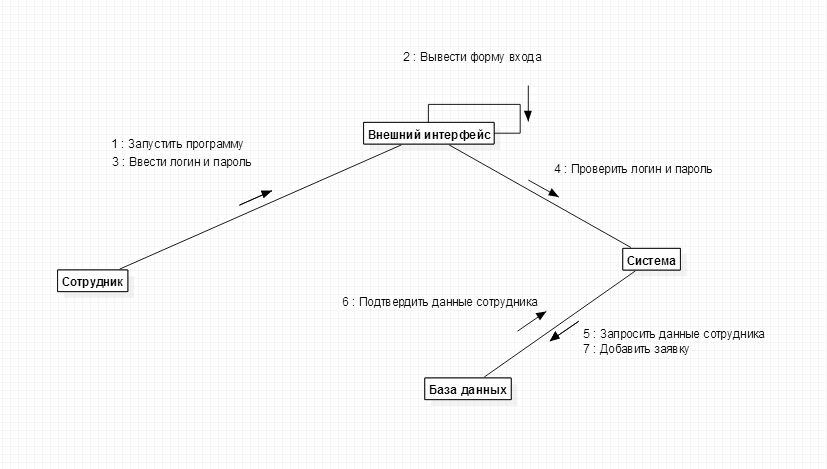
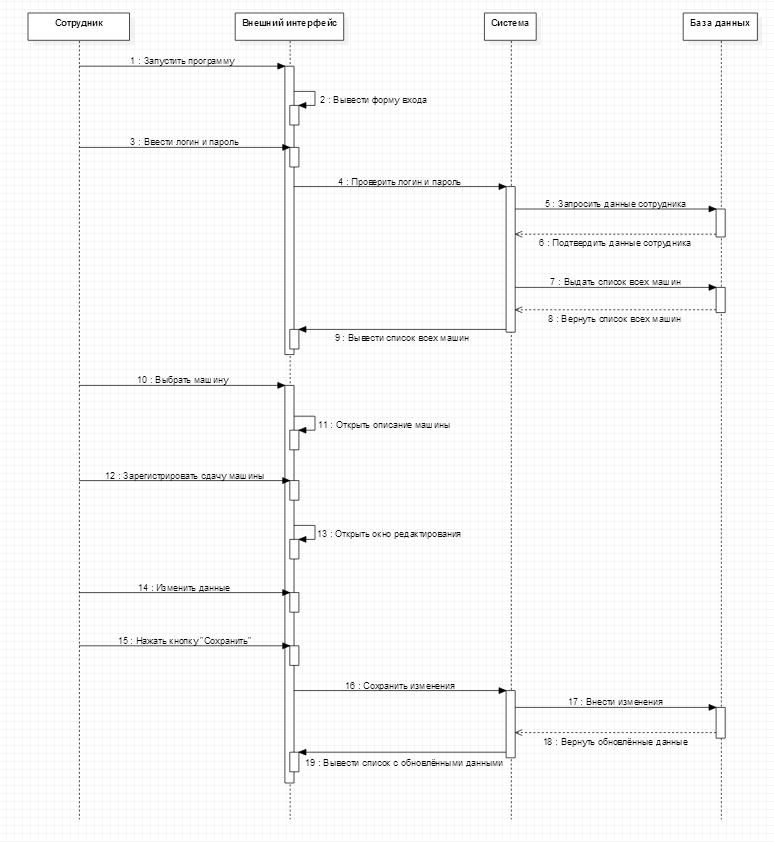


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Зарегистрировать сдачу машины»**

**Диаграмма последовательности**



**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника
7. Выдать список всех машин
8. Вернуть список всех машин
9. Вывести список всех машин
10. Выбрать машину
11. Открыть описание машины
12. Зарегистрировать сдачу машины
13. Открыть окно редактирования
14. Изменить данные
15. Нажать кнопку «Сохранить»
16. Сохранить изменения
17. Внести изменения
18. Вернуть обновленные данные
19. Вывести список с обновленными данными

**Диаграмма кооперации**

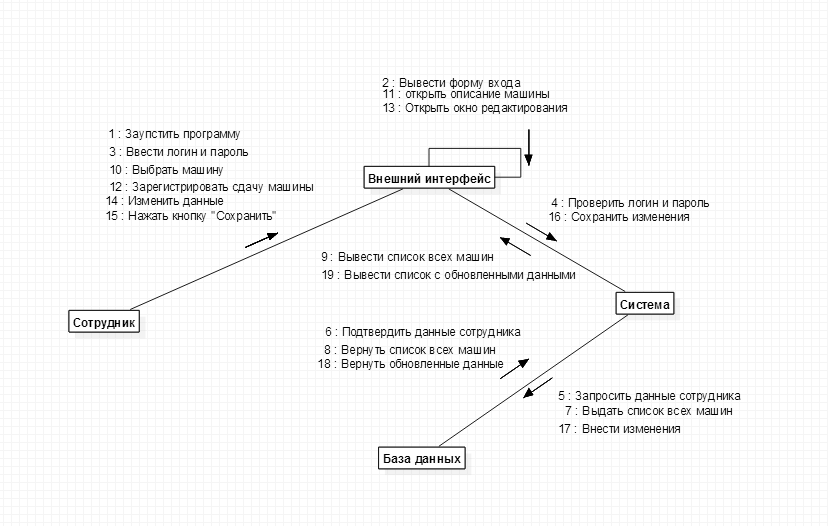
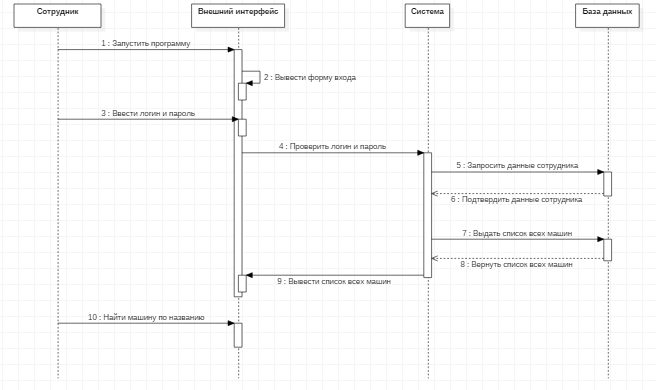


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Найти машину по названию»**

**Диаграмма последовательности**



**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника
7. Выдать список всех машин
8. Вернуть список всех машин
9. Вывести список всех машин
10. Найти машину по названию

**Диаграмма кооперации**

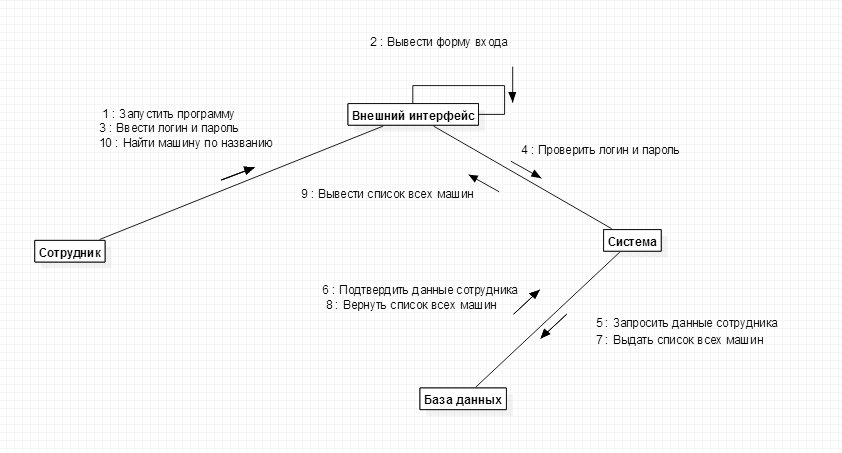
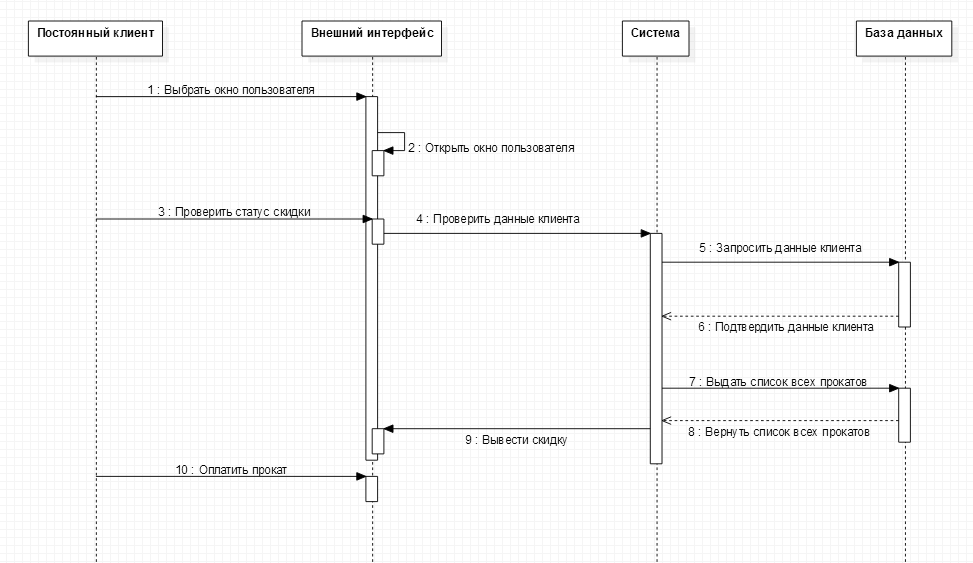


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Оплатить прокат»**

**Диаграмма последовательности**



**Объекты**

1.Постоянный клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Проверить статус скидки
4. Проверить данные клиента
5. Запросить данные клиента
6. Подтвердить данные клиента
7. Выдать список всех прокатов
8. Вернуть список всех прокатов
9. Вывести скидку
10. Оплатить прокат

**Диаграмма кооперации**

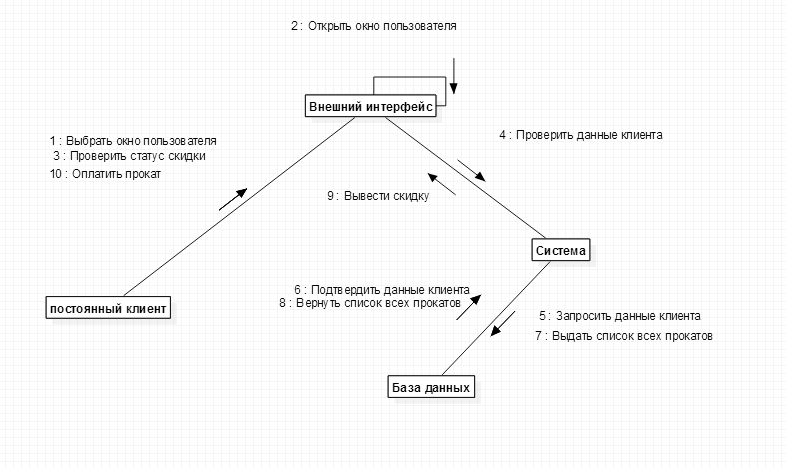
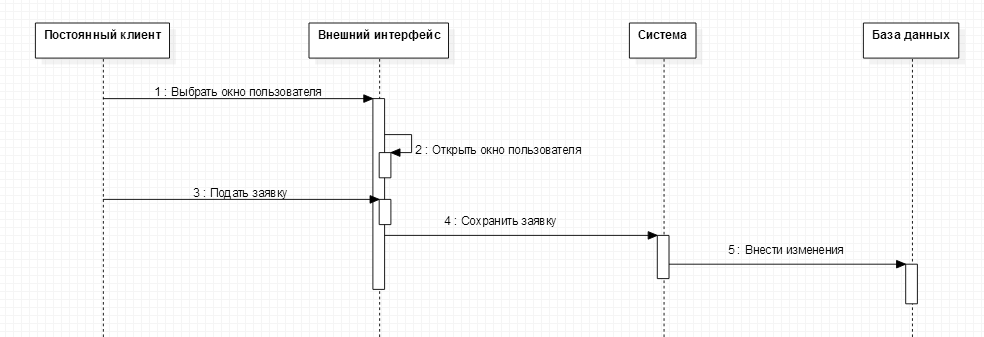


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Подать заявку»**

**Диаграмма последовательности**



**Объекты**

1.Постоянный клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Подать заявку
4. Сохранить заявку
5. Внести изменения

**Диаграмма кооперации**

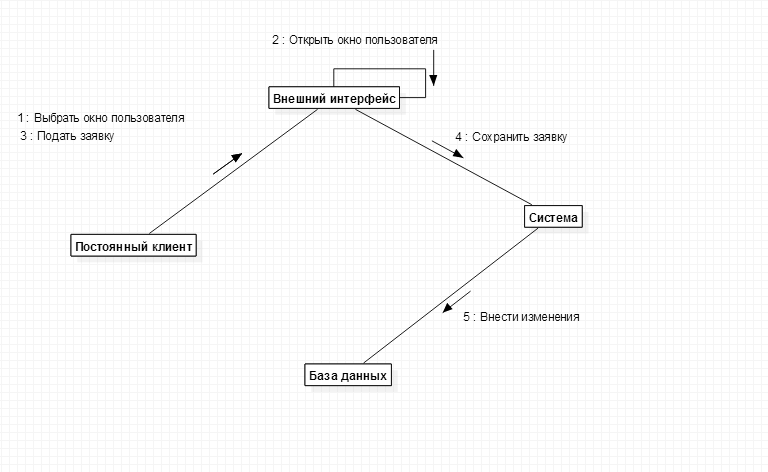
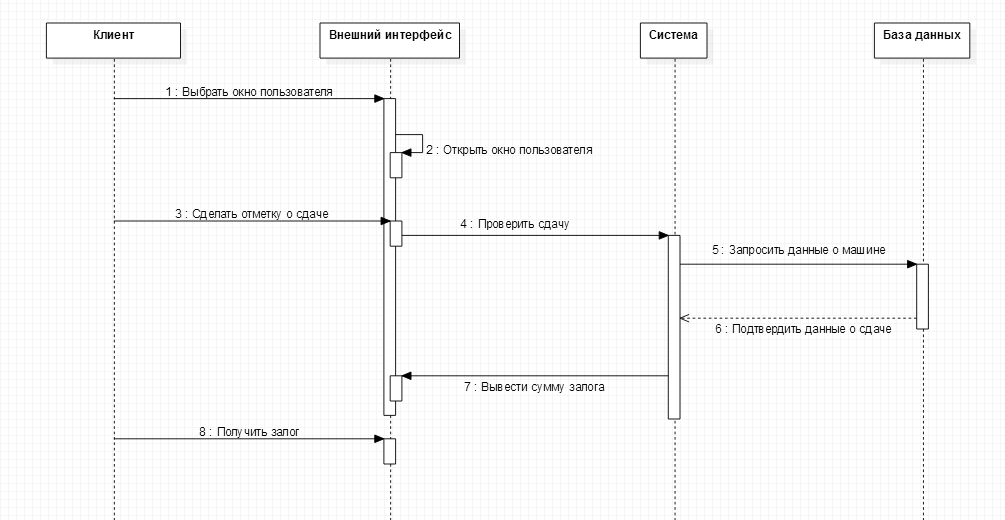


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Получить залог»**

**Диаграмма последовательности**



**Объекты**

1.Клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Сделать отметку о сдаче
4. Проверить сдачу
5. Запросить данные о машине
6. Подтвердить данные о сдаче
7. Вывести сумму залога
8. Получить залог

**Диаграмма кооперации**

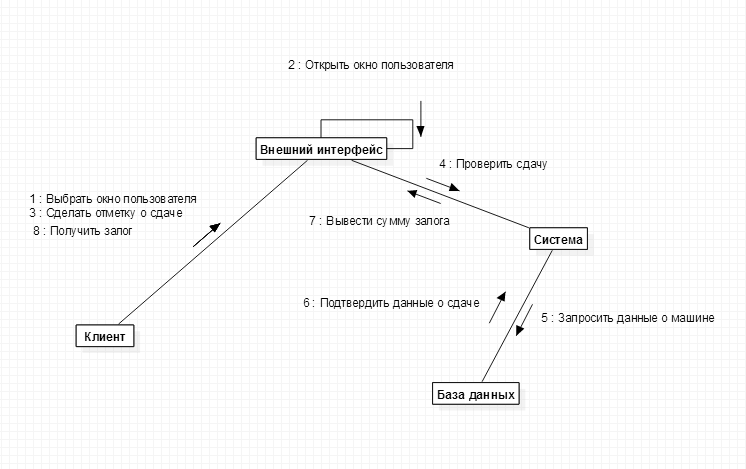
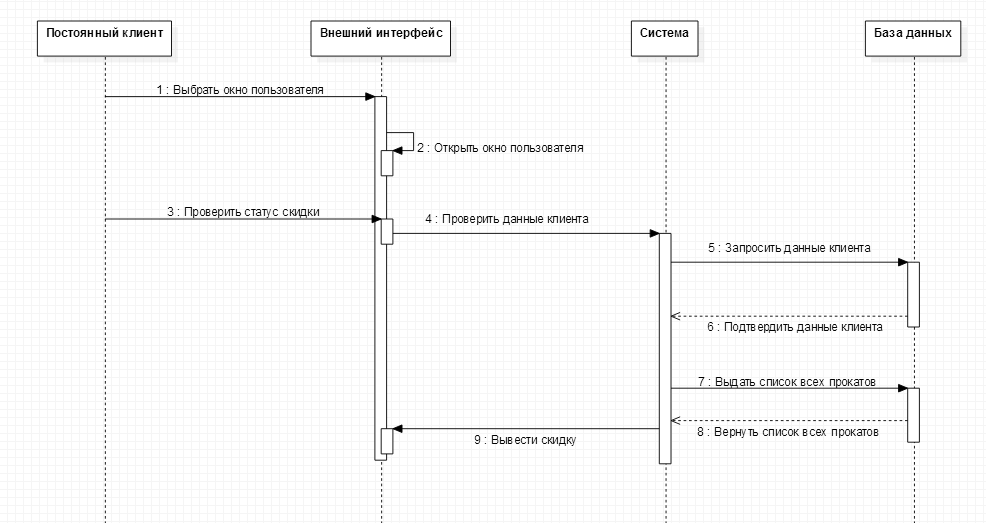


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Получить скидку»**

**Диаграмма последовательности**



**Объекты**

1.Постоянный клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Проверить статус скидки
4. Проверить данные клиента
5. Запросить данные клиента
6. Подтвердить данные клиента
7. Выдать список всех прокатов
8. Вернуть список всех прокатов
9. Вывести скидку

**Диаграмма кооперации**

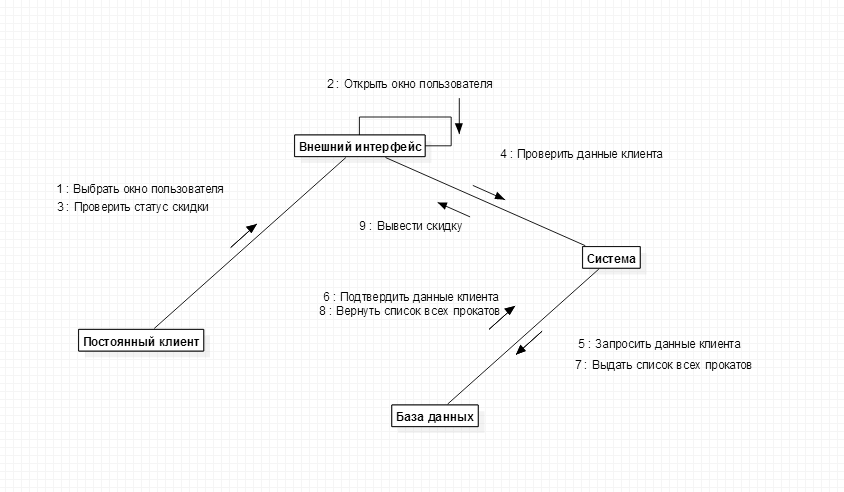
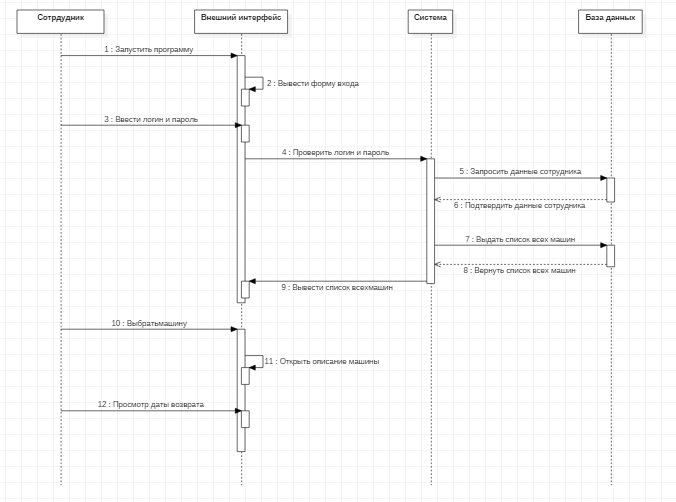


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Посмотреть дату возврата машины»**

**Диаграмма последовательности**



**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника
7. Выдать список всех машин
8. Вернуть список всех машин
9. Вывести список всех машин
10. Выбрать машину
11. Открыть описание машины
12. Посмотреть дату возврата машины

**Диаграмма кооперации**

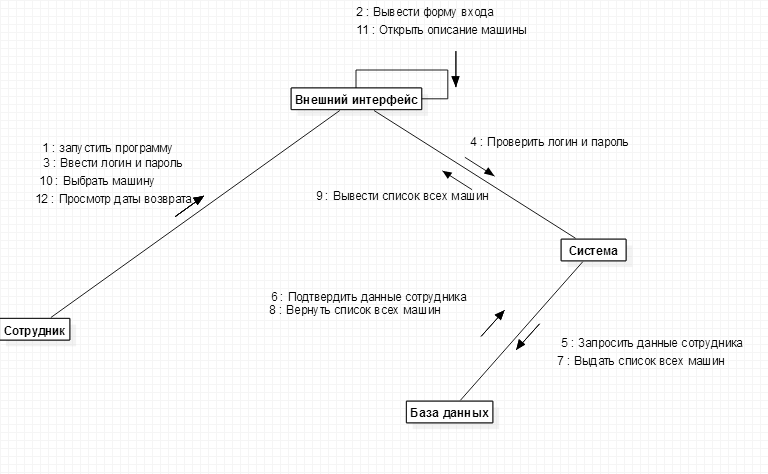
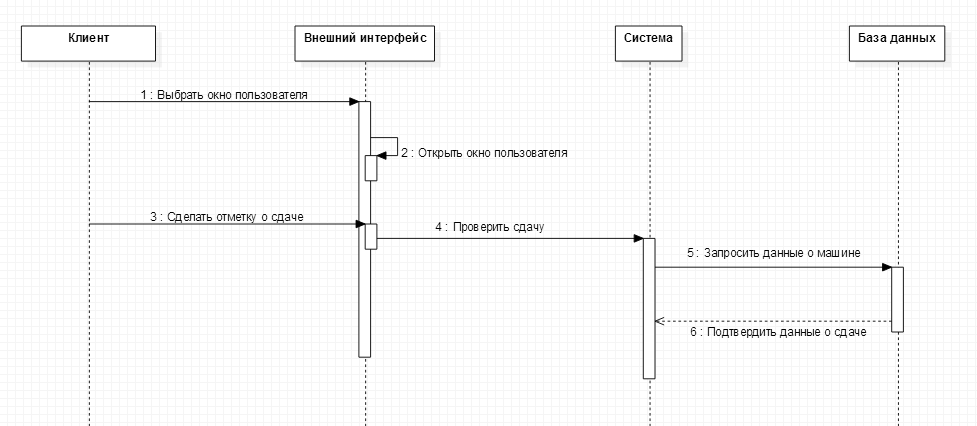


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Сдать машину»**

**Диаграмма последовательности**



**Объекты**

1.Клиент

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Выбрать окно пользователя
2. Открыть окно пользователя
3. Сделать отметку о сдаче
4. Проверить сдачу
5. Запросить данные о машине
6. Подтвердить данные о сдаче

**Диаграмма кооперации**

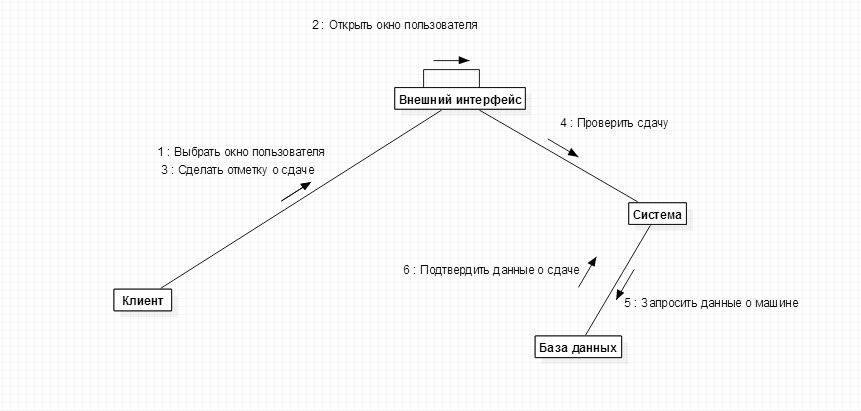
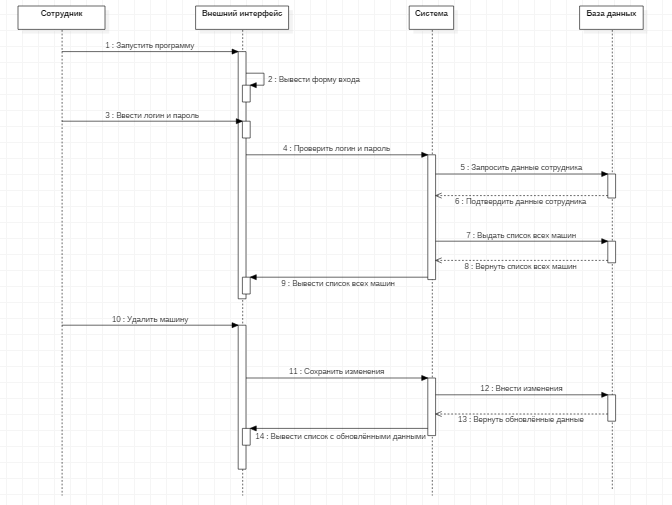


Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**«Удалить машину»**

**Диаграмма последовательности**



**Объекты**

1.Сотрудник

2.Внешний интерфейс

3.Система

4.База данных

**Сообщения между объектами:**

1. Запустить программу
2. Вывести форму входа
3. Ввести логин и пароль
4. Проверить логин и пароль
5. Запросить данные сотрудника
6. Подтвердить данные сотрудника
7. Выдать список всех машин
8. Вернуть список всех машин
9. Вывести список всех машин
10. Удалить машину
11. Сохранить изменения
12. Внести изменения
13. Вернуть обновленные данные
14. Вывести список с обновленными данными

**Диаграмма кооперации**



Диаграмма кооперации отражает те же объекты и процессы, что и диаграмма последовательности.

**Диаграмма пакетов.**

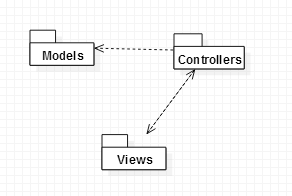
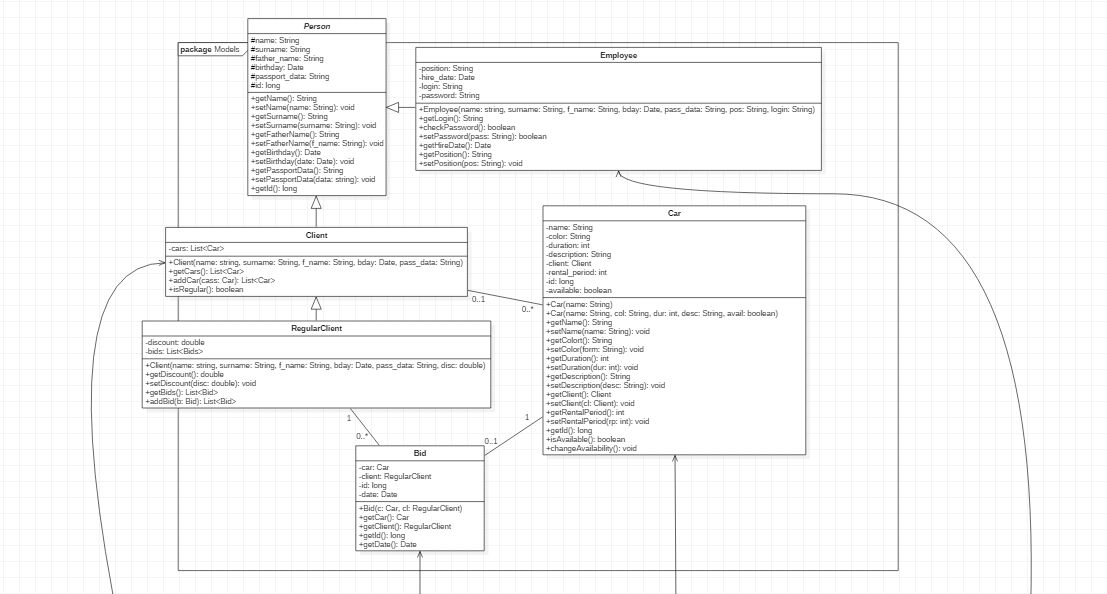
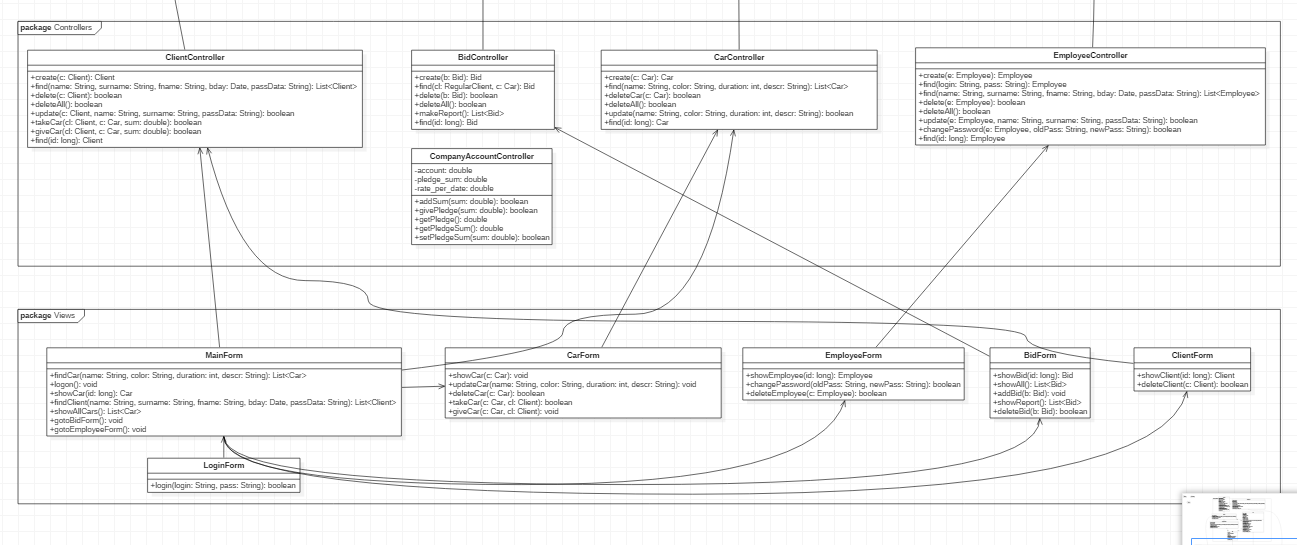


Диаграмма пакетов программы «Прокат машин» построена по принципу группировки классов по стереотипам и приведена в соответствие с системным паттерном проектирования Models-Views-Controllers (MVC). Система разделена на 3 логические части: **модель, представление и контроллер**. Пакет Models содержит классы сущностей предметной области. Пакет Controllers является промежуточным классом, содержащим бизнес-логику программы, обрабатывающим запросы от классов пакета Views и выполняющим действия с сущностями из пакета Models. Таким образом, классы из пакета Controllers посылают сообщения классам пакетов Models и Views, поэтому стрелки от пакета Controllers направлены к обоим соседним пакетам. Классы пакета Models не посылают сообщения ни одному классу из пакетов Controllers или Views, что отражено отсутствием стрелок от пакета Models. Пакет Views предназначен для отображения данных пользователю и обработку его действий. При совершении пользователем некоторого действия класс пакета Views делает вызов соответствующего метода из классов пакета Controllers. В связи с этим на диаграмме указана стрелка от пакета Views к пакету Controllers.

**Диаграмма классов.**

****

****

Пакет **Models** содержит классы сущностей предметной области. В данный пакет входят классы:

* Абстрактный класс Person, который является родительским классом для классов Employee и Client и содержит общие для данных потомков поля и общие методы get и set для доступа и изменения полей: name, surname, father\_name, id, birthday, passport\_data.
* Класс Employee – класс сотрудника, потомок Person, наследующий его поля и методы, содержащий поля position (должность), hire\_date (дата приёма на работу), login и password (учётные данные для входа в систему). Также класс содержит конструктор, методы get и set для доступа и изменения полей, метод checkPassword для проверки правильности учётных данных. Связь между классами Employee и Person показана на диаграмме стрелкой «Обобщение*». Связь между классами осуществляется через поля*(in name:string, in surname:String, in f\_name:String, in bday:Date, in pass\_data:String,)
* Класс Client – класс клиента, потомок Person, наследующий его поля и методы, содержащий поле cars – список всех машин, которые брал напрокат клиент за всё время его пользования прокатом. Также класс содержит конструктор, метод getCars для доступа к полю cars, метод addCar для добавления новой машины в список cars, метод isRegular, показывающий, является ли клиент постоянным. Связь между классами Client и Person показана на диаграмме стрелкой «Обобщение». *Связь между классами осуществляется через поля* (in name:string, in surname:String, in f\_name:String, in bday:Date, in pass\_data:String)
* Класс RegularClient – класс постоянного клиента, потомок Person и Client, наследующий их поля и методы, содержащий поля bids (список заявок клиента) и discount (размер скидки для данного клиента). Также класс содержит конструктор, методы get и set для доступа и изменения полей, метод addBid для добавления новой заявки в список bids. Связь между классами RegularClient и Client показана на диаграмме стрелкой «Обобщение», а связь между классами RegularClient и Person осуществляется через иерархию наследования по классу Client. *Связь между классами RegularClient и Client, RegularClient и Person осуществляется через поля* (in name:string, in surname:String, in f\_name:String, in bday:Date, in pass\_data:String)
* Класс Car – класс машины, содержащий поля name, color, id, duration (время в прокате), description (описание машины, которое может содержать дополнительную информацию о машине), client (ссылка на клиента, который на данный момент пользуется этой машиной), rental\_period (длительность проката в днях), available (доступность машины для взятия её напрокат). Также класс содержит конструкторы, методы get и set для доступа и изменения полей, метод isAvailable для проверки доступности машины для взятия её в прокат, метод changeAvailability для изменения доступности машины. Между классом Car и классом Client существует связь: у каждого клиента от 0 и больше машин в списке cars, а у каждой машины может быть только 1 клиент, если она на данный момент сдана в прокат, или не быть ни одного клиента, если машина находится в пункте автопроката. Данная связь показана на диаграмме стрелкой «Ассоциация» с указанием множественности на обоих её концах.

*Связь между классом Car и классом Client и Bid осуществляется через поле* client.

*Связь между классом Car и классом CarController осуществляется через поля* (in name:String, in color:String, in duration:int, in descr:String)

* Класс Bid – класс заявки, содержащий поля car (машина, которую клиент хочет видеть в прокате), client (постоянный клиент, сделавший заявку), id, date (дата создания заявки). Также класс содержит конструктор, методы get для доступа к полям. Между классом Bid и классом RegularClient существует связь: у каждого постоянного клиента от 0 и больше заявок в списке bids, а у каждой заявки обязательно есть клиент, и только один. Между классом Bid и классом Car также существует связь: в каждой заявке указана одна машина, а у каждой машины может быть одна заявка на неё или не быть заявки вовсе. Данные связи показаны на диаграмме стрелками «Ассоциация» с указанием множественности на обоих концах.

*Между классом Bid и классом RegularClient существует связь через поля bid*

*Между классом Bid и классом Car связь через поле car*

*Между классом Bid и классом BidController через поля (in cl:RegularClient, in c:Car)*

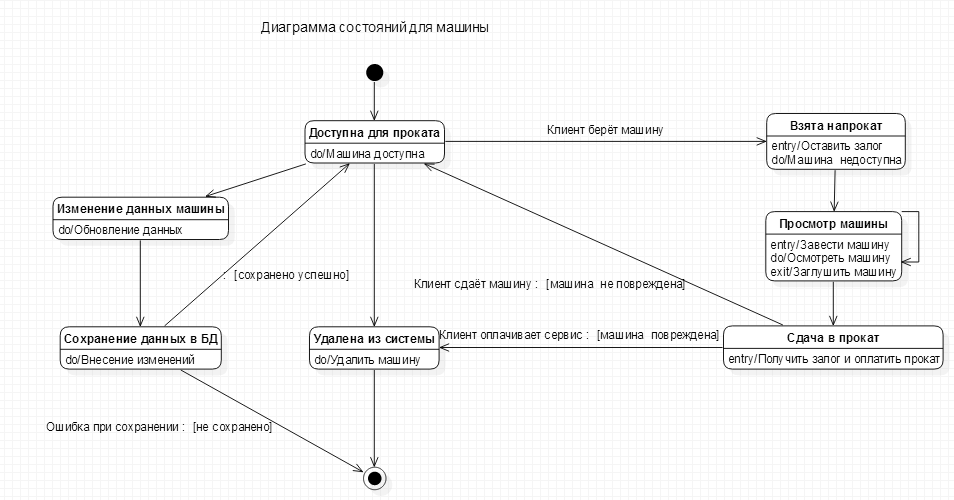
Пакет **Controllers** содержит классы бизнес-логики, выполняющие действия с сущностями из пакета Models. В данный пакет входят классы:

* Класс ClientController – класс, управляющий клиентами (обычными и постоянными). Позволяет создавать клиента, находить клиентов по данным (имени, фамилии и т.д.) или id, изменять информацию о клиенте, удалять одного или всех клиентов, брать машину напрокат и сдавать её. Данный класс управляет классом Client, что отражено на диаграмме стрелкой «Ассоциация» с направлением от ClientController к Client. *Связаны через поля (in name:String, in surname:String, in fname:String, in bday:Date, in passData:String)*
* Класс BidController – класс, управляющий заявками. Позволяет создавать заявку, находить заявки по клиенту и машине или по id, удалять одну или все заявки, составлять отчёт по заявкам. Данный класс управляет классом Bid, что отражено на диаграмме стрелкой «Ассоциация» с направлением от BidController к Bid.
* Класс CarController – класс, управляющий машинами. Позволяет создавать машину, находить машины по данным (название, цвет и т.д.) или id, изменять информацию о машине, удалять одну или все машины. Данный класс управляет классом Car, что отражено на диаграмме стрелкой «Ассоциация» с направлением от CarController к Car. *Через поля (in name:String, in color:String, in duration:int, in descr:String)*
* Класс EmployeeController – класс, управляющий сотрудниками. Позволяет создавать сотрудника, находить сотрудников по данным (имени, фамилии и т.д.), id или учётным данным, изменять информацию о сотруднике, удалять одного или всех сотрудников, изменять пароль сотрудника. Данный класс управляет классом Employee, что отражено на диаграмме стрелкой «Ассоциация» с направлением от EmployeeController к Employee. *Связь через поля (in login:String, in pass:String)*
* Класс CompanyAccountController – класс, управляющий счётом системы проката. Он содержит поля account (собственно счёт системы), pledge\_sum (сумма залога, равная для всех клиентов и машин), rate\_per\_date (стоимость одного дня проката машины). Также класс обладает методами addSum – добавление некоторой суммы на счёт проката, givePledge – внести залог, getPledge – получить залог обратно (при сдаче машины после проката), getPledgeSum и setPledgeSum – получит или изменить сумму залога.

Пакет **Views** предназначен для отображения данных пользователю и обработку его действий. Он представлен классами форм, которые предназначены для взаимодействия с пользователем и представляют собой окна программы. В данный пакет входят классы:

* Класс LoginForm – форма для проверки введённых пользователем (которым является сотрудник) учётных данных. Если логин и пароль совпадают с учётными данными какого-либо сотрудника, осуществляется переход с данной формы на MainForm. Связь между LoginForm и MainForm показана на диаграмме стрелкой «Ассоциация» с направлением от LoginForm к MainForm. *Через поля (in login:String, in pass:String)*
* Класс MainForm – основная форма приложения, с помощью которой можно просмотреть список всех машин или информацию о какой-либо одной машине (при этом перейдя на форму CarForm), найти конкретную машину или конкретного клиента (перейдя на форму ClientForm), выйти из системы, перейти на формы BidForm и EmployeeForm. Связь со всеми формами, классами ClientController и CarController показана на диаграмме направленными от MainForm стрелками «Ассоциация». *Через поля (in name:String, in surname:String, in fname:String, in bday:Date, in passData:String)*
* Класс CarForm – форма для работы с конкретной машиной. С этой формы можно просмотреть информацию о машине, изменить её данные, удалить её, взять машину напрокат или сдать её обратно в прокат. Данная форма связана с классом CarController, что отражено на диаграмме стрелкой «Ассоциация» с направлением от CarForm к CarController. *Через поля (in name:String, in color:String, in duration:int, in descr:String)*
* Класс EmployeeForm – форма для работы с сотрудником. С этой формы можно просмотреть информацию о сотруднике, изменить его пароль, удалить сотрудника. Данная форма связана с классом EmployeeController, что отражено на диаграмме стрелкой «Ассоциация» с направлением от EmployeeForm к EmployeeController. *Через поля (in name:string, in surname:String, in f\_name:String, in bday:Date, in pass\_data:String, in pos:String, in login:String)*
* Класс BidForm – форма для работы с заявкой. С этой формы можно просмотреть информацию о конкретной заявке и всех заявках вместе, добавить новую заявку, удалить заявку, составить отчёт по заявкам. Данная форма связана с классом BidController, что отражено на диаграмме стрелкой «Ассоциация» с направлением от BidForm к BidController. *Через поле Bid*
* Класс ClientForm – форма для работы с конкретным клиентом. С этой формы можно просмотреть информацию о клиенте и удалить его. Данная форма связана с классом ClientController, что отражено на диаграмме стрелкой «Ассоциация» с направлением от ClientForm к ClientController. *Через поле Client.*

**Описание диаграммы состояний для машины**

****

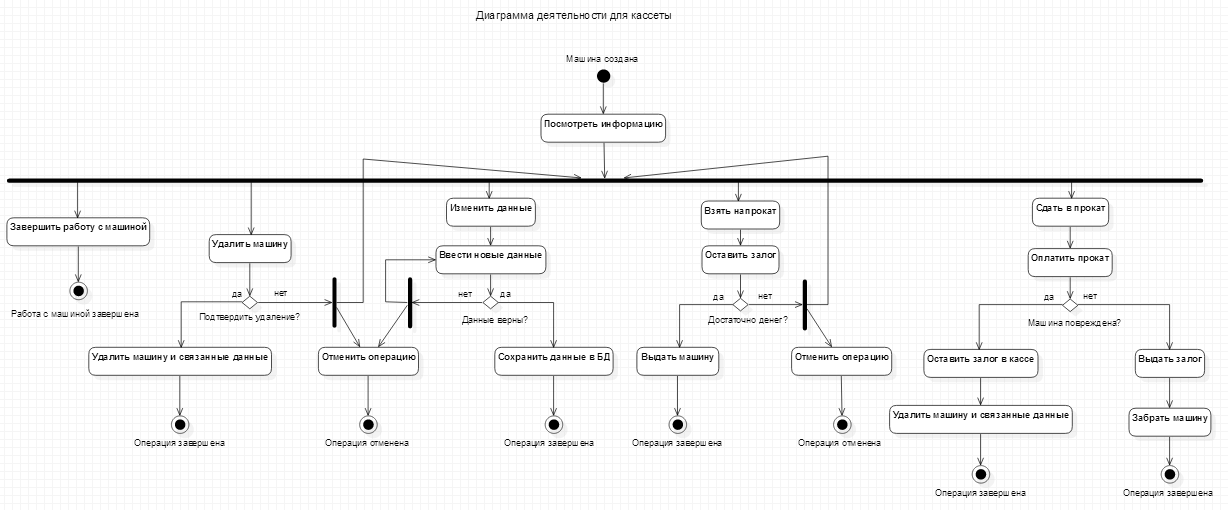
После создания машина находится в состоянии доступности для взятия в прокат. Пока машина находится в данном состоянии, специальное поле класса Car available равно true, т.е. машина доступна.

Когда клиент берёт машину, происходит переход из состояния «Машина доступна» в состояние «Машина взята напрокат». Когда объект переходит в данное состояние, выполняется внесение залога за машину. Пока машина находится в данном состоянии, машина недоступна, т.к. находится у клиента. Находясь у клиента, машина периодически находится в состоянии эксплуатации, когда ею пользуются. При переходе в данное состояние выполняется включение машины, в течение всего состояния осуществляется эксплуатация машины, а после, т.е. при выходе из данного состояния, происходит заглушение машины.

Когда клиент возвращает машину, она переходит в состояние «Сдача машины», при этом клиент должен получить залог и оплатить прокат. Если машины не повреждена, клиент сдаёт её в прокат и машина возвращается в исходное состояние «Доступна для проката». Если же машина повреждена, клиент оплачивает сервис, а сама машина переходит в состояние удаления из системы, при этом выполняются действия по удалению. После удаления машина переходит в конечное состояние.

Если необходимо изменить данные о машине, она переходит из состояния доступности в состояние «Изменение данных о машине», при этом происходит обновление данных. После изменения данных происходит сохранение новых данных в БД, при этом выполняется внесение изменений. Если сохранение происходит успешно, машина возвращается в состояние доступности. Если при сохранении возникает ошибка, данные не сохраняются и машина переходит в конечное состояние.

**Диаграмма деятельности для машины**

****

После создания (что является начальным состоянием машины) пользователь просматривает информацию о ней, и после этого может совершить различные действия: завершить работу с машиной, удалить её, изменить данные, взять её напрокат или сдать обратно.

После просмотра информации пользователь может завершить работу с машиной. После совершения данного действия машина переходит в конечное состояние, работа с машиной завершается.

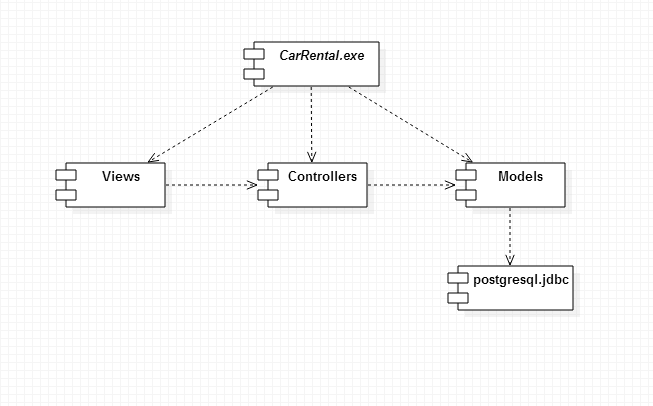
Также пользователь может удалить машину из системы. Система запрашивает у пользователя подтверждение удаления, и если пользователь действительно хочет удалить машину, из системы и базы данных удаляются все данные о машине, включая связанные с ней данные. На этом операция завершается. Если же пользователь случайно нажал на кнопку удаления, он не подтверждает удаление и в этом случае происходит отмена операции и окончание работы или же возврат на стадию выбора операции над машиной.

Пользователь может изменить данные машины. Для этого ему необходимо ввести новые данные и подтвердить внесение изменений. Если пользователь подтверждает изменения, они сохраняются в базу данных и операция на этом завершается. Иначе происходит отмена операции и возврат на стадию выбора операции над машиной.

Пользователю доступно взятие машины напрокат. Для этого нужно внести залог за машину, и если средств пользователя достаточно для данной операции, ему выдается требуемая машина, после чего операция завершается. Если же денег недостаточно, операция отменяется и происходит возврат на стадию выбора операции над машиной.

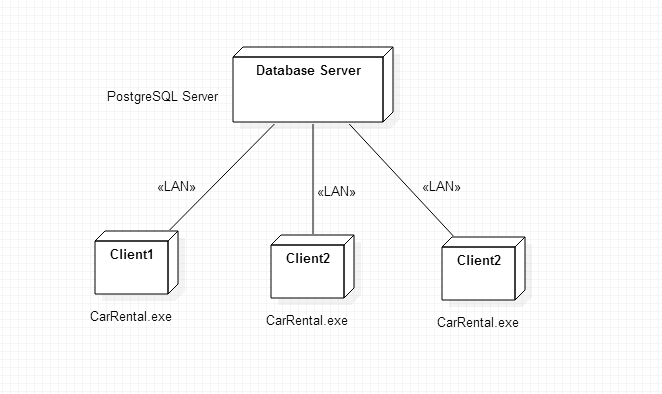
Пользователь, попользовавшись машиной, сдаёт её обратно и оплачивает прокат. Если машина не повреждена, она принимается обратно в пункт проката, а пользователю выдаётся внесённый им ранее залог, и на этом операция завершается. Если же машина повреждена, залог не выдаётся, машина и все связанные с ней данные удаляются из системы и БД, и на этом операция завершается.

**Диаграмма компонентов.**

****

В данном проекте предполагается, что система написана на языке Java, соответствует паттерну MVC и работает с базой данных PostgreSQL. В связи с этими требованиями на диаграмме компонентов отображены составные части проекта: пакеты Models, Controllers и Views, содержащие Java-классы, описанные на диаграмме классов, а также драйвер postgresql.jdbc, позволяющий всему приложению в целом, а конкретнее Java-классам из пакета Models взаимодействовать с СУБД PostgreSQL. Таким образом, для компиляции классов пакета Models необходимо наличие postgresql.jdbc (данная связь отображена стрелкой от Models к postgresql.jdbc). Для компиляции классов из Controllers нужны классы из Models, для компиляции Views нужны готовые классы из Controllers, а для получения исполняемой программы CarRental.exe требуется компиляция всех вышеуказанных компонентов системы. Все связи между компонентами указаны стрелками на диаграмме.

**Диаграмма развертывания**

****

Так как разрабатываемая система предназначена для использования сотрудниками компании, она имеет небольшое количество компонентов. Предполагается наличие общего сервера с развёрнутой на нём базой данных, хранящей данные всех клиентов, когда-либо обращавшихся в автопрокат, и данные обо всех машинах в системе (предполагается размещение автопроката в пределах одного города и возможность доставки требуемой клиентом машины из одного пункта автопроката в другой без больших затрат времени). Второй частью системы являются клиентские станции, на которых и установлена разрабатываемая система, представленная на диаграмме компонентов в виде CarRental.exe. Клиентские компьютеры посредством локальной сети обращаются к серверу базы данных для получения необходимых данных.