

Система совместной работы над документами

Контекст

Компания Zesla Group предоставляет услуги по разработке уникального измерительного оборудования для строительных компаний. В текущий момент времени компания работает со своими клиентами и контрагентами, подписывая бумажную версию договора. Учитывая растущее количество контрактов, клиентов и контрагентов, а также время и стоимость, необходимые для подписания двух и трехсторонних договоров, компания Zesla Group приняла решение разработать систему, позволяющую создавать, редактировать и подписывать контракты в электронном виде.

Задача

Разработать простую систему совместной работы над документами, позволяющую сократить время и стоимость процесса создания и подписания договоров для компании Zesla Group.

Требования к системе

Бизнес-правила

1. В подписании одного договора могут участвовать две или три стороны (компании)
2. Каждый участник процесса (компания) должна иметь возможность подтверждения документа каждой из трех ролей: юрист, экономист, генеральный директор, подтверждение документа всеми ролями - обязательно
3. Утвержденный всеми сторонами документ может быть подписан только пользователем с ролью генеральный директор
4. Каждая компания может иметь в системе до 3 пользователей для ролей юрист и экономист, роль генерального директора - только у одного пользователя
5. Все сотрудники одной компании имеют доступ ко всем договорам, которые находятся в работе у данной компании

Функциональные требования

ReqID	Требование	Приоритет
FR01	Платформа должна обеспечивать возможность создания, редактирования, просмотра (включая список), удаления и архивирования документов	Высокий
FR02	Система должна работать в браузере без установки дополнительного ПО (по аналогии с google docs)	Высокий

FR03	Система должна позволять участникам отслеживать вносимые в документ изменения в режиме реального времени (по аналогии с google docs)	Высокий
FR04	Система должна позволять создавать, просматривать, редактировать и удалять комментарии на части текста (по аналогии с google docs)	Средний
FR05	Система должна позволять сохранять различные версии документов, а также возвращаться к любой выбранной версии (по аналогии с google docs)	Низкий
FR06	Система должна позволять отправлять текстовые сообщения внутри интерфейса редактирования документа всем приглашенным в конкретный документ участникам как индивидуально в режиме “личное сообщение”, так и отправлять групповые текстовые сообщения всем участникам	Средний
FR07	Система должна позволять базовое форматирование: выделение абзацев, выравнивание текста по горизонтали (влево, по центру, вправо), выделение жирным и <i>курсивом</i> блоков текста	Низкий
FR08	Система должна позволять отправлять на согласование версию документа другой компании (клиент, контрагент) для последующего согласования и подписания	Средний
FR09	Система должна позволять отправлять и принимать приглашение на редактирование, согласования или подписание документа	Средний
FR10	Система должна отправлять уведомления на email и в окно редактора в моменты, когда: <ol style="list-style-type: none"> 1. Пользователь получает приглашение на редактирование документа 2. Требуется согласование документа для перевода в статус “Согласован” 3. Статус документа согласован и требуется подписание документа для перевода в статус “Подписан” 4. Статус документа меняется на “В архиве” 	Низкий

Допущения

1. Все роли и пользователи в системе заранее заданы, не предполагается системы управления пользователями и ролями
2. Пользователи могут иметь доступ к документам без запроса пароля, для входа в систему достаточно указать имя пользователя
3. В рамках одной сессии у пользователя может быть только одна роль. Также должна быть предусмотрена возможность смены пользователя (роли) на другую

Жизненный цикл документов

Сущность	Стадии жизненного цикла
Документ	<ol style="list-style-type: none">1. Создается2. Согласование3. Согласован4. Подписание5. Подписан6. В архиве

Требования к реализации

Приложение должно представлять собой SPA и должно быть разработано с использованием фреймворков React и Flask, без использования ORM.

Требования к хранению данных

База данных для хранения документов - MongoDB.

Требований к реализации бэкенд части (Python)

Статически типизировать имеющийся серверный код. Настроить туру на обязательное требование типизированных сигнатур функций, также запретить запись значения None, если тип не помечен как Optional.

Убедиться, что код проходит проверку типов.

Отключить в конфигурации для black приведение строковых литералов к виду в двойных кавычках (т.е. разрешить одинарные).

Убедиться, что настроить black и flake8 не конфликтуют (длина строки и т.д.).

Настроить git hooks, используя пакет pre-commit. Хуки должны использовать все инструменты, перечисленные выше, можно дополнительно добавить и другие.

Убедиться, что dev-зависимости не смешиваются с основными, и что их установку можно пропустить (здесь может быть удобен poetry, но его использовать не обязательно).

В README подробно описать с нуля процесс установки всех требований проекта для использования в production. Привести конкретный набор команд.

Отдельно описать настройку окружения для разработчика. Описать, как вызвать выполнение тестов, хуки pre-commit.

Описание должно быть таким, чтобы разработчики, не знакомые с проектом, могли а) запустить его б) продолжить разработку с текущего этапа.

Требований к реализации фронтэнд части (React)

Вся фронтенд часть должна быть написана на React и быть SPA. В этом задании нет ограничений на используемые библиотеки, но рекомендуется использовать стандартные возможности реакта для хранения и манипуляции данными. То есть использовать state, props, context и useReducer вместо подключения библиотек типа Redux.

Для генерации приложения можно (и рекомендуется) использовать create react app (CRA). CRA уже содержит настройки для ESLint, скорректируйте их под себя и настройте pre-commit hooks по аналогии с backend частью.

Библиотеки, которые могут ускорить разработку:

- Роутинг: react-router
- Стили: material-ui / react-bootstrap
- WYSIWYG редактор:
<https://ourcodeworld.com/articles/read/1065/top-15-best-rich-text-editor-components-wysiwyg-for-reactjs>
- Utils: lodash, classnames