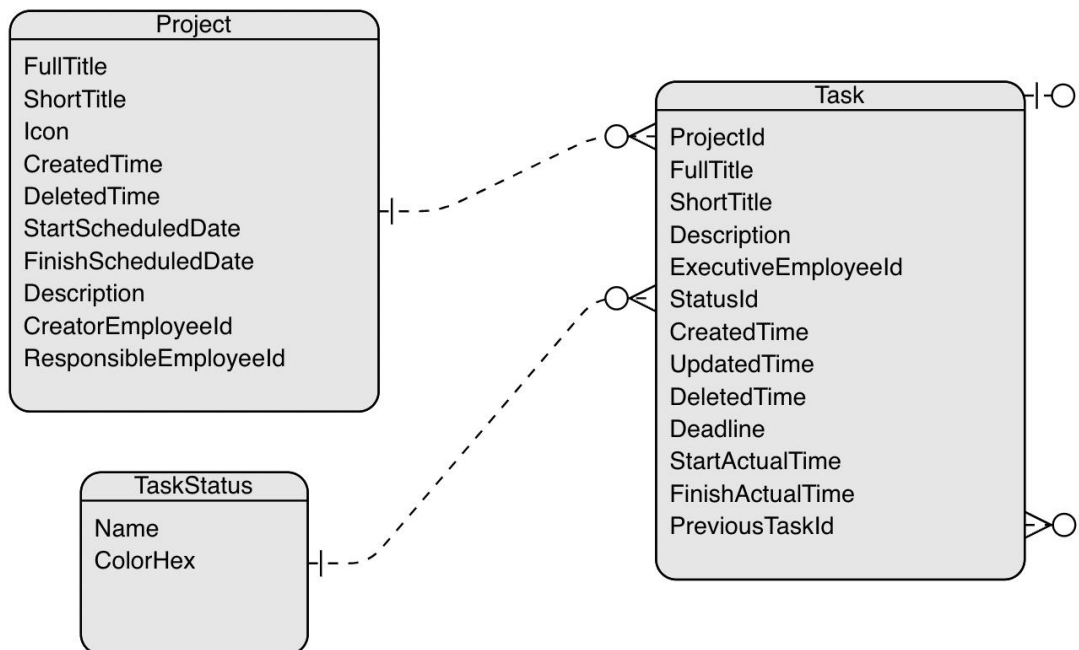


Сессия 1 (Проекты и задачи)

Наименование репозитория/ветки	Раздел задания
Designing/DB	Проектирование БД
Desktop/Project_Management	Desktop-разработка
Designing/API	Документирование API

Проектирование и реализация БД

Базовые сущности были спроектированы ранее: предоставленные наработки необходимо перенести в СУБД без изменений и определить подходящие типы данных. Допускается наращивать структуру БД дополнительными полями, таблицами, связями и другими объектами при необходимости. Итоговый вариант схемы данных загрузите в соответствующий репозиторий в формате .pdf/.jpg/.png.



Проект

- Иконка
- Наименование
- Короткое наименование (чаще всего это аббревиатура, легко и компактно идентифицирующая проект, например, PRO)
- Плановые даты начала/окончания
- Дата создания/удаления (дата создания должна фиксироваться на уровне БД)

- Описание (длинный текст)
- Ответственный сотрудник
- Создатель

Задача

- Наименование
- Короткое наименование (автоматически формируется как “Короткое наименование проекта” + порядковый номер задачи внутри проекта, например, “PRO-13”)
- Описание (длинный текст)
- Исполнитель
- Статус
- Дата создания/удаления/обновления (дата создания и обновления должна фиксироваться на уровне БД)
- Крайний исполнения задачи (Deadline)
- Предшествующая задача (когда задачу нельзя начинать до выполнения предыдущей)
- Фактическое время начала/завершения выполнения

Статус (относительно статичный набор данных)

- Наименование
- Цвет (в формате #ffffff)

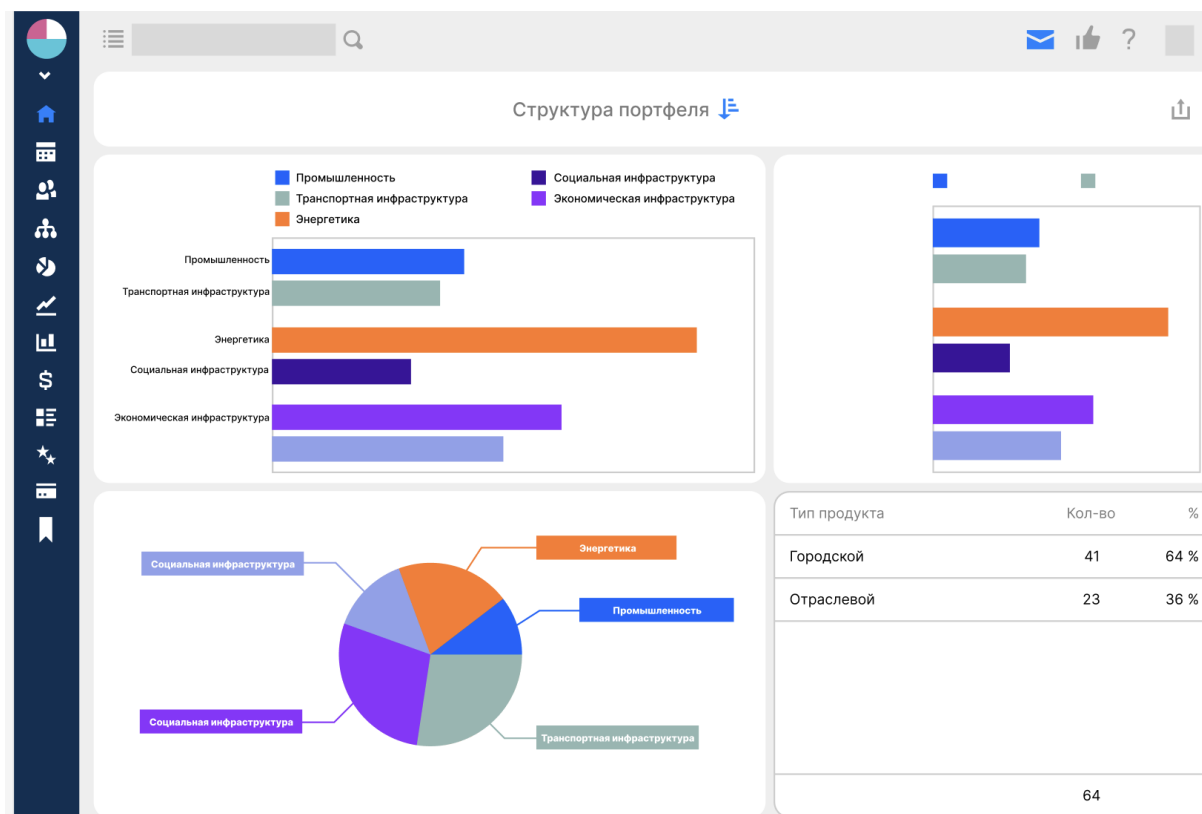
Необходимо дополнить структуру данных для хранения следующей информации:

- Список наблюдателей за продвижением работы над задачей
- Комментарии/обсуждения сотрудников по конкретной задаче
- Набор вложений и/или ссылок к конкретной задаче

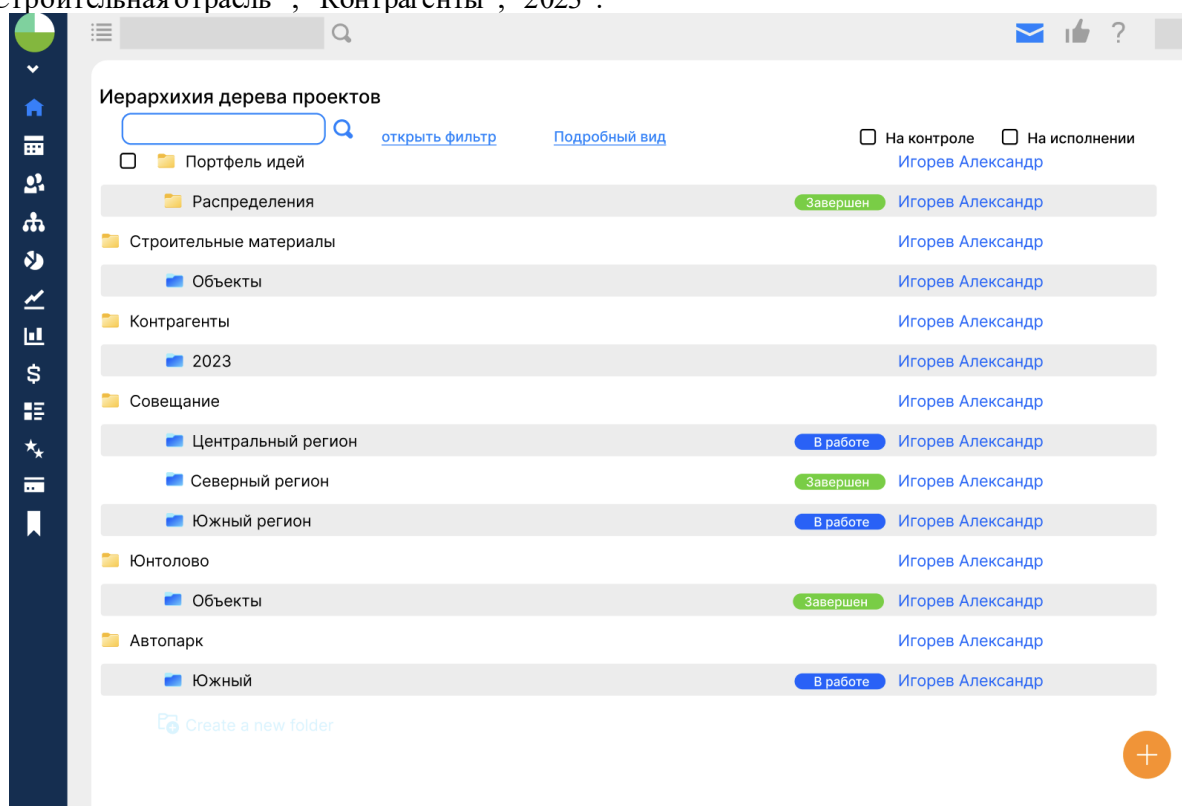
Также требуется спроектировать и реализовать историю изменения статусов задач (на какой статус и когда было изменение). История должна генерироваться на стороне базы данных (проверка будет осуществлена без клиентского приложения) с помощью триггеров.

Для дальнейшего развития системы “Проектный офис” необходимо обеспечить хранение информации о портфелях проекта, где портфель - это объединение группы проектов по региональному, промышленному или другому признаку.

Для определения сущности и полей для портфеля проектов воспользуйтесь проектом интерфейса системы первой итерации проектирования.



Для просмотра портфеля проектов предполагается иерархическая структура по разным признакам объединения проекта в портфель, причем один и тот же проект может отображаться в разных портфелях, например проект “Леруа Мерлен” будет отображаться в портфелях “Строительная отрасль”, “Контрагенты”, “2023”.



Обеспечьте хранение данных для построения аналитики по проектам на основании проекта интерфейса системы первой итерации проектирования.

Импорт

Выполните импорт данных из предоставленных файлов в созданные таблицы БД без потери данных.

Desktop-разработка

Общие требования к приложению

К приложению необходимо добавить иконку из ресурсов.

Необходимо использовать единую цветовую схему во всём приложении в соответствии с макетами.

Левое меню



Необходимо реализовать общее меню для навигации по функционалу приложения. В меню должны быть добавлены следующие пункты:

- Дашборд
- Задачи
- Календарь проектов

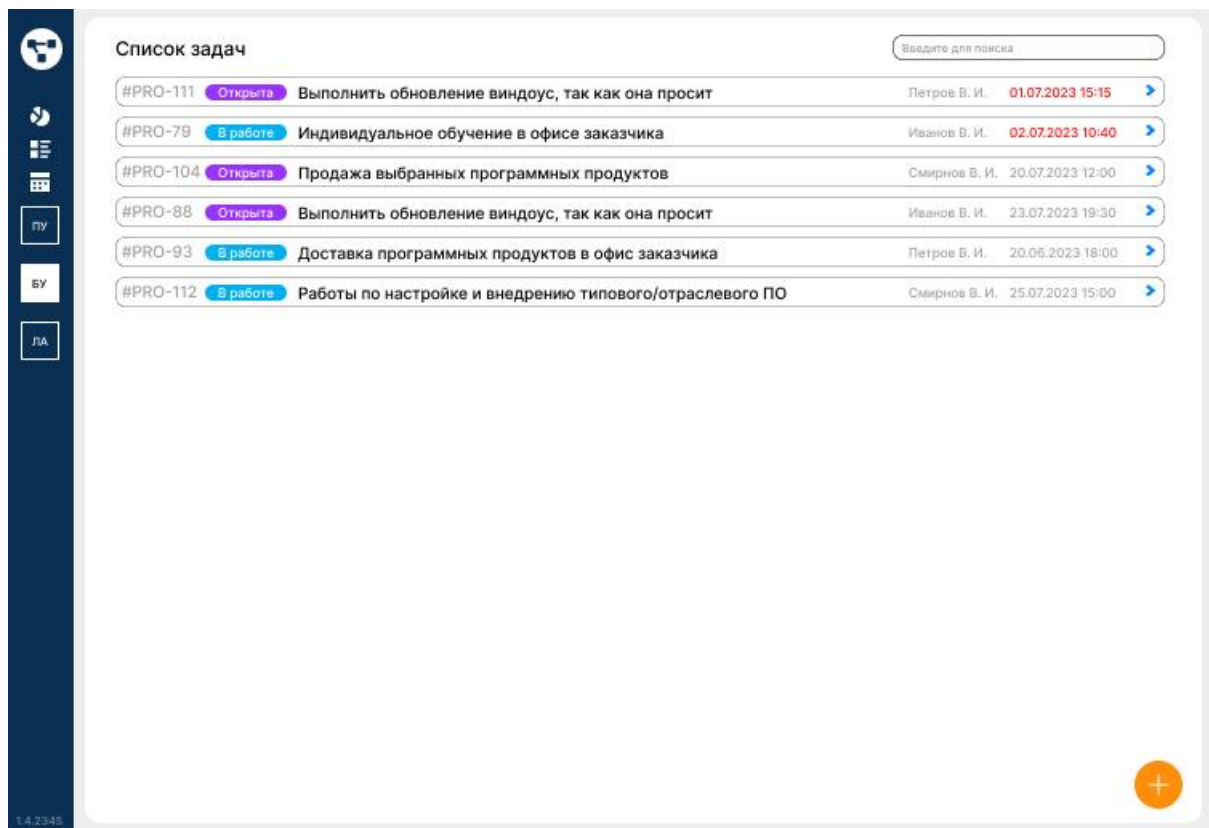
Для элементов меню необходимо подготовить пустые страницы-заготовки для каждого раздела. Приложение должно сохранять последний запущенный раздел в приложении, чтобы при следующем запуске приложения открывать его.

Под элементами меню должны быть отображены проекты в виде двух первых букв из названия проекта. Если название проекта состоит из двух или более слов, то для иконки проекта необходимо использовать первые буквы первых двух слов. При наведении на элементы проектов необходимо показывать пользователю подсказку с полным названием проекта. При нажатии на конкретный проект он должен быть выбран текущим и вся информация в последующих разделах

приложения должна быть получена по выбранному проекту. По умолчанию должен быть выбран первый проект.

В нижней части компонента “меню” необходимо отобразить версию приложения, а также номер сборки в формате 1.0.456 (где 1.0 - версия, а 456 - номер сборки приложения. При каждой сборке приложения через средство разработки номер должен увеличиваться и сохраняться в параметрах проекта.

Список задач



Список задач должен содержать задачи только выбранного проекта. По умолчанию пользователю отображаются только задачи, у которых статус не “закрыта”. Задачи должны быть отсортированы в следующем порядке:

1. Задачи со статусом “в работе”, но у которых дедлайн уже прошел
2. Задачи со статусом “открыта”, но у которых дедлайн уже прошел
3. Задачи со статусом “в работе”
4. Задачи со статусом “открыта”
5. Остальные задачи

Каждая задача в списке должна отображать следующую информацию:

- номер (с коротким названием проекта),
- наименование,
- исполнитель,
- дедлайн.

Статус должен отображаться с корректным цветом в соответствии с импортом. Дедлайн у просроченных задач должен быть обозначен красным цветом текста.

В верхней части списка задач должна быть реализована возможность поиска задач. Поиск должен работать в реальном времени, а также позволять находить задачи у которых наименование или описание задачи содержат текст из поисковой строки. Результаты поиска необходимо сортировать по совпадению сначала в заголовке задачи, а после по описанию.

Подробный просмотр задачи

Список задач

Введите для поиска

#PRO-111 **Открыта** Выполнить обновление виндоус, так как она просит

#PRO-111 **Открыта** Выполнить обновление виндоус, так как она просит

#PRO-111 **Открыта** Подготовить отчёт

#PRO-111 **Открыта** Выполнить обновление виндоус, так как она просит

#PRO-123 **Подготовить отчёт** **В работе**

Создание задачи: 01.07.2023 15:45

Дедлайн: 01.07.2023 15:45

Предшествующая задача

#PRO-100

Фактическое начало выполнения: 01.07.2023 15:45

Фактическое завершение: -

Описание задачи

Исполнитель

Красин Дмитрий Сергеевич

Приложения

Фото с доски.jpg 4.5MB

Котики играют.jpg 14MB

Добавить +

Наблюдатели

Петров Игорь

Добавить +

Сохранить Удалить

При нажатии на определенную задачу список задач сжимается в два раза, а в правой части окна отображается компонент “подробный просмотр задачи”. Выбранная задача в списке, в свою очередь, должна быть выделена цветом.

В компоненте по просмотру/добавлению информации по задаче необходимо предусмотреть вывод и редактирование всех возможных данных задачи (кроме иконки) с последующим сохранением информации в БД.

Создание задачи

В правой нижней части окна списка задач должна быть отображена кнопка для добавления новой задачи, которая открывает компонент “подробный просмотр задачи” для заполнения данных новой задачи.

Реализуйте функционал по созданию задачи с сохранением данных в БД. Необходимо учесть не только поля у объекта “Задача”, но и ряд зависимых наборов данных, в частности:

- список файлов-приложений;
- наблюдателей.

Удаление задачи

Задачу, которую еще фактически не завершили, допускается пометить как удаленную. Удаленные задачи не должны отображаться в списке. При удалении задачи необходимо учесть, что она могла быть задействована в качестве предшествующей, запросить у пользователя подтверждение действия и только после этого удалить ссылки на задачу и саму задачу. При удалении задачи из цепочки задач, необходимо выполнить перепривязку.

Документирование API

На основе предоставленных макетов по всем четырем сессиям продумайте и подготовьте документацию по API, которая будет вам полезна для реализации взаимодействия с базой данных. Убедитесь, что ваша документация API содержит следующую информацию:

- путь для вызова каждой конечной точки;
- какие методы HTTP можно использовать с каждой конечной точкой;
- описание и назначение каждого метода;
- поля данных запроса и куда они идут, например, путь, строка запроса или тело;
- объяснение того, какие данные запроса требуются, а какие являются необязательными;
- список потенциальных ошибок;
- описание типа возвращаемого объекта при успешном выполнении запроса.
- Результат необходимо предоставить в формате .xlsx.