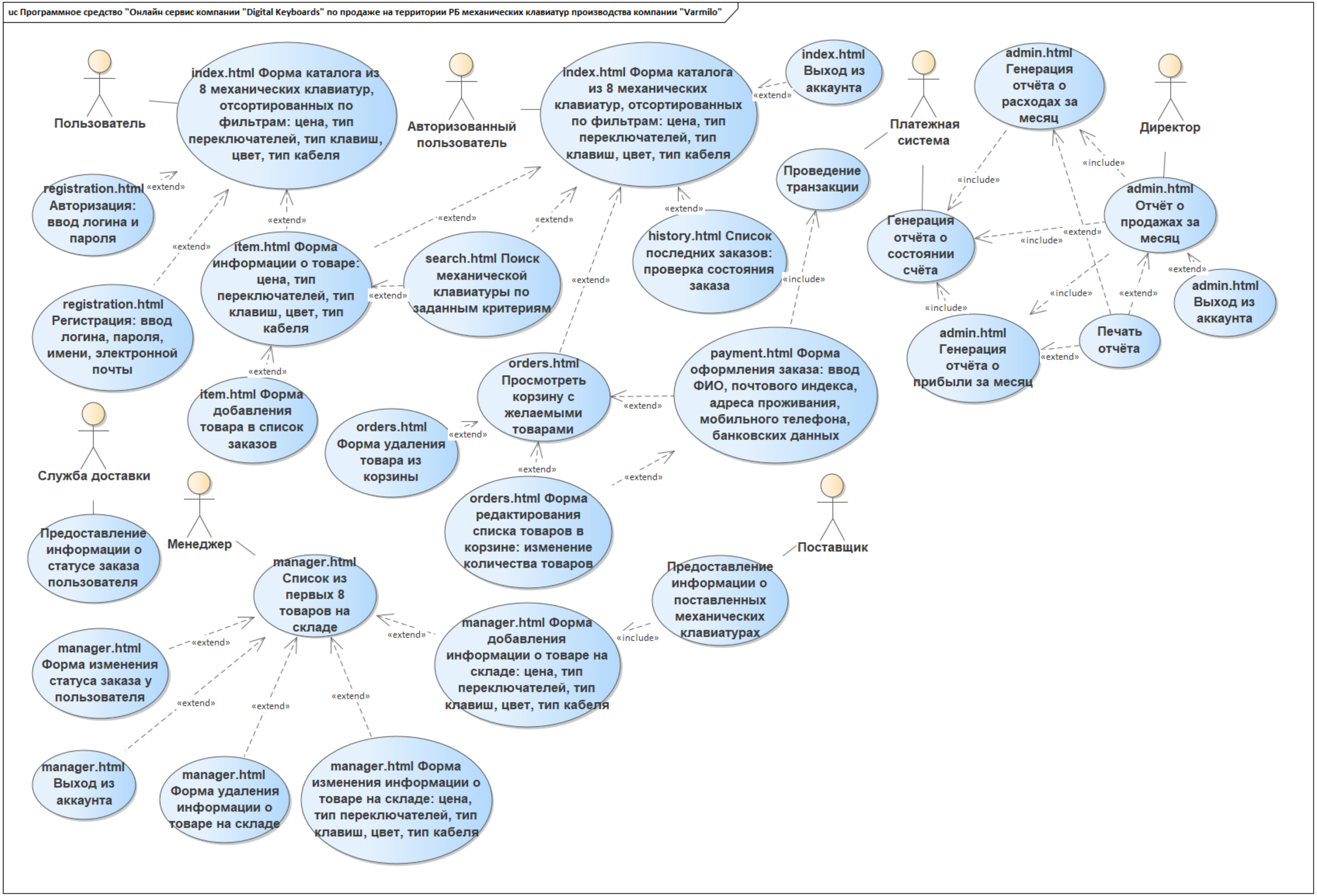
Белорусский государственный университет
информатики и радиоэлектроники
Кафедра ПОИТ

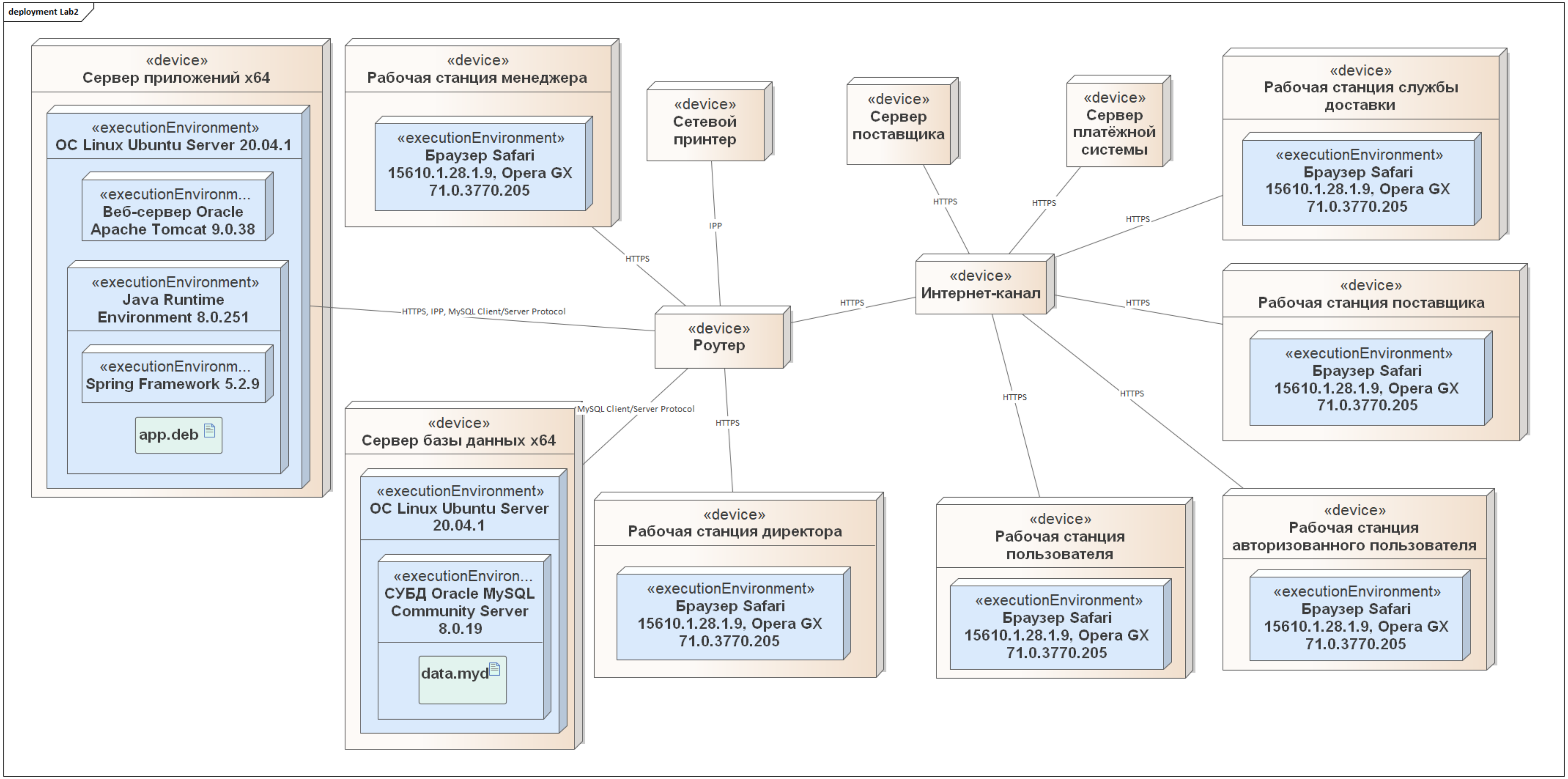
Отчет по лабораторным работам
по дисциплине
«Объектно-ориентированные технологии программирования и стандарты проектирования»

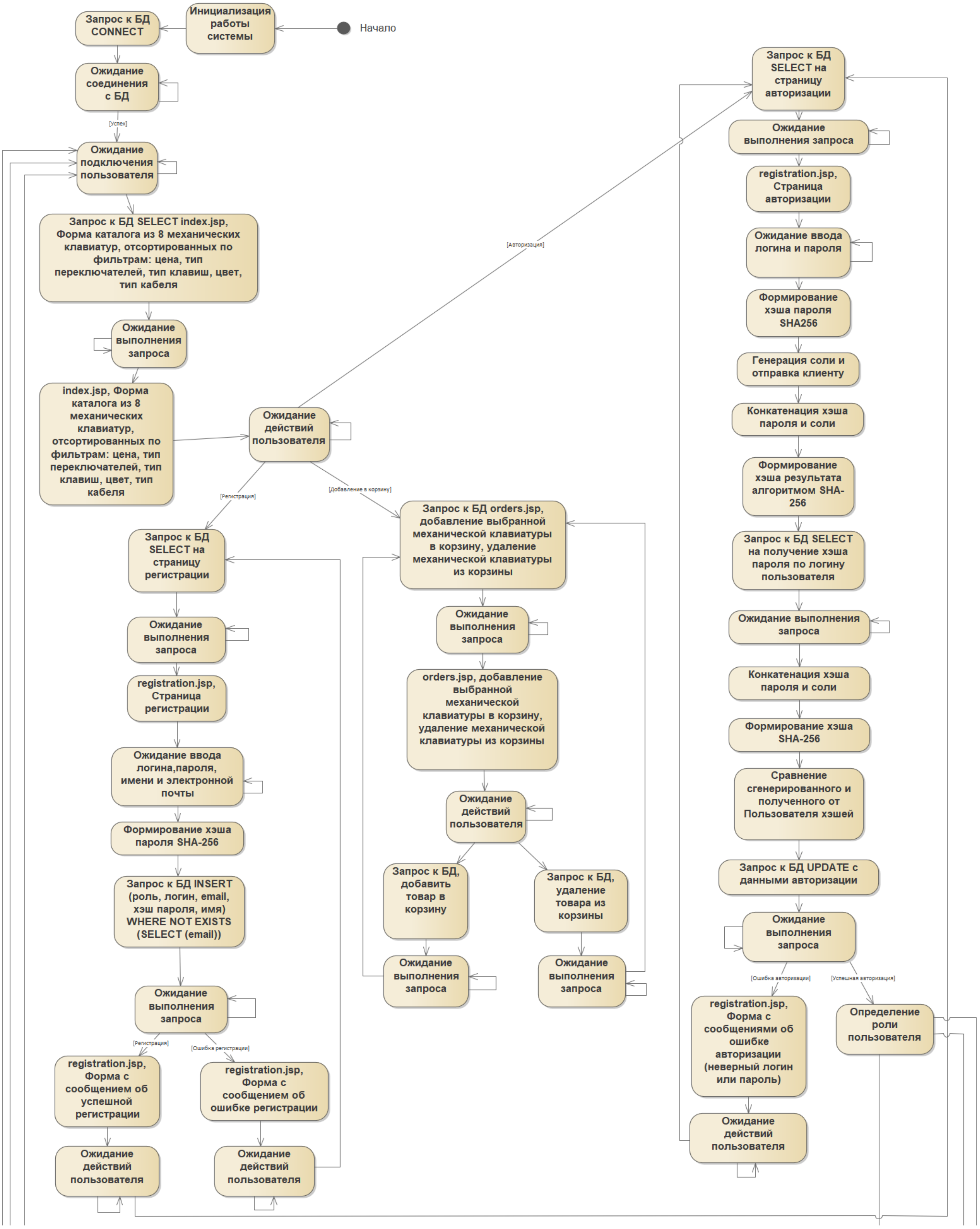
Выполнил:
Гладкий М.Г.
группа 851005

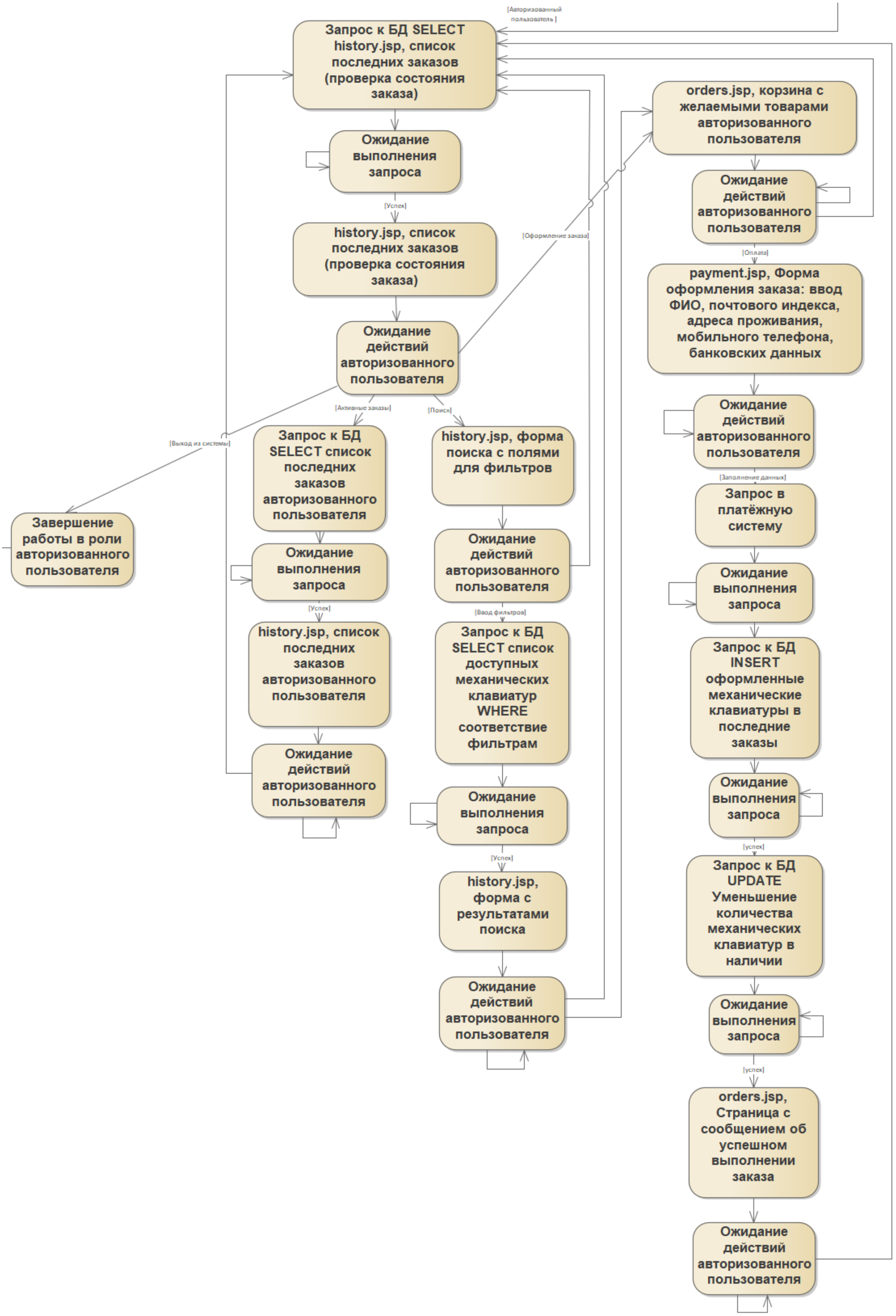
Проверил:
Алексеев И.Г.

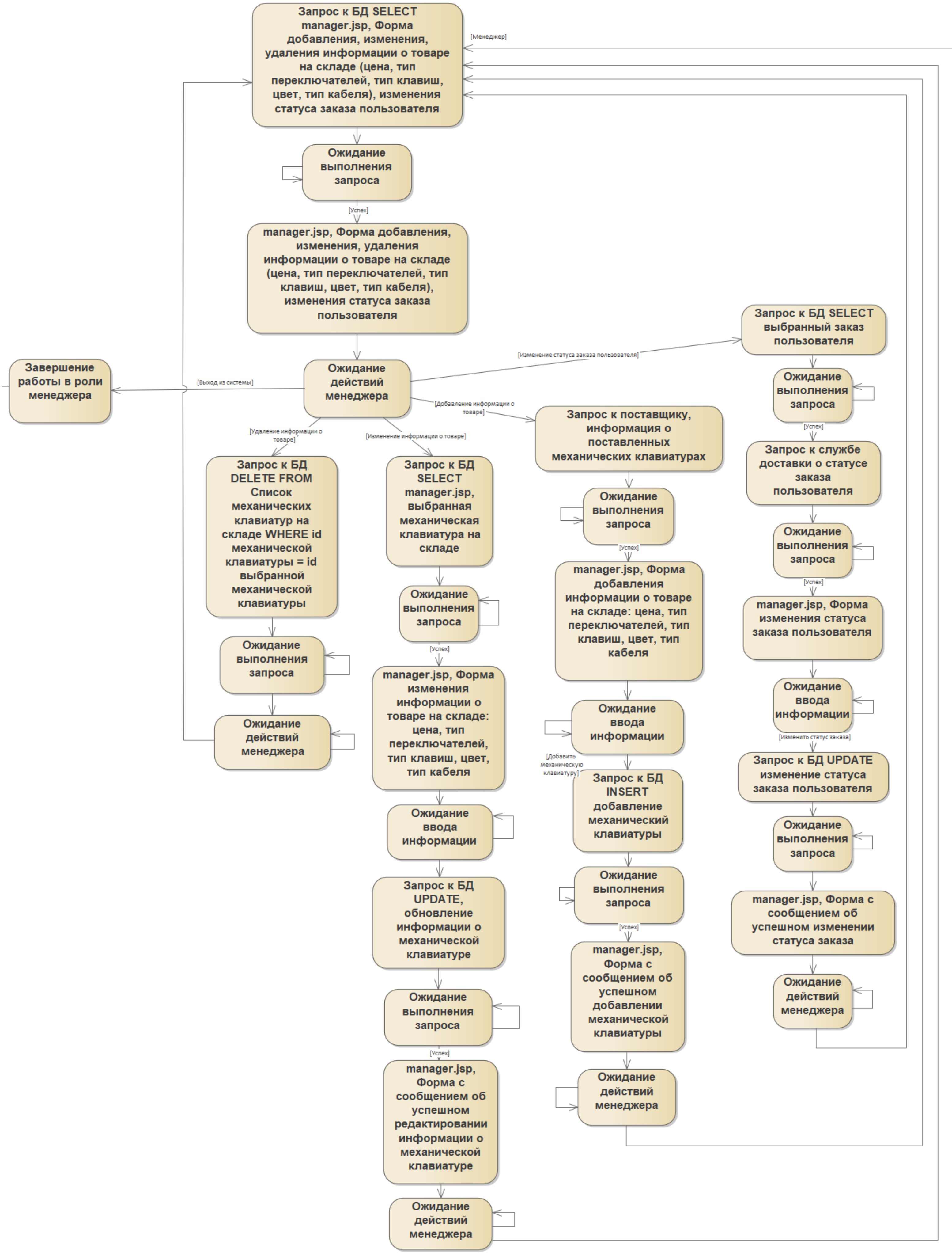
Минск, 2020

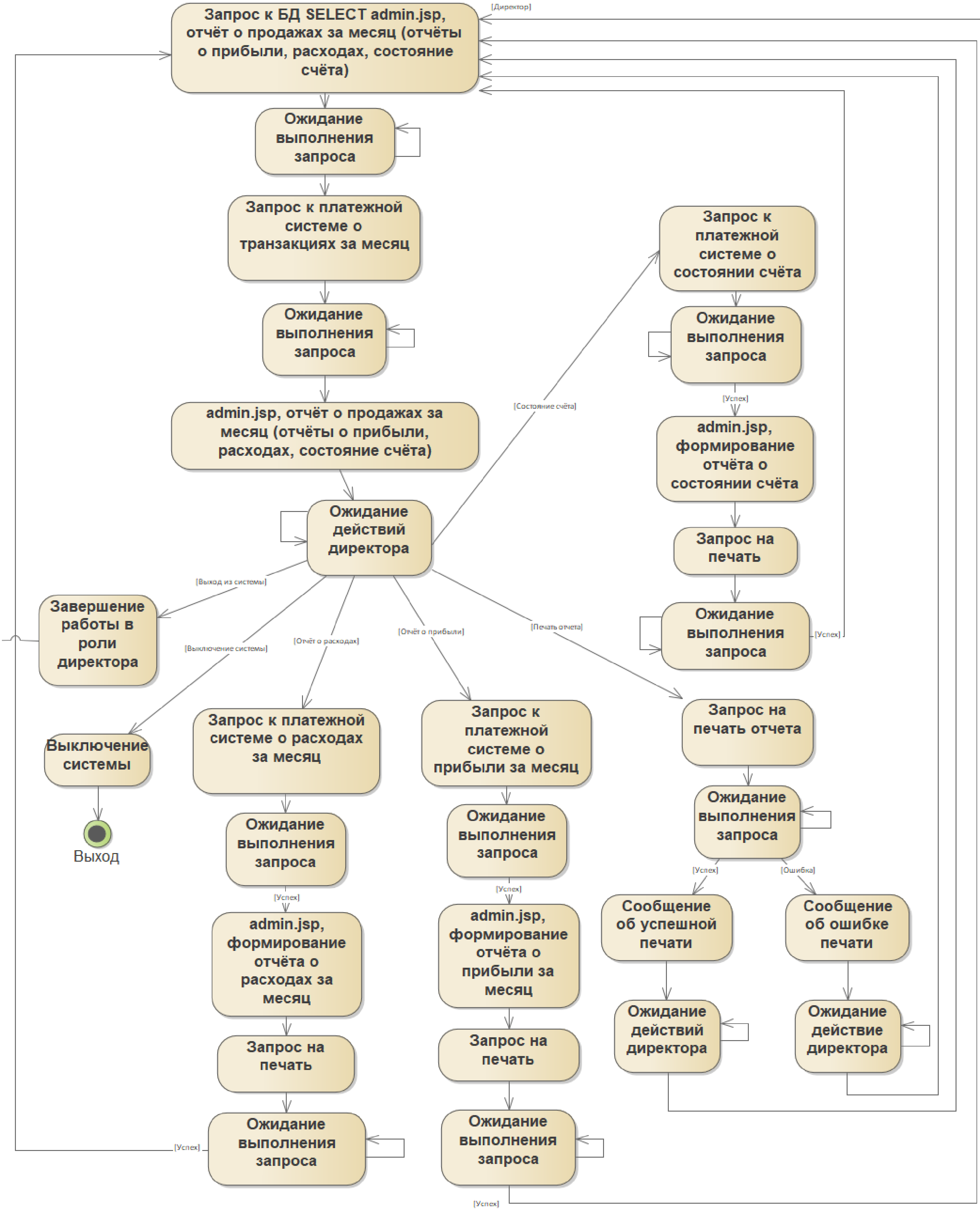

uc Программное средство "Онлайн сервис компании "Digital Keyboards" по продаже на территории РБ механических клавиатур производства компании "Varmilo"
Пользователь
index.html Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Авторизованный пользователь
index.html Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
index.html Выход из аккаунта
Платежная система
admin.html Генерация отчёта о расходах за месяц
Директор
registration.html Авторизация: ввод логина и пароля
item.html Форма информации о товаре: цена, тип переключателей, тип клавиш, цвет, тип кабеля
search.html Поиск механической клавиатуры по заданным критериям
history.html Список последних заказов: проверка состояния заказа
Проведение транзакции
Генерация отчёта о состоянии счёта
admin.html Отчёт о продажах за месяц
admin.html Выход из аккаунта
Печать отчёта
admin.html Генерация отчёта о прибыли за месяц
registration.html Регистрация: ввод логина, пароля, имени, электронной почты
item.html Форма добавления товара в список заказов
orders.html Просмотреть корзину с желаемыми товарами
payment.html Форма оформления заказа: ввод ФИО, почтового индекса, адреса проживания, мобильного телефона, банковских данных
orders.html Форма удаления товара из корзины
orders.html Форма редактирования списка товаров в корзине: изменение количества товаров
Служба доставки
Предоставление информации о статусе заказа пользователя
Менеджер
manager.html Список из первых 8 товаров на складе
Поставщик
Предоставление информации о поставленных механических клавиатурах
manager.html Форма добавления информации о товаре на складе: цена, тип переключателей, тип клавиш, цвет, тип кабеля
manager.html Форма изменения статуса заказа у пользователя
manager.html Выход из аккаунта
manager.html Форма удаления информации о товаре на складе
manager.html Форма изменения информации о товаре на складе: цена, тип переключателей, тип клавиш, цвет, тип кабеля
«extend»
«include»

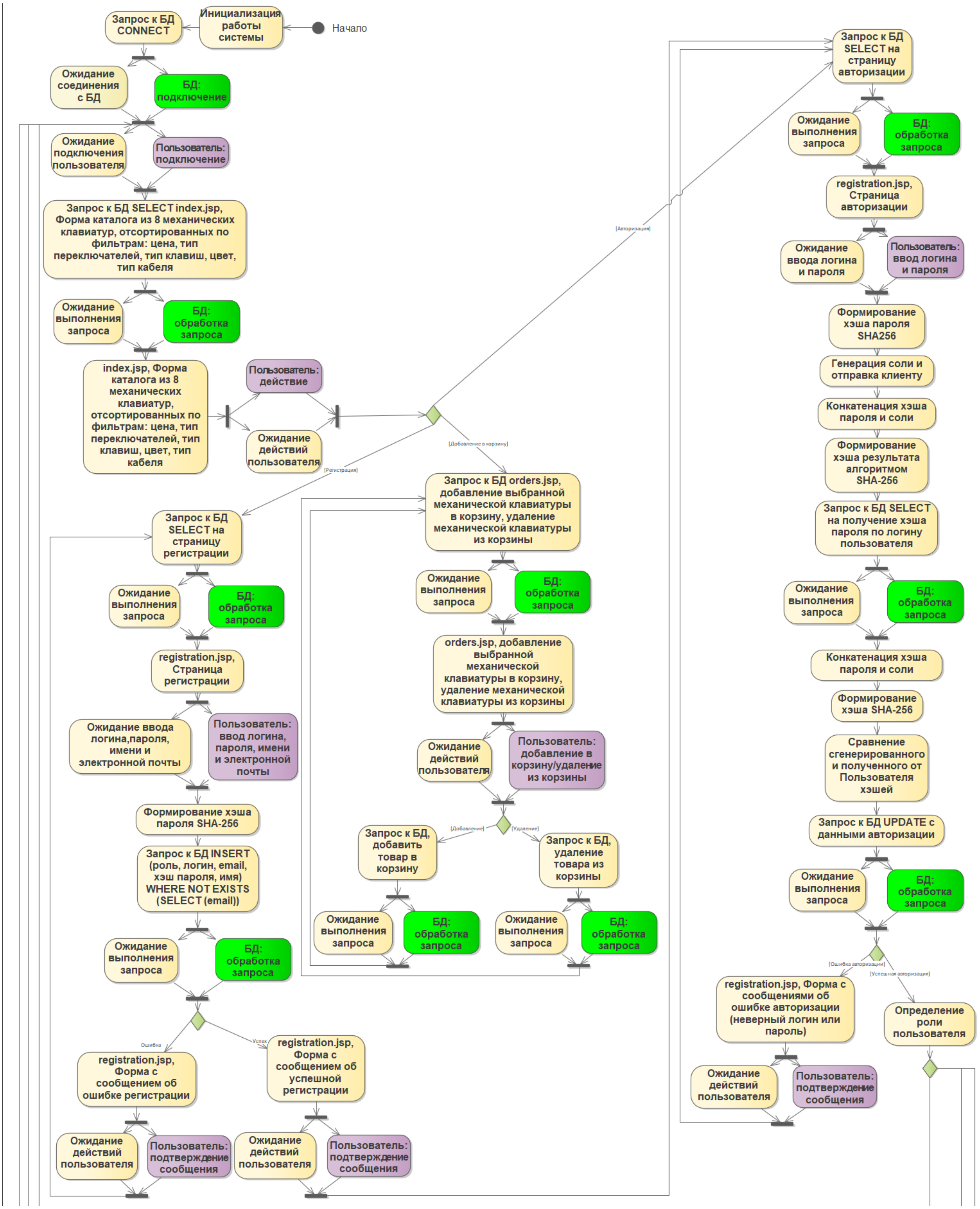

deployment Lab2
«device»
Сервер приложений x64
«executionEnvironment»
ОС Linux Ubuntu Server 20.04.1
«executionEnvironm...
Веб-сервер Oracle Apache Tomcat 9.0.38
«executionEnvironment»
Java Runtime Environment 8.0.251
«executionEnvironm...
Spring Framework 5.2.9
app.deb
«device»
Рабочая станция менеджера
«executionEnvironment»
Браузер Safari 15610.1.28.1.9, Opera GX 71.0.3770.205
«device»
Сетевой принтер
«device»
Сервер поставщика
«device»
Сервер платёжной системы
«device»
Рабочая станция службы доставки
«executionEnvironment»
Браузер Safari 15610.1.28.1.9, Opera GX 71.0.3770.205
HTTPS
IPP
HTTPS
HTTPS
HTTPS
HTTPS, IPP, MySQL Client/Server Protocol
«device»
Роутер
HTTPS
«device»
Интернет-канал
HTTPS
«device»
Рабочая станция поставщика
«executionEnvironment»
Браузер Safari 15610.1.28.1.9, Opera GX 71.0.3770.205
MySQL Client/Server Protocol
HTTPS
HTTPS
HTTPS
«device»
Сервер базы данных x64
«executionEnvironment»
ОС Linux Ubuntu Server 20.04.1
«executionEnviron...
СУБД Oracle MySQL Community Server 8.0.19
data.myd
«device»
Рабочая станция директора
«executionEnvironment»
Браузер Safari 15610.1.28.1.9, Opera GX 71.0.3770.205
«device»
Рабочая станция пользователя
«executionEnvironment»
Браузер Safari 15610.1.28.1.9, Opera GX 71.0.3770.205
«device»
Рабочая станция авторизованного пользователя
«executionEnvironment»
Браузер Safari 15610.1.28.1.9, Opera GX 71.0.3770.205

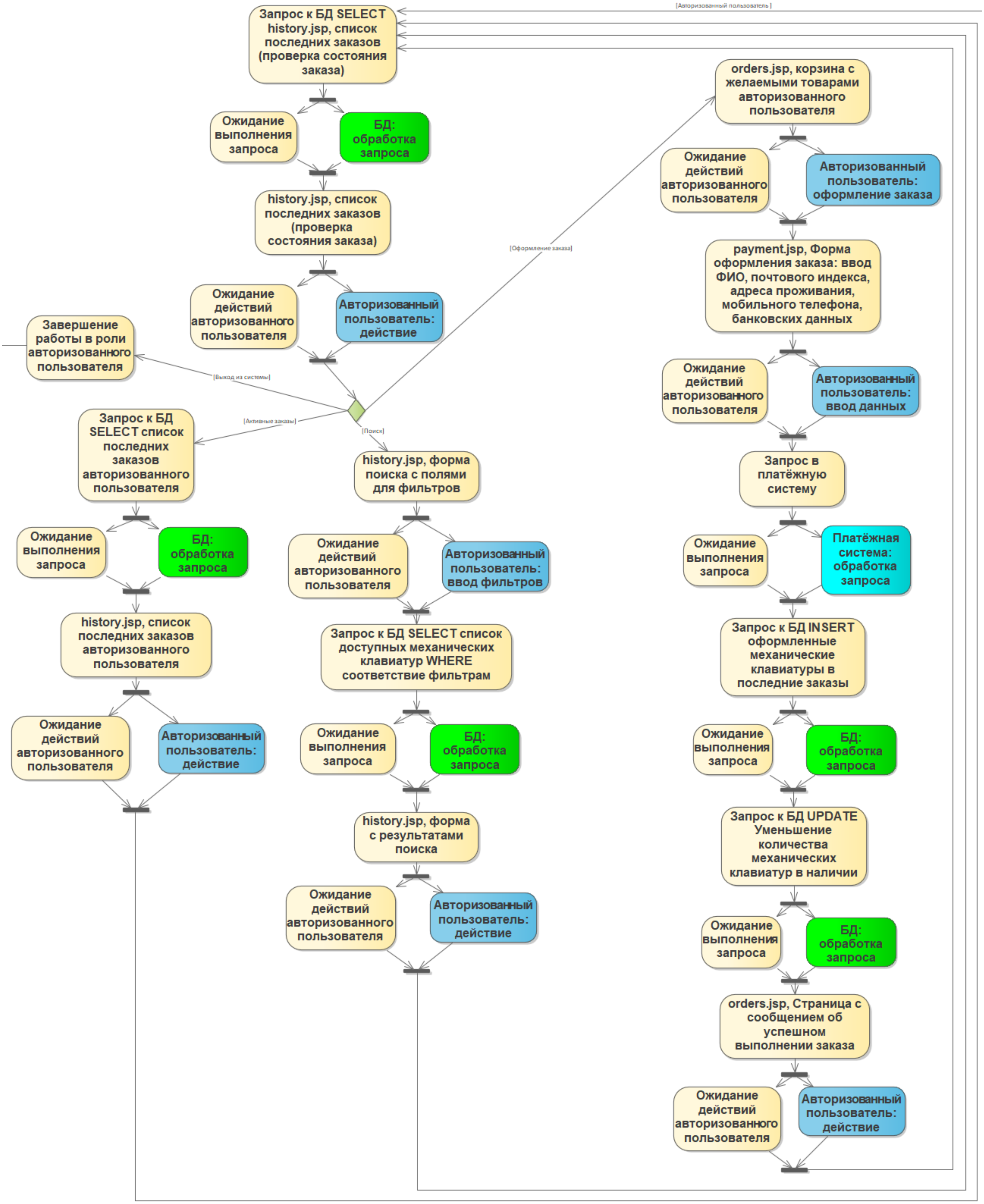

Начало
Инициализация работы системы
Запрос к БД CONNECT
Ожидание соединения с БД
[Успех]
Ожидание подключения пользователя
Запрос к БД SELECT index.jsp, Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Ожидание выполнения запроса
index.jsp, Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Ожидание действий пользователя
[Авторизация]
[Регистрация]
[Добавление в корзину]
Запрос к БД SELECT на страницу регистрации
Ожидание выполнения запроса
registration.jsp, Страница регистрации
Ожидание ввода логина,пароля, имени и электронной почты
Формирование хэша пароля SHA-256
Запрос к БД INSERT (роль, логин, email, хэш пароля, имя) WHERE NOT EXISTS (SELECT (email))
Ожидание выполнения запроса
[Регистрация]
[Ошибка регистрации]
registration.jsp, Форма с сообщением об успешной регистрации
registration.jsp, Форма с сообщением об ошибке регистрации
Ожидание действий пользователя
Ожидание действий пользователя
Запрос к БД orders.jsp, добавление выбранной механической клавиатуры в корзину, удаление механической клавиатуры из корзины
Ожидание выполнения запроса
orders.jsp, добавление выбранной механической клавиатуры в корзину, удаление механической клавиатуры из корзины
Ожидание действий пользователя
Запрос к БД, добавить товар в корзину
Запрос к БД, удаление товара из корзины
Ожидание выполнения запроса
Ожидание выполнения запроса
Запрос к БД SELECT на страницу авторизации
Ожидание выполнения запроса
registration.jsp, Страница авторизации
Ожидание ввода логина и пароля
Формирование хэша пароля SHA256
Генерация соли и отправка клиенту
Конкатенация хэша пароля и соли
Формирование хэша результата алгоритмом SHA-256
Запрос к БД SELECT на получение хэша пароля по логину пользователя
Ожидание выполнения запроса
Конкатенация хэша пароля и соли
Формирование хэша SHA-256
Сравнение сгенерированного и полученного от Пользователя хэшей
Запрос к БД UPDATE с данными авторизации
Ожидание выполнения запроса
[Ошибка авторизации]
[Успешная авторизация]
registration.jsp, Форма с сообщениями об ошибке авторизации (неверный логин или пароль)
Определение роли пользователя
Ожидание действий пользователя

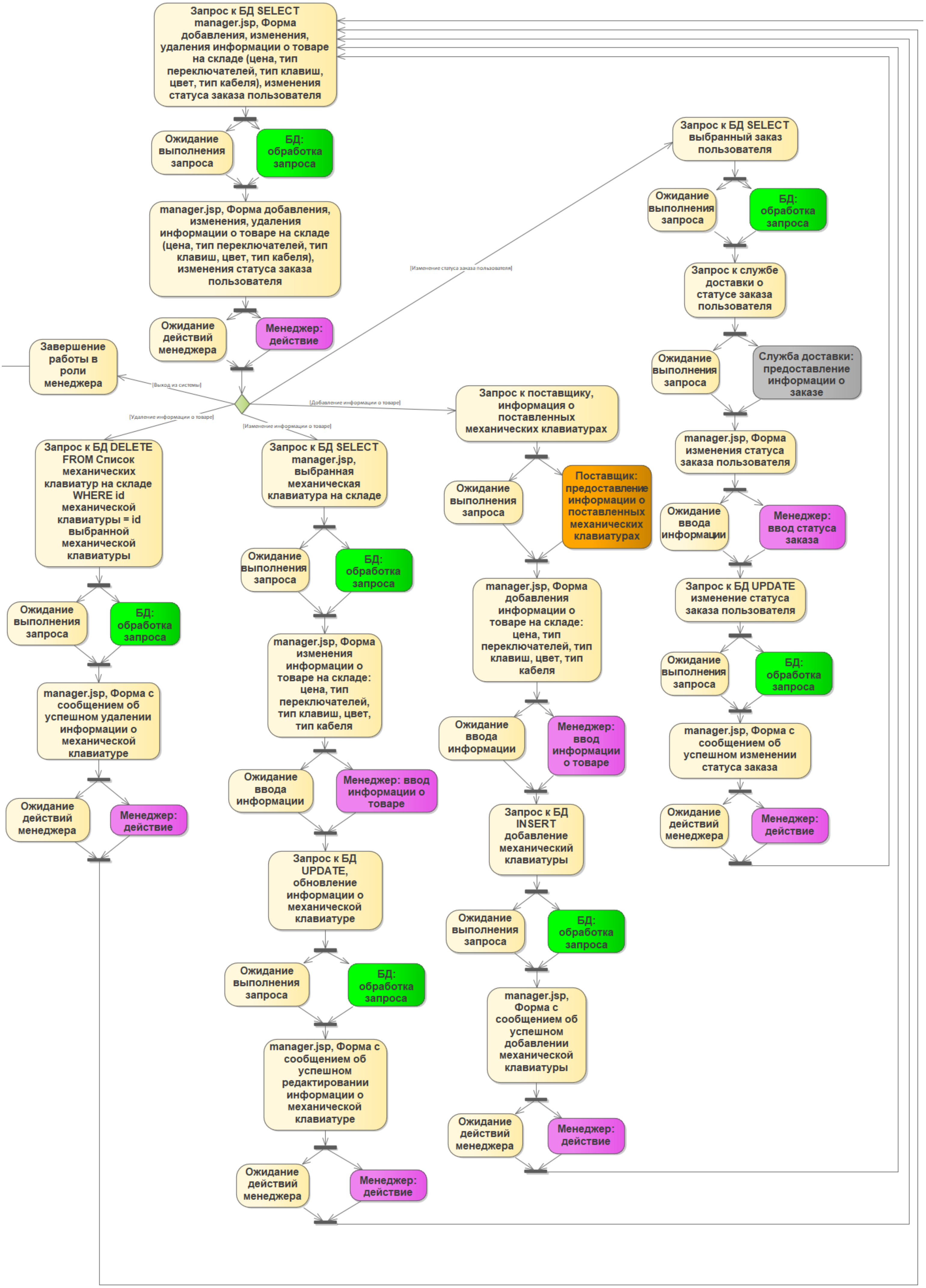

[Авторизованный пользователь]
Запрос к БД SELECT history.jsp, список последних заказов (проверка состояния заказа)
Ожидание выполнения запроса
[Успех]
history.jsp, список последних заказов (проверка состояния заказа)
Ожидание действий авторизованного пользователя
[Оформление заказа]
[Активные заказы]
[Поиск]
[Выход из системы]
Завершение работы в роли авторизованного пользователя
Запрос к БД SELECT список последних заказов авторизованного пользователя
Ожидание выполнения запроса
[Успех]
history.jsp, список последних заказов авторизованного пользователя
Ожидание действий авторизованного пользователя
history.jsp, форма поиска с полями для фильтров
Ожидание действий авторизованного пользователя
[Ввод фильтров]
Запрос к БД SELECT список доступных механических клавиатур WHERE соответствие фильтрам
Ожидание выполнения запроса
[Успех]
history.jsp, форма с результатами поиска
Ожидание действий авторизованного пользователя
orders.jsp, корзина с желаемыми товарами авторизованного пользователя
Ожидание действий авторизованного пользователя
[Оплата]
payment.jsp, Форма оформления заказа: ввод ФИО, почтового индекса, адреса проживания, мобильного телефона, банковских данных
Ожидание действий авторизованного пользователя
[Заполнение данных]
Запрос в платёжную систему
Ожидание выполнения запроса
Запрос к БД INSERT оформленные механические клавиатуры в последние заказы
Ожидание выполнения запроса
[успех]
Запрос к БД UPDATE Уменьшение количества механических клавиатур в наличии
Ожидание выполнения запроса
[успех]
orders.jsp, Страница с сообщением об успешном выполнении заказа
Ожидание действий авторизованного пользователя

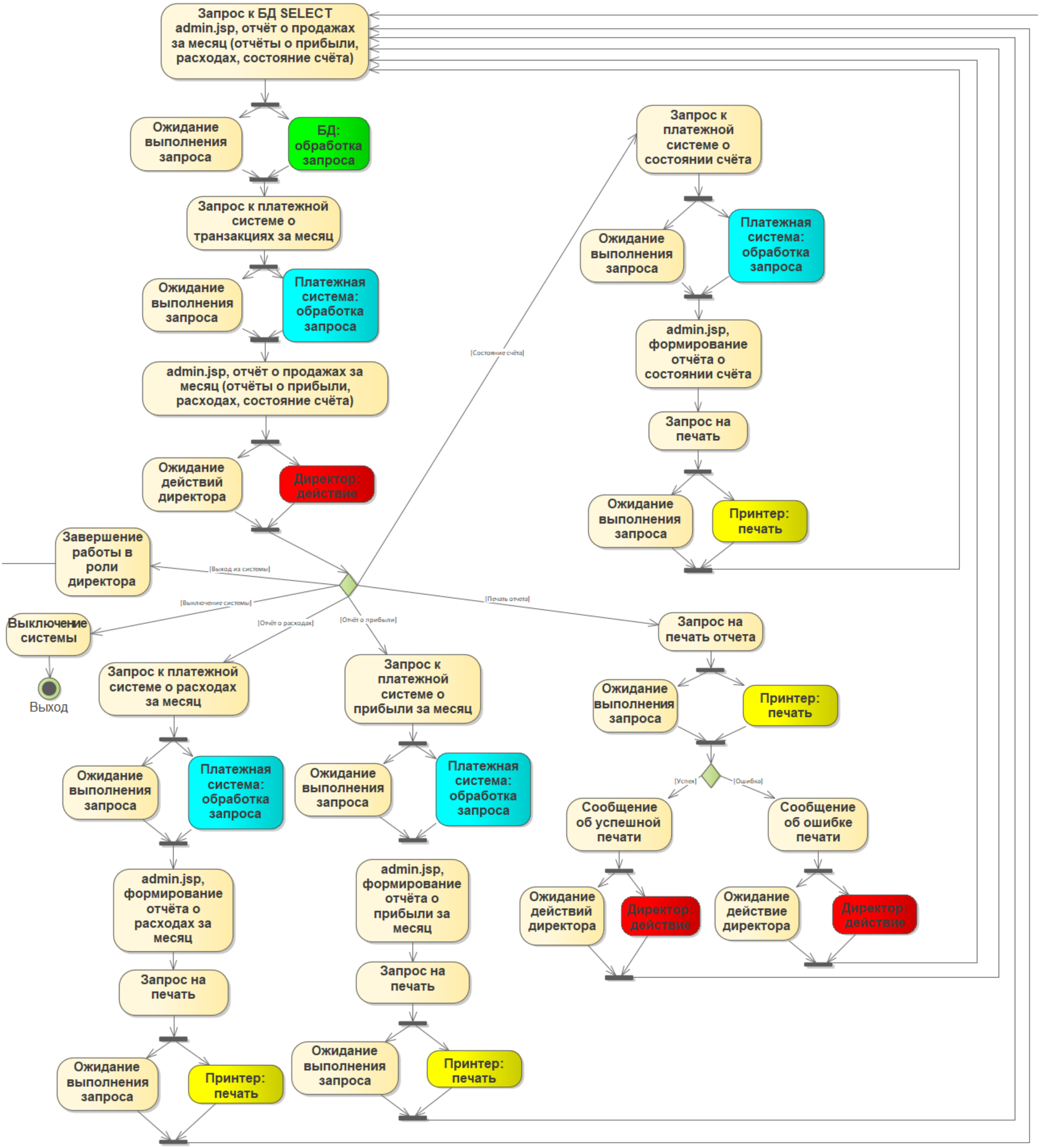

Запрос к БД SELECT manager.jsp, Форма добавления, изменения, удаления информации о товаре на складе (цена, тип переключателей, тип клавиш, цвет, тип кабеля), изменения статуса заказа пользователя
[Менеджер]
Ожидание выполнения запроса
[Успех]
manager.jsp, Форма добавления, изменения, удаления информации о товаре на складе (цена, тип переключателей, тип клавиш, цвет, тип кабеля), изменения статуса заказа пользователя
Завершение работы в роли менеджера
[Выход из системы]
Ожидание действий менеджера
[Изменение статуса заказа пользователя]
Запрос к БД SELECT выбранный заказ пользователя
Ожидание выполнения запроса
[Успех]
Запрос к службе доставки о статусе заказа пользователя
Ожидание выполнения запроса
[Успех]
manager.jsp, Форма изменения статуса заказа пользователя
Ожидание ввода информации
[Изменить статус заказа]
Запрос к БД UPDATE изменение статуса заказа пользователя
Ожидание выполнения запроса
manager.jsp, Форма с сообщением об успешном изменении статуса заказа
Ожидание действий менеджера
[Добавление информации о товаре]
Запрос к поставщику, информация о поставленных механических клавиатурах
Ожидание выполнения запроса
[Успех]
manager.jsp, Форма добавления информации о товаре на складе: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Ожидание ввода информации
[Добавить механическую клавиатуру]
Запрос к БД INSERT добавление механический клавиатуры
Ожидание выполнения запроса
[Успех]
manager.jsp, Форма с сообщением об успешном добавлении механической клавиатуры
Ожидание действий менеджера
[Удаление информации о товаре]
Запрос к БД DELETE FROM Список механических клавиатур на складе WHERE id механической клавиатуры = id выбранной механической клавиатуры
Ожидание выполнения запроса
Ожидание действий менеджера
[Изменение информации о товаре]
Запрос к БД SELECT manager.jsp, выбранная механическая клавиатура на складе
Ожидание выполнения запроса
[Успех]
manager.jsp, Форма изменения информации о товаре на складе: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Ожидание ввода информации
Запрос к БД UPDATE, обновление информации о механической клавиатуре
Ожидание выполнения запроса
[Успех]
manager.jsp, Форма с сообщением об успешном редактировании информации о механической клавиатуре
Ожидание действий менеджера

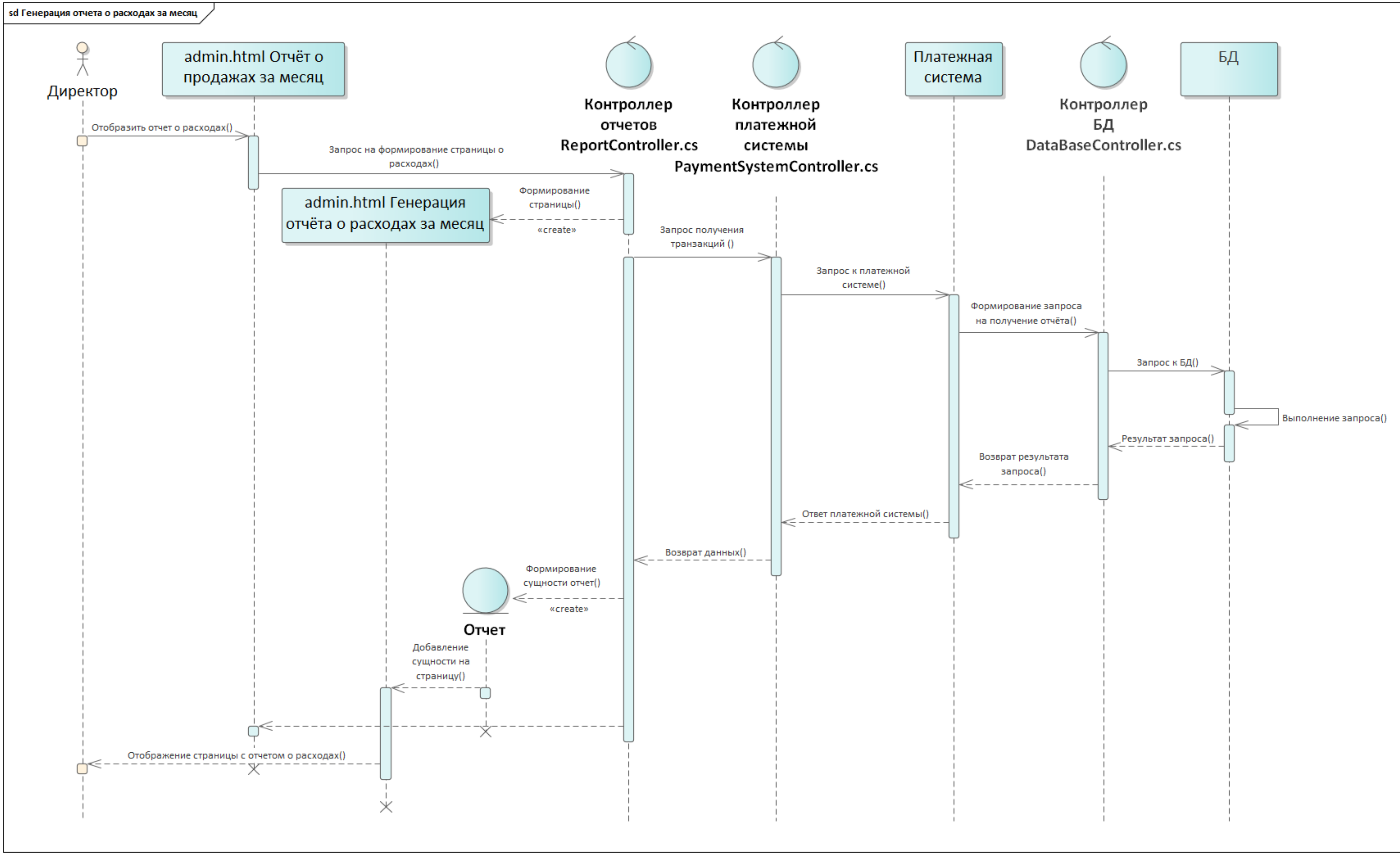

[Директор]
Запрос к БД SELECT admin.jsp, отчёт о продажах за месяц (отчёты о прибыли, расходах, состояние счёта)
Ожидание выполнения запроса
Запрос к платежной системе о транзакциях за месяц
Ожидание выполнения запроса
admin.jsp, отчёт о продажах за месяц (отчёты о прибыли, расходах, состояние счёта)
Ожидание действий директора
[Состояние счёта]
Запрос к платежной системе о состоянии счёта
Ожидание выполнения запроса
[Успех]
admin.jsp, формирование отчёта о состоянии счёта
Запрос на печать
Ожидание выполнения запроса
[Успех]
[Выход из системы]
Завершение работы в роли директора
[Выключение системы]
Выключение системы
Выход
[Отчёт о расходах]
Запрос к платежной системе о расходах за месяц
Ожидание выполнения запроса
[Успех]
admin.jsp, формирование отчёта о расходах за месяц
Запрос на печать
Ожидание выполнения запроса
[Успех]
[Отчёт о прибыли]
Запрос к платежной системе о прибыли за месяц
Ожидание выполнения запроса
[Успех]
admin.jsp, формирование отчёта о прибыли за месяц
Запрос на печать
Ожидание выполнения запроса
[Успех]
[Печать отчета]
Запрос на печать отчета
Ожидание выполнения запроса
[Успех]
[Ошибка]
Сообщение об успешной печати
Сообщение об ошибке печати
Ожидание действий директора
Ожидание действие директора

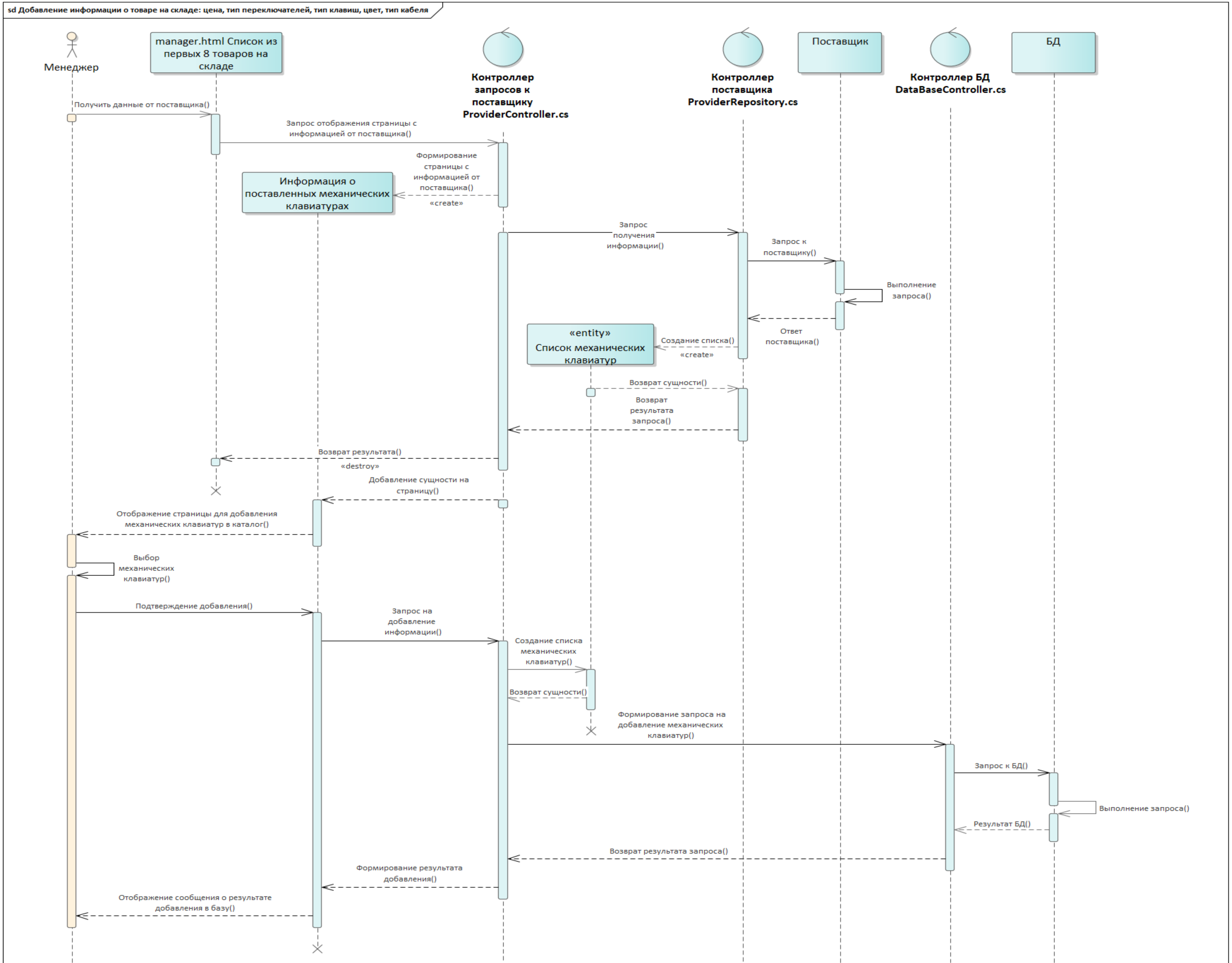

Начало
Инициализация работы системы
Запрос к БД CONNECT
Ожидание соединения с БД
БД: подключение
Ожидание подключения пользователя
Пользователь: подключение
Запрос к БД SELECT index.jsp, Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Ожидание выполнения запроса
БД: обработка запроса
index.jsp, Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Пользователь: действие
Ожидание действий пользователя
[Авторизация]
[Добавление в корзину]
[Регистрация]
Запрос к БД SELECT на страницу регистрации
Ожидание выполнения запроса
БД: обработка запроса
registration.jsp, Страница регистрации
Ожидание ввода логина,пароля, имени и электронной почты
Пользователь: ввод логина, пароля, имени и электронной почты
Формирование хэша пароля SHA-256
Запрос к БД INSERT (роль, логин, email, хэш пароля, имя) WHERE NOT EXISTS (SELECT (email))
Ожидание выполнения запроса
БД: обработка запроса
Ошибка
Успех
registration.jsp, Форма с сообщением об ошибке регистрации
registration.jsp, Форма с сообщением об успешной регистрации
Ожидание действий пользователя
Пользователь: подтверждение сообщения
Ожидание действий пользователя
Пользователь: подтверждение сообщения
Запрос к БД orders.jsp, добавление выбранной механической клавиатуры в корзину, удаление механической клавиатуры из корзины
Ожидание выполнения запроса
БД: обработка запроса
orders.jsp, добавление выбранной механической клавиатуры в корзину, удаление механической клавиатуры из корзины
Ожидание действий пользователя
Пользователь: добавление в корзину/удаление из корзины
[Добавление]
[Удаление]
Запрос к БД, добавить товар в корзину
Запрос к БД, удаление товара из корзины
Ожидание выполнения запроса
БД: обработка запроса
Ожидание выполнения запроса
БД: обработка запроса
Запрос к БД SELECT на страницу авторизации
Ожидание выполнения запроса
БД: обработка запроса
registration.jsp, Страница авторизации
Ожидание ввода логина и пароля
Пользователь: ввод логина и пароля
Формирование хэша пароля SHA256
Генерация соли и отправка клиенту
Конкатенация хэша пароля и соли
Формирование хэша результата алгоритмом SHA-256
Запрос к БД SELECT на получение хэша пароля по логину пользователя
Ожидание выполнения запроса
БД: обработка запроса
Конкатенация хэша пароля и соли
Формирование хэша SHA-256
Сравнение сгенерированного и полученного от Пользователя хэшей
Запрос к БД UPDATE с данными авторизации
Ожидание выполнения запроса
БД: обработка запроса
[Ошибка авторизации]
[Успешная авторизация]
registration.jsp, Форма с сообщениями об ошибке авторизации (неверный логин или пароль)
Определение роли пользователя
Ожидание действий пользователя
Пользователь: подтверждение сообщения

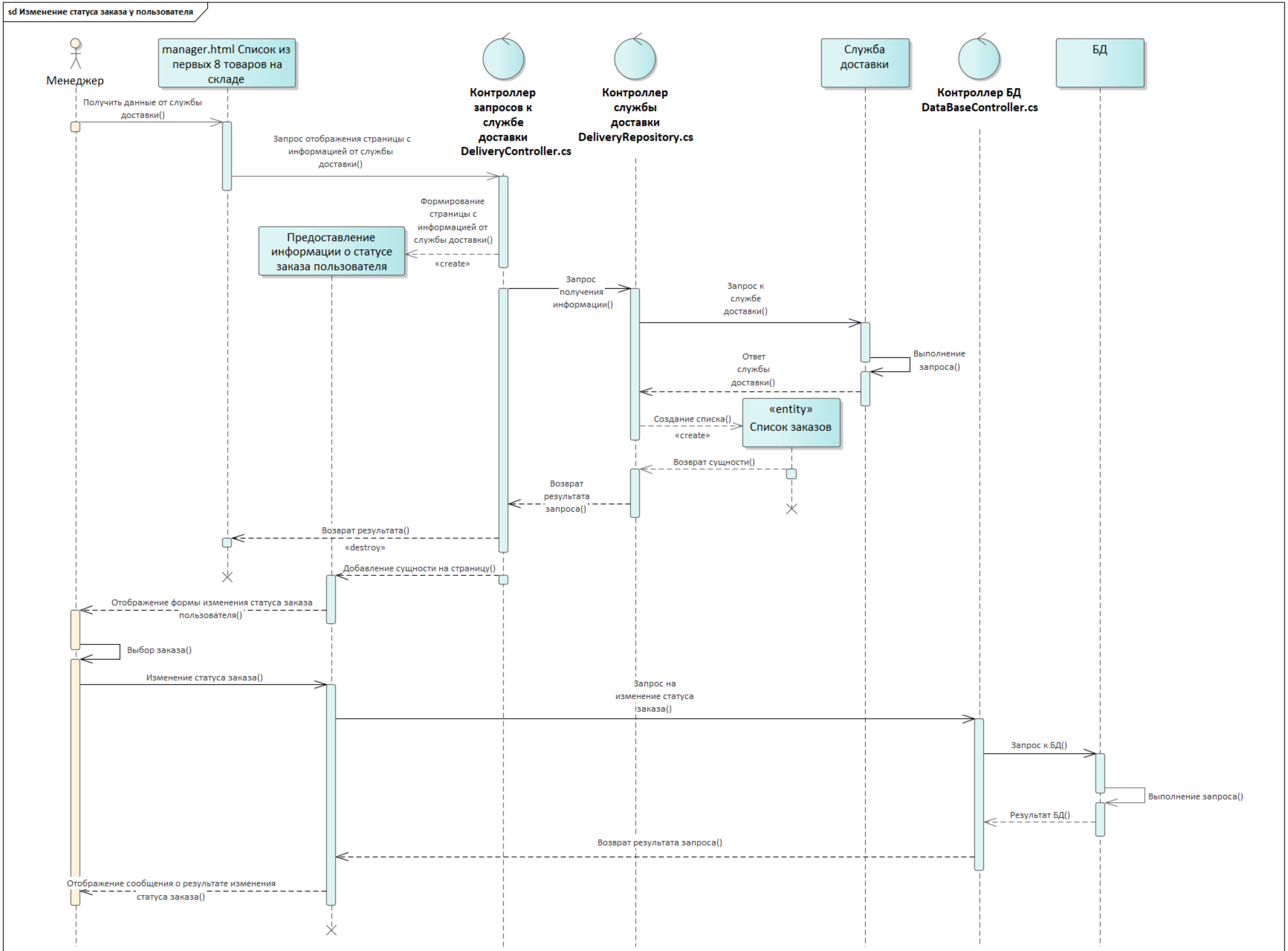

[Авторизованный пользователь]
Запрос к БД SELECT history.jsp, список последних заказов (проверка состояния заказа)
Ожидание выполнения запроса
БД: обработка запроса
history.jsp, список последних заказов (проверка состояния заказа)
Ожидание действий авторизованного пользователя
Авторизованный пользователь: действие
Завершение работы в роли авторизованного пользователя
[Выход из системы]
[Оформление заказа]
[Активные заказы]
[Поиск]
Запрос к БД SELECT список последних заказов авторизованного пользователя
Ожидание выполнения запроса
БД: обработка запроса
history.jsp, список последних заказов авторизованного пользователя
Ожидание действий авторизованного пользователя
Авторизованный пользователь: действие
history.jsp, форма поиска с полями для фильтров
Ожидание действий авторизованного пользователя
Авторизованный пользователь: ввод фильтров
Запрос к БД SELECT список доступных механических клавиатур WHERE соответствие фильтрам
Ожидание выполнения запроса
БД: обработка запроса
history.jsp, форма с результатами поиска
Ожидание действий авторизованного пользователя
Авторизованный пользователь: действие
orders.jsp, корзина с желаемыми товарами авторизованного пользователя
Ожидание действий авторизованного пользователя
Авторизованный пользователь: оформление заказа
payment.jsp, Форма оформления заказа: ввод ФИО, почтового индекса, адреса проживания, мобильного телефона, банковских данных
Ожидание действий авторизованного пользователя
Авторизованный пользователь: ввод данных
Запрос в платёжную систему
Ожидание выполнения запроса
Платёжная система: обработка запроса
Запрос к БД INSERT оформленные механические клавиатуры в последние заказы
Ожидание выполнения запроса
БД: обработка запроса
Запрос к БД UPDATE Уменьшение количества механических клавиатур в наличии
Ожидание выполнения запроса
БД: обработка запроса
orders.jsp, Страница с сообщением об успешном выполнении заказа
Ожидание действий авторизованного пользователя
Авторизованный пользователь: действие

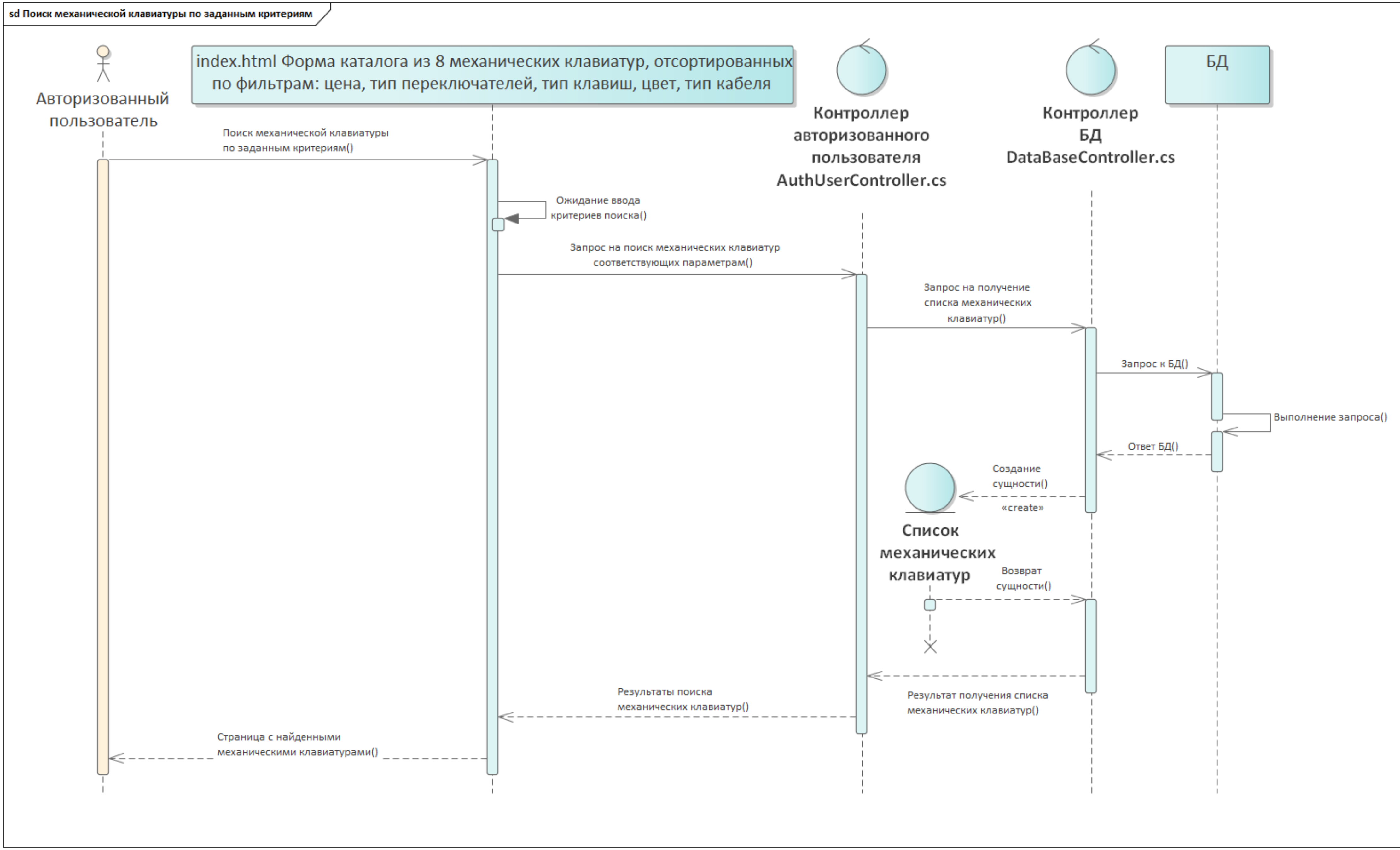

Запрос к БД SELECT manager.jsp, Форма добавления, изменения, удаления информации о товаре на складе (цена, тип переключателей, тип клавиш, цвет, тип кабеля), изменения статуса заказа пользователя
Ожидание выполнения запроса
БД: обработка запроса
manager.jsp, Форма добавления, изменения, удаления информации о товаре на складе (цена, тип переключателей, тип клавиш, цвет, тип кабеля), изменения статуса заказа пользователя
Ожидание действий менеджера
Менеджер: действие
Завершение работы в роли менеджера
[Выход из системы]
[Удаление информации о товаре]
[Изменение информации о товаре]
[Добавление информации о товаре]
[Изменение статуса заказа пользователя]
Запрос к БД DELETE FROM Список механических клавиатур на складе WHERE id механической клавиатуры = id выбранной механической клавиатуры
Ожидание выполнения запроса
БД: обработка запроса
manager.jsp, Форма с сообщением об успешном удалении информации о механической клавиатуре
Ожидание действий менеджера
Менеджер: действие
Запрос к БД SELECT manager.jsp, выбранная механическая клавиатура на складе
Ожидание выполнения запроса
БД: обработка запроса
manager.jsp, Форма изменения информации о товаре на складе: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Ожидание ввода информации
Менеджер: ввод информации о товаре
Запрос к БД UPDATE, обновление информации о механической клавиатуре
Ожидание выполнения запроса
БД: обработка запроса
manager.jsp, Форма с сообщением об успешном редактировании информации о механической клавиатуре
Ожидание действий менеджера
Менеджер: действие
Запрос к поставщику, информации о поставленных механических клавиатурах
Ожидание выполнения запроса
Поставщик: предоставление информации о поставленных механических клавиатурах
manager.jsp, Форма добавления информации о товаре на складе: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Ожидание ввода информации
Менеджер: ввод информации о товаре
Запрос к БД INSERT добавление механический клавиатуры
Ожидание выполнения запроса
БД: обработка запроса
manager.jsp, Форма с сообщением об успешном добавлении механической клавиатуры
Ожидание действий менеджера
Менеджер: действие
Запрос к БД SELECT выбранный заказ пользователя
Ожидание выполнения запроса
БД: обработка запроса
Запрос к службе доставки о статусе заказа пользователя
Ожидание выполнения запроса
Служба доставки: предоставление информации о заказе
manager.jsp, Форма изменения статуса заказа пользователя
Ожидание ввода информации
Менеджер: ввод статуса заказа
Запрос к БД UPDATE изменение статуса заказа пользователя
Ожидание выполнения запроса
БД: обработка запроса
manager.jsp, Форма с сообщением об успешном изменении статуса заказа
Ожидание действий менеджера
Менеджер: действие

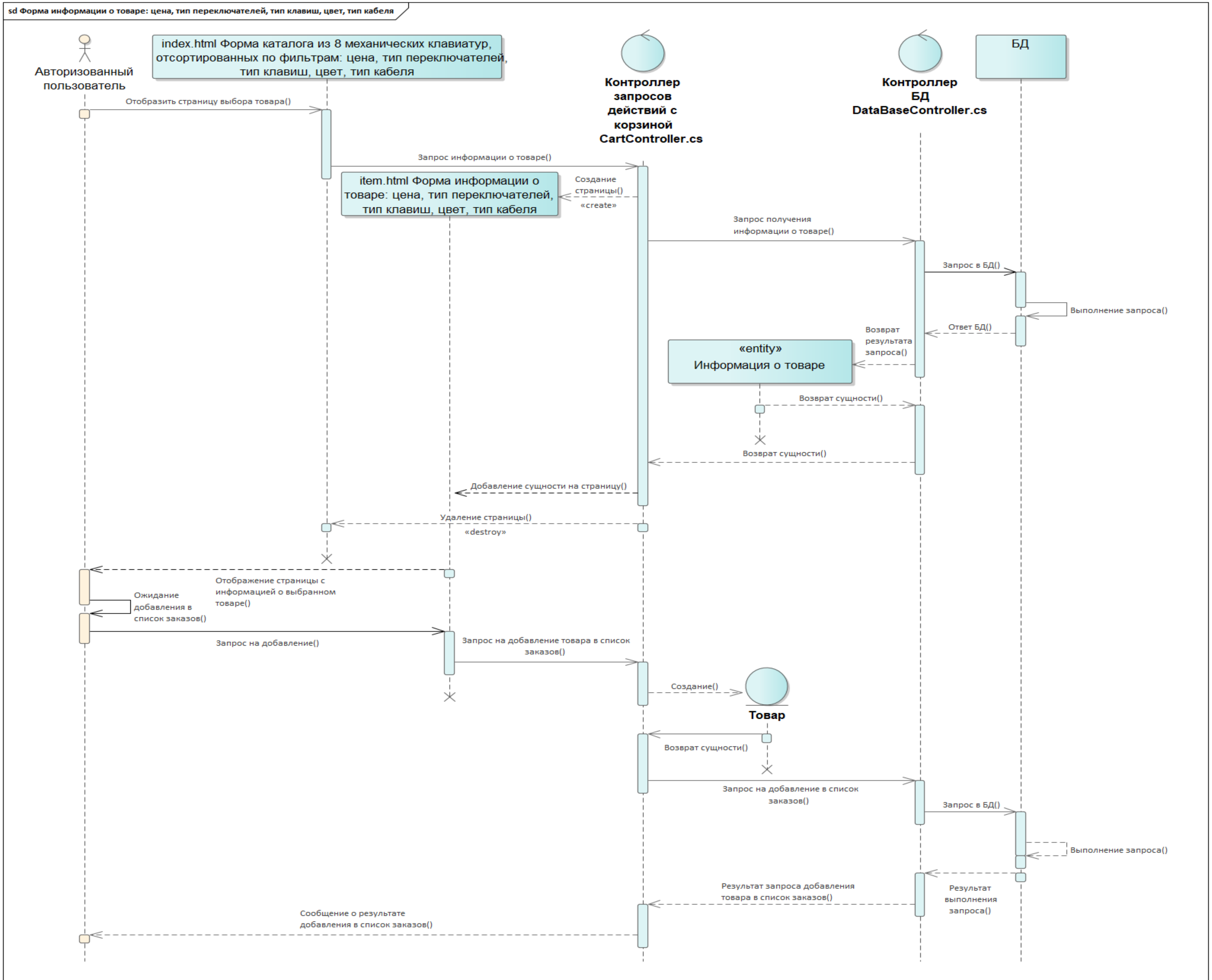

Запрос к БД SELECT admin.jsp, отчёт о продажах за месяц (отчёты о прибыли, расходах, состояние счёта)
Ожидание выполнения запроса
БД: обработка запроса
Запрос к платежной системе о транзакциях за месяц
Ожидание выполнения запроса
Платежная система: обработка запроса
admin.jsp, отчёт о продажах за месяц (отчёты о прибыли, расходах, состояние счёта)
Ожидание действий директора
Директор: действие
Завершение работы в роли директора
[Выход из системы]
[Выключение системы]
Выключение системы
Выход
[Отчёт о расходах]
[Отчёт о прибыли]
[Состояние счёта]
[Печать отчета]
Запрос к платежной системе о состоянии счёта
Ожидание выполнения запроса
Платежная система: обработка запроса
admin.jsp, формирование отчёта о состоянии счёта
Запрос на печать
Ожидание выполнения запроса
Принтер: печать
Запрос к платежной системе о расходах за месяц
Ожидание выполнения запроса
Платежная система: обработка запроса
admin.jsp, формирование отчёта о расходах за месяц
Запрос на печать
Ожидание выполнения запроса
Принтер: печать
Запрос к платежной системе о прибыли за месяц
Ожидание выполнения запроса
Платежная система: обработка запроса
admin.jsp, формирование отчёта о прибыли за месяц
Запрос на печать
Ожидание выполнения запроса
Принтер: печать
Запрос на печать отчета
Ожидание выполнения запроса
Принтер: печать
[Успех]
[Ошибка]
Сообщение об успешной печати
Сообщение об ошибке печати
Ожидание действий директора
Директор: действие
Ожидание действие директора
Директор: действие

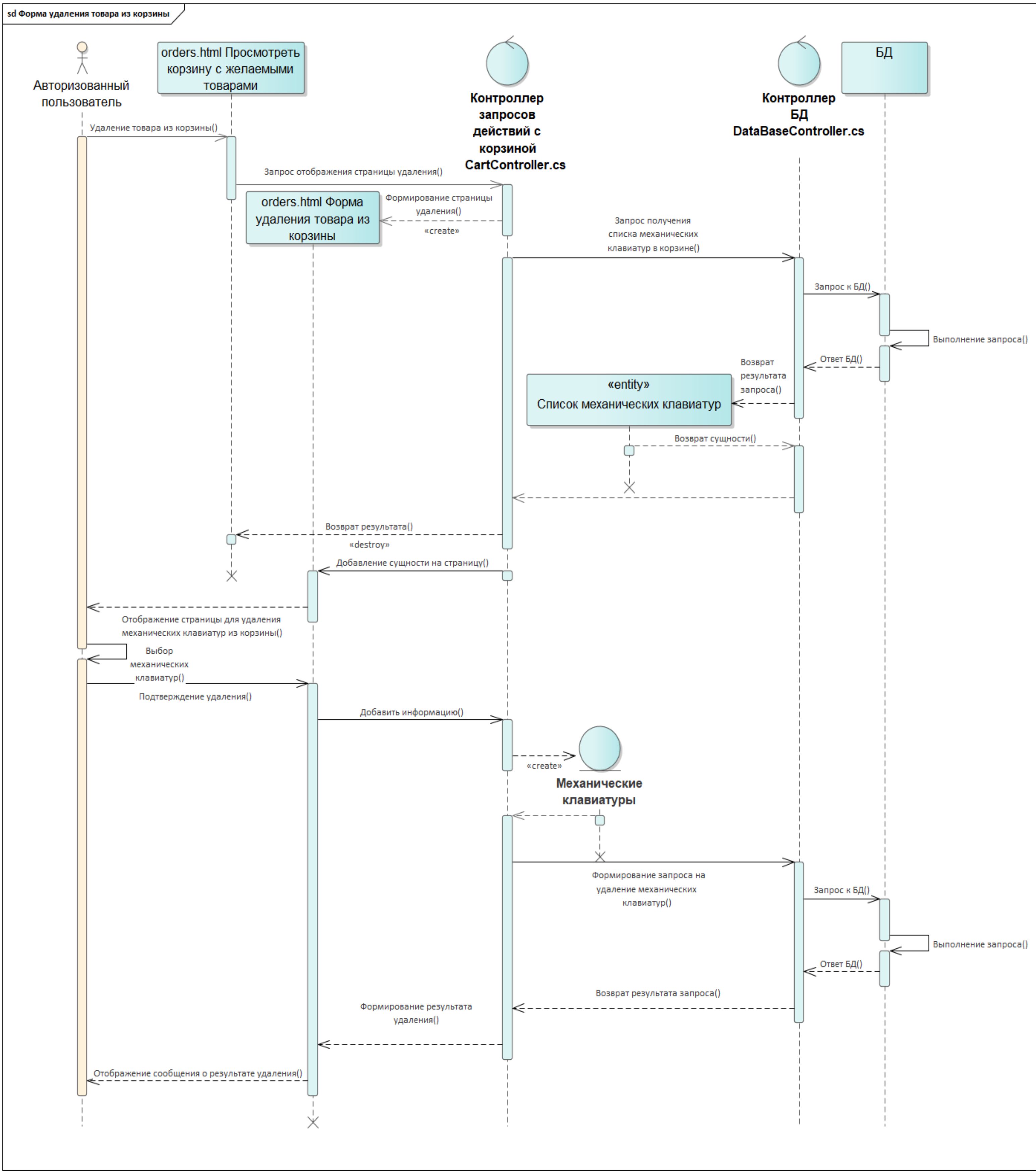

sd Генерация отчета о расходах за месяц
Директор
admin.html Отчёт о продажах за месяц
Контроллер отчетов ReportController.cs
Контроллер платежной системы PaymentSystemController.cs
Платежная система
Контроллер БД DataBaseController.cs
БД
Отобразить отчет о расходах()
Запрос на формирование страницы о расходах()
admin.html Генерация отчёта о расходах за месяц
Формирование страницы()
«create»
Запрос получения транзакций ()
Запрос к платежной системе()
Формирование запроса на получение отчёта()
Запрос к БД()
Выполнение запроса()
Результат запроса()
Возврат результата запроса()
Ответ платежной системы()
Возврат данных()
Формирование сущности отчет()
«create»
Отчет
Добавление сущности на страницу()
Отображение страницы с отчетом о расходах()

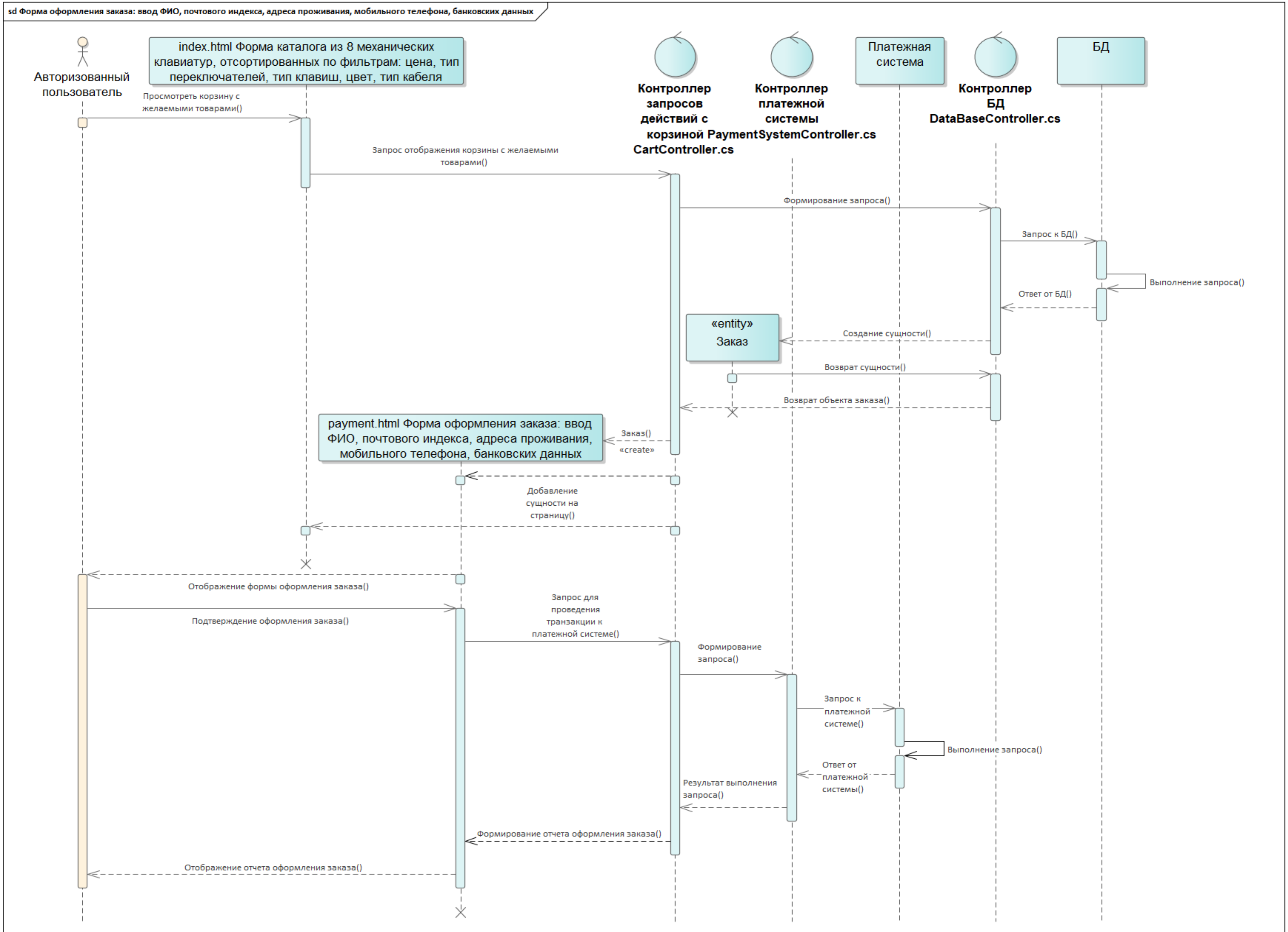

sd Добавление информации о товаре на складе: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Менеджер
manager.html Список из первых 8 товаров на складе
Контроллер запросов к поставщику ProviderController.cs
Контроллер поставщика ProviderRepository.cs
Поставщик
Контроллер БД DataBaseController.cs
БД
Получить данные от поставщика()
Запрос отображения страницы с информацией от поставщика()
Формирование страницы с информацией от поставщика()
«create»
Информация о поставленных механических клавиатурах
Запрос получения информации()
Запрос к поставщику()
Выполнение запроса()
Ответ поставщика()
Создание списка()
«create»
«entity» Список механических клавиатур
Возврат сущности()
Возврат результата запроса()
Возврат результата()
«destroy»
Добавление сущности на страницу()
Отображение страницы для добавления механических клавиатур в каталог()
Выбор механических клавиатур()
Подтверждение добавления()
Запрос на добавление информации()
Создание списка механических клавиатур()
Возврат сущности()
Формирование запроса на добавление механических клавиатур()
Запрос к БД()
Выполнение запроса()
Результат БД()
Возврат результата запроса()
Формирование результата добавления()
Отображение сообщения о результате добавления в базу()

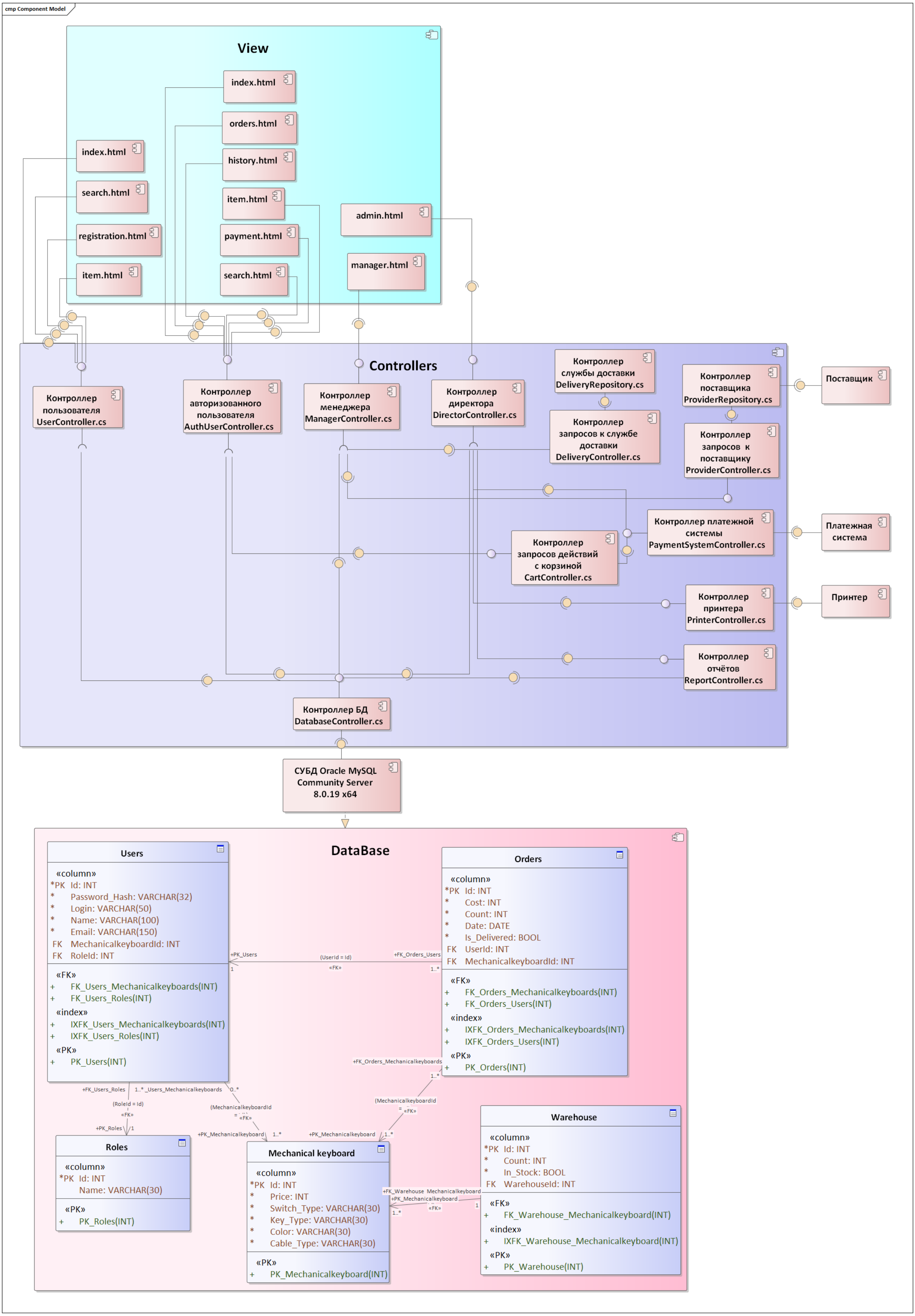

sd Изменение статуса заказа у пользователя
Менеджер
manager.html Список из первых 8 товаров на складе
Контроллер запросов к службе доставки DeliveryController.cs
Контроллер службы доставки DeliveryRepository.cs
Служба доставки
Контроллер БД DataBaseController.cs
БД
Получить данные от службы доставки()
Запрос отображения страницы с информацией от службы доставки()
Формирование страницы с информацией от службы доставки()
«create»
Предоставление информации о статусе заказа пользователя
Запрос получения информации()
Запрос к службе доставки()
Выполнение запроса()
Ответ службы доставки()
«entity»
Список заказов
Создание списка()
«create»
Возврат сущности()
Возврат результата запроса()
Возврат результата()
«destroy»
Добавление сущности на страницу()
Отображение формы изменения статуса заказа пользователя()
Выбор заказа()
Изменение статуса заказа()
Запрос на изменение статуса заказа()
Запрос к БД()
Выполнение запроса()
Результат БД()
Возврат результата запроса()
Отображение сообщения о результате изменения статуса заказа()

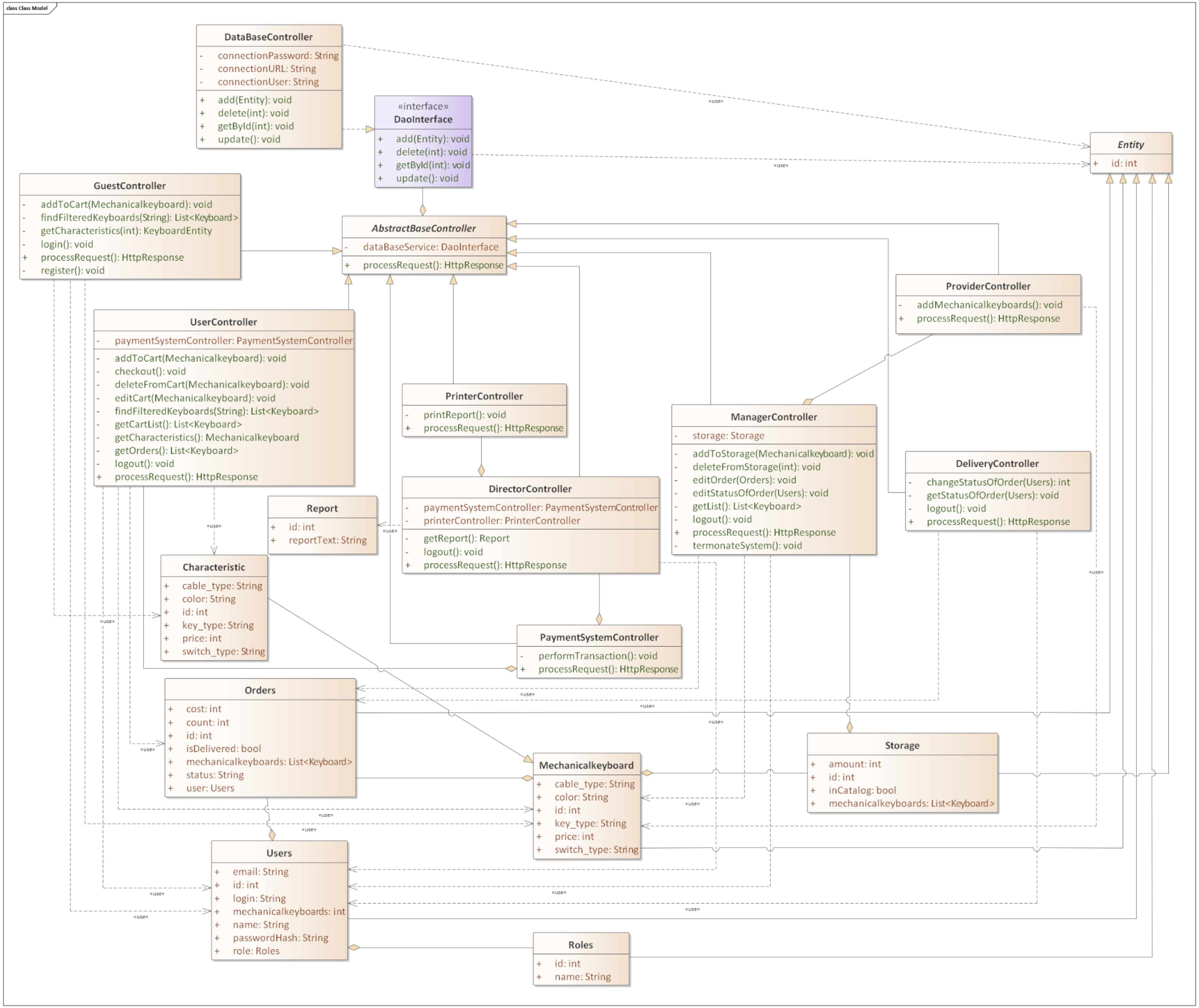

sd Поиск механической клавиатуры по заданным критериям
Авторизованный пользователь
index.html Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Контроллер авторизованного пользователя AuthUserController.cs
Контроллер БД DataBaseController.cs
БД
Поиск механической клавиатуры по заданным критериям()
Ожидание ввода критериев поиска()
Запрос на поиск механических клавиатур соответствующих параметрам()
Запрос на получение списка механических клавиатур()
Запрос к БД()
Выполнение запроса()
Ответ БД()
Создание сущности()
«create»
Список механических клавиатур
Возврат сущности()
Результат получения списка механических клавиатур()
Результаты поиска механических клавиатур()
Страница с найденными механическими клавиатурами()


sd Форма информации о товаре: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Авторизованный пользователь
index.html Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Контроллер запросов действий с корзиной CartController.cs
Контроллер БД DataBaseController.cs
БД
Отобразить страницу выбора товара()
Запрос информации о товаре()
item.html Форма информации о товаре: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Создание страницы()
«create»
Запрос получения информации о товаре()
Запрос в БД()
Выполнение запроса()
Ответ БД()
Возврат результата запроса()
«entity»
Информация о товаре
Возврат сущности()
Возврат сущности()
Добавление сущности на страницу()
Удаление страницы()
«destroy»
Отображение страницы с информацией о выбранном товаре()
Ожидание добавления в список заказов()
Запрос на добавление()
Запрос на добавление товара в список заказов()
Создание()
Товар
Возврат сущности()
Запрос на добавление в список заказов()
Запрос в БД()
Выполнение запроса()
Результат выполнения запроса()
Результат запроса добавления товара в список заказов()
Сообщение о результате добавления в список заказов()


sd Форма удаления товара из корзины
Авторизованный пользователь
orders.html Просмотреть корзину с желаемыми товарами
Контроллер запросов действий с корзиной CartController.cs
Контроллер БД DataBaseController.cs
БД
Удаление товара из корзины()
Запрос отображения страницы удаления()
orders.html Форма удаления товара из корзины
Формирование страницы удаления()
«create»
Запрос получения списка механических клавиатур в корзине()
Запрос к БД()
Выполнение запроса()
Ответ БД()
Возврат результата запроса()
«entity»
Список механических клавиатур
Возврат сущности()
Возврат результата()
«destroy»
Добавление сущности на страницу()
Отображение страницы для удаления механических клавиатур из корзины()
Выбор механических клавиатур()
Подтверждение удаления()
Добавить информацию()
«create»
Механические клавиатуры
Формирование запроса на удаление механических клавиатур()
Запрос к БД()
Выполнение запроса()
Ответ БД()
Возврат результата запроса()
Формирование результата удаления()
Отображение сообщения о результате удаления()


sd Форма оформления заказа: ввод ФИО, почтового индекса, адреса проживания, мобильного телефона, банковских данных
Авторизованный пользователь
index.html Форма каталога из 8 механических клавиатур, отсортированных по фильтрам: цена, тип переключателей, тип клавиш, цвет, тип кабеля
Контроллер запросов действий с корзиной CartController.cs
Контроллер платежной системы PaymentSystemController.cs
Платежная система
Контроллер БД DataBaseController.cs
БД
Просмотреть корзину с желаемыми товарами()
Запрос отображения корзины с желаемыми товарами()
Формирование запроса()
Запрос к БД()
Выполнение запроса()
Ответ от БД()
«entity» Заказ
Создание сущности()
Возврат сущности()
Возврат объекта заказа()
payment.html Форма оформления заказа: ввод ФИО, почтового индекса, адреса проживания, мобильного телефона, банковских данных
Заказ()
«create»
Добавление сущности на страницу()
Отображение формы оформления заказа()
Подтверждение оформления заказа()
Запрос для проведения транзакции к платежной системе()
Формирование запроса()
Запрос к платежной системе()
Выполнение запроса()
Ответ от платежной системы()
Результат выполнения запроса()
Формирование отчета оформления заказа()
Отображение отчета оформления заказа()


cmp Component Model
View
index.html
orders.html
history.html
item.html
payment.html
search.html
index.html
search.html
registration.html
item.html
admin.html
manager.html
Controllers
Контроллер пользователя UserController.cs
Контроллер авторизованного пользователя AuthUserController.cs
Контроллер менеджера ManagerController.cs
Контроллер директора DirectorController.cs
Контроллер службы доставки DeliveryRepository.cs
Контроллер запросов к службе доставки DeliveryController.cs
Контроллер поставщика ProviderRepository.cs
Контроллер запросов к поставщику ProviderController.cs
Поставщик
Контроллер платежной системы PaymentSystemController.cs
Платежная система
Контроллер запросов действий с корзиной CartController.cs
Контроллер принтера PrinterController.cs
Принтер
Контроллер отчётов ReportController.cs
Контроллер БД DatabaseController.cs
СУБД Oracle MySQL Community Server 8.0.19 x64
DataBase
Users
«column»
*PK Id: INT
* Password_Hash: VARCHAR(32)
* Login: VARCHAR(50)
* Name: VARCHAR(100)
* Email: VARCHAR(150)
FK MechanicalkeyboardId: INT
FK RoleId: INT
«FK»
+ FK_Users_Mechanicalkeyboards(INT)
+ FK_Users_Roles(INT)
«index»
+ IXFK_Users_Mechanicalkeyboards(INT)
+ IXFK_Users_Roles(INT)
«PK»
+ PK_Users(INT)
Orders
«column»
*PK Id: INT
* Cost: INT
* Count: INT
* Date: DATE
* Is_Delivered: BOOL
FK UserId: INT
FK MechanicalkeyboardId: INT
«FK»
+ FK_Orders_Mechanicalkeyboards(INT)
+ FK_Orders_Users(INT)
«index»
+ IXFK_Orders_Mechanicalkeyboards(INT)
+ IXFK_Orders_Users(INT)
«PK»
+ PK_Orders(INT)
+PK_Users
(UserId = Id)
«FK»
+FK_Orders_Users
1
1..*
+FK_Orders_Mechanicalkeyboards
1..*
(MechanicalkeyboardId = Id)
«FK»
+PK_Mechanicalkeyboard
1..*
+FK_Users_Roles
1..*
+_Users_Mechanicalkeyboards
0..*
(RoleId = Id)
«FK»
+PK_Roles
1
(MechanicalkeyboardId = Id)
«FK»
+PK_Mechanicalkeyboard
1..*
Roles
«column»
*PK Id: INT
Name: VARCHAR(30)
«PK»
+ PK_Roles(INT)
Mechanical keyboard
«column»
*PK Id: INT
* Price: INT
* Switch_Type: VARCHAR(30)
* Key_Type: VARCHAR(30)
* Color: VARCHAR(30)
* Cable_Type: VARCHAR(30)
«PK»
+ PK_Mechanicalkeyboard(INT)
+FK_Warehouse_Mechanicalkeyboard
+PK_Mechanicalkeyboard
«FK»
1..*
1
Warehouse
«column»
*PK Id: INT
* Count: INT
* In_Stock: BOOL
FK WarehouseId: INT
«FK»
+ FK_Warehouse_Mechanicalkeyboard(INT)
«index»
+ IXFK_Warehouse_Mechanicalkeyboard(INT)
«PK»
+ PK_Warehouse(INT)


class Class Model
DataBaseController
- connectionPassword: String
- connectionURL: String
- connectionUser: String
+ add(Entity): void
+ delete(int): void
+ getById(int): void
+ update(): void
«interface»
DaoInterface
+ add(Entity): void
+ delete(int): void
+ getById(int): void
+ update(): void
«use»
«use»
Entity
+ id: int
GuestController
- addToCart(Mechanicalkeyboard): void
- findFilteredKeyboards(String): List<Keyboard>
- getCharacteristics(int): KeyboardEntity
- login(): void
+ processRequest(): HttpResponse
- register(): void
AbstractBaseController
- dataBaseService: DaoInterface
+ processRequest(): HttpResponse
ProviderController
- addMechanicalkeyboards(): void
+ processRequest(): HttpResponse
UserController
- paymentSystemController: PaymentSystemController
- addToCart(Mechanicalkeyboard): void
- checkout(): void
- deleteFromCart(Mechanicalkeyboard): void
- editCart(Mechanicalkeyboard): void
- findFilteredKeyboards(String): List<Keyboard>
- getCartList(): List<Keyboard>
- getCharacteristics(): Mechanicalkeyboard
- getOrders(): List<Keyboard>
- logout(): void
+ processRequest(): HttpResponse
PrinterController
- printReport(): void
+ processRequest(): HttpResponse
ManagerController
- storage: Storage
- addToStorage(Mechanicalkeyboard): void
- deleteFromStorage(int): void
- editOrder(Orders): void
- editStatusOfOrder(Users): void
- getList(): List<Keyboard>
- logout(): void
+ processRequest(): HttpResponse
- termonateSystem(): void
DeliveryController
- changeStatusOfOrder(Users): int
- getStatusOfOrder(Users): void
- logout(): void
+ processRequest(): HttpResponse
Report
+ id: int
+ reportText: String
«use»
«use»
DirectorController
- paymentSystemController: PaymentSystemController
- printerController: PrinterController
- getReport(): Report
- logout(): void
+ processRequest(): HttpResponse
Characteristic
+ cable_type: String
+ color: String
+ id: int
+ key_type: String
+ price: int
+ switch_type: String
«use»
«use»
PaymentSystemController
- performTransaction(): void
+ processRequest(): HttpResponse
Orders
+ cost: int
+ count: int
+ id: int
+ isDelivered: bool
+ mechanicalkeyboards: List<Keyboard>
+ status: String
+ user: Users
«use»
«use»
«use»
«use»
Storage
+ amount: int
+ id: int
+ inCatalog: bool
+ mechanicalkeyboards: List<Keyboard>
Mechanicalkeyboard
+ cable_type: String
+ color: String
+ id: int
+ key_type: String
+ price: int
+ switch_type: String
«use»
«use»
«use»
Users
+ email: String
+ id: int
+ login: String
+ mechanicalkeyboards: int
+ name: String
+ passwordHash: String
+ role: Roles
«use»
«use»
«use»
«use»
Roles
+ id: int
+ name: String

Users.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
public class Users : Entity {

	public String email;
	public int id;
	public String login;
	public int mechanicalkeyboards;
	public String name;
	public String passwordHash;
	public Roles role;
	public Roles m_Roles;
	public Orders m_Orders;

	public Users(){

	}

	~Users(){

	}

}//end Users
```

UserController.cs

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



public class UserController : AbstractBaseController {

	private PaymentSystemController paymentSystemController;

	public UserController(){

	}

	~UserController(){

	}

	///
	/// <param name="keyboard"></param>
	private void addToCart(Mechanicalkeyboard keyboard){

	}

	private void checkout(){

	}

	///
	/// <param name="keyboard"></param>
	private void deleteFromCart(Mechanicalkeyboard keyboard){

	}

	///
	/// <param name="keyboard"></param>
	private void editCart(Mechanicalkeyboard keyboard){

	}
```

```
    ///
    /// <param name="filter"></param>
    private List<Keyboard> findFilteredKeyboards(String filter){

        return null;
    }

    private List<Keyboard> getCartList(){

        return null;
    }

    private Mechanicalkeyboard getCharacteristics(){

        return null;
    }

    private List<Keyboard> getOrders(){

        return null;
    }

    private void logout(){

    }

    public override HttpResponse processRequest(){

        return null;
    }

}//end UserController
```

Storage.cs

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



public class Storage : Entity {

    public int amount;
    public int id;
    public bool inCatalog;
    public List<Keyboard> mechanicalkeyboards;
    public ManagerController m_ManagerController;

    public Storage(){

    }

    ~Storage(){

    }

}//end Storage
```

Roles.cs

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

```
public class Roles : Entity {

	public int id;
	public String name;

	public Roles(){

	}

	~Roles(){

	}

}//end Roles
```

Report.cs

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



public class Report {

	public int id;
	public String reportText;

	public Report(){

	}

	~Report(){

	}

}//end Report
```

ProviderController.cs

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



public class ProviderController : AbstractBaseController {

	public ProviderController(){

	}

	~ProviderController(){

	}

	private void addMechanicalkeyboards(){

	}

	public override HttpResponse processRequest(){

		return null;
	}

}//end ProviderController
```

PrinterController.cs

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



public class PrinterController : AbstractBaseController {

      public PrinterController(){

      }

      ~PrinterController(){

      }

      private void printReport(){

      }

      public override HttpResponse processRequest(){

            return null;
      }

}//end PrinterController
```

PaymentSystemController.cs

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



public class PaymentSystemController : AbstractBaseController {

      public UserController m_UserController;
      public DirectorController m_DirectorController;

      public PaymentSystemController(){

      }

      ~PaymentSystemController(){

      }

      private void performTransaction(){

      }

      public HttpResponse processRequest(){

            return null;
      }

}//end PaymentSystemController
```

Orders.cs

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

```
public class Orders : Entity {

	public int cost;
	public int count;
	public int id;
	public bool isDelivered;
	public List<Keyboard> mechanicalkeyboards;
	public String status;
	public Users user;

	public Orders(){

	}

	~Orders(){

	}

}//end Orders
```

Mechanicalkeyboard.cs

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



public class Mechanicalkeyboard : Entity {

	public String cable_type;
	public String color;
	public int id;
	public String key_type;
	public int price;
	public String switch_type;
	public Orders m_Orders;
	public Storage m_Storage;

	public Mechanicalkeyboard(){

	}

	~Mechanicalkeyboard(){

	}

	/* Begin - EA generated code for Parts and Ports */

	public string Port1 ;

	/* End - EA generated code for Parts and Ports */

}//end Mechanicalkeyboard
```

ManagerController.cs

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



public class ManagerController : AbstractBaseController {

	public class PaymentController {
        public PaymentController(){

        }

        ~PaymentController(){

        }

        public void GetPaymentReport(){

        }

    }//end PaymentController

    private Storage storage;
    public ProducerController m_ProducerController;
    public ProviderController m_ProviderController;

    public ManagerController(){

    }

    ~ManagerController(){

    }

    ///
    /// <param name="keyboard"></param>
    private void addToStorage(Mechanicalkeyboard keyboard){

    }

    ///
    /// <param name="id"></param>
    private void deleteFromStorage(int id){

    }

    ///
    /// <param name="order"></param>
    private void editOrder(Orders order){

    }

    ///
    /// <param name="user"></param>
    private void editStatusOfOrder(Users user){

    }

    private List<Keyboard> getList(){

        return null;
    }

    private void logout(){

    }

    public HttpResponse processRequest(){

        return null;
    }

    private void terminateSystem(){

    }

}//end ManagerController
```

GuestController.cs

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



public class GuestController : AbstractBaseController {

	public GuestController(){

	}

	~GuestController(){

	}

	///
	/// <param name="keyboard"></param>
	private void addToCart(Mechanicalkeyboard keyboard){

	}

	///
	/// <param name="filter"></param>
	private List<Keyboard> findFilteredKeyboards(String filter){

		return null;
	}

	///
	/// <param name="id"></param>
	private KeyboardEntity getCharacteristics(int id){

		return null;
	}

	private void login(){

	}

	public override HttpResponse processRequest(){

		return null;
	}

	private void register(){

	}

}//end GuestController
```

Entity.cs

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



public abstract class Entity {

	public int id;

	public Entity(){

	}
	~Entity(){

	}

}//end Entity
```

DirectorController.cs

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



public class DirectorController : AbstractBaseController {

	private PaymentSystemController paymentSystemController;
	private PrinterController printerController;
	public PrinterController m_PrinterController;

	public DirectorController(){

	}

	~DirectorController(){

	}

	private Report getReport(){

		return null;
	}

	private void logout(){

	}

	public override HttpResponse processRequest(){

		return null;
	}

}//end DirectorController
```

DeliveryController.cs

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



public class DeliveryController : AbstractBaseController {

	public DeliveryController(){

	}

	~DeliveryController(){

	}

	///
	/// <param name="user"></param>
	private int changeStatusOfOrder(Users user){

		return 0;
```

```
        }

        ///
        /// <param name="user"></param>
        private void getStatusOfOrder(Users user){

        }

        private void logout(){

        }

        public override HttpResponse processRequest(){

                return null;
        }

}//end DeliveryController
```

DataBaseController.cs

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



public class DataBaseController : DaoInterface {

        private String connectionPassword;
        private String connectionURL;
        private String connectionUser;

        public DataBaseController(){

        }

        ~DataBaseController(){

        }

        ///
        /// <param name="entity"></param>
        public void add(Entity entity){

        }

        ///
        /// <param name="id"></param>
        public void delete(int id){

        }

        ///
        /// <param name="id"></param>
        public void getById(int id){

        }

        public void update(){

        }

}//end DataBaseController
```

DaoInterface.cs

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



public interface DaoInterface  {

	///
	/// <param name="entity"></param>
	void add(Entity entity);

	///
	/// <param name="id"></param>
	void delete(int id);

	///
	/// <param name="id"></param>
	void getById(int id);

	void update();
}//end DaoInterface
```

Characteristic.cs

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



public class Characteristic : Mechanicalkeyboard {

	public String cable_type;
	public String color;
	public int id;
	public String key_type;
	public int price;
	public String switch_type;

	public Characteristic(){

	}

	~Characteristic(){

	}

}//end Characteristic
```

AbstractBaseController.cs

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



public abstract class AbstractBaseController {

	private DaoInterface dataBaseService;
	public DaoInterface m_DaoInterface;

	public AbstractBaseController(){

	}

	~AbstractBaseController(){
	}

	public HttpResponse processRequest(){

		return null;
	}

}//end AbstractBaseController
```