

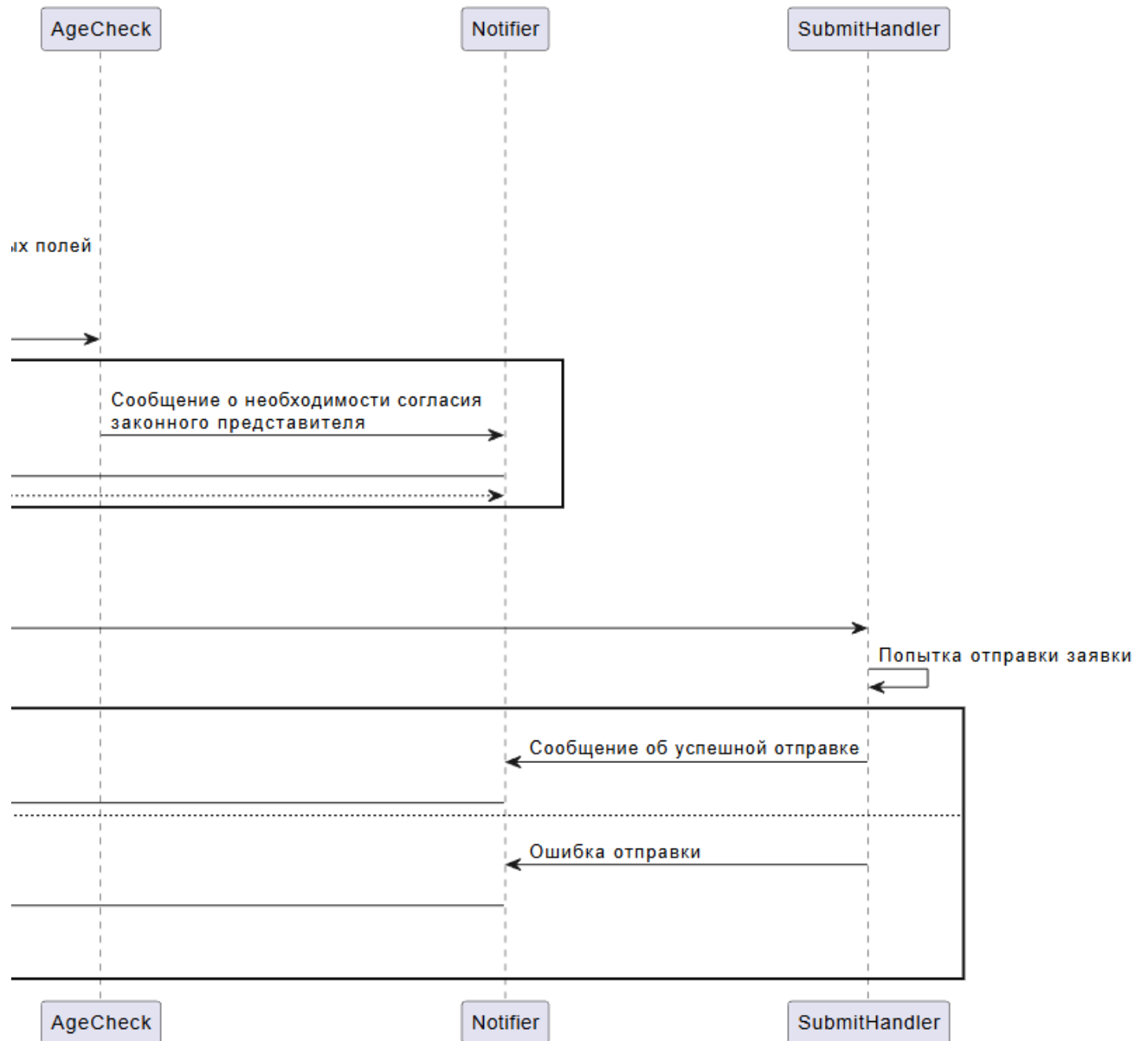


Рисунок 4.2 — продолжение диаграммы взаимодействия

Источник: собственная разработка

4.3 Диаграмма состояний

Сценарии вариантов использования и диаграммы взаимодействия описывают поведение информационного ресурса в целом, показывая взаимодействие его объектов. При проектировании сайта иногда возникает необходимость детально изучить поведение отдельных сущностей — например, жизненный цикл формы заявки. Для таких целей применяются диаграммы состояний.

Диаграмма состояний представляет собой ориентированный граф специального вида, моделирующий поведение некоторого автомата. Вершинами такого графа являются состояния сущности (стадии её существования), а дуги обозначают переходы между состояниями, возникающие под действием событий или по завершении операций. Диаграммы состояний могут быть вложены, что позволяет детализировать поведение отдельных подсистем или компонентов. Для корректного построения диаграммы состояний важно понимать особенности моделируемой сущности и базовые положения теории автоматов.

Диаграммы состояний целесообразно строить для тех классов или компонентов системы, поведение которых необходимо изучить — в рассматриваемом проекте это, прежде всего, интерактивная форма заявки (она же «форма оформления заявки»). Состояние может содержать только имя или имя и спецификацию внутренних действий:

- **entry** — действие при входе в состояние (выполняется независимо от того, по какому переходу осуществлён вход);
- **exit** — действие при выходе из состояния;
- **do** — длительная деятельность, выполняемая в состоянии (например, автосохранение, валидация, ожидание события).

Ниже представлены перечень состояний формы заявки (табл. 4.3) и спецификация переходов между этими состояниями (табл. 4.4), адаптированные под предметную область — сайт тату-мастера.

Таблица 4.3 – Перечень состояний

Состояние	Характеристика состояния
Открыта	entry/open — форма отрисована в интерфейсе, пользователь может начать ввод данных.
Заполнена данными	do/add — пользователь ввёл данные о тату-идее и контактную информацию; данные находятся в форме (не сохранены на сервере).
Проверена	do/check — выполнена проверка корректности введённых полей (валидность e-mail/телефона, обязательные поля заполнены, возраст подтверждён и т.д.).

Скорректирована	do/change — после обнаружения ошибок пользователь внёс изменения в форму.
Отправлена	do/send — инициирована процедура отправки заявки (имитация отправки на клиенте или отправка через внешнее API/мессенджер).
Подтверждена (успех)	entry/success — система отобразила уведомление об успешной отправке; форма может быть очищена или сохранена локально.
Ошибка отправки	entry/error — при попытке отправки возникла ошибка (нет соединения, сбой API и т.п.); пользователь информирован.
Сохранена локально	do/save — при необходимости данные временно сохраняются в LocalStorage (опция); состояние означает, что данные сохранены на стороне клиента.
Удалена	exit/delete — форма удалена (пользователь отменил заполнение или очистил поля); удаление может быть выполнено на любой стадии.
Закрыта	exit/close — форма свернута или закрыта пользователем; при закрытии может происходить сохранение/удаление временных данных.

Источник: собственная разработка

Таблица 4.4 – Спецификация переходов

Выход (Новое состояние)	Вход (Предыдущее состояние)	Название перехода / Условие
Заполнена данными	Открыта	Пользователь вводит данные в поля формы (информация о тату-идее, референсы, контактные данные).
Проверена	Заполнена данными	После заполнения форма запускает валидацию полей (клиентская проверка JavaScript).
Сохранена локально	Проверена / Скорректирована	Если данные корректны и пользователь выбрал опцию сохранения, данные записываются в LocalStorage; либо после корректировки форма сохраняется локально.

Отправлена	Проверена	Пользователь нажимает кнопку «Отправить»; инициируется процесс отправки заявки (имитация отправки или вызов внешнего API).
Подтверждена (успех)	Отправлена	Если отправка прошла успешно (получен ответ об успешной доставке или сообщение успешно сформировано), система показывает уведомление об успешной отправке.
Ошибка отправки	Отправлена	При отсутствии соединения или сбое API отправка не завершается успешно; система формирует уведомление об ошибке.
Скорректирована	Проверена	Если проверка выявила некорректные поля, пользователь вносит правки → состояние «Скорректирована».

Диаграмму состояний можно видеть на рисунке 4.3

