Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

ОТЧЕТ
по лабораторной работе №5

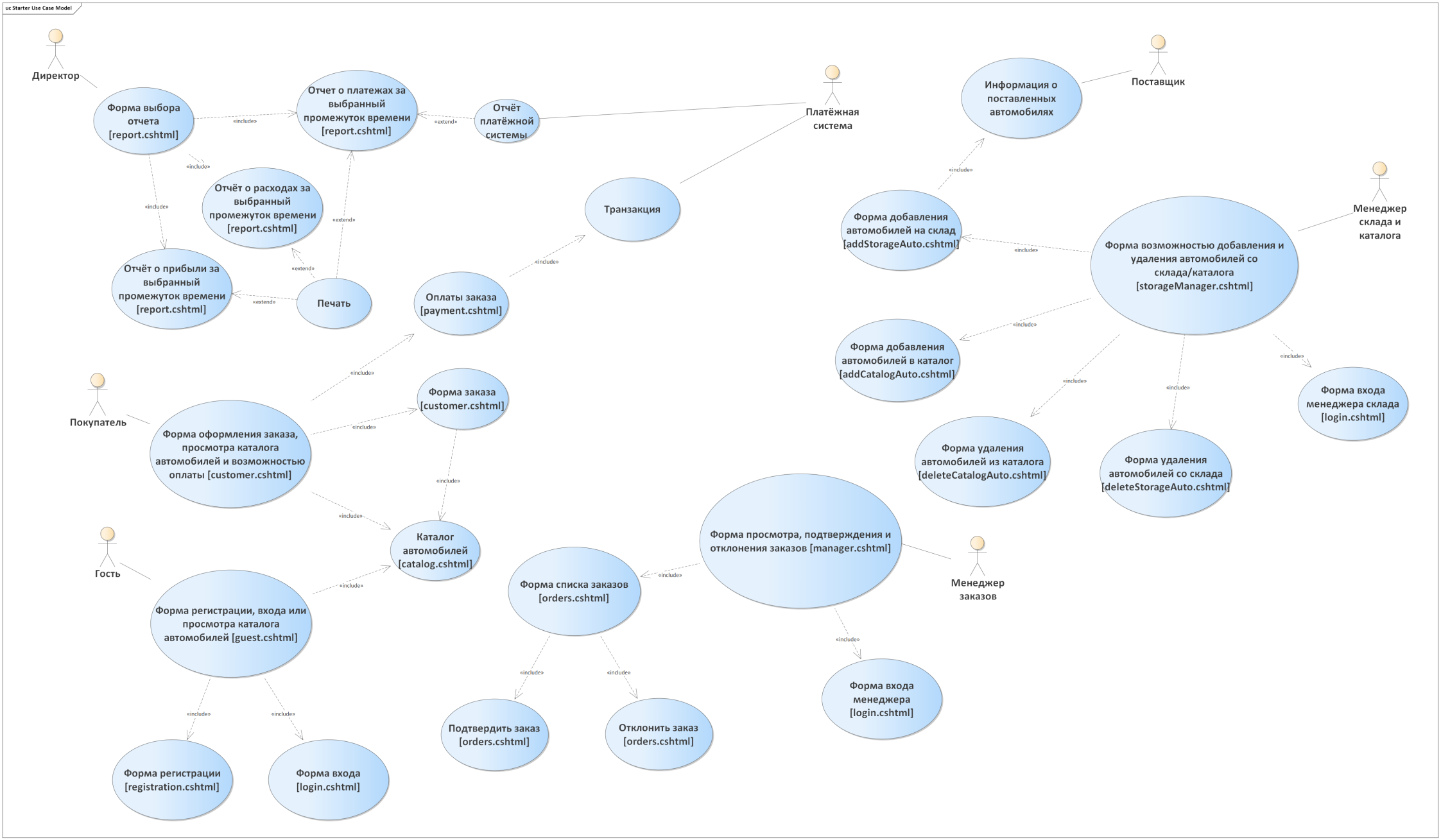
по дисциплине:

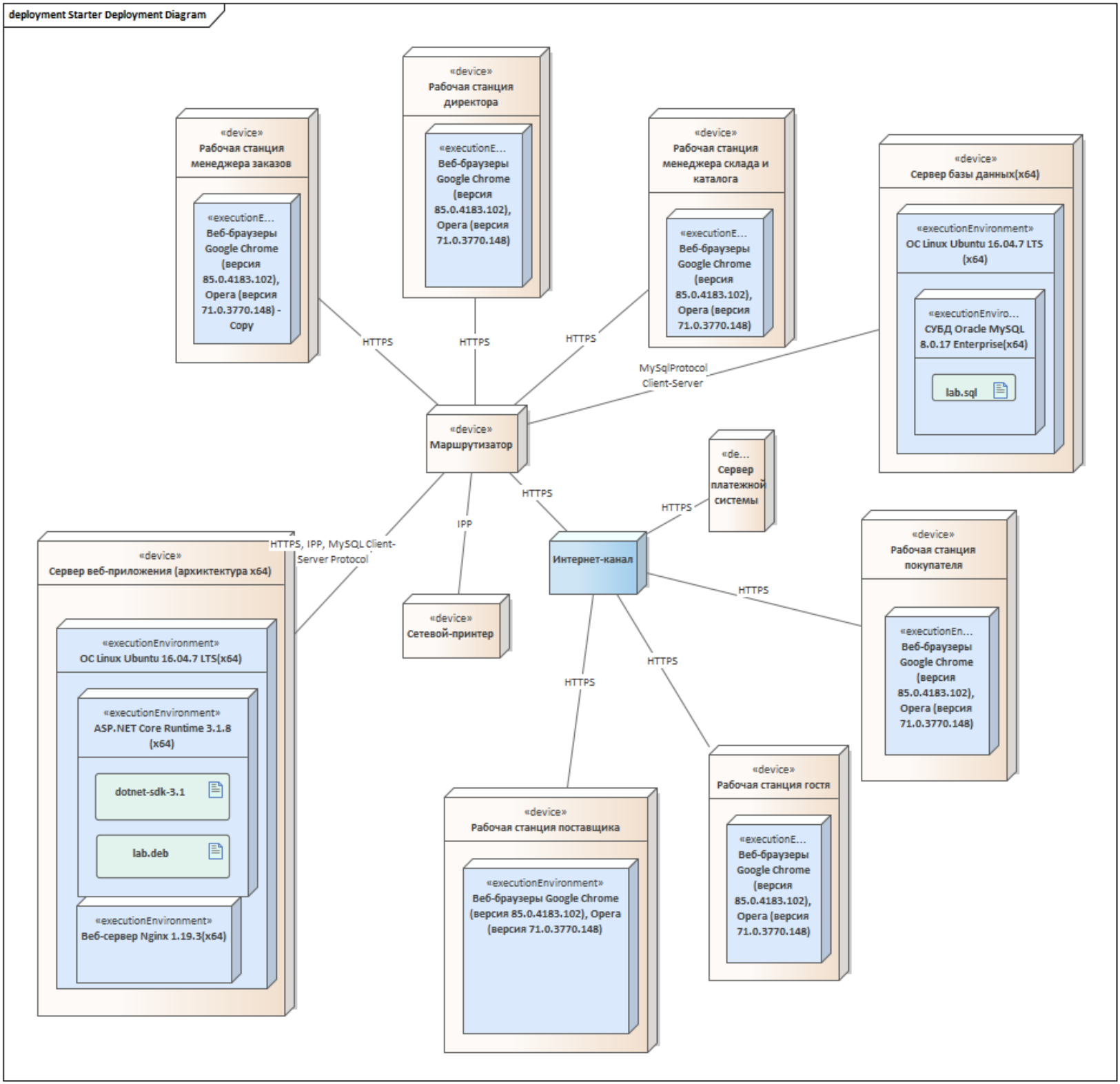
Объектно-ориентированные технологии программирования и стандарты проектирования

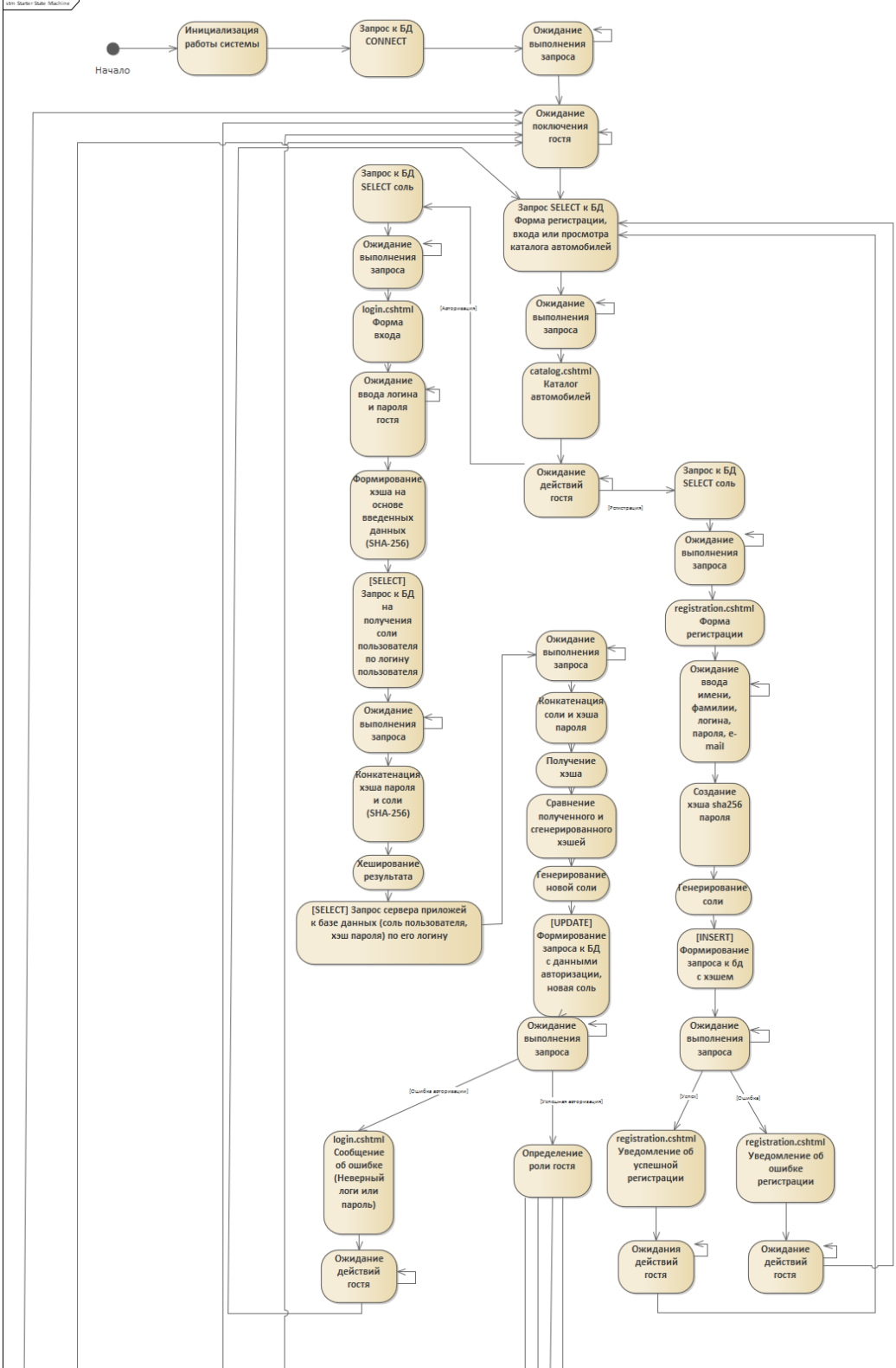
Выполнил:
Студент гр. 851005
Ермолович И. В.

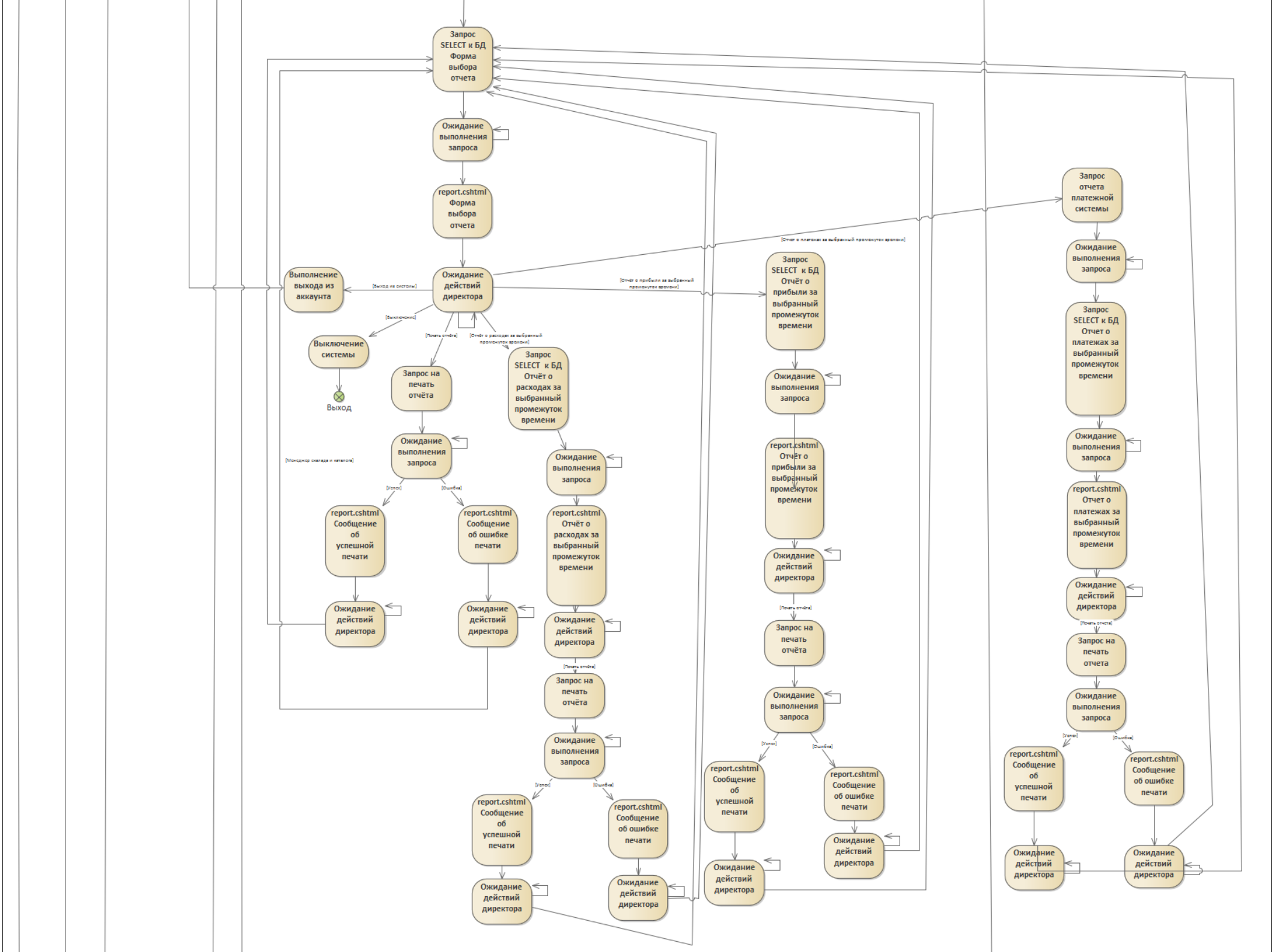
Проверил:
Алексеев И.Г.

Минск 2020

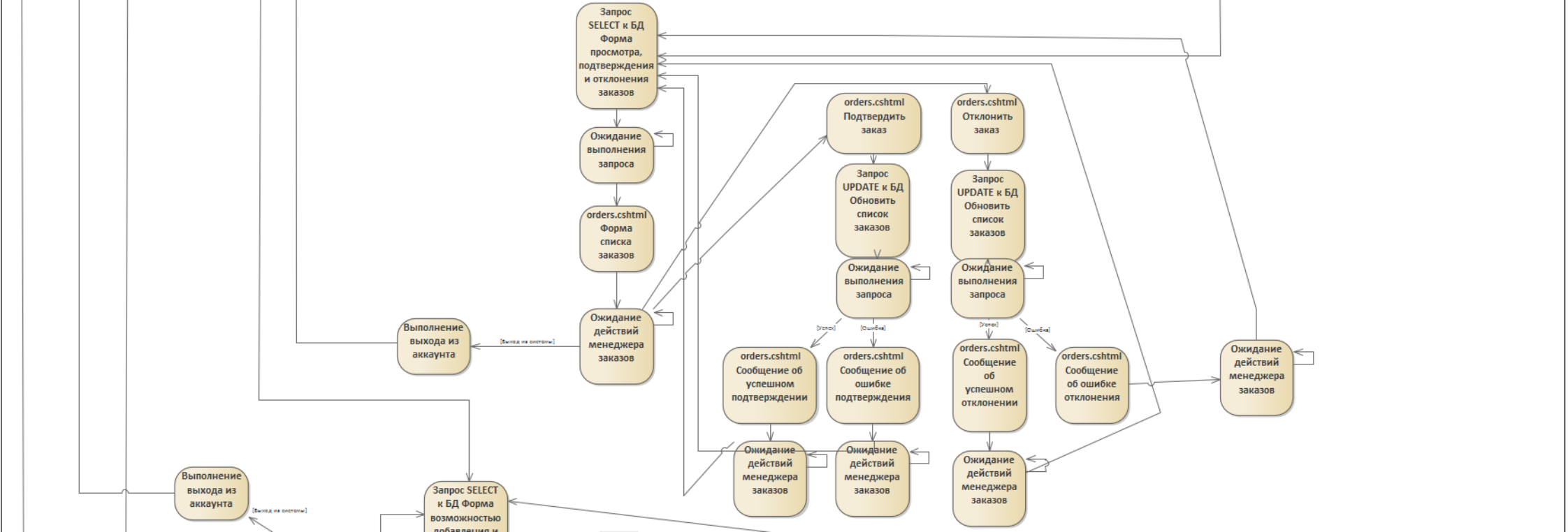

uc Starter Use Case Model
Директор
Форма выбора отчета [report.cshtml]
Отчет о платежах за выбранный промежуток времени [report.cshtml]
Отчёт платёжной системы
Платёжная система
Отчёт о расходах за выбранный промежуток времени [report.cshtml]
Транзакция
Отчёт о прибыли за выбранный промежуток времени [report.cshtml]
Печать
Оплаты заказа [payment.cshtml]
Информация о поставленных автомобилях
Поставщик
Форма добавления автомобилей на склад [addStorageAuto.cshtml]
Форма возможностью добавления и удаления автомобилей со склада/каталога [storageManager.cshtml]
Менеджер склада и каталога
Форма добавления автомобилей в каталог [addCatalogAuto.cshtml]
Форма входа менеджера склада [login.cshtml]
Форма удаления автомобилей из каталога [deleteCatalogAuto.cshtml]
Форма удаления автомобилей со склада [deleteStorageAuto.cshtml]
Покупатель
Форма заказа [customer.cshtml]
Форма оформления заказа, просмотра каталога автомобилей и возможностью оплаты [customer.cshtml]
Гость
Каталог автомобилей [catalog.cshtml]
Форма регистрации, входа или просмотра каталога автомобилей [guest.cshtml]
Форма списка заказов [orders.cshtml]
Форма просмотра, подтверждения и отклонения заказов [manager.cshtml]
Менеджер заказов
Форма входа менеджера [login.cshtml]
Подтвердить заказ [orders.cshtml]
Отклонить заказ [orders.cshtml]
Форма регистрации [registration.cshtml]
Форма входа [login.cshtml]
«include»
«extend»

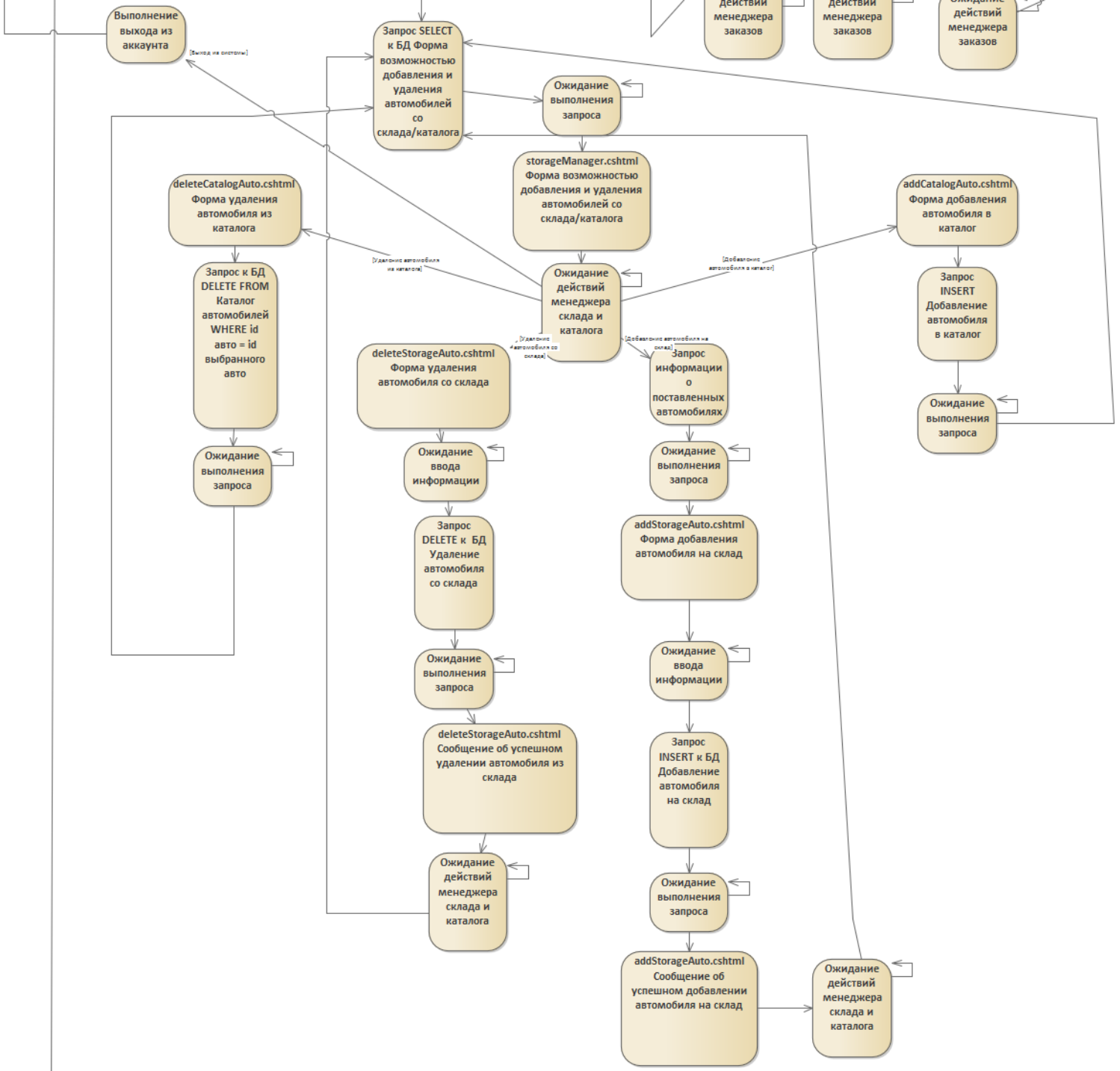

deployment Starter Deployment Diagram
«device»
Рабочая станция директора
«executionE...
Веб-браузеры Google Chrome (версия 85.0.4183.102), Opera (версия 71.0.3770.148)
«device»
Рабочая станция менеджера заказов
«executionE...
Веб-браузеры Google Chrome (версия 85.0.4183.102), Opera (версия 71.0.3770.148) - Copy
«device»
Рабочая станция менеджера склада и каталога
«executionE...
Веб-браузеры Google Chrome (версия 85.0.4183.102), Opera (версия 71.0.3770.148)
«device»
Сервер базы данных(x64)
«executionEnvironment»
ОС Linux Ubuntu 16.04.7 LTS (x64)
«executionEnviro...
СУБД Oracle MySQL 8.0.17 Enterprise(x64)
lab.sql
HTTPS
HTTPS
HTTPS
MySqlProtocol Client-Server
«device»
Маршрутизатор
«de...
Сервер платежной системы
HTTPS
HTTPS
IPP
HTTPS, IPP, MySQL Client-Server Protocol
«device»
Сервер веб-приложения (архиктектура x64)
«executionEnvironment»
ОС Linux Ubuntu 16.04.7 LTS(x64)
«executionEnvironment»
ASP.NET Core Runtime 3.1.8 (x64)
dotnet-sdk-3.1
lab.deb
«executionEnvironment»
Веб-сервер Nginx 1.19.3(x64)
«device»
Сетевой-принтер
Интернет-канал
HTTPS
«device»
Рабочая станция покупателя
«executionEn...
Веб-браузеры Google Chrome (версия 85.0.4183.102), Opera (версия 71.0.3770.148)
HTTPS
HTTPS
HTTPS
«device»
Рабочая станция поставщика
«executionEnvironment»
Веб-браузеры Google Chrome (версия 85.0.4183.102), Opera (версия 71.0.3770.148)
«device»
Рабочая станция гостя
«executionE...
Веб-браузеры Google Chrome (версия 85.0.4183.102), Opera (версия 71.0.3770.148)

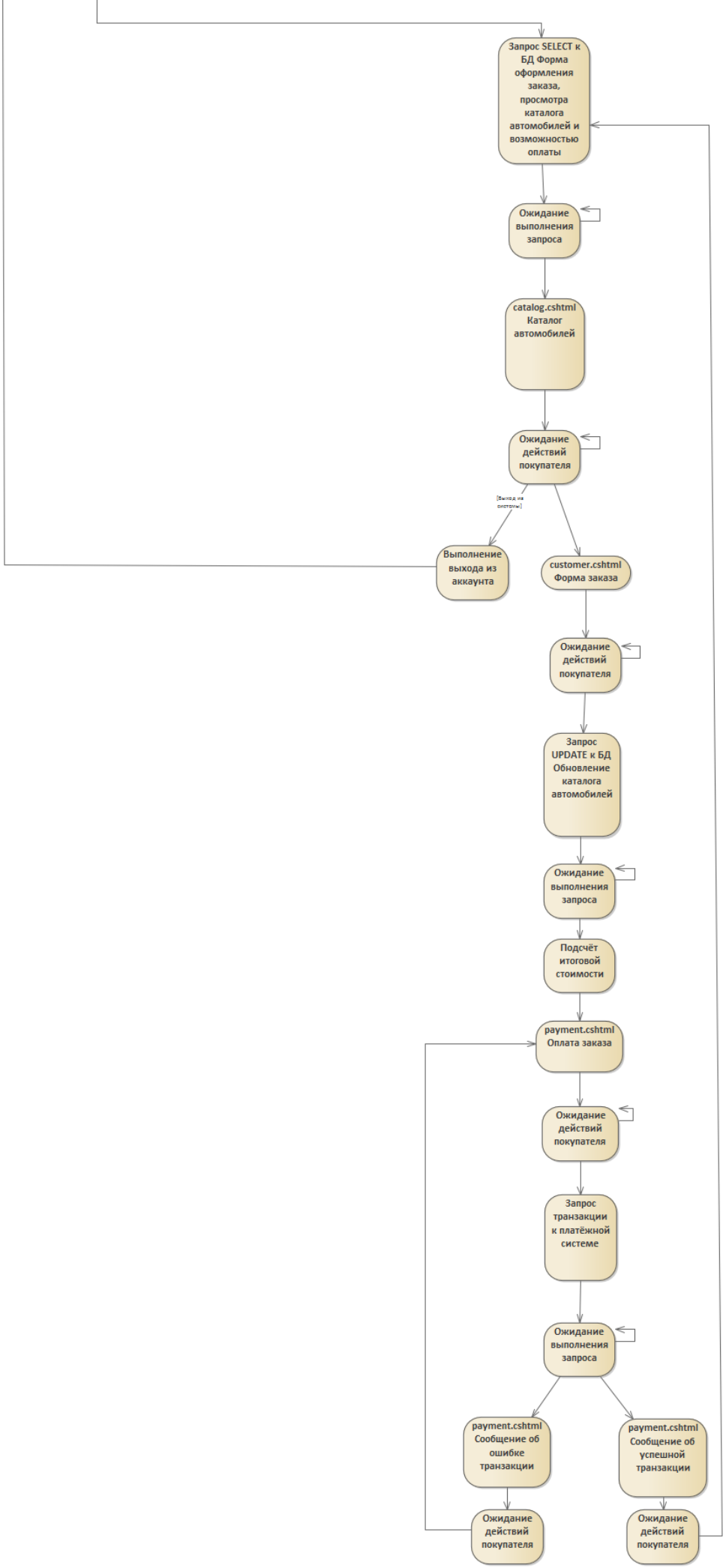

Начало
Инициализация работы системы
Запрос к БД CONNECT
Ожидание выполнения запроса
Ожидание поключения гостя
Запрос к БД SELECT соль
Ожидание выполнения запроса
login.cshtml Форма входа
Ожидание ввода логина и пароля гостя
Формирование хэша на основе введенных данных (SHA-256)
[SELECT] Запрос к БД на получения соли пользователя по логину пользователя
Ожидание выполнения запроса
Конкатенация хэша пароля и соли (SHA-256)
Хеширование результата
[SELECT] Запрос сервера приложей к базе данных (соль пользователя, хэш пароля) по его логину
Ожидание выполнения запроса
Конкатенация соли и хэша пароля
Получение хэша
Сравнение полученного и сгенерированного хэшей
Генерирование новой соли
[UPDATE] Формирование запроса к БД с данными авторизации, новая соль
Ожидание выполнения запроса
[Ошибка авторизации]
[Успешная авторизация]
login.cshtml Сообщение об ошибке (Неверный логи или пароль)
Ожидание действий гостя
Определение роли гостя
Запрос SELECT к БД Форма регистрации, входа или просмотра каталога автомобилей
Ожидание выполнения запроса
[Авторизация]
catalog.cshtml Каталог автомобилей
Ожидание действий гостя
[Регистрация]
Запрос к БД SELECT соль
Ожидание выполнения запроса
registration.cshtml Форма регистрации
Ожидание ввода имени, фамилии, логина, пароля, e-mail
Создание хэша sha256 пароля
Генерирование соли
[INSERT] Формирование запроса к бд с хэшем
Ожидание выполнения запроса
[Успех]
[Ошибка]
registration.cshtml Уведомление об успешной регистрации
Ожидания действий гостя
registration.cshtml Уведомление об ошибке регистрации
Ожидание действий гостя

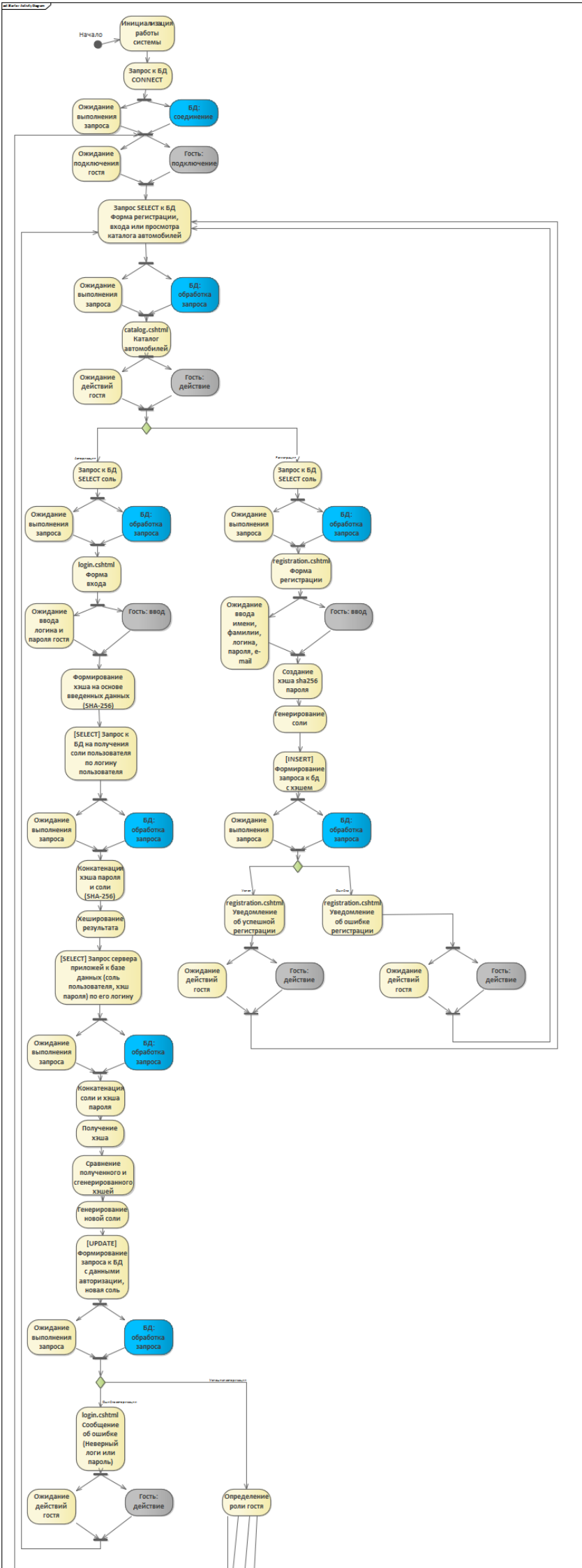


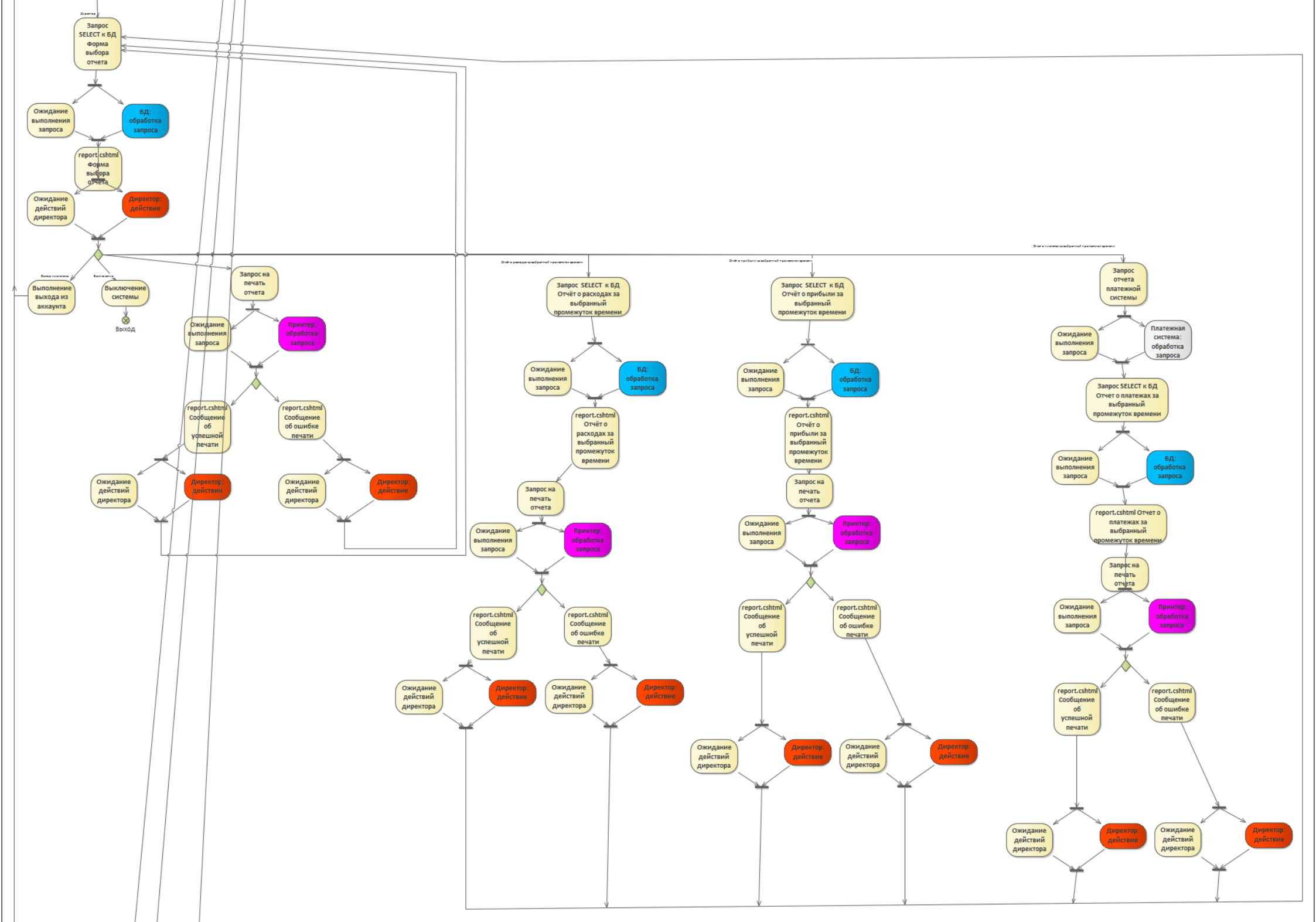
Запрос SELECT к БД Форма выбора отчета
Ожидание выполнения запроса
report.cshtml Форма выбора отчета
Ожидание действий директора
Выполнение выхода из аккаунта
[Выход из системы]
[Выключение]
Выключение системы
Выход
[Почать отчёта]
[Отчёт о расходах за выбранный промежуток времени]
Запрос на печать отчёта
Ожидание выполнения запроса
[Менеджер скалада и каталога]
[Успех]
[Ошибка]
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Запрос SELECT к БД Отчёт о расходах за выбранный промежуток времени
Ожидание выполнения запроса
report.cshtml Отчёт о расходах за выбранный промежуток времени
Ожидание действий директора
[Почать отчёта]
Запрос на печать отчёта
Ожидание выполнения запроса
[Успех]
[Ошибка]
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Ожидание действий директора
[Отчёт о прибыли за выбранный промежуток времени]
Запрос SELECT к БД Отчёт о прибыли за выбранный промежуток времени
Ожидание выполнения запроса
report.cshtml Отчёт о прибыли за выбранный промежуток времени
Ожидание действий директора
[Почать отчёта]
Запрос на печать отчёта
Ожидание выполнения запроса
[Успех]
[Ошибка]
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Ожидание действий директора
[Отчёт о платежах за выбранный промежуток времени]
Запрос отчета платежной системы
Ожидание выполнения запроса
Запрос SELECT к БД Отчет о платежах за выбранный промежуток времени
Ожидание выполнения запроса
report.cshtml Отчет о платежах за выбранный промежуток времени
Ожидание действий директора
[Почать отчета]
Запрос на печать отчета
Ожидание выполнения запроса
[Успех]
[Ошибка]
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Ожидание действий директора

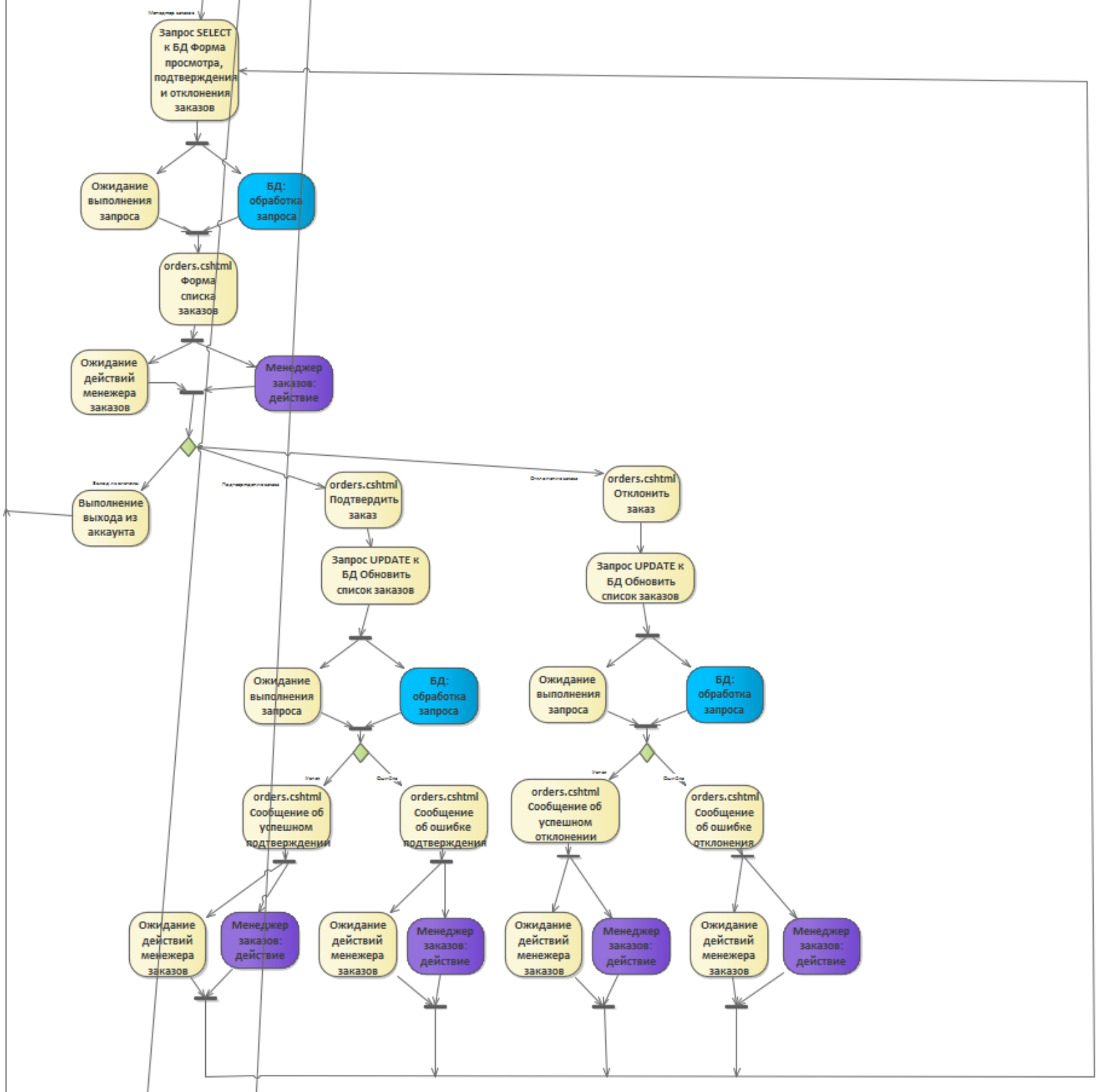

Запрос SELECT к БД Форма просмотра, подтверждения и отклонения заказов
Ожидание выполнения запроса
orders.cshtml Форма списка заказов
Ожидание действий менеджера заказов
Выполнение выхода из аккаунта
[Выход из системы]
orders.cshtml Подтвердить заказ
Запрос UPDATE к БД Обновить список заказов
Ожидание выполнения запроса
[Успех]
[Ошибка]
orders.cshtml Сообщение об успешном подтверждении
orders.cshtml Сообщение об ошибке подтверждения
Ожидание действий менеджера заказов
Ожидание действий менеджера заказов
orders.cshtml Отклонить заказ
Запрос UPDATE к БД Обновить список заказов
Ожидание выполнения запроса
[Успех]
[Ошибка]
orders.cshtml Сообщение об успешном отклонении
orders.cshtml Сообщение об ошибке отклонения
Ожидание действий менеджера заказов
Ожидание действий менеджера заказов
Выполнение выхода из аккаунта
[Выход из системы]
Запрос SELECT к БД Форма возможностью

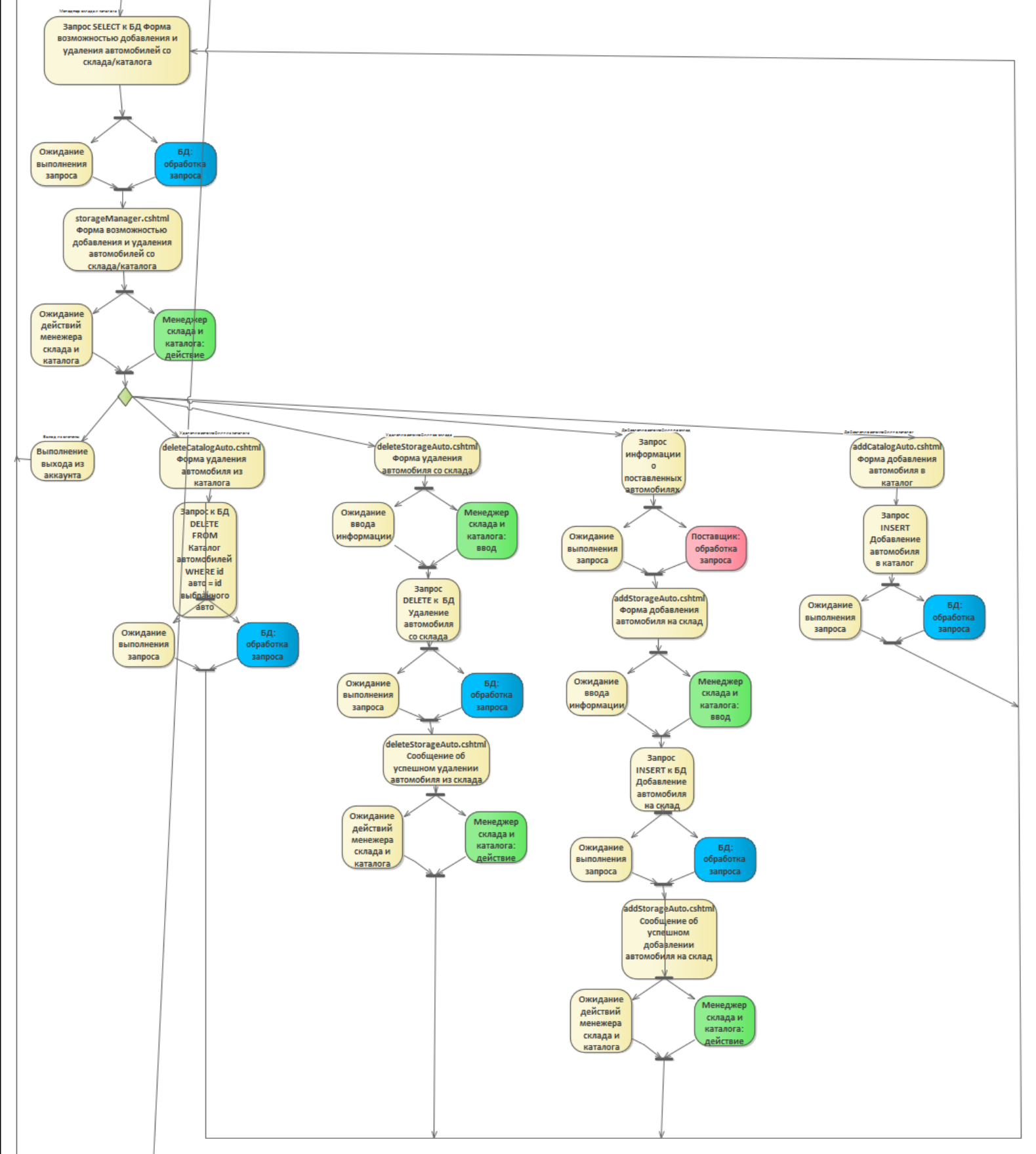

Выполнение выхода из аккаунта
[Выход из системы]
Запрос SELECT к БД Форма возможностью добавления и удаления автомобилей со склада/каталога
Ожидание выполнения запроса
storageManager.cshtml Форма возможностью добавления и удаления автомобилей со склада/каталога
deleteCatalogAuto.cshtml Форма удаления автомобиля из каталога
addCatalogAuto.cshtml Форма добавления автомобиля в каталог
[Удаление автомобиля из каталога]
[Добавление автомобиля в каталог]
Ожидание действий менеджера склада и каталога
Запрос к БД DELETE FROM Каталог автомобилей WHERE id авто = id выбранного авто
Запрос INSERT Добавление автомобиля в каталог
Ожидание выполнения запроса
Ожидание выполнения запроса
deleteStorageAuto.cshtml Форма удаления автомобиля со склада
[Удаление автомобиля со склада]
[Добавление автомобиля на склад]
Запрос информации о поставленных автомобилях
Ожидание ввода информации
Ожидание выполнения запроса
addStorageAuto.cshtml Форма добавления автомобиля на склад
Запрос DELETE к БД Удаление автомобиля со склада
Ожидание выполнения запроса
Ожидание ввода информации
deleteStorageAuto.cshtml Сообщение об успешном удалении автомобиля из склада
Запрос INSERT к БД Добавление автомобиля на склад
Ожидание действий менеджера склада и каталога
Ожидание выполнения запроса
addStorageAuto.cshtml Сообщение об успешном добавлении автомобиля на склад
Ожидание действий менеджера склада и каталога

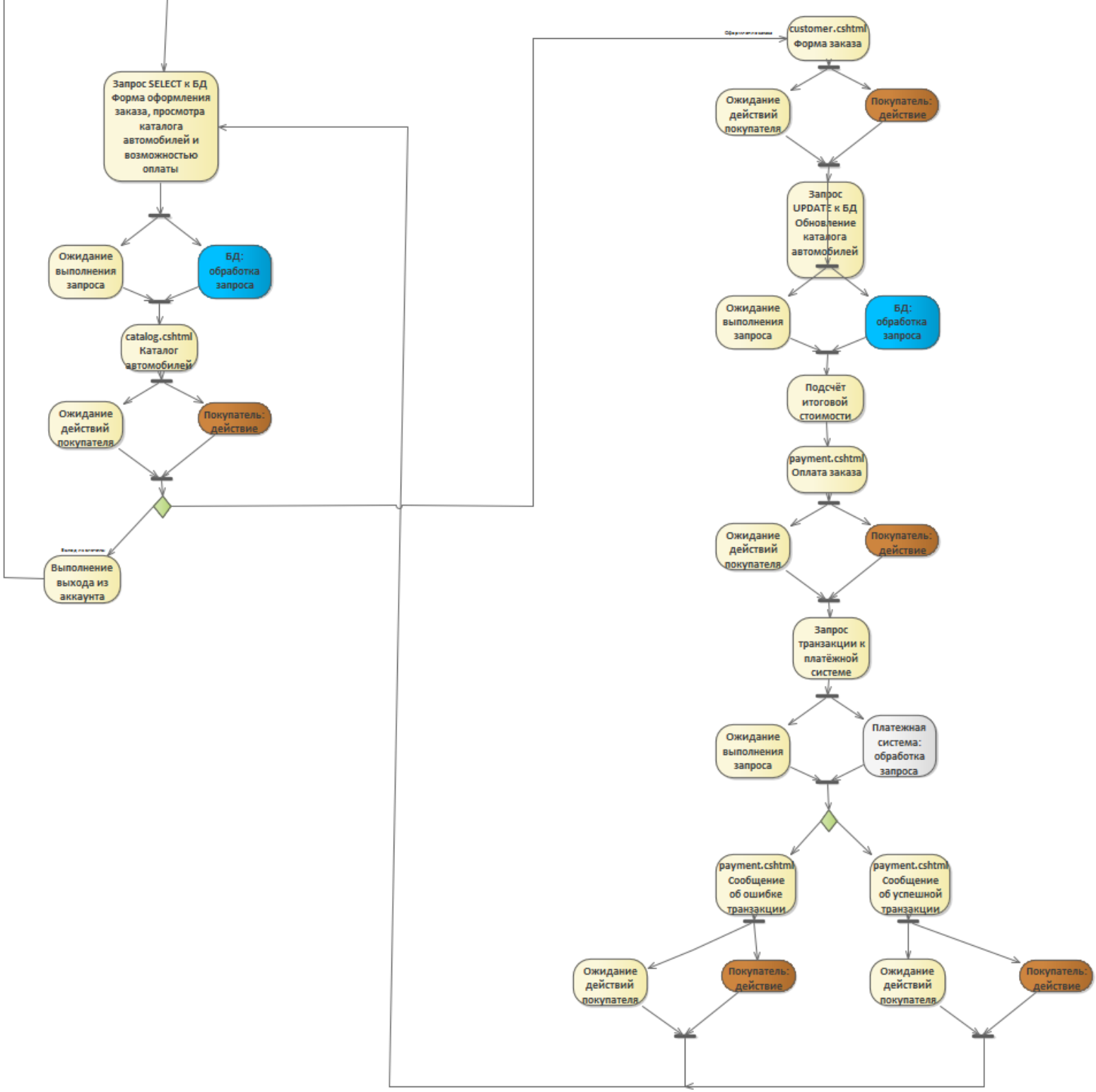

Запрос SELECT к БД Форма оформления заказа, просмотра каталога автомобилей и возможностью оплаты
Ожидание выполнения запроса
catalog.cshtml Каталог автомобилей
Ожидание действий покупателя
[Выход из системы]
Выполнение выхода из аккаунта
customer.cshtml Форма заказа
Ожидание действий покупателя
Запрос UPDATE к БД Обновление каталога автомобилей
Ожидание выполнения запроса
Подсчёт итоговой стоимости
payment.cshtml Оплата заказа
Ожидание действий покупателя
Запрос транзакции к платёжной системе
Ожидание выполнения запроса
payment.cshtml Сообщение об ошибке транзакции
payment.cshtml Сообщение об успешной транзакции
Ожидание действий покупателя
Ожидание действий покупателя


Начало
Инициализация работы системы
Запрос к БД CONNECT
Ожидание выполнения запроса
БД: соединение
Ожидание подключения гостя
Гость: подключение
Запрос SELECT к БД Форма регистрации, входа или просмотра каталога автомобилей
Ожидание выполнения запроса
БД: обработка запроса
catalog.cshtml Каталог автомобилей
Ожидание действий гостя
Гость: действие
Запрос к БД SELECT соль
Ожидание выполнения запроса
БД: обработка запроса
login.cshtml Форма входа
Ожидание ввода логина и пароля гостя
Гость: ввод
Формирование хэша на основе введенных данных (SHA-256)
[SELECT] Запрос к БД на получения соли пользователя по логину пользователя
Ожидание выполнения запроса
БД: обработка запроса
Конкатенация хэша пароля и соли (SHA-256)
Хеширование результата
[SELECT] Запрос сервера приложей к базе данных (соль пользователя, хэш пароля) по его логину
Ожидание выполнения запроса
БД: обработка запроса
Конкатенация соли и хэша пароля
Получение хэша
Сравнение полученного и сгенерированного хэшей
Генерирование новой соли
[UPDATE] Формирование запроса к БД с данными авторизации, новая соль
Ожидание выполнения запроса
БД: обработка запроса
login.cshtml Сообщение об ошибке (Неверный логи или пароль)
Ожидание действий гостя
Гость: действие
Определение роли гостя
Запрос к БД SELECT соль
Ожидание выполнения запроса
БД: обработка запроса
registration.cshtml Форма регистрации
Ожидание ввода имени, фамилии, логина, пароля, e-mail
Гость: ввод
Создание хэша sha256 пароля
Генерирование соли
[INSERT] Формирование запроса к бд с хэшем
Ожидание выполнения запроса
БД: обработка запроса
registration.cshtml Уведомление об успешной регистрации
registration.cshtml Уведомление об ошибке регистрации
Ожидание действий гостя
Гость: действие
Ожидание действий гостя
Гость: действие


Запрос SELECT к БД Форма выбора отчета
Ожидание выполнения запроса
БД: обработка запроса
report.cshtml Форма выбора отчета
Ожидание действий директора
Директор: действие
Выполнение выхода из аккаунта
Выключение системы
Выход
Запрос на печать отчета
Ожидание выполнения запроса
Принтер: обработка запроса
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Директор: действие
Ожидание действий директора
Директор: действие
Запрос SELECT к БД Отчёт о расходах за выбранный промежуток времени
Ожидание выполнения запроса
БД: обработка запроса
report.cshtml Отчёт о расходах за выбранный промежуток времени
Запрос на печать отчета
Ожидание выполнения запроса
Принтер: обработка запроса
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Директор: действие
Ожидание действий директора
Директор: действие
Запрос SELECT к БД Отчёт о прибыли за выбранный промежуток времени
Ожидание выполнения запроса
БД: обработка запроса
report.cshtml Отчёт о прибыли за выбранный промежуток времени
Запрос на печать отчета
Ожидание выполнения запроса
Принтер: обработка запроса
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Директор: действие
Ожидание действий директора
Директор: действие
Запрос отчета платежной системы
Ожидание выполнения запроса
Платежная система: обработка запроса
Запрос SELECT к БД Отчет о платежах за выбранный промежуток времени
Ожидание выполнения запроса
БД: обработка запроса
report.cshtml Отчет о платежах за выбранный промежуток времени
Запрос на печать отчета
Ожидание выполнения запроса
Принтер: обработка запроса
report.cshtml Сообщение об успешной печати
report.cshtml Сообщение об ошибке печати
Ожидание действий директора
Директор: действие
Ожидание действий директора
Директор: действие

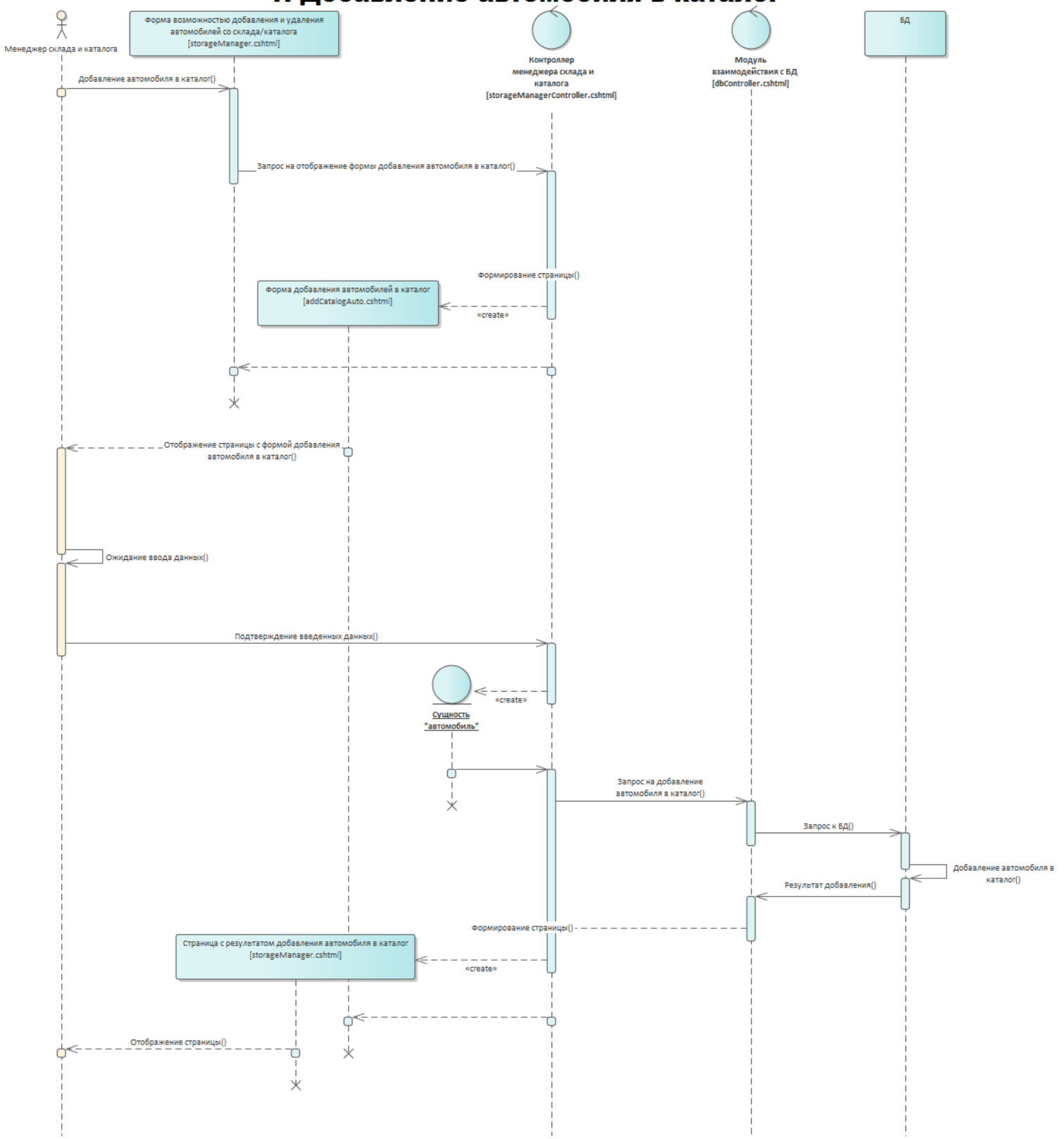

Запрос SELECT к БД Форма просмотра, подтверждения и отклонения заказов
Ожидание выполнения запроса
БД: обработка запроса
orders.cshtml Форма списка заказов
Ожидание действий менежера заказов
Менеджер заказов: действие
Выполнение выхода из аккаунта
orders.cshtml Подтвердить заказ
orders.cshtml Отклонить заказ
Запрос UPDATE к БД Обновить список заказов
Запрос UPDATE к БД Обновить список заказов
Ожидание выполнения запроса
БД: обработка запроса
Ожидание выполнения запроса
БД: обработка запроса
orders.cshtml Сообщение об успешном подтверждении
orders.cshtml Сообщение об ошибке подтверждения
orders.cshtml Сообщение об успешном отклонении
orders.cshtml Сообщение об ошибке отклонения
Ожидание действий менежера заказов
Менеджер заказов: действие
Ожидание действий менежера заказов
Менеджер заказов: действие
Ожидание действий менежера заказов
Менеджер заказов: действие
Ожидание действий менежера заказов
Менеджер заказов: действие

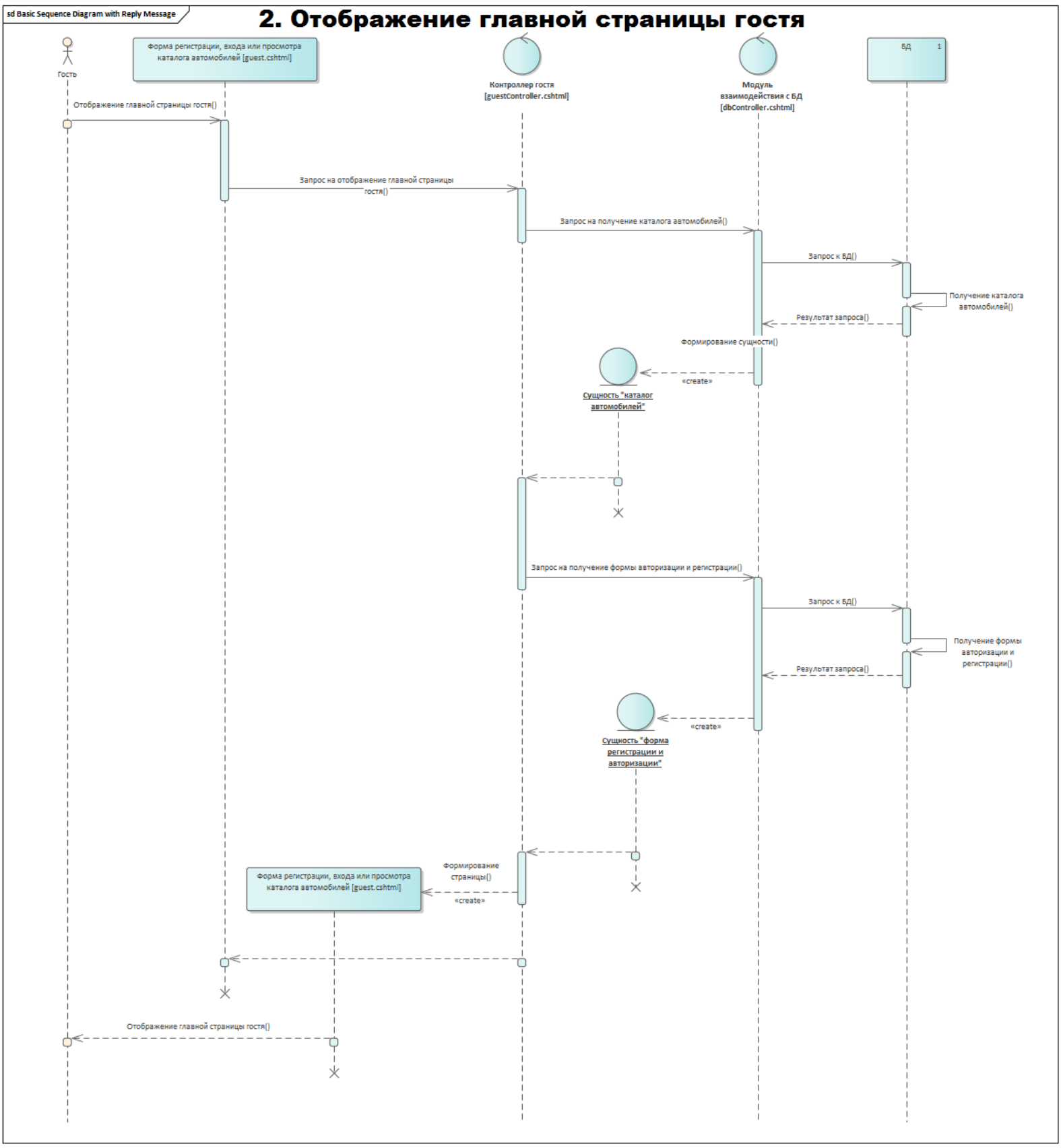

Запрос SELECT к БД Форма возможностью добавления и удаления автомобилей со склада/каталога
Ожидание выполнения запроса
БД: обработка запроса
storageManager.cshtml Форма возможностью добавления и удаления автомобилей со склада/каталога
Ожидание действий менежера склада и каталога
Менеджер склада и каталога: действие
Выполнение выхода из аккаунта
deleteCatalogAuto.cshtml Форма удаления автомобиля из каталога
Запрос к БД DELETE FROM Каталог автомобилей WHERE id авто = id выбранного авто
Ожидание выполнения запроса
БД: обработка запроса
deleteStorageAuto.cshtml Форма удаления автомобиля со склада
Ожидание ввода информации
Менеджер склада и каталога: ввод
Запрос DELETE к БД Удаление автомобиля со склада
Ожидание выполнения запроса
БД: обработка запроса
deleteStorageAuto.cshtml Сообщение об успешном удалении автомобиля из склада
Ожидание действий менежера склада и каталога
Менеджер склада и каталога: действие
Запрос информации о поставленных автомобилях
Ожидание выполнения запроса
Поставщик: обработка запроса
addStorageAuto.cshtml Форма добавления автомобиля на склад
Ожидание ввода информации
Менеджер склада и каталога: ввод
Запрос INSERT к БД Добавление автомобиля на склад
Ожидание выполнения запроса
БД: обработка запроса
addStorageAuto.cshtml Сообщение об успешном добавлении автомобиля на склад
Ожидание действий менежера склада и каталога
Менеджер склада и каталога: действие
addCatalogAuto.cshtml Форма добавления автомобиля в каталог
Запрос INSERT Добавление автомобиля в каталог
Ожидание выполнения запроса
БД: обработка запроса


Запрос SELECT к БД Форма оформления заказа, просмотра каталога автомобилей и возможностью оплаты
Ожидание выполнения запроса
БД: обработка запроса
catalog.cshtml Каталог автомобилей
Ожидание действий покупателя
Покупатель: действие
Выполнение выхода из аккаунта
customer.cshtml Форма заказа
Ожидание действий покупателя
Покупатель: действие
Запрос UPDATE к БД Обновление каталога автомобилей
Ожидание выполнения запроса
БД: обработка запроса
Подсчёт итоговой стоимости
payment.cshtml Оплата заказа
Ожидание действий покупателя
Покупатель: действие
Запрос транзакции к платёжной системе
Ожидание выполнения запроса
Платежная система: обработка запроса
payment.cshtml Сообщение об ошибке транзакции
payment.cshtml Сообщение об успешной транзакции
Ожидание действий покупателя
Покупатель: действие
Ожидание действий покупателя
Покупатель: действие

sd Basic Sequence Diagram with Reply Message

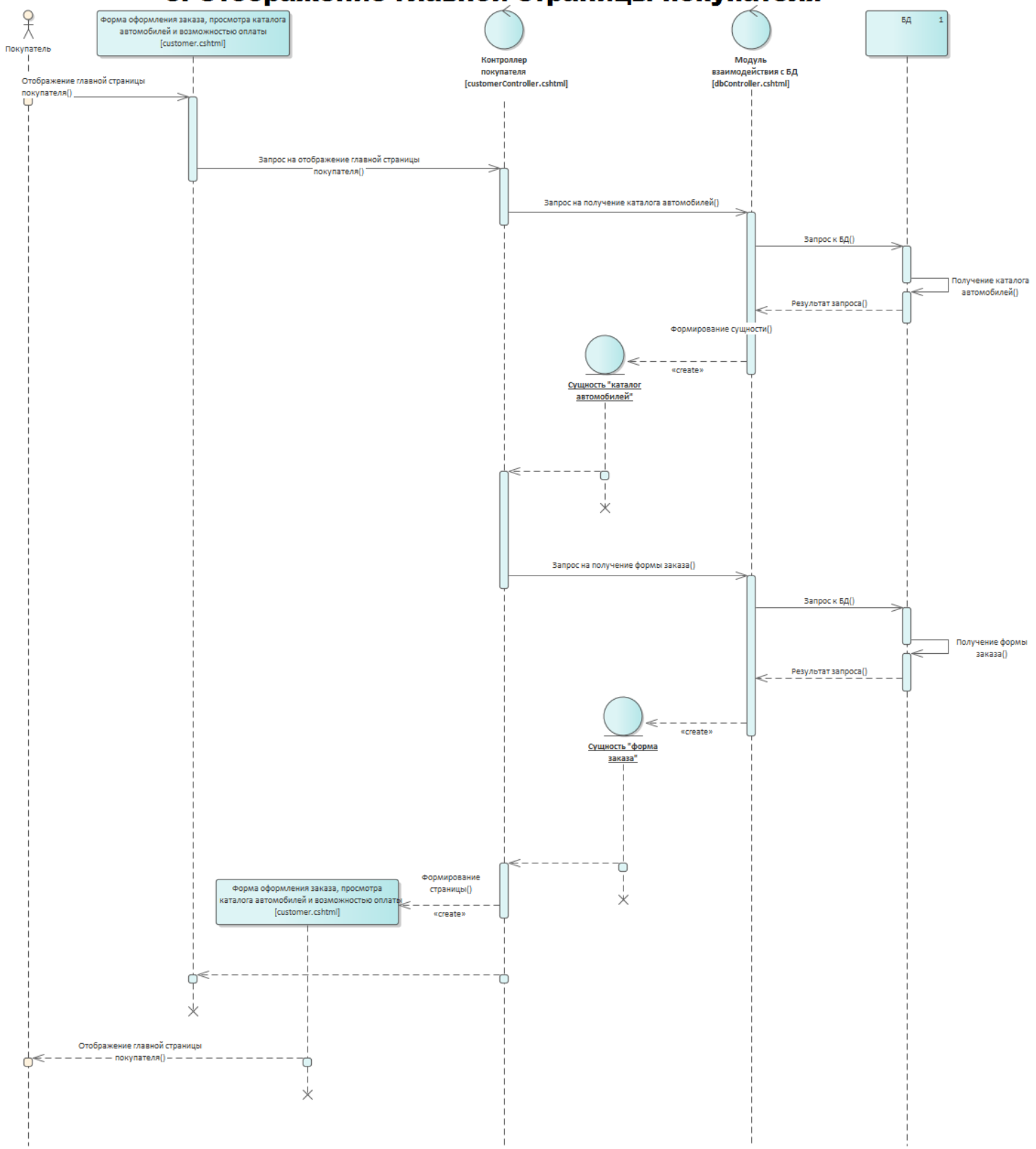
1. Добавление автомобиля в каталог




sd Basic Sequence Diagram with Reply Message
2. Отображение главной страницы гостя
Гость
Форма регистрации, входа или просмотра каталога автомобилей [guest.cshtml]
Контроллер гостя [guestController.cshtml]
Модуль взаимодействия с БД [dbController.cshtml]
БД
1
Отображение главной страницы гостя()
Запрос на отображение главной страницы гостя()
Запрос на получение каталога автомобилей()
Запрос к БД()
Получение каталога автомобилей()
Результат запроса()
Формирование сущности()
«create»
Сущность "каталог автомобилей"
Запрос на получение формы авторизации и регистрации()
Запрос к БД()
Получение формы авторизации и регистрации()
Результат запроса()
«create»
Сущность "форма регистрации и авторизации"
Формирование страницы()
Форма регистрации, входа или просмотра каталога автомобилей [guest.cshtml]
«create»
Отображение главной страницы гостя()

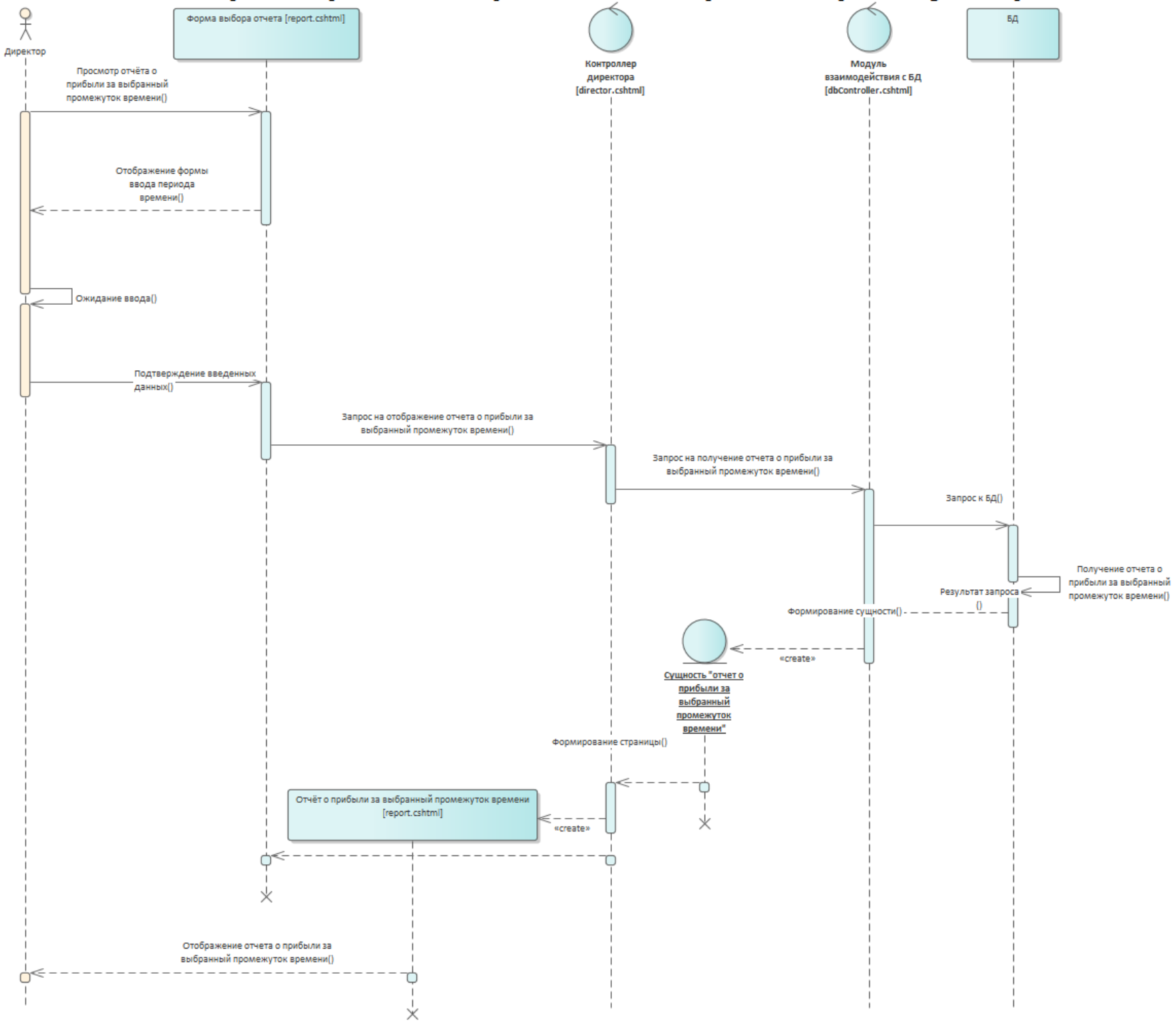
sd Basic Sequence Diagram with Reply Message

3. Отображение главной страницы покупателя



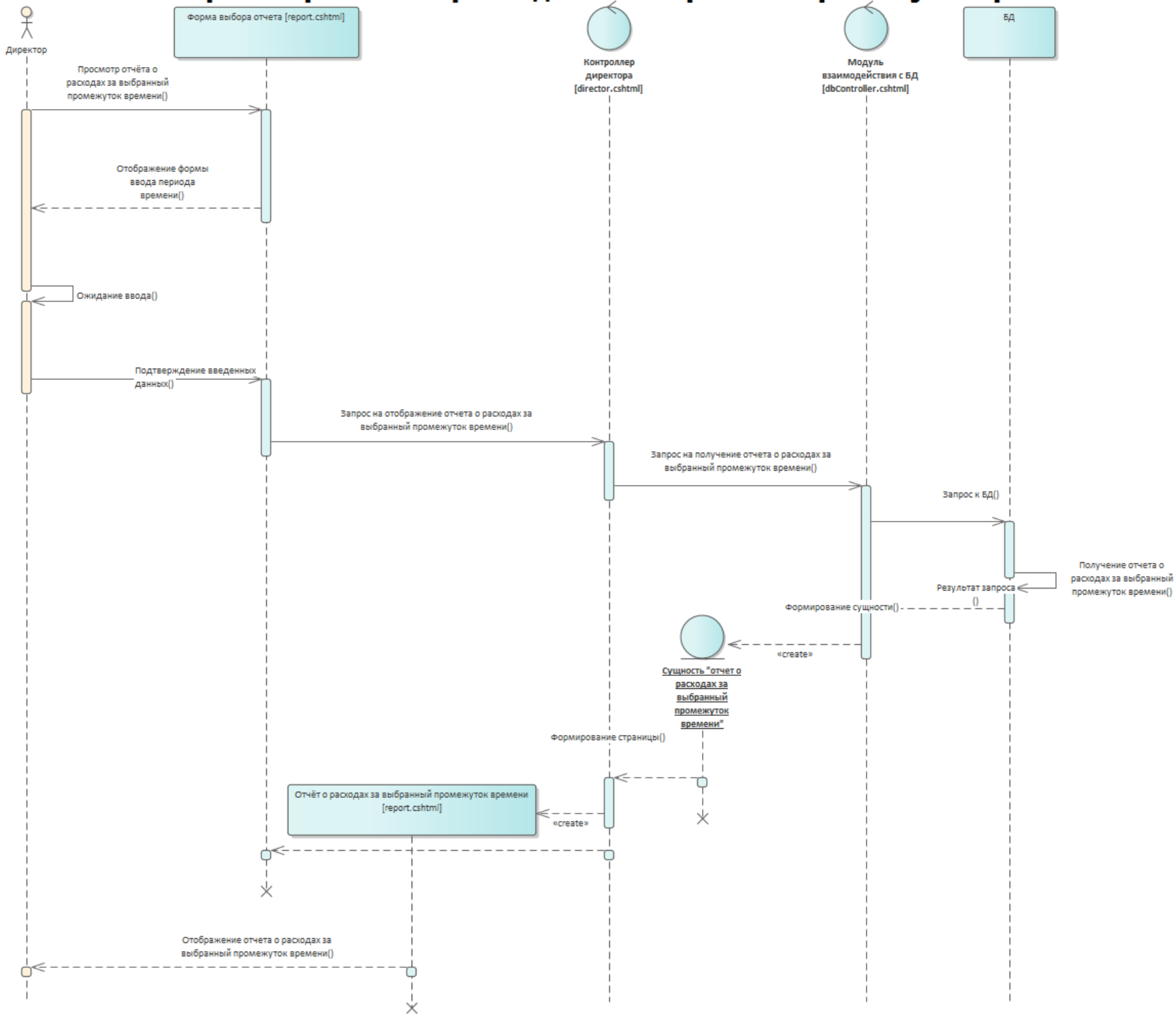
sd Basic Sequence Diagram with Reply Message

4. Просмотр отчета о прибыли за выбранный промежуток времени



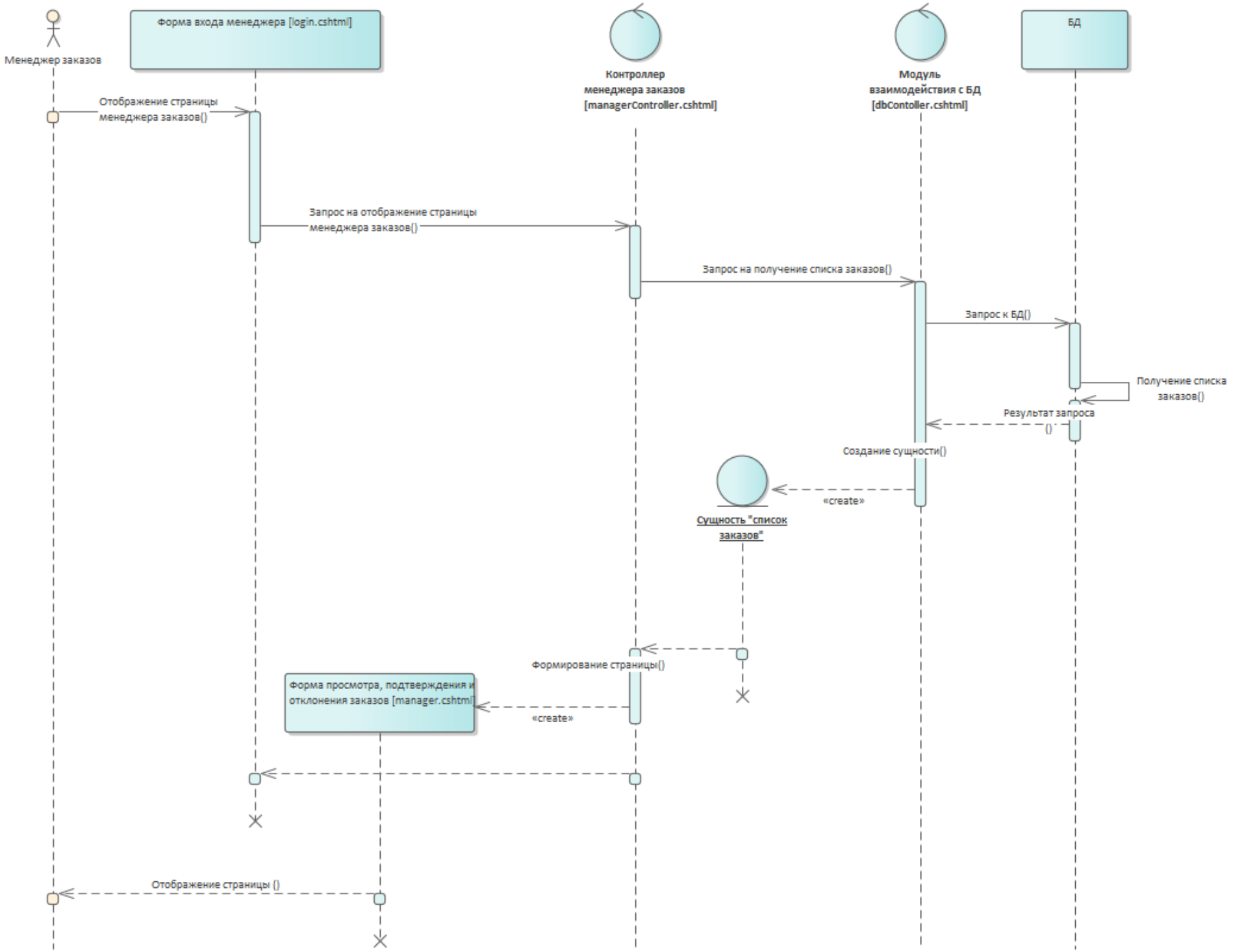
sd Basic Sequence Diagram with Reply Message

5. Просмотр отчета о расходах за выбранный промежуток времени



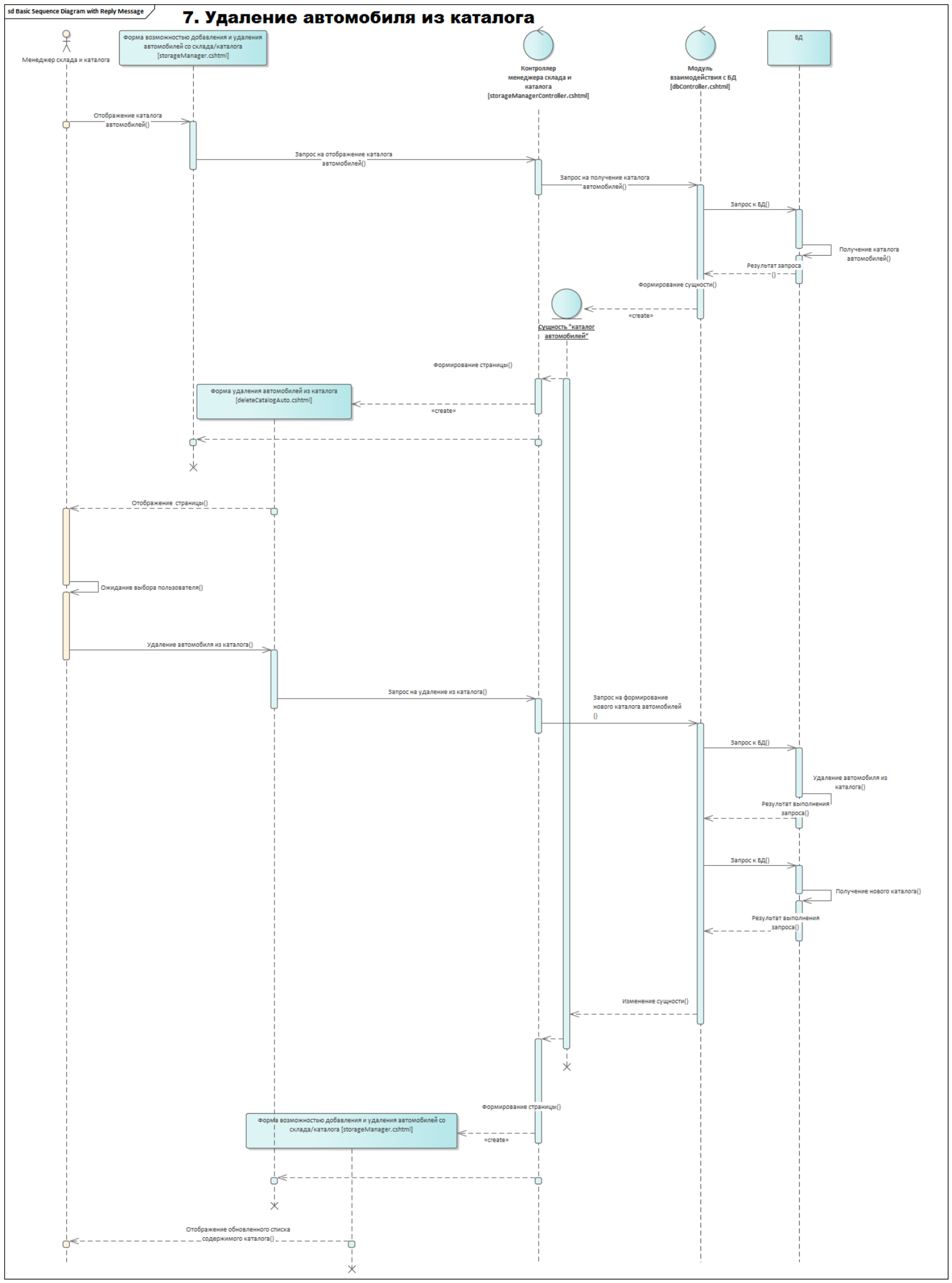
sd Basic Sequence Diagram with Reply Message

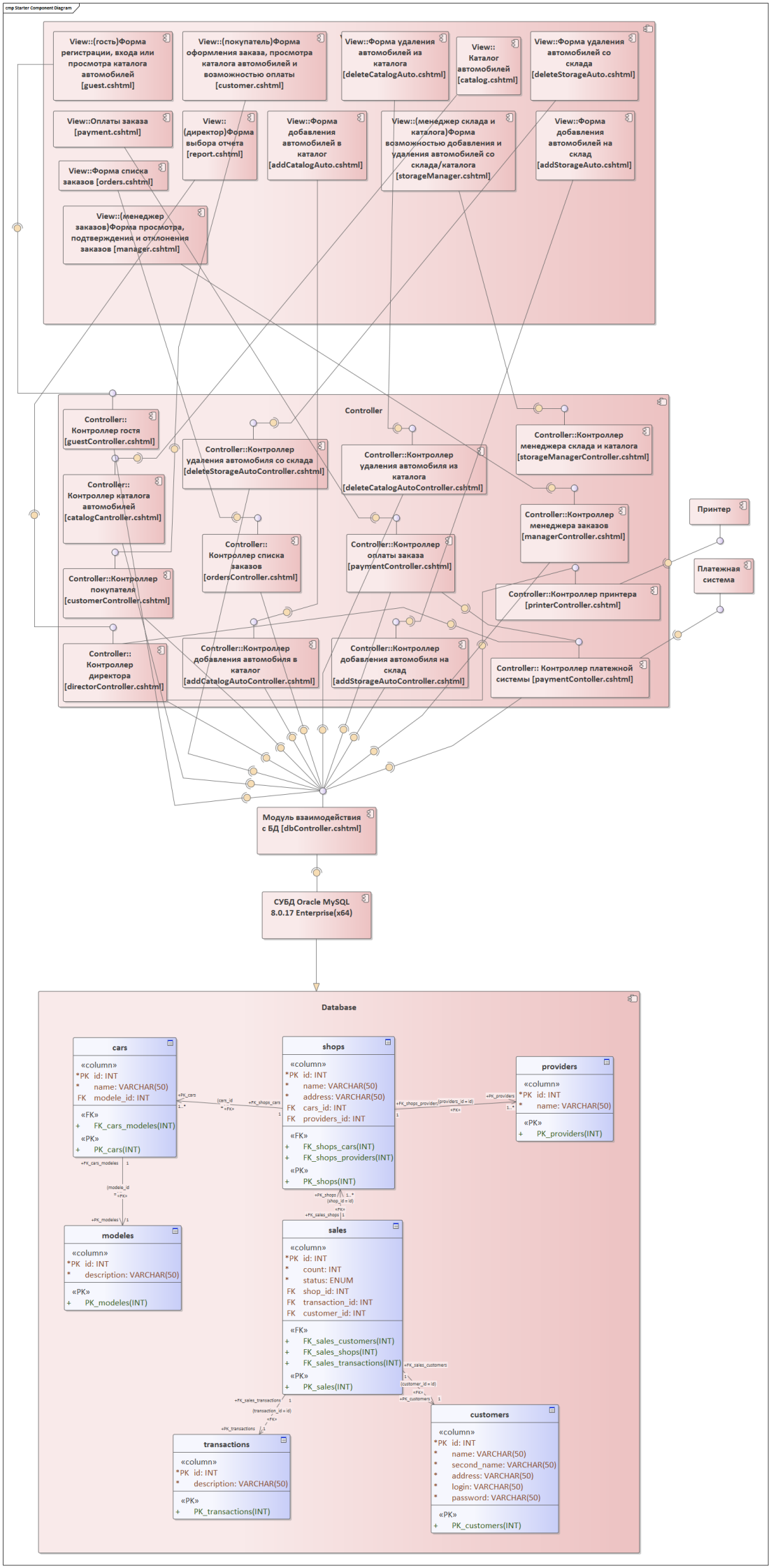
6. Отображение главной страницы менеджера заказов

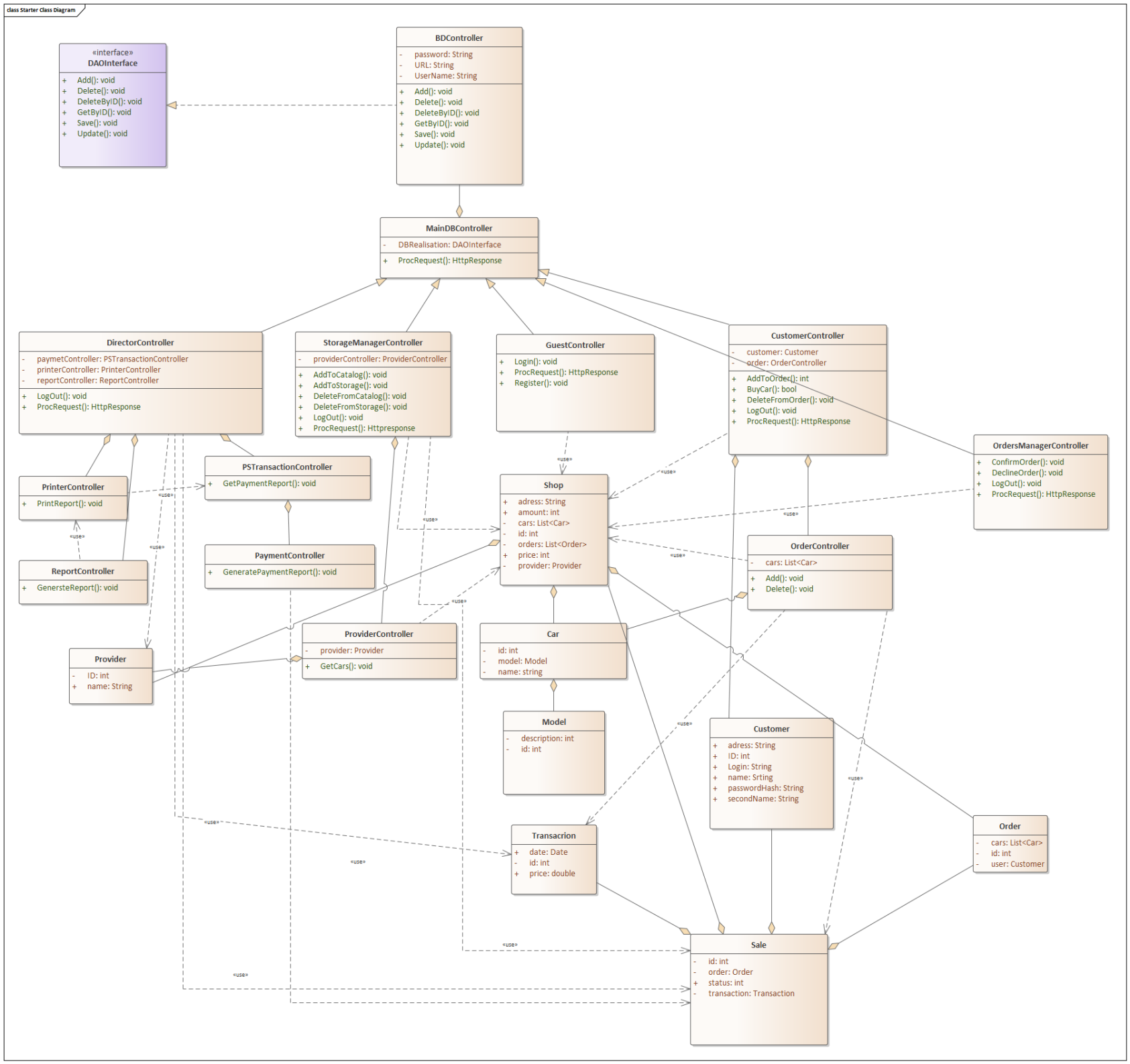


sd Basic Sequence Diagram with Reply Message

7. Удаление автомобиля из каталога




cmp Starter Component Diagram
View::(гость)Форма регистрации, входа или просмотра каталога автомобилей [guest.cshtml]
View::(покупатель)Форма оформления заказа, просмотра каталога автомобилей и возможностью оплаты [customer.cshtml]
View::Форма удаления автомобилей из каталога [deleteCatalogAuto.cshtml]
View:: Каталог автомобилей [catalog.cshtml]
View::Форма удаления автомобилей со склада [deleteStorageAuto.cshtml]
View::Оплаты заказа [payment.cshtml]
View:: (директор)Форма выбора отчета [report.cshtml]
View::Форма добавления автомобилей в каталог [addCatalogAuto.cshtml]
View::(менеджер склада и каталога)Форма возможностью добавления и удаления автомобилей со склада/каталога [storageManager.cshtml]
View::Форма добавления автомобилей на склад [addStorageAuto.cshtml]
View::Форма списка заказов [orders.cshtml]
View::(менеджер заказов)Форма просмотра, подтверждения и отклонения заказов [manager.cshtml]
Controller
Controller:: Контроллер гостя [guestController.cshtml]
Controller::Контроллер удаления автомобиля со склада [deleteStorageAutoController.cshtml]
Controller::Контроллер удаления автомобиля из каталога [deleteCatalogAutoController.cshtml]
Controller::Контроллер менеджера склада и каталога [storageManagerController.cshtml]
Controller:: Контроллер каталога автомобилей [catalogCantroller.cshtml]
Controller:: Контроллер списка заказов [ordersController.cshtml]
Controller::Контроллер оплаты заказа [paymentController.cshtml]
Controller::Контроллер менеджера заказов [managerController.cshtml]
Принтер
Платежная система
Controller::Контроллер покупателя [customerController.cshtml]
Controller::Контроллер принтера [printerController.cshtml]
Controller:: Контроллер директора [directorController.cshtml]
Controller::Контроллер добавления автомобиля в каталог [addCatalogAutoController.cshtml]
Controller::Контроллер добавления автомобиля на склад [addStorageAutoController.cshtml]
Controller:: Контроллер платежной системы [paymentContoller.cshtml]
Модуль взаимодействия с БД [dbController.cshtml]
СУБД Oracle MySQL 8.0.17 Enterprise(x64)
Database
cars
«column»
*PK id: INT
* name: VARCHAR(50)
FK modele_id: INT
«FK»
+ FK_cars_modeles(INT)
«PK»
+ PK_cars(INT)
shops
«column»
*PK id: INT
* name: VARCHAR(50)
* address: VARCHAR(50)
FK cars_id: INT
FK providers_id: INT
«FK»
+ FK_shops_cars(INT)
+ FK_shops_providers(INT)
«PK»
+ PK_shops(INT)
providers
«column»
*PK id: INT
* name: VARCHAR(50)
«PK»
+ PK_providers(INT)
modeles
«column»
*PK id: INT
* description: VARCHAR(50)
«PK»
+ PK_modeles(INT)
sales
«column»
*PK id: INT
* count: INT
* status: ENUM
FK shop_id: INT
FK transaction_id: INT
FK customer_id: INT
«FK»
+ FK_sales_customers(INT)
+ FK_sales_shops(INT)
+ FK_sales_transactions(INT)
«PK»
+ PK_sales(INT)
customers
«column»
*PK id: INT
* name: VARCHAR(50)
* second_name: VARCHAR(50)
* address: VARCHAR(50)
* login: VARCHAR(50)
* password: VARCHAR(50)
«PK»
+ PK_customers(INT)
transactions
«column»
*PK id: INT
* description: VARCHAR(50)
«PK»
+ PK_transactions(INT)


class Starter Class Diagram
«interface»
DAOInterface
+ Add(): void
+ Delete(): void
+ DeleteByID(): void
+ GetByID(): void
+ Save(): void
+ Update(): void
BDController
- password: String
- URL: String
- UserName: String
+ Add(): void
+ Delete(): void
+ DeleteByID(): void
+ GetByID(): void
+ Save(): void
+ Update(): void
MainDBController
- DBRealisation: DAOInterface
+ ProcRequest(): HttpResponse
DirectorController
- paymetController: PSTransactionController
- printerController: PrinterController
- reportController: ReportController
+ LogOut(): void
+ ProcRequest(): HttpResponse
StorageManagerController
- providerController: ProviderController
+ AddToCatalog(): void
+ AddToStorage(): void
+ DeleteFromCatalog(): void
+ DeleteFromStorage(): void
+ LogOut(): void
+ ProcRequest(): Httpresponse
GuestController
+ Login(): void
+ ProcRequest(): HttpResponse
+ Register(): void
CustomerController
- customer: Customer
- order: OrderController
+ AddToOrder(): int
+ BuyCar(): bool
+ DeleteFromOrder(): void
+ LogOut(): void
+ ProcRequest(): HttpResponse
OrdersManagerController
+ ConfirmOrder(): void
+ DeclineOrder(): void
+ LogOut(): void
+ ProcRequest(): HttpResponse
PrinterController
+ PrintReport(): void
PSTransactionController
+ GetPaymentReport(): void
Shop
+ adress: String
+ amount: int
- cars: List<Car>
- id: int
- orders: List<Order>
+ price: int
- provider: Provider
OrderController
- cars: List<Car>
+ Add(): void
+ Delete(): void
ReportController
+ GenersteReport(): void
PaymentController
+ GeneratePaymentReport(): void
Provider
- ID: int
+ name: String
ProviderController
- provider: Provider
+ GetCars(): void
Car
- id: int
- model: Model
- name: string
Model
- description: int
- id: int
Customer
+ adress: String
+ ID: int
+ Login: String
+ name: Srting
+ passwordHash: String
+ secondName: String
Transacrion
+ date: Date
- id: int
+ price: double
Order
- cars: List<Car>
- id: int
- user: Customer
Sale
- id: int
- order: Order
+ status: int
- transaction: Transaction
«use»

Сгенерированный код:

```
///////////////////////////////////////////////////////////
//  AssistentController.cs
//  Implementation of the Class AssistentController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:48
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class AssistentController : MainDBController {

	public AssistentController(){

	}

	~AssistentController(){

	}

	public List<Car> GetCars(){

		return null;
	}

	public void LogOut(){

	}

	public HttpResponse ProcRequest(){

		return null;
	}

	public List<Car> Search(){

		return null;
	}

	public void SellCar(){

	}

}//end AssistentController

///////////////////////////////////////////////////////////
//  BDController.cs
//  Implementation of the Class BDController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:48
//  Original author: Ilya
```

```
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class BDController : DAOInterface {

	private String password;
	private String URL;
	private String UserName;

	public BDController(){

	}

	~BDController(){

	}

	public void Add(){

	}

	public void Delete(){

	}

	public void DeleteByID(){

	}

	public void GetByID(){

	}

	public void Save(){

	}

	public void Update(){

	}

}//end BDController

///////////////////////////////////////////////////////////
//  Car.cs
//  Implementation of the Class Car
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:48
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
```

```
using System.Text;
using System.IO;



public class Car {

	private int id;
	private Model model;
	private string name;
	public Model m_Model;

	public Car(){

	}

	~Car(){

	}

}//end Car

///////////////////////////////////////////////////////////
//  Customer.cs
//  Implementation of the Class Customer
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:48
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class Customer {

	public String adress;
	public int ID;
	public String Login;
	public Srting name;
	/// <summary>
	/// ame
	/// </summary>
	public String passwordHash;
	public String secondName;

	public Customer(){

	}

	~Customer(){

	}

}//end Customer
```

```
///////////////////////////////////////////////////////////
//  CustomerController.cs
//  Implementation of the Class CustomerController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:48
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class CustomerController : MainDBController {

	private Customer customer;
	private OrderController order;
	public OrderController m_OrderController;
	public Customer m_Customer;

	public CustomerController(){

	}

	~CustomerController(){

	}

	public int AddToOrder(){

		return 0;
	}

	public bool BuyCar(){

		return false;
	}

	public void DeleteFromOrder(){

	}

	public void LogOut(){

	}

	public HttpResponse ProcRequest(){

		return null;
	}

}//end CustomerController

///////////////////////////////////////////////////////////
//  DAOInterface.cs
//  Implementation of the Interface DAOInterface
//  Generated by Enterprise Architect
```

```
// Created on:      08-дек-2020 18:58:48
// Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public interface DAOInterface  {

        void Add();

        void Delete();

        void DeleteByID();

        void GetByID();

        void Save();

        void Update();
}//end DAOInterface

///////////////////////////////////////////////////////////
// DirectorController.cs
// Implementation of the Class DirectorController
// Generated by Enterprise Architect
// Created on:      08-дек-2020 18:58:48
// Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class DirectorController : MainDBController {

        private PSTransactionController paymetController;
        private PrinterController printerController;
        private ReportController reportController;
        public PSTransactionController m_PSTransactionController;
        public PrinterController m_PrinterController;
        public ReportController m_ReportController;

        public DirectorController(){

        }

        ~DirectorController(){

        }

        public void LogOut(){
```

```
	}

	public HttpResponse ProcRequest(){

		return null;
	}

}//end DirectorController

///////////////////////////////////////////////////////////
//  GuestController.cs
//  Implementation of the Class GuestController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class GuestController : MainDBController {

	public GuestController(){

	}

	~GuestController(){

	}

	public void Login(){

	}

	public HttpResponse ProcRequest(){

		return null;
	}

	public void Register(){

	}

}//end GuestController

///////////////////////////////////////////////////////////
//  MainDBController.cs
//  Implementation of the Class MainDBController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
```

```
using System.Collections.Generic;
using System.Text;
using System.IO;



public class MainDBController {

	private DAOInterface DBRealisation;
	public BDController m_BDController;

	public MainDBController(){

	}

	~MainDBController(){

	}

	public HttpResponse ProcRequest(){

		return null;
	}

}//end MainDBController

///////////////////////////////////////////////////////////
//  Model.cs
//  Implementation of the Class Model
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class Model {

	private int description;
	private int id;

	public Model(){

	}

	~Model(){

	}

}//end Model

///////////////////////////////////////////////////////////
//  Order.cs
//  Implementation of the Class Order
```

```
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class Order {

	private List<Car> cars;
	private int id;
	private Customer user;

	public Order(){

	}

	~Order(){

	}

}//end Order

///////////////////////////////////////////////////////////
//  OrderController.cs
//  Implementation of the Class OrderController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class OrderController {

	private List<Car> cars;
	public Car m_Car;

	public OrderController(){

	}

	~OrderController(){

	}

	public void Add(){

	}
```

```
		public void Delete(){

		}

}//end OrderController

///////////////////////////////////////////////////////////
//  OrdersManagerController.cs
//  Implementation of the Class OrdersManagerController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class OrdersManagerController : MainDBController {

		public OrdersManagerController(){

		}

		~OrdersManagerController(){

		}

		public void ConfirmOrder(){

		}

		public void DeclineOrder(){

		}

		public void LogOut(){

		}

		public HttpResponse ProcRequest(){

			return null;
		}

}//end OrdersManagerController

///////////////////////////////////////////////////////////
//  PaymentController.cs
//  Implementation of the Class PaymentController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////
```

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class PaymentController {

	public PaymentController(){

	}

	~PaymentController(){

	}

	public void GeneratePaymentReport(){

	}

}//end PaymentController

///////////////////////////////////////////////////////////
//  PrinterController.cs
//  Implementation of the Class PrinterController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class PrinterController {

	public PrinterController(){

	}

	~PrinterController(){

	}

	public void PrintReport(){

	}

}//end PrinterController

///////////////////////////////////////////////////////////
//  Provider.cs
//  Implementation of the Class Provider
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
```

```
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class Provider {

	private int ID;
	public String name;

	public Provider(){

	}

	~Provider(){

	}

}//end Provider

///////////////////////////////////////////////////////////
//  ProviderController.cs
//  Implementation of the Class ProviderController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class ProviderController {

	private Provider provider;
	public Provider m_Provider;

	public ProviderController(){

	}

	~ProviderController(){

	}

	public void GetCars(){

	}

}//end ProviderController
```

```
///////////////////////////////////////////////////////////
//  PSTransactionController.cs
//  Implementation of the Class PSTransactionController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class PSTransactionController {

	public PaymentController m_PaymentController;

	public PSTransactionController(){

	}

	~PSTransactionController(){

	}

	public void GetPaymentReport(){

	}

}//end PSTransactionController

///////////////////////////////////////////////////////////
//  ReportController.cs
//  Implementation of the Class ReportController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class ReportController {

	public ReportController(){

	}

	~ReportController(){

	}

	public void GenersteReport(){
```

```
        }

}//end ReportController

///////////////////////////////////////////////////////////
//  Sale.cs
//  Implementation of the Class Sale
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class Sale {

        private int id;
        private Order order;
        public int status;
        private Transaction transaction;
        public Shop m_Shop;
        public Customer m_Customer;
        public Order m_Order;
        public Transacrion m_Transacrion;

        public Sale(){

        }

        ~Sale(){

        }

}//end Sale

///////////////////////////////////////////////////////////
//  Shop.cs
//  Implementation of the Class Shop
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class Shop {

        public String adress;
```

```
        public int amount;
        private List<Car> cars;
        private int id;
        private List<Order> orders;
        public int price;
        private Provider provider;
        public Car m_Car;
        public Order m_Order;
        public Provider m_Provider;

        public Shop(){

        }

        ~Shop(){

        }

}//end Shop

///////////////////////////////////////////////////////////
//  StorageManagerController.cs
//  Implementation of the Class StorageManagerController
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class StorageManagerController : MainDBController {

        private ProviderController providerController;
        public ProviderController m_ProviderController;

        public StorageManagerController(){

        }

        ~StorageManagerController(){

        }

        public void AddToCatalog(){

        }

        public void AddToStorage(){

        }

        public void DeleteFromCatalog(){

        }
```

```
	public void DeleteFromStorage(){

	}

	public void LogOut(){

	}

	public Httpresponse ProcRequest(){

		return null;
	}

}//end StorageManagerController

///////////////////////////////////////////////////////////
//  Transacrion.cs
//  Implementation of the Class Transacrion
//  Generated by Enterprise Architect
//  Created on:      08-дек-2020 18:58:49
//  Original author: Ilya
///////////////////////////////////////////////////////////

using System;
using System.Collections.Generic;
using System.Text;
using System.IO;



public class Transacrion {

	public Date date;
	private int id;
	public double price;

	public Transacrion(){

	}

	~Transacrion(){

	}

}//end Transacrion
```