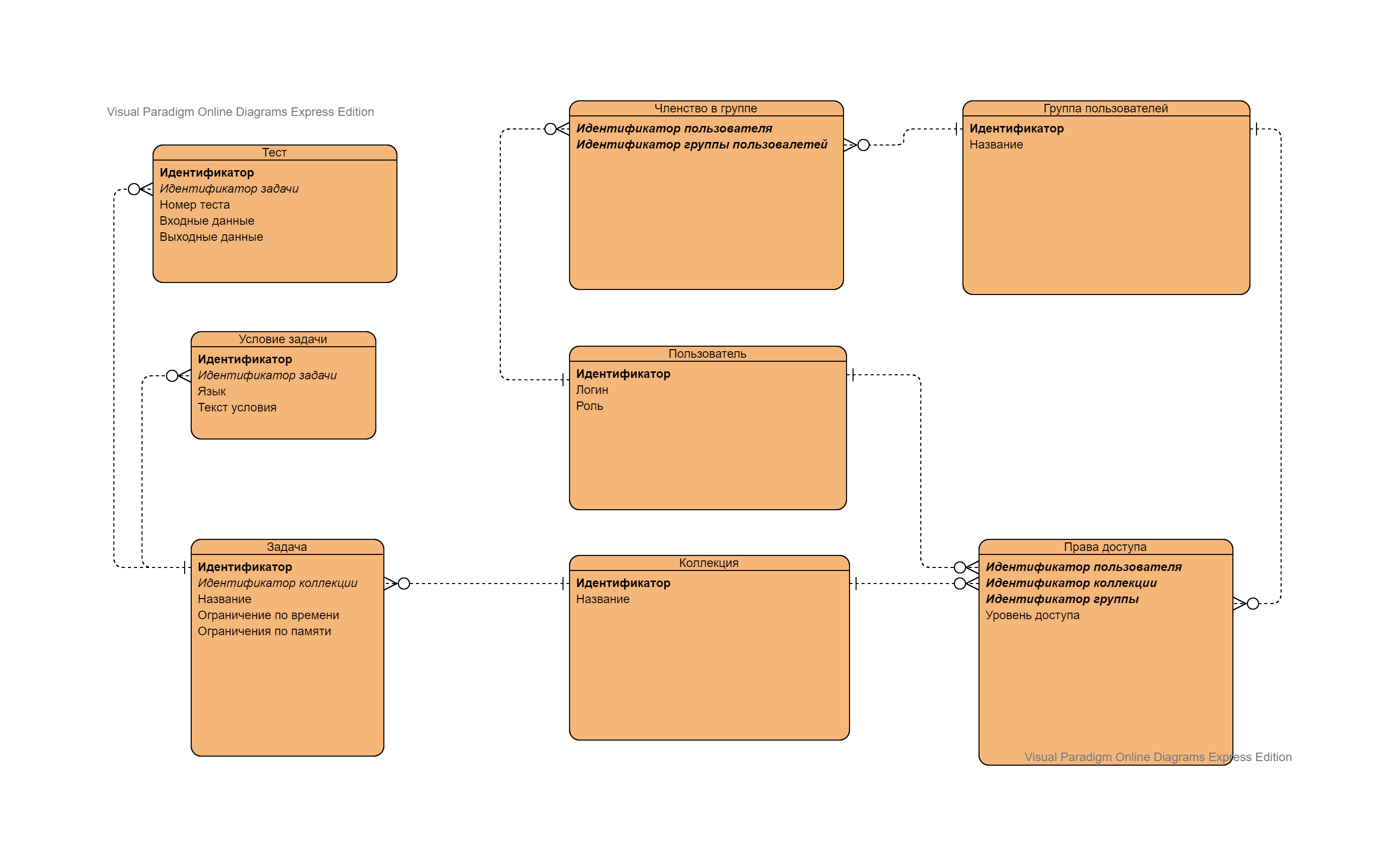
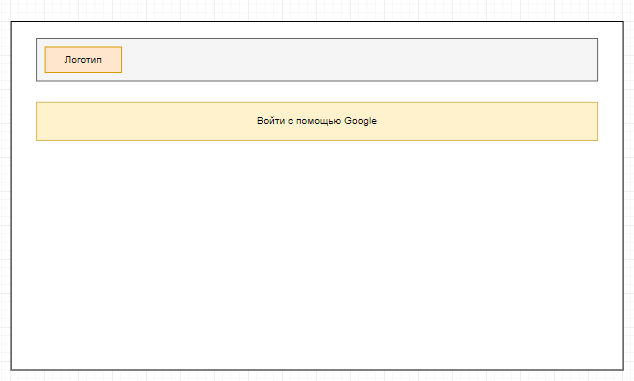
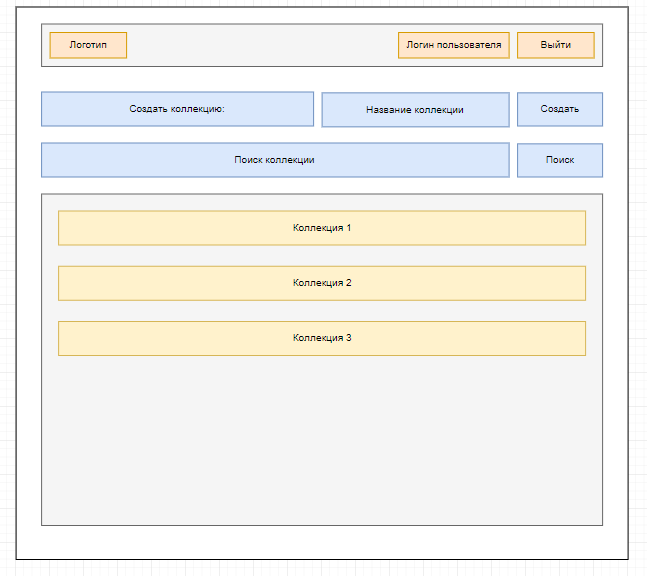
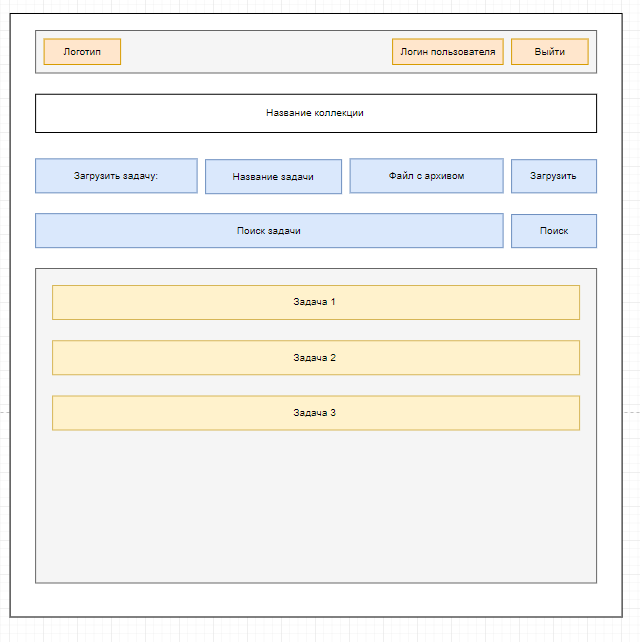
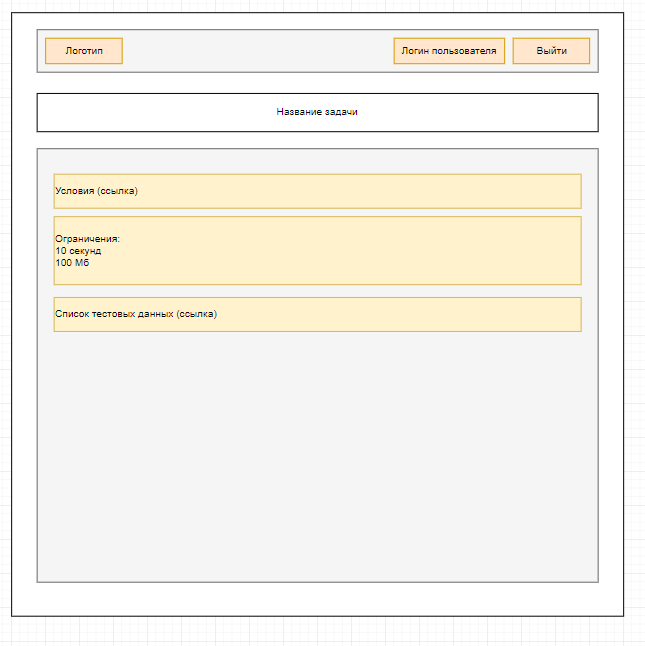
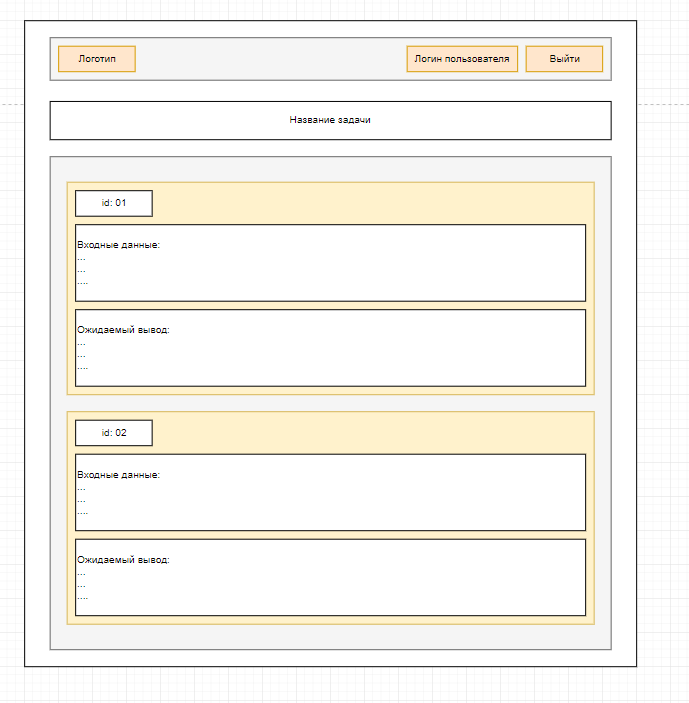
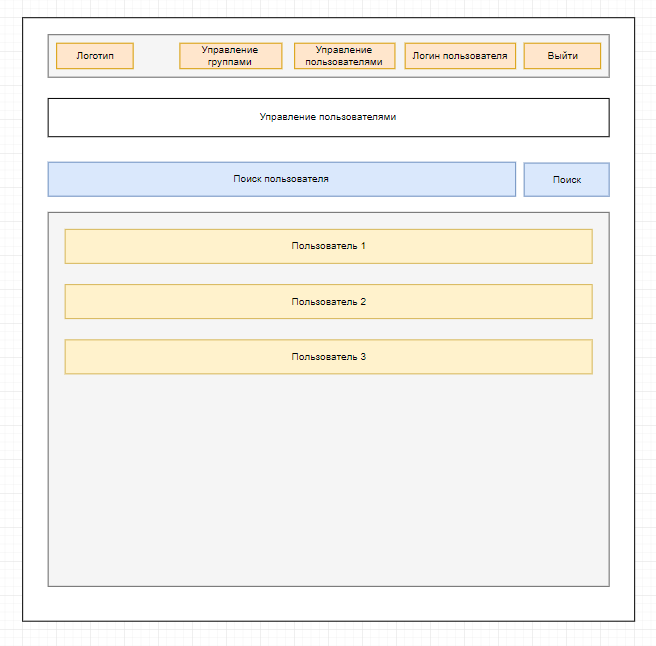
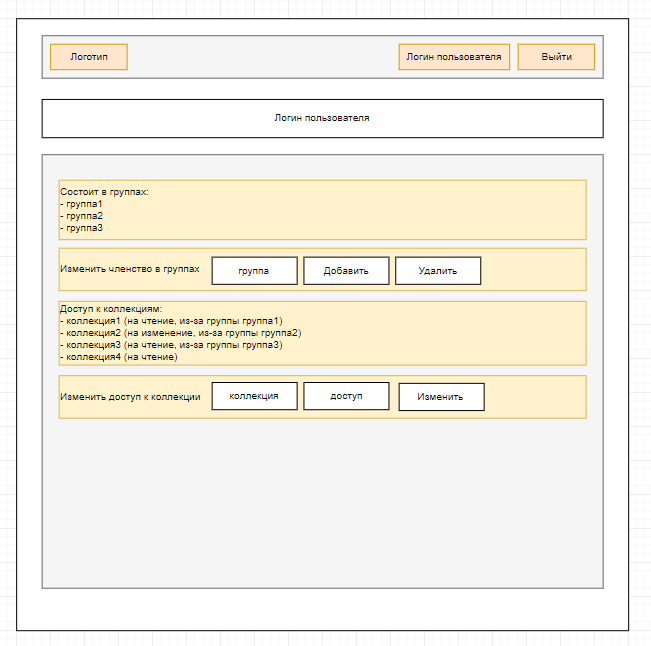
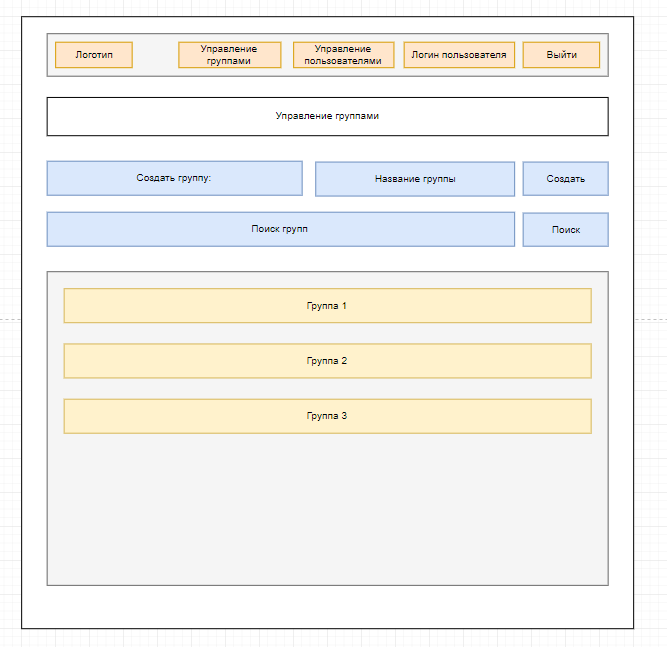
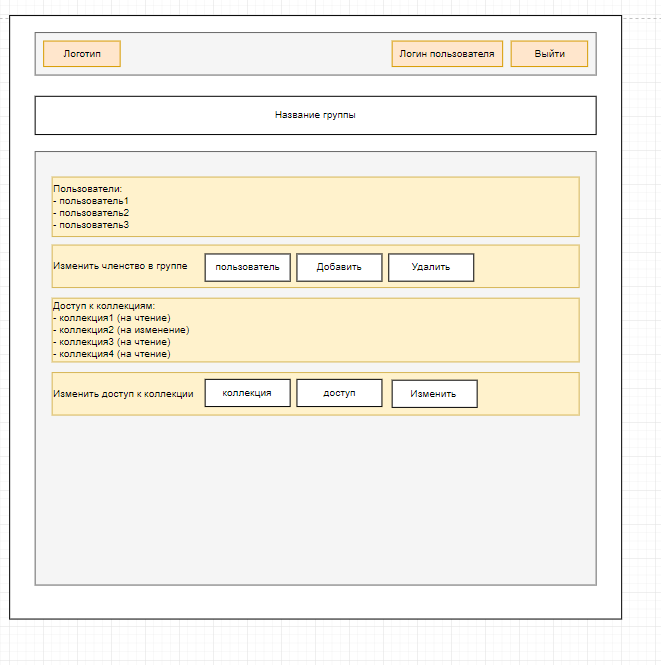
Диаграмма сущностей.



Прототипы экранных форм

1. Форма входа. Форма, которая показывается всем неавторизованным пользователям и содержит кнопку, ведущую на авторизацию через Google OAuth.  
   
2. Форма списка коллекций. Содержит список коллекций, поле и кнопку для фильтрации, поле и кнопку для добавления новой коллекции.  
   
3. Форма списка задач. Содержит список задач конкретной коллекции, поле и кнопку для фильтрации, поля и кнопку для добавления новой задачи.  
   
4. Форма с информацией о задаче. Содержит информацию об ограничениях задачи, ссылку на условия задачи, ссылку на список тестовых данных.  
   
5. Форма списка тестовых данных. Содержит информацию о тестах задачи: порядковый номер, входные данные, выходные данные.  
   
6. Форма списка пользователей. Содержит список пользователей и поля для его фильтрации.   
   
7. Форма редактирования прав пользователя. Содержит информацию о членстве пользователя в группах и кнопки для его изменения. Также содержит список доступов к коллекциям с указанием типа (чтение, изменение) и источника доступа (доступ группы или доступ отдельного пользователя), и кнопки для его изменения.  
   
8. Форма списка групп пользователей. Содержит список групп пользователей с полями для фильтрации и создания новой группы.  
   
9. Форма редактирования группы. Содержит информацию о членах группы с полями для изменения. Также содержит изменяемый список доступов к коллекциям.  
   

Разработка API системы.

1. Начало авторизации пользователя api/authorize/begin.  
   Входные данные: нет.  
   Выходные данные: перенаправление на страницу авторизации Google OAuth.
2. Завершение авторизации пользователя api/authorize/commit.  
   Входные данные: данные от Google OAuth.  
   Выходные данные: userId, token.

**token необходимо передавать в заголовке Authorization всех запросах ниже.**

1. Получение списка коллекций api/collections?filter=фильтр  
   Входные данные: фильтр по которому искать коллекции.  
   Выходные данные: массив с объектами коллекций {id: 1, title: ``}.
2. Добавление новой коллекции POST api/collections  
   Входные данные: название коллекции.  
   Выходные данные: id новой коллекции.
3. Получение списка задач внутри коллекции api/collections/{id}/problems?filter=фильтр  
   Входные данные: id коллекции, фильтр по которому искать задачи.  
   Выходные данные: массив с объектами задач {id: 1, title: ``}.
4. Добавление новой задачи POST api/collections/{id}/problems  
   Входные данные: структурированный zip-архив с информацией о задаче в виде набора байт, название задачи.  
   Выходные данные: id новой задачи.
5. Получение информации о задаче api/collections/{id}/problems/{problemId}  
   Входные данные: id коллекции, id задачи  
   Выходные данные: объект задачи {id: 1, title:``, statementUrl: ``, limits: {`memory`: `1024`, `time`: `10`}}
6. Получение информации о тестовых данных для задачи api/collections/{id}/problems/{problemId}/tests  
   Входные данные: id коллекции, id задачи  
   Выходные данные: массив с объектами тестов {id: 1, input:``, output:``}
7. Получение списка пользователей api/users?filter=фильтр  
   Входные данные: фильтр, по которому искать пользователей.  
   Выходные данные: массив с объектами пользователей {id: 1, login: ``}
8. Получение данных о пользователе api/users/{userId}

Входные данные: userId пользователя

Выходные данные: объект пользователя:   
{id: 1, login: ``,   
groups: [{id: 123, name: ``, roles: [{id: 124, collectionId: 456, access: `Read`},] }, …],  
 roles: [ {id: 123, collectionId: 345, access: `Read`}, …]  
}

1. Изменение членства в группах \ ролях: PATCH api/users/{userId}  
   Входные данные: userId пользователя, объект пользователя {groups: […], roles: […]}

Выходные данные: нет.

1. Получение списка групп пользователей api/groups?filter=фильтр  
   Входные данные: фильтр, по которому искать группы.  
   Выходные данные: массив с объектами групп {id: 1, name: ``}
2. Получение данных о группе api/groups/{groupId}  
   Входные данные: groupId группы.

Выходные данные: объект группы {id:1, name: ``, users: [{id: 123, login: ``}, …],roles: [ {id: 123, collectionId: 345, access: `Read`}, …]}

1. Изменение принадлежности группы \ ролей: PATCH api/groups/{groupId}  
   Входные данные: groupId группы, объект группы {users: […], roles: […]}

Выходные данные: нет.

Иерархическая структура работ.

1. Требования:
   1. Сбор требований.
   2. Составление требований.
   3. Согласование и утверждение требований.
2. Проектирование:
   1. Проектирование UI.
   2. Проектирование API.
   3. Проектирование схемы БД.
3. Разработка:
   1. Разработка сущностей БД.
   2. Разработка API:
      1. Разработка API авторизации (1 – 2).
      2. Разработка API коллекций (3 – 4).
      3. Разработка API задач. (4 – 8)
      4. Разработка API пользователей и ролей (9 – 11)
      5. Разработка API групп (10-14)
   3. Разработка UI:
      1. Разработка UI авторизации.
      2. Разработка UI коллекций.
      3. Разработка UI задач.
      4. Разработка UI пользователей.
      5. Разработка UI групп.
4. Тестирование.
5. Создание документации.
6. Внедрение.

Beta-версия:

1. Разработка авторизации: сущности, API и UI.
2. Разработка подсистемы задач: сущности, API и UI.
3. Тестирование.

Финальная версия:

1. Разработка подсистемы коллекций: сущности, API и UI.
2. Разработка подсистемы пользователей и ролей: сущности, API и UI.
3. Разработка подсистемы групп: сущности, API и UI.
4. Тестирование.
5. Создание документации.
6. Внедрение.

PERT.

Сущностей: 8

Элементов интерфейса: 9

API: 14

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | M (наиболее вероятно) | O (минимально возможное) | P (пессимистическая оценка) | СКО | E (средняя трудоёмкость) | Количество работ |
| UI | 6 | 3 | 10 | 1.2 | 6.2 | 9 |
| API | 5 | 2 | 14 | 2.0 | 6.0 | 14 |
| Сущность | 2 | 1 | 4 | 0.5 | 2.2 | 8 |
| Итого: |  |  |  | 8.4 | 156.8 |  |

Суммарная оценка, которую не превысим с вероятностью 95% = 156.8 + 2 \* 8.4 = 173.6 чел. \* час.

Среднее число рабочих дней в месяц – 21, среднее число часов, затраченных на проект в рабочий день – 5, отсюда среднее число часов, затраченных на проект, в месяц – 105. Продолжительность проекта составит 1.6 месяца.

