ОГЛАВЛЕНИЕ

[1 Тематика курсовой работы #](#__RefHeading___1)

[2 Требования к выполнению и представлению результатов курсовой работы #](#__RefHeading___2)

# Тематика курсовой работы

Разработать программную клиент-серверную систему защищенного документооборота, реализующую следующие сценарии:

Пусть в системе документооборота есть 3 пользователя, образующие следующую иерархию:

**Директор** может:

1. Давать поручения своим подчиненным – Главе отдела и Специалисту отдела.
2. Подписывать *документы*, связанные с поручениями. *Документ*, имеющий подпись Директора, считается «принятым к исполнению», а поручение считается «закрытым».
3. Не подписывать *документы*, связанные с поручениями, тем самым либо отменяя его (поручение просто считается «закрытым», никаких дальнейших действий по нему не требуется), либо отправляя *документ* на доработку (поручение остается активным, а *документ* отправляется обратно по маршруту)

**Глава отдела** может:

1. Давать поручения своему подчиненному – Специалисту отдела.
2. Делегировать существующие поручения своему подчиненному – Специалисту отдела.
3. Загружать *документ* в ответ на соответствующее поручение.
4. Подписывать *документы*, связанные с поручениями. *Документ*, имеющий подпись Главы отдела, отправляется дальше по маршруту – к Директору.
5. Не подписывать *документы*, связанные с поручениями, тем самым либо отменяя свое поручение (оно просто считается «закрытым», никаких дальнейших действий по нему не требуется), либо отправляя *документ* на доработку (поручение остается активным, а *документ* отправляется обратно по маршруту – к предыдущему подписанту). Обратите внимание, что Глава отдела не может отменить поручение Директора.
6. Удалять *документ*, который был им ошибочно загружен в ответ на какое-либо поручение, но еще не имеет подписи.

**Специалист отдела** может:

1. Загружать *документ* в ответ на соответствующее поручение.
2. Подписывать *документы*, связанные с поручениями. *Документ*, имеющий подпись Специалиста отдела, отправляется дальше по маршруту – к Главе отдела.
3. Удалять *документ*, который был им ошибочно загружен в ответ на какое-либо поручение, но еще не имеет подписи.

Информация должна включать числовые и строковые данные, а также целиком файлы некоторого формата (например, документы).

Информация должна храниться в базе данных, для этого обучающемуся необходимо выбрать СУБД и реализовать базу данных в соответствии с предметной областью.

Серверная часть должна обеспечивать обмен данными по сети с клиентской частью, проводить аутентификацию и авторизацию пользователя (token, cookies, ...). Сетевой обмен должен быть криптографически защищен. Серверная часть должна быть способной одновременно обслуживать несколько клиентов.

Клиентская часть ответственна за обмен данными с серверной частью, предоставлять пользователям графический интерфейс (посредством WEB-приложения, десктопного или мобильного приложения), предоставлять средства аутентификации пользователя в системе.

# Требования к выполнению и представлению результатов курсовой работы

Работа выполняется индивидуально или в составе групп, с использованием языка программирования и стека технологий на выбор учащегося. В процессе построения программной системы должны быть использованы методы сетевого программирования, параллельного программирования, взаимодействия с базами данных, построения пользовательских интерфейсов.

Рекомендуемый стек технологий (можно использовать альтернативы, например, стек Java EE):

Spring(MVC, Boot, Security, Data, WebFlux, …), PostgreSQL/MySQL, Junit, …

Результаты представляются в виде отчета, содержащего следующие элементы:

* Описание примененного инструментария, обоснование выбора тех или иных технологий для выполнения целей проекта.
* Подробное описание архитектуры программной системы;
* программный код пакетов и классов программной системы;
* примеры тестовых сценариев и результаты их работы;
* ссылку на git репозиторий с проектом (опционально).