­­ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

КУРСОВАЯ РАБОТА

Тема: Система планирования хода выполнения проекта

Работу выполнил студент Гончаров Владимир Андреевич группы K3140

(фамилия, имя, отчество) (номер группы)

Руководитель Белозубов Александр Владимирович

(фамилия, имя, отчество)

Работа защищена " " 2018 г. с оценкой

Подписи членов комиссии:

САНКТ-ПЕТЕРБУРГ, 2018 г.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

**АННОТАЦИЯ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | | Гончаров Владимир Андреевич | | | | | |
| (Фамилия, И.О.) | | | | | | | |
| Факультет | Инфокоммуникационных технологий | | | | | | |
| Кафедра | ИКТ | | | | | Группа | K3140 |
| Направление (специальность) | | | | | 09.03.03 | | |
| Руководитель | | | Белозубов А.В., Университет ИТМО, доцент фак. ПииКТ, К.Т.Н | | | | |
| ( Фамилия, И.О., место работы, должность, ученое звание, степень ) | | | | | | | |
| Дисциплина | | Информатика | | | | | |
|  | | | | | | | |
| Наименование темы | | | | Система планирования хода выполнения проекта | | | |
|  | | | |  | | | |

**ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1. Цель и задачи работы** | Предложены студентом | | | Сформулированы при участии студента | |
| Определены руководителем | |
| Требуется разработать модуль системы планирования хода выполнения проекта и структуру JSON-файла, содержащего данные объектов системы. Для каждого проекта необходимо хранить его название, дату начала, планируемую дату окончания, информацию о его руководителе, включающую имя, фамилию, отчество при наличии и дату рождения, список исполнителей проекта, характеризующихся краткой информацией, включающей имя, фамилию, отчество при наличии и дату рождения. Каждый проект характеризуется этапами выполнения, каждый из которых включает название этапа, дату начала, дату окончания, статус выполнения этапа, информацию об исполнителе этапа и краткое текстовое описание. Модуль должен предоставлять возможность получения данных о списке проектов, хранящихся в базе данных, возможность перехода к интерфейсу просмотра общей информации о каждом проекте с указанием хода его выполнения, а также перехода к просмотру информации об отдельном этапе. База данных представляется JSON-файлом. Средства доступа к базе данных: Python + Django. | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **2. Характер работы** | | | | | |
|  | Расчет | | | Конструирование | |
|  | Моделирование | | | Другое, | |
| Разработка ПО | |
| **4. Содержание работы** | |  | | | |
| Введение; Анализ и уточнение задания; Инфологическая модель базы данных; Описание | | | | | |
| интерфейсов пользователей системы; Описание архитектуры системы; Описание | | | | | |
| программного кода системы; Заключение | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **5. Выводы** | | |  | |  |
| Создана база данных организаций, пользователей, заказов; разработан | | | | | |
| модуль системы, предоставляющий доступ к данным о компании и ее заказах. | | | | | |
|  | | | | | |

|  |  |  |
| --- | --- | --- |
| Руководитель | |  |
| подпись, дата | | |
| Студент |  | |
| подпись, дата | | |

Оглавление

[Введение 6](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935313)

[1. Анализ и уточнение задания 7](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935314)

[2. Инфологическая модель базы данных 8](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935315)

[3. Описание интерфейсов пользователей системы 10](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935316)

[4. Описание архитектуры системы 20](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935317)

[5. Описание программного кода системы 25](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935318)

[Заключение 28](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935319)

[Используемая документация 29](file:///C:\Users\mamuH\Downloads\Пояснительная%20записка%20(2).docx#_Toc515935320)

# Введение

Требуется разработать модуль системы планирования хода выполнения проекта и структуру JSON-файла, содержащего данные объектов системы. Для каждого проекта необходимо хранить его название, дату начала, планируемую дату окончания, информацию о его руководителе, включающую имя, фамилию, отчество при наличии и дату рождения, список исполнителей проекта, характеризующихся краткой информацией, включающей имя, фамилию, отчество при наличии и дату рождения. Каждый проект характеризуется этапами выполнения, каждый из которых включает название этапа, дату начала, дату окончания, статус выполнения этапа, информацию об исполнителе этапа и краткое текстовое описание. Модуль должен предоставлять возможность получения данных о списке проектов, хранящихся в базе данных, возможность перехода к интерфейсу просмотра общей информации о каждом проекте с указанием хода его выполнения, а также перехода к просмотру информации об отдельном этапе. База данных представляется JSON-файлом. Средства доступа к базе данных: Python + Django.

# Анализ и уточнение задания

Требования по заданию:

* Модуль должен предоставлять возможность получения данных о списке проектов, хранящихся в базе данных
* Возможность перехода к интерфейсу просмотра общей информации о каждом руководителе, исполнителе и этапе проекта.

После уточнения задания были получены следующие дополнительные требования к системе:

* Осуществление добавление заказов;
* Осуществление добавления организаций;
* Осуществление авторизации пользователей;

**Аналоги:**

Асана: https://asana.com/  
Функции:

* Возможность создание заданий для каждого проекта
* Возможность назначения ответственного лица за выполнение каждого задания
* Наличие календаря крайних сроков
* Возможность просмотра текущих задач

Недостатки:

* Сервис является платным.
* Избыточность функционала

# Инфологическая модель базы данных

На этапе разработки инфологической модели выделено 4 сущности:

* Администратор (менеджер);
* Проект;
* Этап;
* Участник;

Сущности имеют следующую структуру:

**Администратор (менеджер)**

* Логин;
* Пароль;
* Уровень доступа;
* Проект;

**Проект**

* ID;
* Название;
* Дата начала;
* Дата окончания;

**Участник**

* ID;
* Список проектов;
* Имя;
* Фамилия;
* Отчество;
* Дата рождения;
* Роль в проектах;

**Этап**

* ID;
* Проект;
* Номер;
* Название;
* Описание;
* Дата начала;
* Дата конца;
* Статус

# Описание интерфейсов пользователей системы

В системе имеется 2 роли — администратор и менеджер системы.

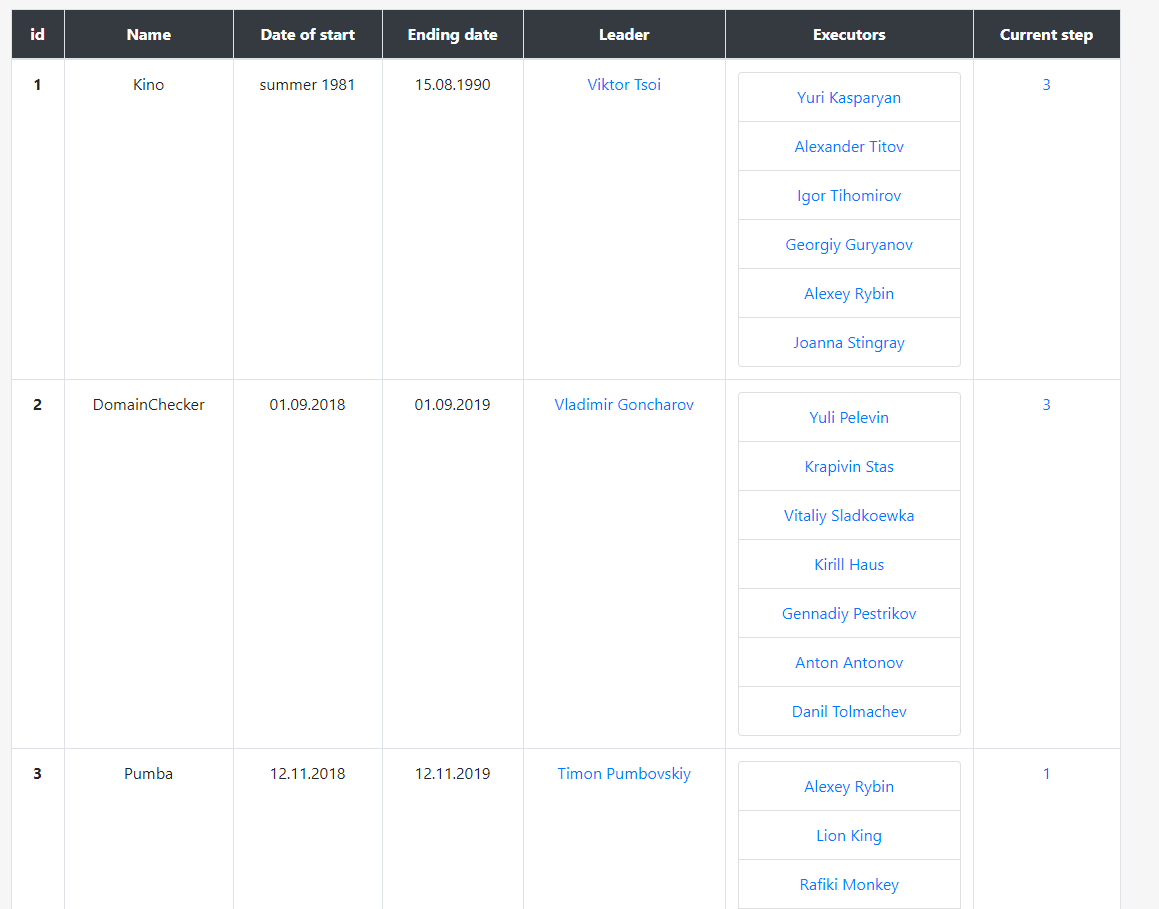
**Администратор системы**

* Добавление любой информации в систему.

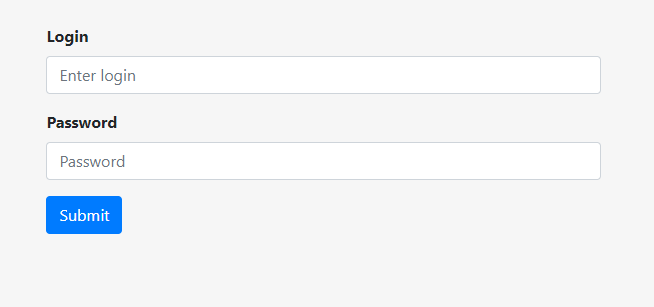
**Менеджер системы**

* Добавление участника и этапа в проект.

На *Рисунке 1.1* представлен вид главной страницы.

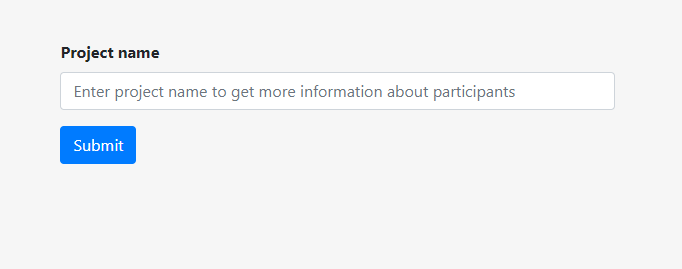
  
*Рисунок 1.1 – главная страница*

На *Рисунке 1.2* представлен вид страницы авторизации. Чтобы авторизоваться, введите данные, а затем нажмите кнопку «Submit»



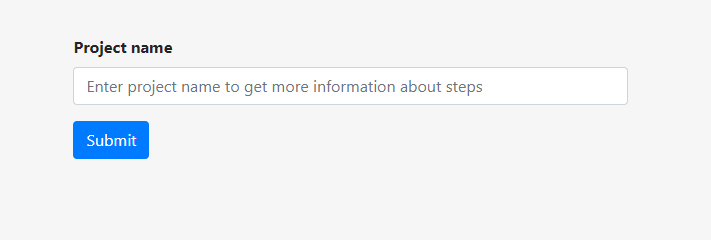
*Рисунок 1.2 – страница авторизации*

На *Рисунке 1.3* представлен вид страницы поиска информации по участникам проекта.



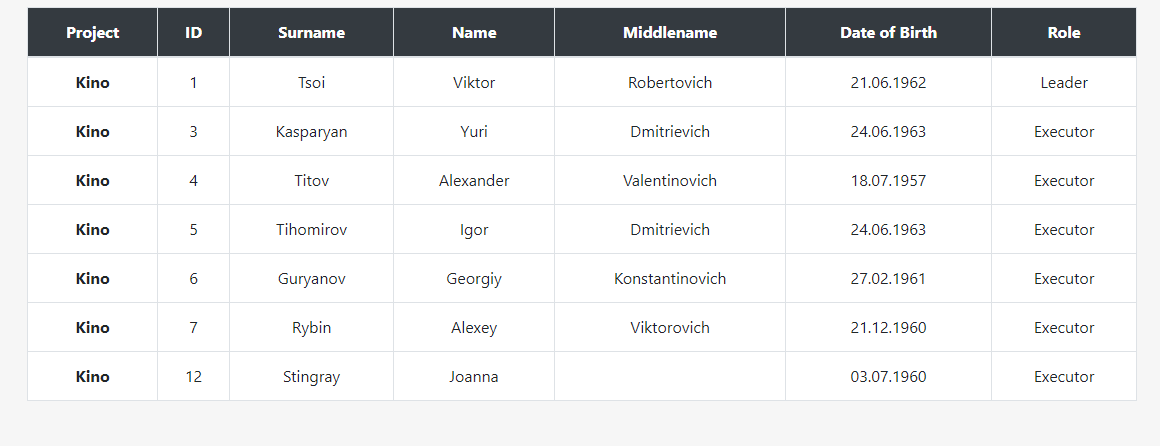
*Рисунок 1.3 – страница поиска информации об участниках*

На *Рисунке 1.4* представлен вид страницы поиска информации по этапам проекта.



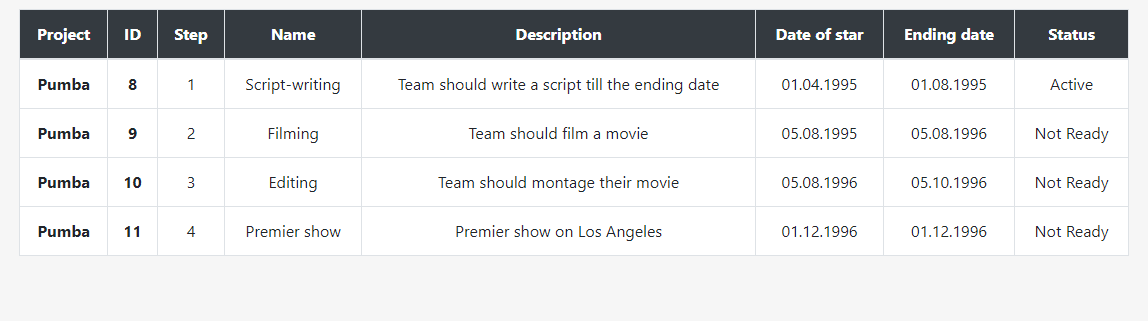
*Рисунок 1.4 – страница поиска информации об этапах*

На *Рисунке 1.5* представлен вид страницы вывода информации об участниках проекта



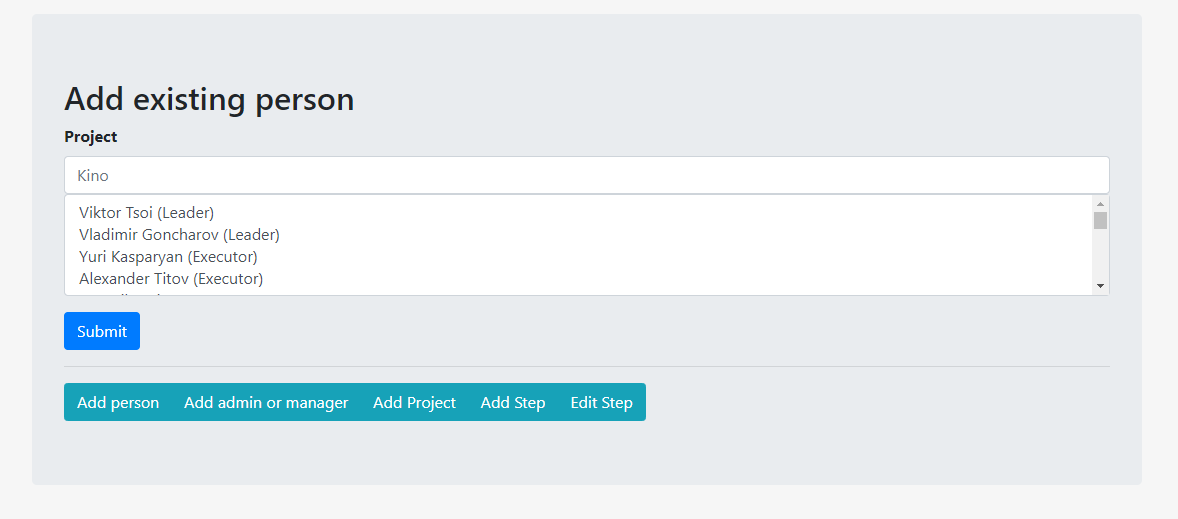
*Рисунок 1.5 – страница вывода информации об участниках*

На *Рисунке 1.6* представлен вид страницы вывода информации об этапах проекта

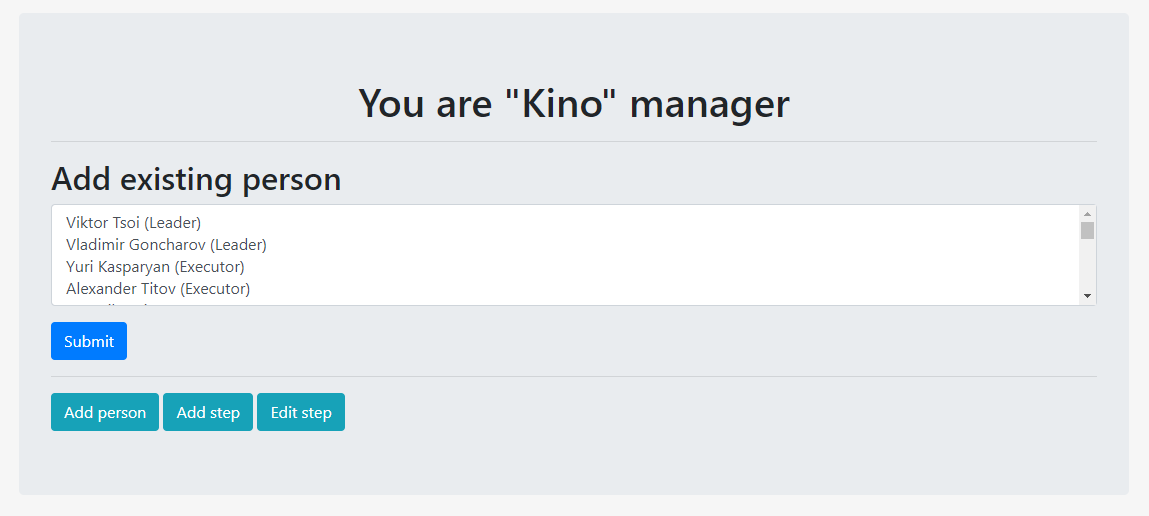


*Рисунок 1.6 – страница вывода информации об этапах*

На *Рисунке 2* представлен вид страницы администратора сервиса.

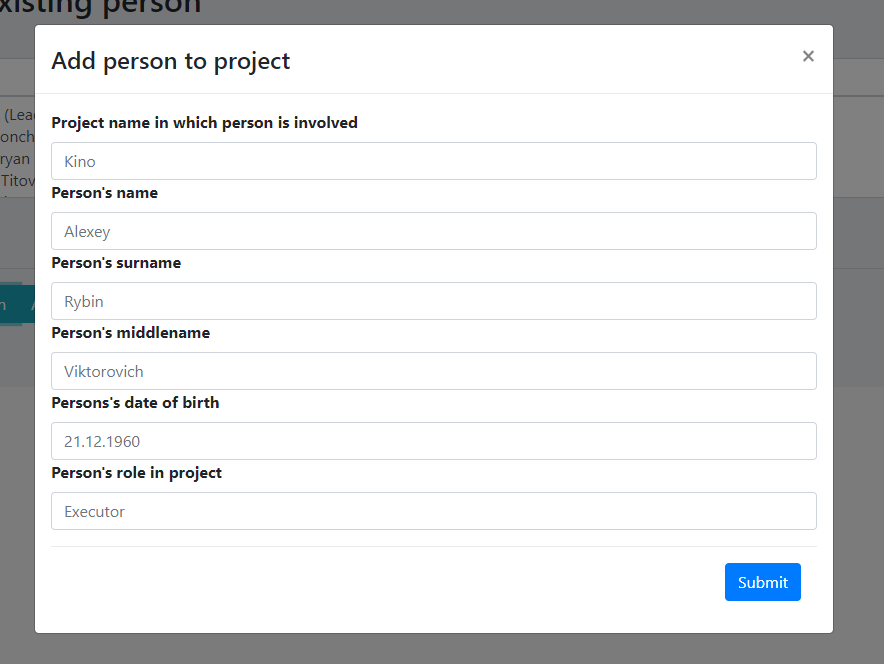
  
*Рисунок 2 – страница администратора сайта*

На *Рисунке 3* представлен вид страницы менеджера проекта.



*Рисунок 3 – страница менеджера проекта*

На *Рисунке 4* представлен вид добавления нового человека в проект.

  
*Рисунок 4 – Карточка добавления человека*

На *Рисунке 5* представлен вид добавления нового администратора или менеджера.



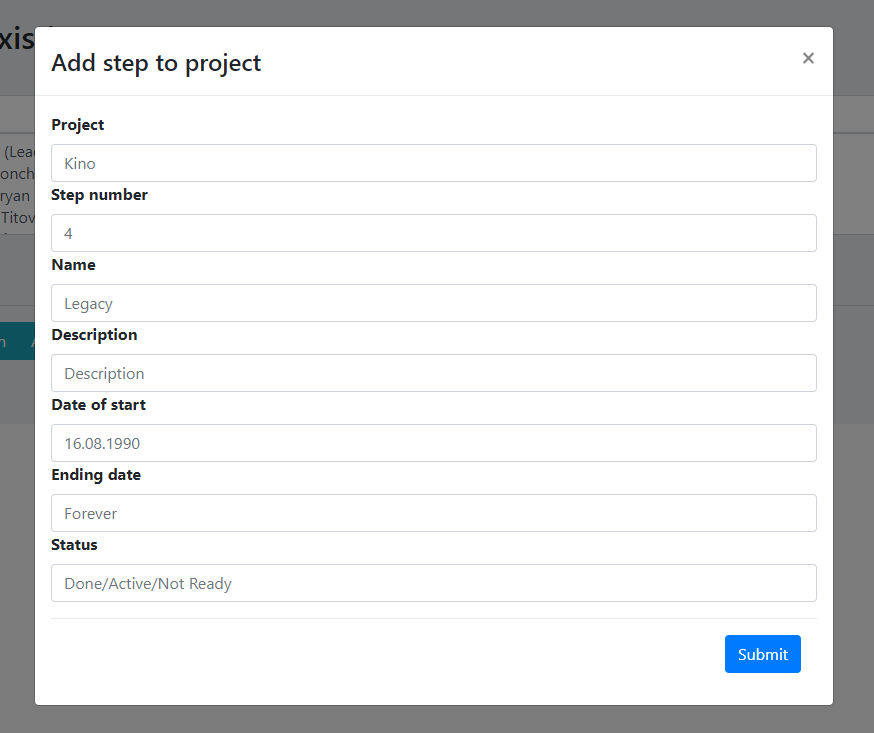
*Рисунок 5 – Карточка добавления администратора (менеджера)*

На *Рисунке 6* представлен вид добавления нового проекта.



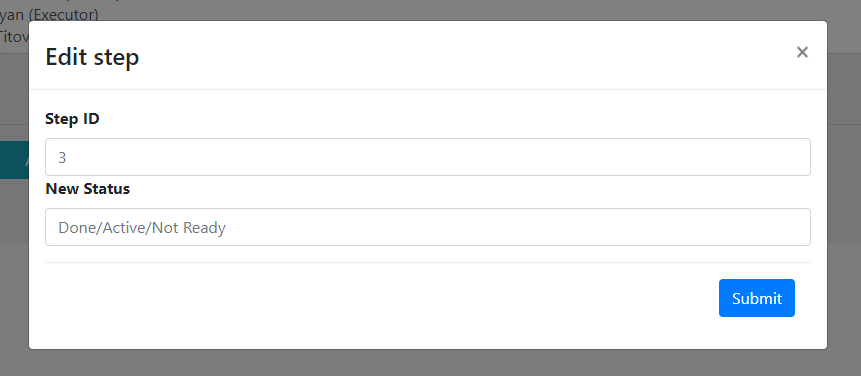
*Рисунок 6 – Карточка добавления проекта*

На *Рисунке 7* представлен вид добавления нового этапа.



*Рисунок 7 – Карточка добавления этапа*

На *Рисунке 8* представлен вид редактирования существующего этапа.



*Рисунок 8 – Карточка редактирования этапа*

# Описание архитектуры системы

Итоговая архитектура системы представлена в *Таблице 1*.

|  |  |  |
| --- | --- | --- |
| Путь | Имя файла | Назначение |
| projectcom | settings.py | Хранение настроек Django. |
|  | urls.py | Подключение файла “main/urls.py” |
| main | static/ | Хранение файлов фреймворка Bootstrap |
|  | urls.py | Ассоциация URL-адресов и вызываемых функций |
|  | views.py | Обеспечение функциональности системы |
| main/templates/main | adminpage.html | Шаблон страницы администратора (менеджера) |
|  | loginpage.html | Шаблон страницы авторизации |
|  | mainpage.html | Шаблон главной страницы |
|  | persons.html | Шаблон страницы поиска информации по участникам |
|  | steps.html | Шаблон страницы поиска информации по этпам |
|  | stepsinfo.html | Шаблон страницы вывода информации об этапах |
|  | wrapper.html | Общий родительский шаблон |
|  | personsinfo.html | Шаблон страницы вывода информации об участниках |
| main | projects.json | База данных проектов |
|  | admininfo.json | База данных администратор |
|  | persons.json | База данных участников |
|  | steps.json | База данных этапов |
|  | idspers.json | Список зарегистрированных id участников |
|  | idsproj.json | Список зарегистрированных id проектов |
|  | idsst.json | Список зарегистрированных id этапов |

На основе инфологической модели была сформирована структура хранения данных. Каждый тип сущности хранится в отдельном json-файле со следующими структурами:

|  |
| --- |
| **projects.json** |
| [  {  "id": "1",  "name": "Kino",  "start": "summer 1981",  "end": "15.08.1990"  },  ...  ] |
| **persons.json** |
| [  {  "id": 1,  "project": [  "Kino"  ],  "name": "Viktor",  "surname": "Tsoi",  "fath\_name": "Robertovich",  "dob": "21.06.1962",  "role": "Leader"  },  ...  ] |
| **admininfo.json** |

|  |
| --- |
| [  {  "login": "admin",  "password": "admin",  "acclevel": "admin",  "project": ""  },  {  "login": "manager",  "password": "manager",  "acclevel": "manager",  "project": "Kino"  },  ...  ] |
| **steps.json** |
| [  {  "project": "Kino",  "id": "2",  "num": "2",  "name": "Music writing",  "desc": "Band is writing music",  "dos": "01.09.1981",  "doe": "15.08.1990",  "status": "Done"  },  ...  ] |

# Описание программного кода системы

Для ускорения разработки интерфейсов был использован фреймворк Bootstrap.

**Взаимодействие с системой осуществляется следующим образом:**

Пользователь попадает на главную страницу, которая соответствует функции *index()* и шаблону mainpage.*html*.

Пользователь может перейти на страницу поиска информации об участниках проекта тогда будет вызвана функция persons() и шаблон persons.html. После ввода названия проекта будет вызвана функция personsinfo() и шаблон personsinfo.html

Пользователь может перейти на страницу поиска информации об этапах проекта тогда будет вызвана функция steps() и шаблон steps.html. После ввода названия проекта будет вызвана функция stepsinfo() и шаблон stepsinfo.html

Пользователь может перейти на страницу авторизации, тогда будет вызвана функция login() и шаблон loginpage.html

Пользователь может представиться системе с использованием логина и пароля, и перейти на страницу администратора или менеджера. При этом будет вызвана функция *auth()*. В случае ввода верных данных будет выведен шаблон adminpage.html, в случае ввода неверных данных будет возвращен шаблон loginpage.html

Находясь на странице администратора **или** менеджера пользователь может:

* добавить нового участника в проект, тогда будет вызвана функция addperson(), которая добавит нового в базу данных, и шаблон adminpage.html.
* добавить этап в проект, тогда будет вызвана функция addstep(), которая добавит новый этап в базу данных, и шаблон adminpage.html.
* изменить существующий этап, тогда будет вызвана функция editstep(), которая внесет изменения в базу данных, и шаблон adminpage.html.
* добавить существующего участника, тогда будет вызвана функция addexperson(), которая внесет изменения в базу данных, и шаблон adminpage.html.

Находясь на странице **только** администратора пользователь может:

* добавить проект, тогда будет вызвана функция addproject(), которая добавит проект в базу данных, и шаблон adminpage.html.
* добавить администратора или менеджера, тогда будет вызвана функция addadmin(), которая добавит новый этап в базу данных, и шаблон adminpage.html

# Заключение

В процессе работы был разработан модуль системы планирования хода выполнения проекта и структуру JSON-файла, содержащего данные объектов системы. Для каждого проекта хранится его название, дата начала, планируемая дата окончания, информация о его руководителе, включающая имя, фамилию, отчество при наличии и дату рождения, список исполнителей проекта, характеризующийся краткой информацией, включающей имя, фамилию, отчество при наличии и дату рождения. Каждый проект характеризуется этапами выполнения, каждый из которых включает название этапа, дату начала, дату окончания, статус выполнения этапа, информацию об исполнителе этапа и краткое текстовое описание. Модуль предоставляет возможность получения данных о списке проектов, хранящихся в базе данных, возможность перехода к интерфейсу просмотра общей информации о каждом проекте с указанием хода его выполнения, а также перехода к просмотру информации об отдельном этапе. База данных представляется JSON-файлом. Средства доступа к базе данных: Python + Django.

Общее количество строк кода: 1610, из них 503 – файлы python, 657 – html, 450 – json.

Всего разработано 8 HTML-страниц, 13 функций. Пояснительная записка содержит 20 страниц, 2 таблицы, 13 рисунков.

# Используемая документация

* Лямин А.В., Череповская Е.Н. Объектно-ориентированное программирование. Компьютерный практикум. – СПб.: Университет ИТМО, 2017. – 143 с.