# ДИАГРАММА USE CASE

Составлена диаграмма прецедентов (use case diagram).

На данной диаграмме отображены все возможные действия внешних актёров. Актёрами являются роли пользователей: оператор и администратор.

## Оператор

Оператору доступны следующие действия:

1. Отправка сообщения;
2. Просмотр истории сообщений.

При отправке сообщений оператор должен заполнить поле «Получатели». **Дополнительно** он может выбрать сервис доставки, время доставки или период, во время которого необходимо доставить сообщение. Также он может прикрепить к сообщению файлы. Из диаграммы видно, что ввод темы сообщения, а также ввод текста *опциональны*.

Также оператору доступен просмотр сообщений в своём журнале сообщений. Здесь оператор может проверить статус доставки, чтобы убедиться, что сообщение доставлено, и просмотреть время доставки.

## Администратор

Администратор наследуется от оператора, т.е. ему доступен весь доступный оператору функционал.

Отличие администратора от оператора заключается в возможности просмотра списка операторов, где администратор может добавить или удалить оператора.

Схема диаграммы Use Case изображена на Рис. 1.

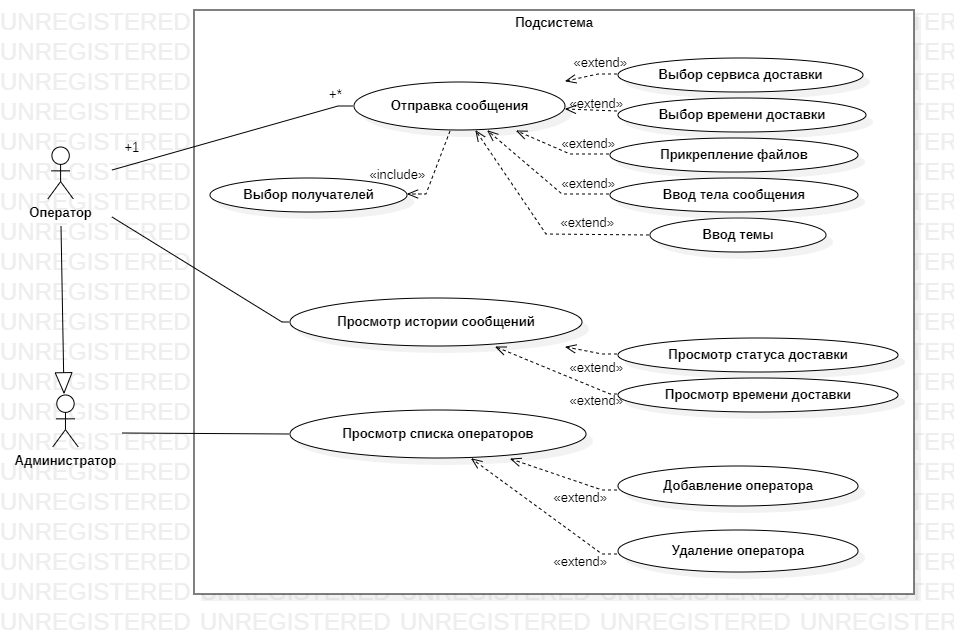


Рис. . Диаграмма Use case

# ДИАГРАММА КЛАССОВ

Структурная диаграмма классов демонстрирует общую структуру иерархии классов и интерфейсов системы.

Для взаимодействия с таблицами в базе данных были созданы соответствующие классы:

1. Message – класс, хранящий информацию об экземпляре сообщения. Помимо отправителя, получателей, темы, тела и прикрепленных файлов, класс предоставляет доступ к вспомогательной информации: запланированное время доставки, журнал, очередь, выбранный пользователем сервис доставки и сервис, с помощью которого сообщение было доставлено на самом деле.
2. DeliveryService – класс, хранящий информацию о внешних сервисах доставки.
3. File – класс, хранящий информацию о прикрепленных к сообщению файлах.
4. FileTypes – перечисление типов файлов.
5. User – класс, хранящий информацию об отправителе: его имя, почту, хэш пароля, статус активности, журнал и доступы.
6. Access – класс, хранящий информацию о доступах пользователей, в том числе роль пользователя.
7. Roles – перечисление ролей пользователей.
8. Tariff – класс, хранящий информацию о существующих тарифах: название тарифа, его цену и возможные скидки.
9. Sale – класс, хранящий информацию о скидках: название и значение.
10. DeliveryStatuses – перечисление статусов доставки.
11. Queue – класс, хранящий информацию об очередях: сообщения, находящиеся в очереди, а также интервал асинхронной отправки сообщений.
12. Journal – класс журнала. Здесь хранится информация о пользователе, которому принадлежит данный журнал, и сообщениях, хранящихся в данном журнале.

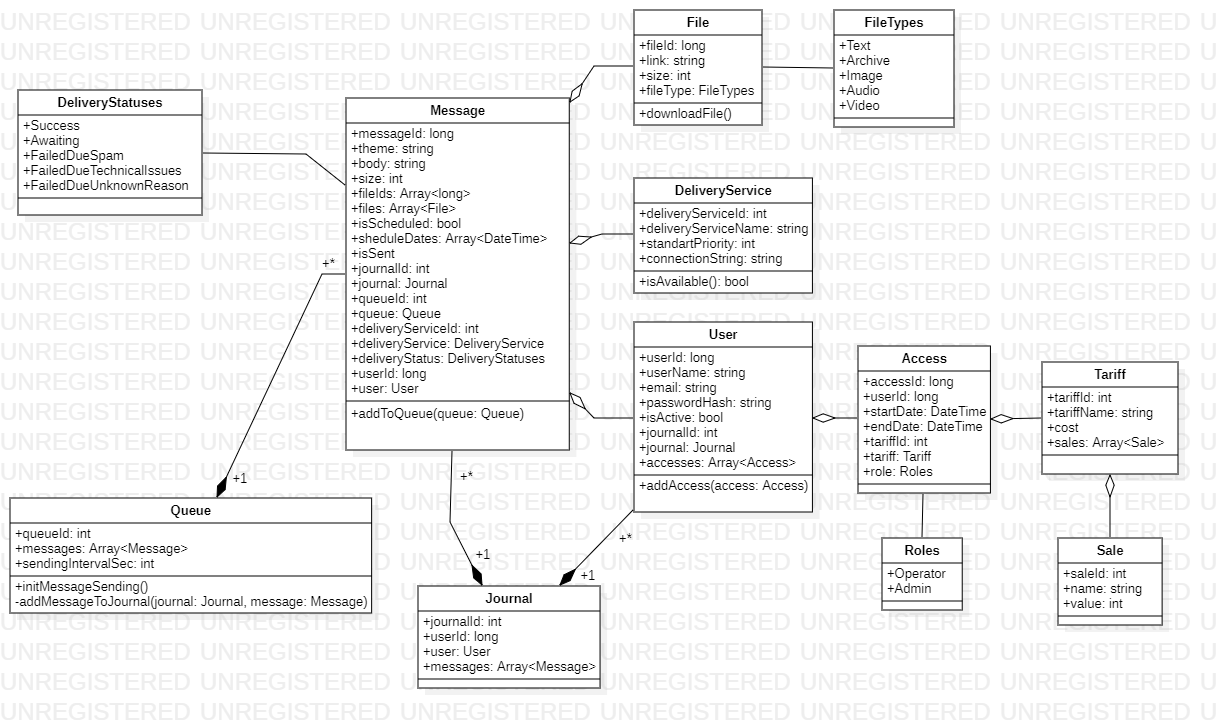


Рис. . Диаграмма классов

# ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ

На данной диаграмме изображен процесс создания сообщения.

Начинается процесс с перехода на страницу создания сообщения и заполнения всех сообщений. Затем осуществляется отправка запроса к серверной части.

Серверная часть обрабатывает запрос, добавляет сообщение в хранилище данных, формирует ответ и возвращает его клиентской части.

После получения ответа клиентская часть осуществляет переход на страницу с деталями созданного сообщения.

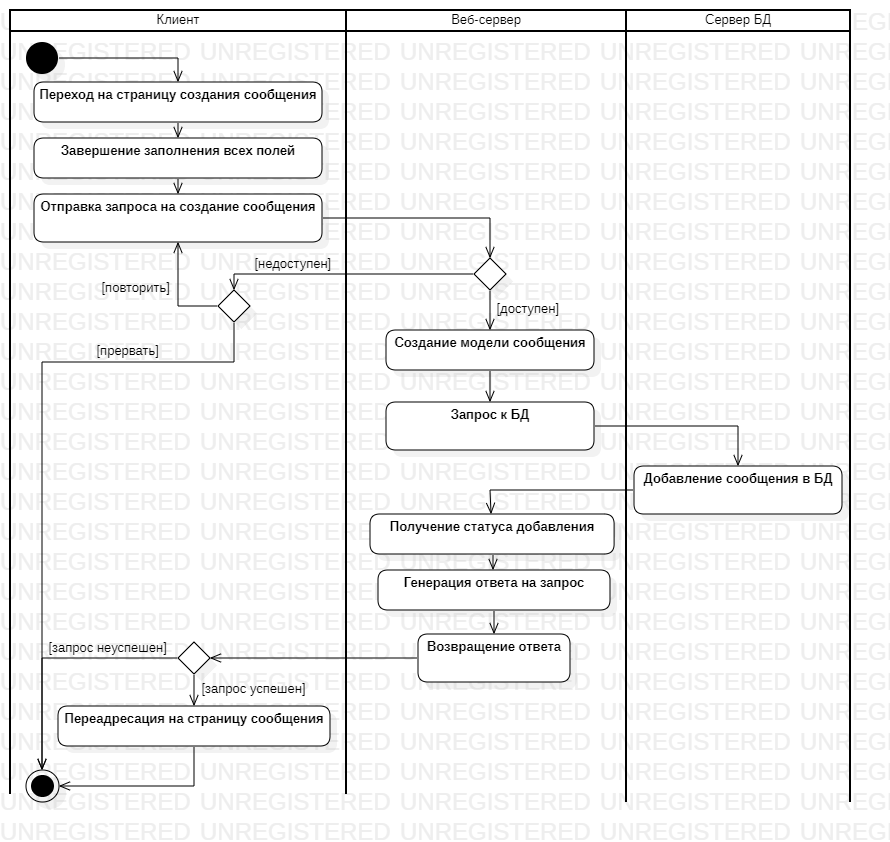


Рис. . Диаграмма последовательности

# ДИАГРАММА СХЕМ СОСТОЯНИЙ

Диаграмма схем состояний отображает поток управления, следующий от состояния к состоянию.

На данной диаграмме изображены процессы авторизации и отправки сообщения.

Процесс авторизации включает в себя ввод логина, пароля и проверку на корректность введенных данных.

Процесс отправки сообщения состоит из составления сообщения, проверки корректности заполненных полей сообщения. Затем письмо заносится в очередь для отправки сообщений, где оно ожидает отправки. При этом клиентская часть осуществляет переход на страницу этого сообщения, где пользователь сможет следить за статусом доставки сообщения.

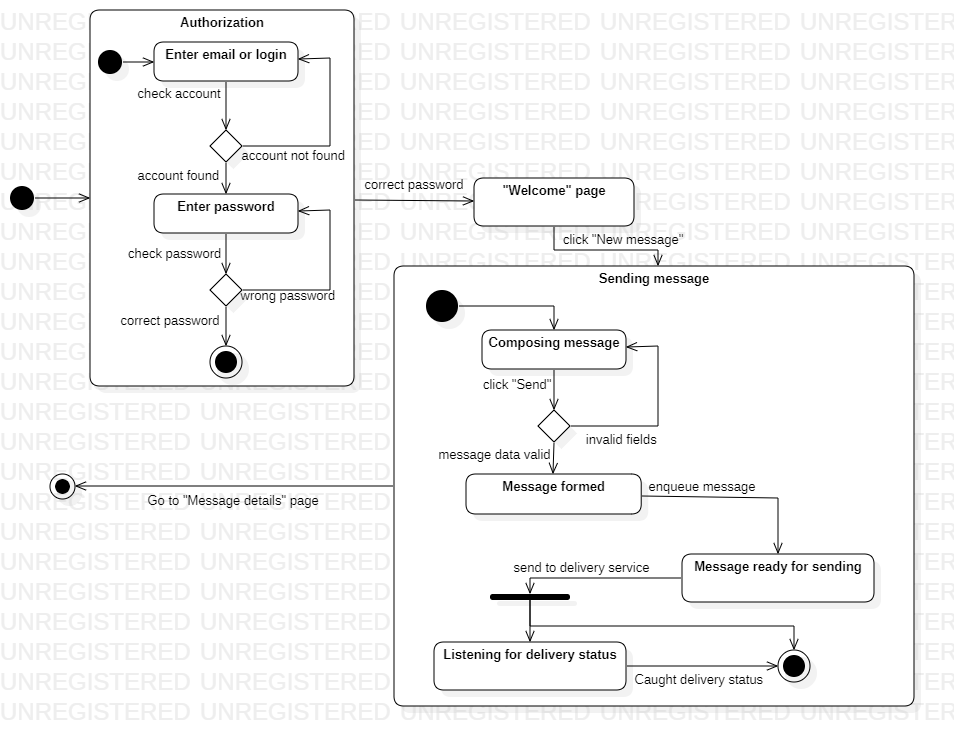


Рис. . Диаграмма схем состояний