**IMS Impuls**

Система управления информацией “Импульс” — это платформа для совместной работы, обеспечивает взаимодействие между архитектором и стейкхолдерами, которые вносят свой вклад в формирование модели данных проекта и обращаются к информации о ключевых бизнес-процессах, архитектуре программного обеспечения и ИТ-архитектуре предприятия.

Потребители используют информацию, полученную из модели данных, для долгосрочного планирования, принятия решений, управления, авторизации и многого другого.

Полностью совместимо с решением Sparx Systems Enterprise Architect, которое фактически может выступать в качестве толстого клиент, для работы с моделью данных проекта.

Частично идея для создания проекта, была позаимствована на сайте - https://prolaborate.sparxsystems.com/prolaborate5

Также может рассматривается как упрощенный аналог системы управления инженерными данными от Неолант "Неосинтез".

**ER диаграмма базы данных проекта**

[**https://www.figma.com/file/vewdlcyelMrWVBgxarOSlN/DB-diagram?type=whiteboard&node-id=0-1&t=2PxWHLlg8fztJO48-0**](https://www.figma.com/file/vewdlcyelMrWVBgxarOSlN/DB-diagram?type=whiteboard&node-id=0-1&t=2PxWHLlg8fztJO48-0)

**Визуальное представление**

[**https://www.figma.com/file/oMGQxCqr40KSGW0XREpRHE/Макет?type=design&mode=design&t=2PxWHLlg8fztJO48-0**](https://www.figma.com/file/oMGQxCqr40KSGW0XREpRHE/Макет?type=design&mode=design&t=2PxWHLlg8fztJO48-0)

**Онлайн проект для последующего показа**

**<https://codesandbox.io/p/github/DenSakharov/impuls/main?embed=1&file=%2Fclient%2Fsrc%2FApp.tsx>**

**Базовые функции:**

1. Авторизация, регистрация, смена пароля на сайте.

2. Профиль пользователя и его роль.

3. Список объектов модели данных проекта в виде структуры (Папки и объекты).

4. Карточка объекта с базовым набором полей, с расширенным функционалом:

- Инструмент параметризации объекта.

- Фиксация истории изменений объектов.

- Обсуждения объектов модели данных.

- Инструмент для трассировки зависимостей объектов.

- Инструмент для формирования задач в рамках проработки объекта модели данных.

- Поддержка типизации объектов.

5. Вложение документов в карточку объекта, с возможностью редактирования (без онлайн редакторов, в много пользовательском режиме). Редактировании документа онлайн несколькими пользователями.

6. Простейший инструмент утверждения объекта с блокировкой карточки.

7. Комментарии в виде форума, по каждому объекту

8. Импорт / экспорт через Excel файл списка всех объектов из модели данных проекта.

9. Создание простейшего отчета PDF.

10. Базовые функции управления этапами и задачами проекта.

**Развитие проекта**

- Интеграция с системой по управлению задачами Jira, Битрикс24 для экспорта задач.

- Графическое отображение диаграмм в UML нотации, спроектированных в Sparx Systems Enterprise Architect.

- Реализация настройки скрытия и шифрования объекта или групп объектов в соответствии с матрицей доступа.

**Описание концепта в части «Управления требованиями»**

Реализовать тонкий клиент (ВЕБ интерфейс) для управления данными проекта в БД Postgresql.

Беклог:

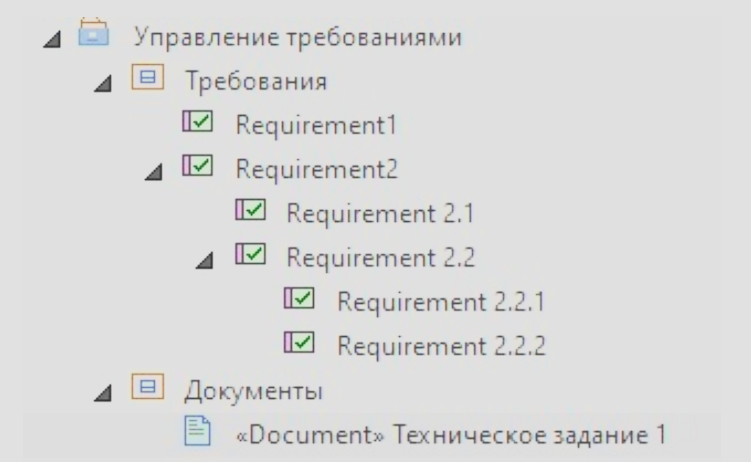
• Реализовать инструменты авторизации пользователей. Предусмотреть матрицу ролей для пользователей:

− manager (утверждающий) – утверждает статус объекта;

− design (проектировщик) – создание и удаление объектов;

− guest (гость) – только просмотр данных.

• Реализовать форму навигации по дереву объектов в проекте. Предусмотреть наличие папок и объектов, структуры подчинения родительских и дочерних объектов.



* Форма карточки объекта для различных типов (Документ, Требование, Связь).

Наличие обязательных полей:

− Название.

− Тип объекта.

− Статус.

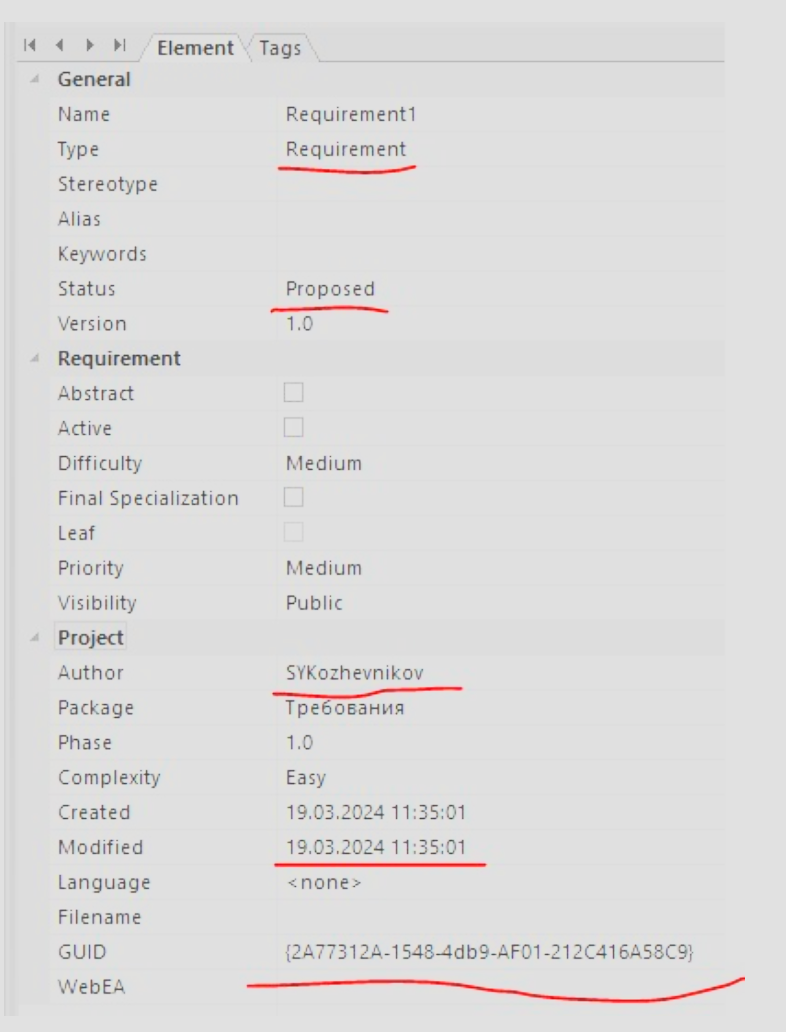
− Автор.

− Дата изменения.

− GUID.

− Детальное описание, с поддержкой минимального форматирования.

− Ссылка на документ.

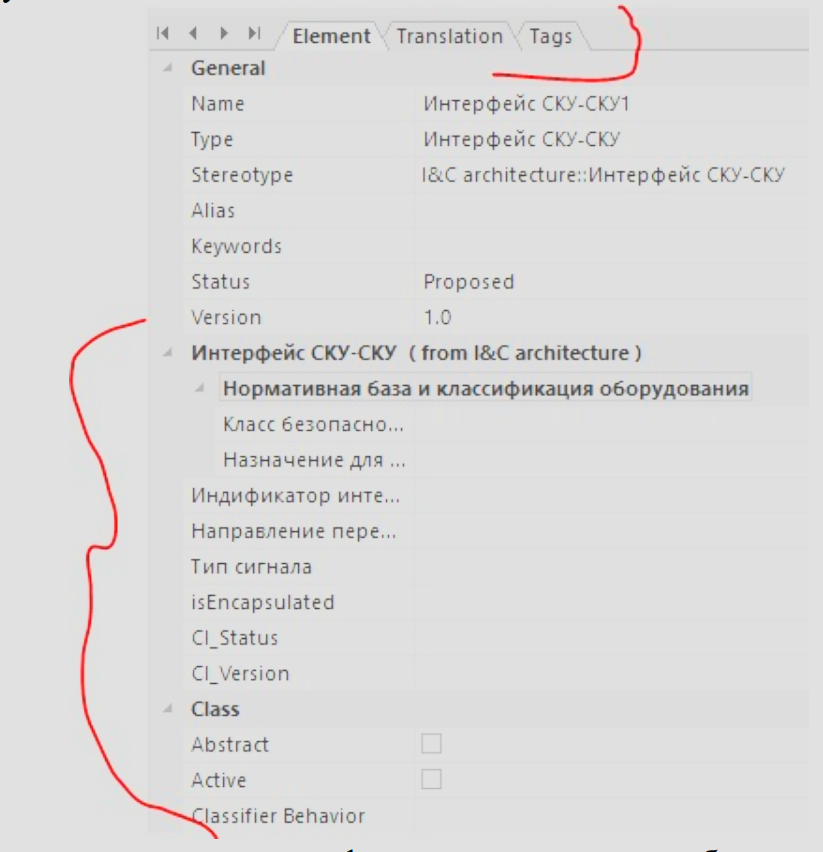


* Реализовать инструмент утверждения объектов.

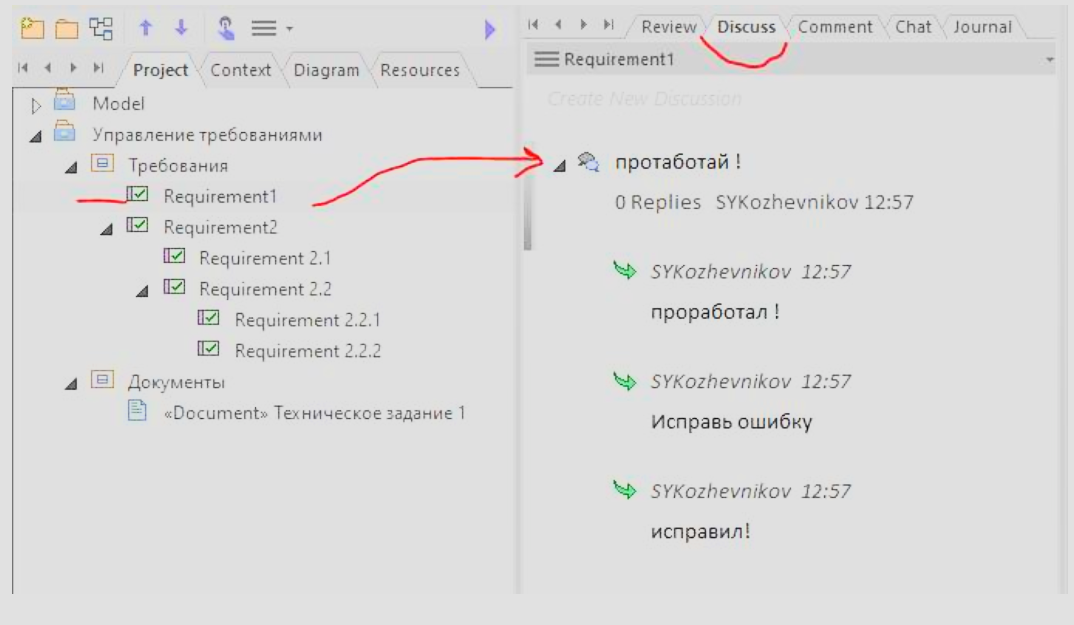
Созданные объекты выводятся в специальном представлении для утверждения пользователем, с определенной ролью (Manager). После утверждения объект блокируется от изменений.

* Реализовать инструменты параметризации объектов.

Реализовать инструмент добавления для каждого объекта кастомных полей для ввода параметров.

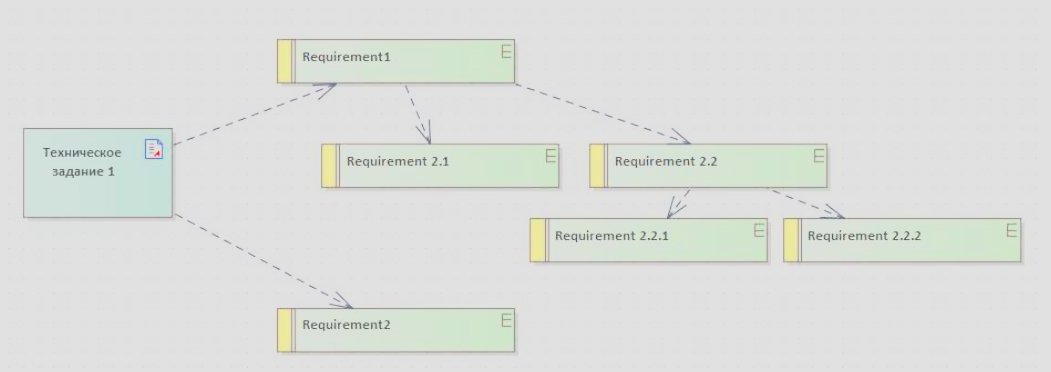


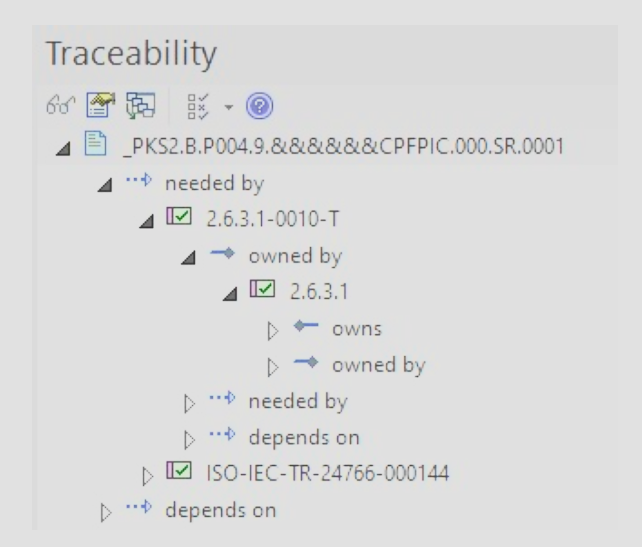
* Реализовать инструмент типа форум или чат, для обсуждения и проработки данных по каждому объекту.



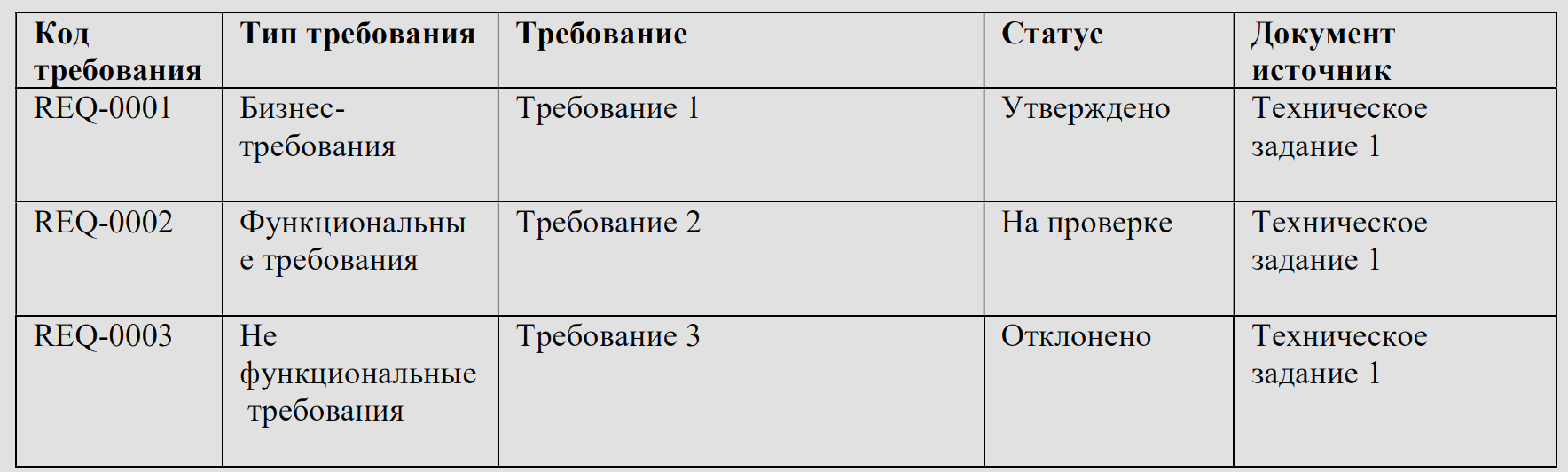
* Реализовать инструменты для трассировки зависимостей (связей) между объектами.

Реализовать представление, которое позволит визуально отображать связи между объектами, от стартового объекта, ко всем связанным объектам. На рисунках представлены зависимости.



****

* Реализовать инструмент экспорта – в формате Excel и csv файла.
* Реализовать отчет – «Спецификация требований»

****

**Ответы на вопросы**

1.Если мы живем в парадигме, того что у нас в системе может быть уровни доступа, которые строго иерархичны (админ(редактор)- читатель), то не понимаю зачем нам группы пользователей хватит одной таблицы с пользователями и атрибутом роль.

>> Возможно и хватит. Не принципиально. Я обычно сталкивался с разделёнными таблицами пользователи, группы, роли. Так система становится более гибкой, в плане реализации матрицы доступов. На этапе разработки, тот, кто будет пилить эту часть, поправит.

2.Что такое стереотип объекта? t\_stereotypes

>> Таблица стереотип содержит перечень всех возможных типов объектов со своими атрибутами. Для заполнения значениями поле объекта - "Тип". Например: Документ, Требование, Риски, Тест кейс и т.д.

3.Таблица задач проекта, о какой сущности идет речь, что такое проект? t\_tasks

>> Таблица с задачами, например, по каждому объекту можно создать задачу для отработки и контролировать выполнение. (В идеале, делать экспорт задач в Jira, Ёжку, Bitrix).

Например, задача "Разработать тест кейс для проверки Требования 1", такая задача привязывается к объекту Требование 1 и отрабатывается. Но в нашем проекте мы эту тему делать не будем.

4.Что такое системные параметры? usys\_system

>> таблица для хранения глобальных настроек проекта.

5.У объекта есть атрибут author, который не является FK для таблицы t\_secuser, автор может находится вне контура системы и мы его можем задать "вручную"? ну и до кучи.

>> Автор подставляется автоматически из справочника, который формируется отдельно от таблицы с пользователями. Так как пользователи увольняются и удаляются из системы, а Автор объекта, должен оставаться навсегда + должна быть возможность добавлять автора вручную из справочника, так как автор не всегда является пользователем системы.

Почему у всех таблиц имеется префикс t\_ ?

>> Привык к таким обозначениям, не принципиально.

t\_ - таблицы

v - представления

seq - порядок нумерации ключей

sys - системные таблицы

sec - безопасность