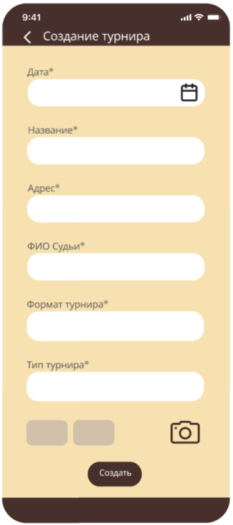
1. Прототипы экранных форм (не менее 7)

По каждой форме:

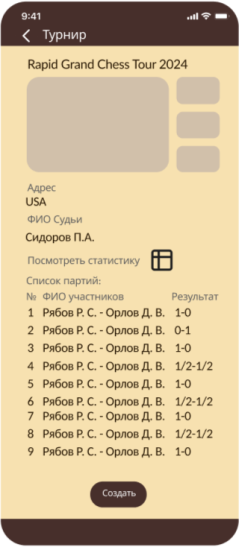
* название,
* краткое описание для чего нужна, какие элементы для чего нужны,
* картинка

1. Экран создания турниров
   * Дата турнира;
   * Название;
   * Адрес проведения;
   * ФИО судьи;
   * Формат турнира;
   * Тип турнира.



1. Экран турнира (отображает текущее состояние турнира, пары игроков и статистику внутри турнира)

* Список участников;
  + - ФИО участника;
    - Кол-во очков;
    - Бухгольц первичный и вторичный(Для некоторых типов турнира, используется для определения победителя, в случае равенства очков у нескольких претендентов на место);
    - Адрес проживания.
* Список игровых пар;
* Номера игровых столов.



1. Экран регистрации участников на турнир
   * ФИО участника;

* Рейтинг;
* Адрес проживания;
* Команда участника(если командный турнир).



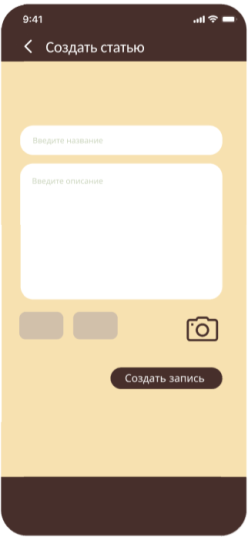
1. Экран статистики игр
   * ФИО участника;
   * Наличие подписки;

* Рейтинг;
* Адрес проживания;
* Команда участника(если командный турнир);
* Список партий с результатом игры;
* ФИО соперников.



1. Экран создания статей (новостная лента)

* Название статьи;
* Текст статьи;
* Фотографии(если надо).



1. Экран справочной информации (справочная информация о правилах шахмат и правилах турниров)

* Название статьи;
* Текст статьи;
* Фотографии.

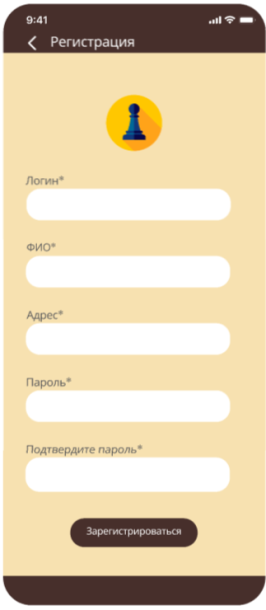


1. Экран анализа партий

* ФИО участников;
* Партия;
* Список ходов;
* Оценка текущей позиции.

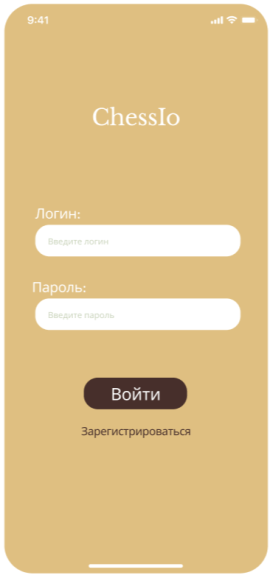
1. Экран регистрации пользователя

* логин;
* пароль;
* ФИО;
* Адрес.



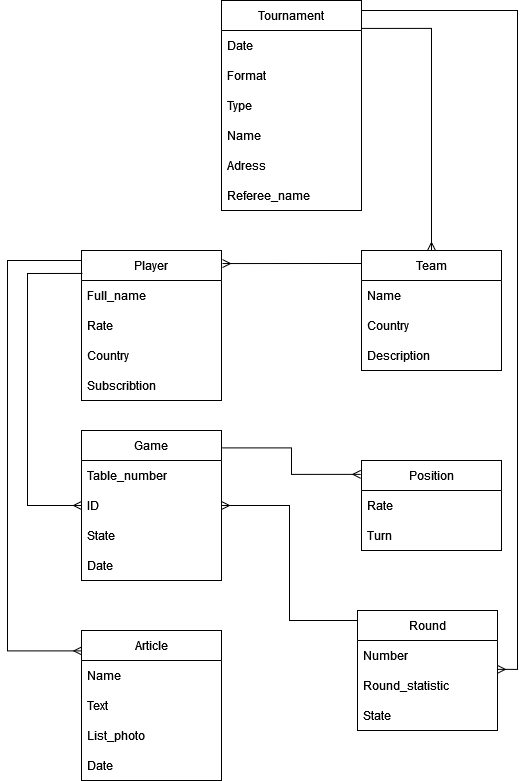
1. Экран авторизации пользователя

* логин;
* пароль.



1. Диаграмма сущностей (ER) (не менее 7 сущностей)

* указать основные атрибуты сущностей
* Связи М:М реализовать через сущность-связку



1. Разработка api системы (не менее 10 функций)

* название функции
* описание действий
* входная информация
* выходная

**Авторизация.**

Описание: введенный пользователем пароль хешируется и сравнивается с сохраненным на сервере хеш-кодом для введенного логина.

Входная информация:

* Логин;
* Пароль.

Выходная информация:

* Токен авторизации;
* Ошибка.

**Регистрация.**

Описание: проверяется существование введенного логина. Введенный пользователем пароль хешируется и сохраняется на сервере.

Входная информация:

* Логин;
* Пароль;
* ФИО;
* Адрес.

Выходная информация:

* Сообщение об успешной регистрации;
* Ошибка.

**Создать турнир.**

Описание: регистрация турнира в системе.

Входная информация:

* + Дата турнира;
  + Название;
  + Адрес проведения;
  + ФИО судьи;
  + Формат турнира;
  + Тип турнира.

Выходная информация:

* Сообщение об успешном создании турнира;
* Ошибка.

**Записать игрока в турнир.**

Описание: система отправляет заявку на участие в турнире, судья турнира получает уведомление.

Входная информация:

* + ФИО участника;
* Рейтинг;
* Адрес проживания;
* Команда участника(если командный турнир).
* ID турнира

Выходная информация:

* Сообщение об успешной отправке;
* Ошибка.

**Подтвердить запись игрока в турнир.**

Описание: подтверждение заявки на участие в турнире.

Входная информация:

* Номер заявки;
* Вердикт.

Выходная информация:

* Сообщение об успешной регистрации;
* Ошибка.

**Записать игрока в команду.**

Описание: система отправляет заявку на вступление в команду, капитан команды получает уведомление.

Входная информация:

* + ФИО участника;
* Рейтинг;
* Адрес проживания;
* ID команды

Выходная информация:

* Сообщение об успешной отправке;
* Ошибка.

**Подтвердить запись игрока в команду.**

Описание: подтверждение заявки на участие в команде.

Входная информация:

* Номер заявки игрока в команду;
* Вердикт.

Выходная информация:

* Сообщение об успешной регистрации;
* Ошибка.

**Записать результат партии.**

Описание: система сохраняет результат партии.

Входная информация:

* Результат партии;
* Список ходов.

Выходная информация:

* Сообщение об успешной записи;
* Ошибка.

**Начать новый раунд.**

Описание: система генерирует новые игровые пары на основе типа турнира.

Входная информация: -

* ID турнира

Выходная информация:

* Турнирная сетка.

**Создать статью.**

Описание: система проводи цензурирование статьи и выдает вердикт.

Входная информация:

* Название статьи;
* Текст статьи;
* Фотографии(если надо).

Выходная информация:

* Сообщение об успешной публикации;
* Ошибка.

**Получить статьи для просмотра.**

Описание: система ищет статьи соответствующие заданным параметрам.

Входная информация:

* Список параметров фильтрации и сортировки.

Выходная информация:

* Список статей.

**Анализ партии.**

Описание: система отправляет запрос к api stockfish.

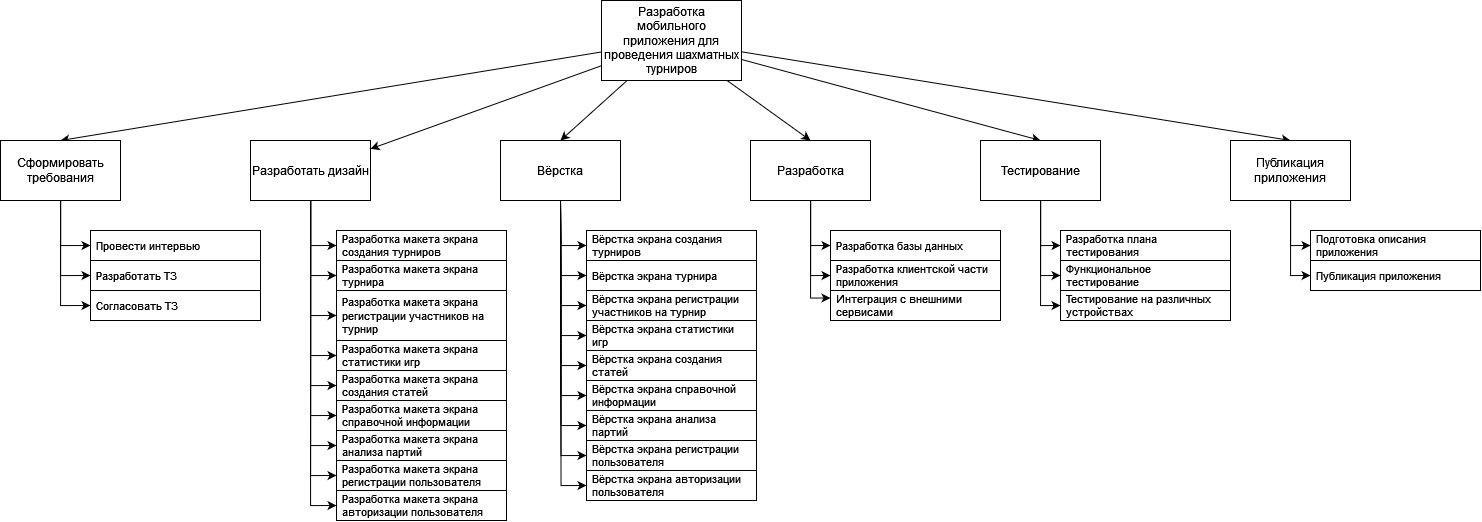
Входная информация:

* Список ходов;

Выходная информация:

* Партия;
* Список ходов;
* Оценка текущей позиции.

1. Иерархическая структура работ (ИСР) (не менее 3 уровней). В ней должны быть указана разработка всех элементов из пункта выше + сопутсвующие работы (разработка ТЗ, приемо-сдаточные испытания и т.д.)



1. Оценить время выполнения проекта по методу PERT.

* Для оценки необходимо количественно оценить состав работ: указать кол-во сущностей, форм и методов api
* Для каждого вида элементов определить пессимистиченые, оптимистиченые и средние трудозатраты
* Вывести общие трудозатраты проекта в чел. x мес.

*чел\*мес*

1. Базовое расписание в виде диаграммы Ганта. В базовом расписании должны быть отражены все элементы ИСР

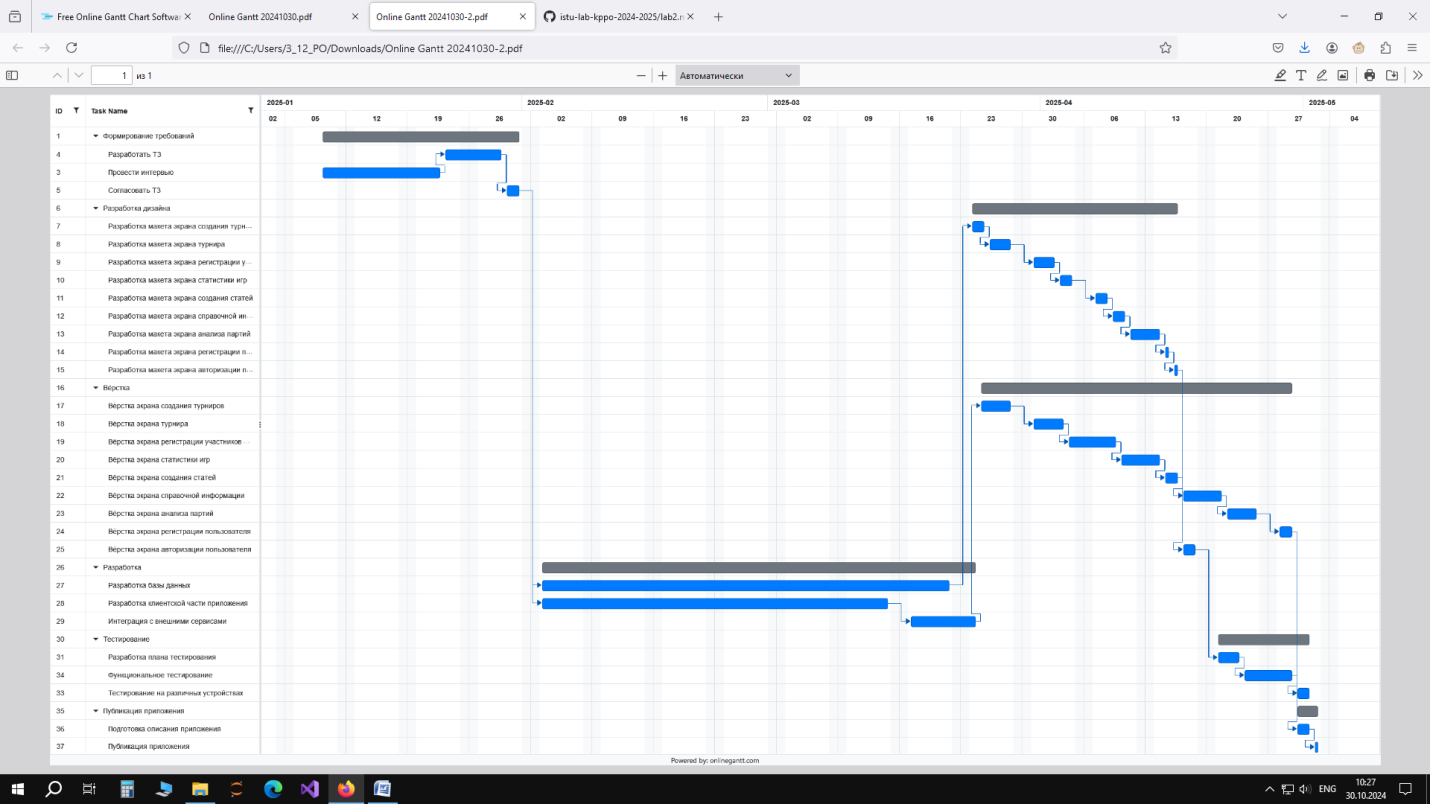


Рисунок Диаграмма Ганта

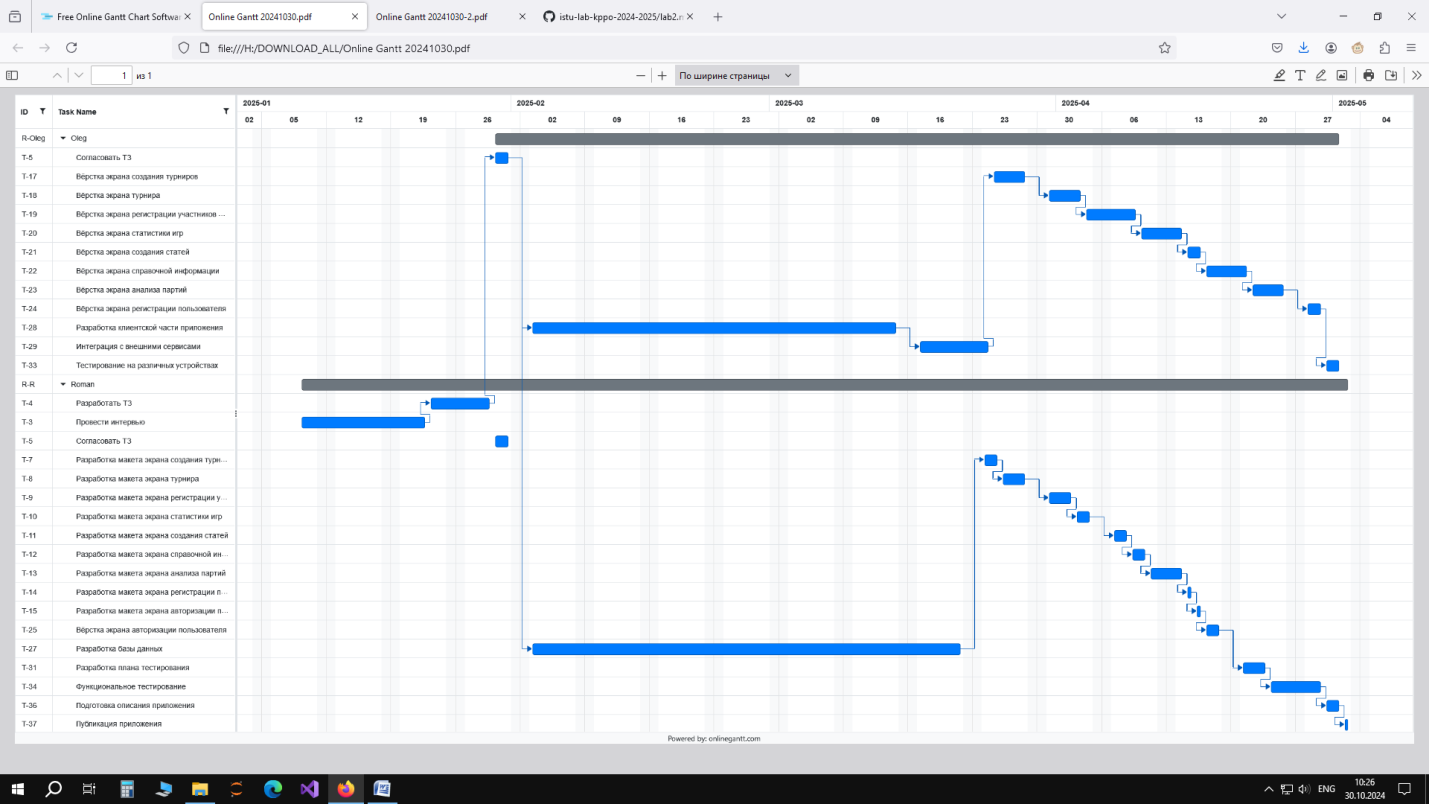


Рисунок Диаграмма Ганта (представление ресурсов)