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

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

ДИРЕКТОР

ГЛАВА ОТДЕЛА

СПЕЦИАЛИСТ ОТДЕЛА

**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.