Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет Программной инженерии и компьютерной техники

**Лабораторная работа №1**

по дисциплине «Архитектура программных систем»

**Выполнил**

Ореховский А.,

группа P3417

**Преподаватель**

Перл И. А.

Санкт-Петербург,

2020

Оглавление

[Задание 2](#_Toc59995308)

[Описание системы 2](#_Toc59995309)

[Требования 2](#_Toc59995310)

[Функциональные 2](#_Toc59995311)

[Use-case диаграмма 2](#_Toc59995312)

[Описание прецедентов 2](#_Toc59995313)

[Вывод 2](#_Toc59995314)

# Задание

Выбрать любую реально существующую систему и описать её в терминах UML. Желательно, чтобы система была не полностью информационной, но опиралась на информационную систему как показано в примере на лекции (Point of sale). Необходимо описать границы системы на разных уровнях, а также описать сценарии использования для нескольких Actor.

Отчёт по работе должен содержать:

1. Титульный лист с указанием автора и номера группы
2. Само задание
3. Описание рассматриваемой системы с требованиями к ней
4. Формальное описание системы с необходимым количеством UML диаграмм

# Описание системы

Для описания системы я решил использовать систему распределения заявок. Данная система применяется в полиграфическом агентстве. Концепция похожа на систему распределения заявок условного ресторана быстрого питания, например Mc.Donald's.

В целом, в агентстве имеются операторы, которые выполняют заказы, а также администратор, который эти заказы будет составлять исходя из нужд клиента. Система автоматически сохраняет информацию о заказе, автоматически выбирает исполнителя заказа, а также подсчитывает зарплату исполнителя, исходя из выполненных им заказов.

После выполнения заказа составляется акт выполненных работ, счет фактура, а также подсчитывается процент дохода оператору. Так же после создания заказа, клиент получает талон с уникальным ИД заказа, благодаря которому он сможет отслеживать состояние готовности данного заказа. Роль клиента так же состоит в составлении оценки обслуживания, исходя из которой будет корректироваться выбор операторов (чем лучше среднее значение оценок, тем вероятнее новое получение заказа), что должно стимулировать оных к лучшему обслуживанию клиентов.

# Требования

## Функциональные требования

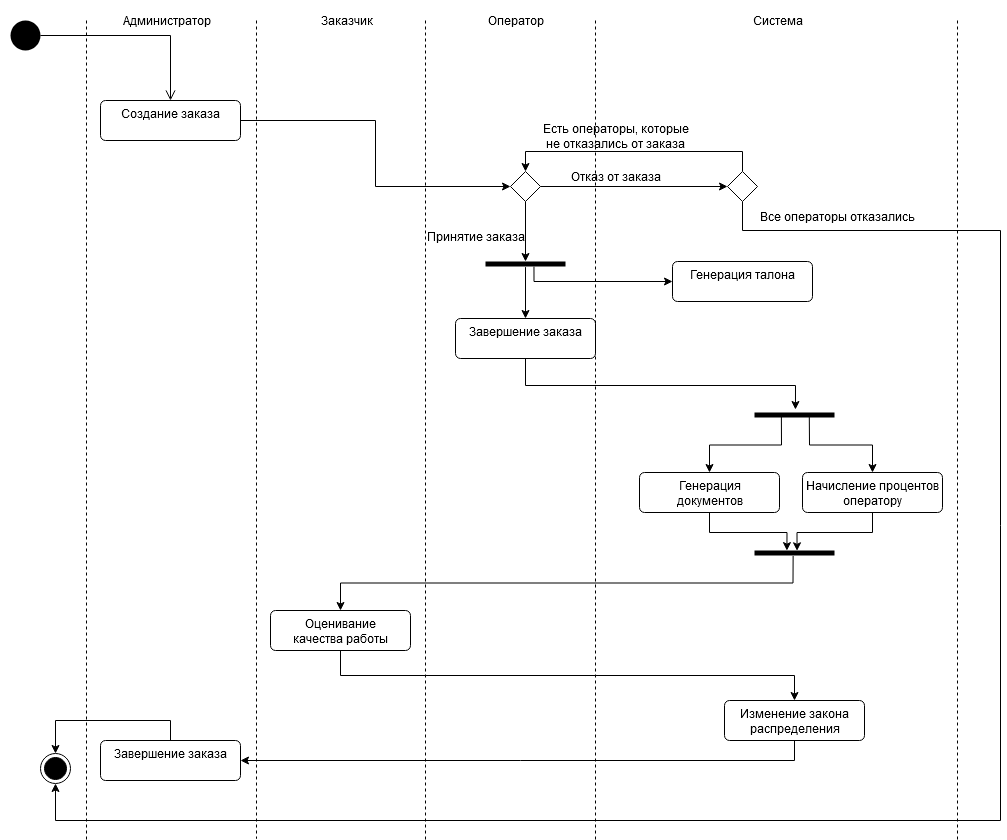
1. Система должна предоставлять возможность создания заказа администратором
2. Система должна предоставлять возможность редактирования заказа администратором
3. Система должна предоставлять возможность отмены заказа администратором
4. Система должна предоставлять возможность принятия заказа оператором
5. Система должна предоставлять возможность отказа от задания оператором
6. Система должна предоставлять возможность завершение заказа оператором
7. Система должна предоставлять возможность заказчику отслеживать состояние заказа по его ИД
8. Система должна предоставлять возможность заказчику оставлять отзыв о качестве выполненных работ

## Не функциональные требования

1. Система должна сохранять все данные о заказе в процессе его выполнения, а также после его завершения
2. Система должна автоматически генерировать необходимые документы после завершения заказа (акт выполненных работ, счет фактура)
3. Система должна автоматически начислять проценты от дохода агентства оператору, выполнившему заказ, на условный счет
4. Система должна автоматически создавать талон после составления заказа администратором
5. После генерации нового заказа заказ должен помещаться в очередь заказов
6. Система должна автоматически выбирать оператора из списка свободных
7. Если свободных операторов не имеется, заказ должен оставаться в очереди до появления свободного оператора
8. В случае, если все операторы откажутся от заказа, система должна отменить заказ
9. Система должна автоматически менять закон распределения заявок после нового отзыва
10. В случае если выбранный системой оператор отказывается от заказа, необходимо выбрать нового заказчика

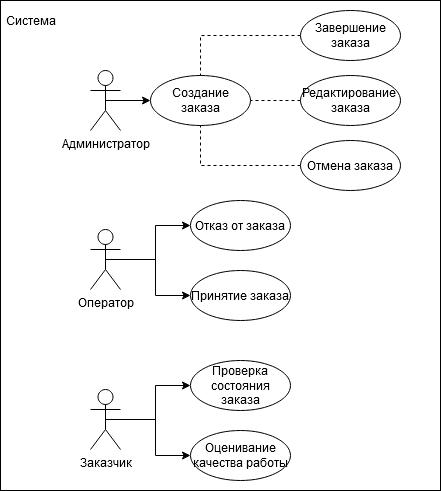
# UML

На рисунке 1 показана диаграмма деятельности системы. Временные рамки ограничены созданием и завершением заказа.



*Рисунок 1*

На рисунке 2 показана use-case диаграмма системы.



*Рисунок 2*

# Вывод

Не функциональных требований получилось больше, чем функциональных, так как данная систем призвана автоматизировать около-бухгалтерский учет. Все нефункциональные требования я посчитал таковыми так как по моему мнению они подходят под описание «бизнес-правил».

Авторизация является важной частью системы. Оператор не может генерировать заказы, как и администратор выполнять заказы. Несмотря на это, я опустил процесс аутентификации и авторизации пользователей, так как к системе не авторизованный/аутентифицированный пользователь будет иметь доступ лишь к части системы, которая отвечает за состояния заказов. Такого пользователя можно считать за заказчика. Заказчиц же сможет оставить отзыв о заказе на оборудовании самого агентства, например на специальном терминале. Наверняка такая практика не есть хорошая.

На диаграмме деятельности системы, как по мне, все расписано достаточно понятно. Данная диаграмма дает представление о том, как система должна работать. На use-case диаграмме показаны роли и доступные им действия, но не показана очередность этих действий (разумеется, для этого и была использована диаграмма деятельности). На диаграммах это не показано, но заказчик может проверит состояние заказа в любой момент времени, начина с момента, когда талон был сгенерирован.

Описание прецедентов я посчитал избыточным и поэтому не добавил в отчет.