|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

**Факультет «Радиотехнический»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

Расчетно-пояснительная записка

Тема: «Удаленная поддержка»

Студент: Цыпышев Т.А.

Группа ИУ5-51Б

Преподаватель: Канев А.И.

2024 г.

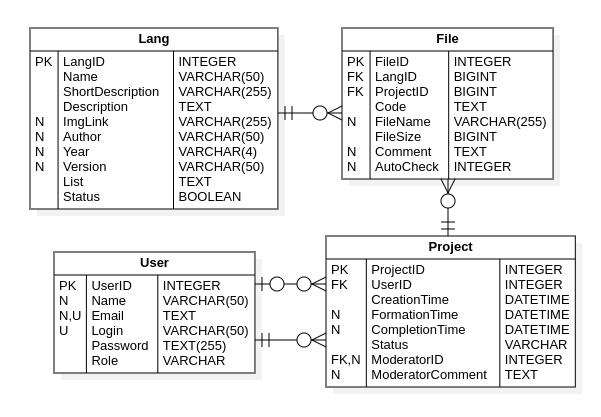


Рисунок 1 - ER диаграмма

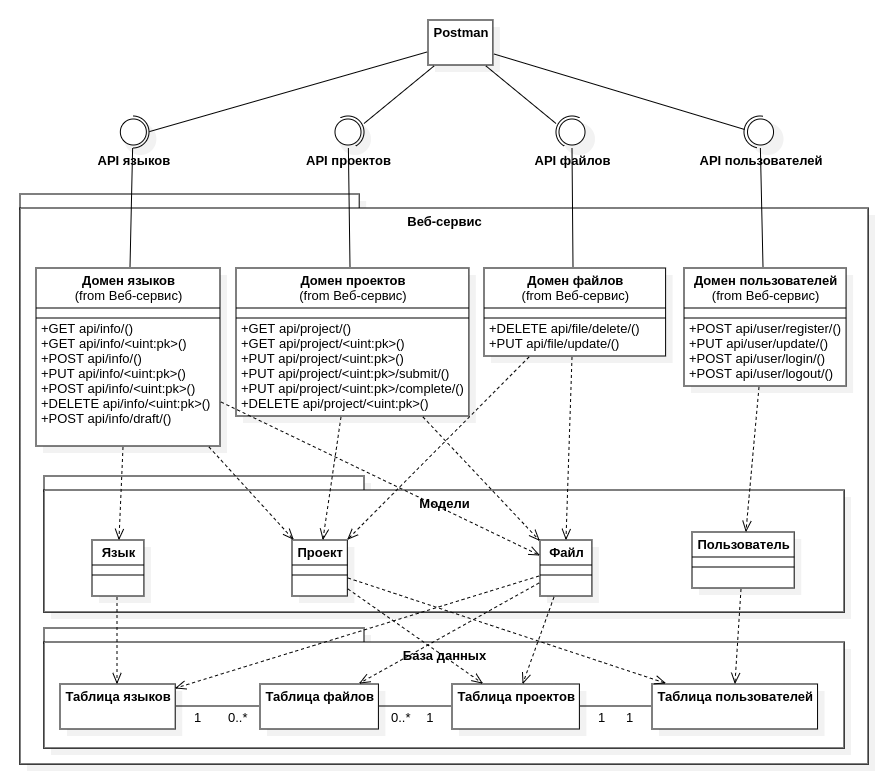


Рисунок 2 - Диаграмма классов

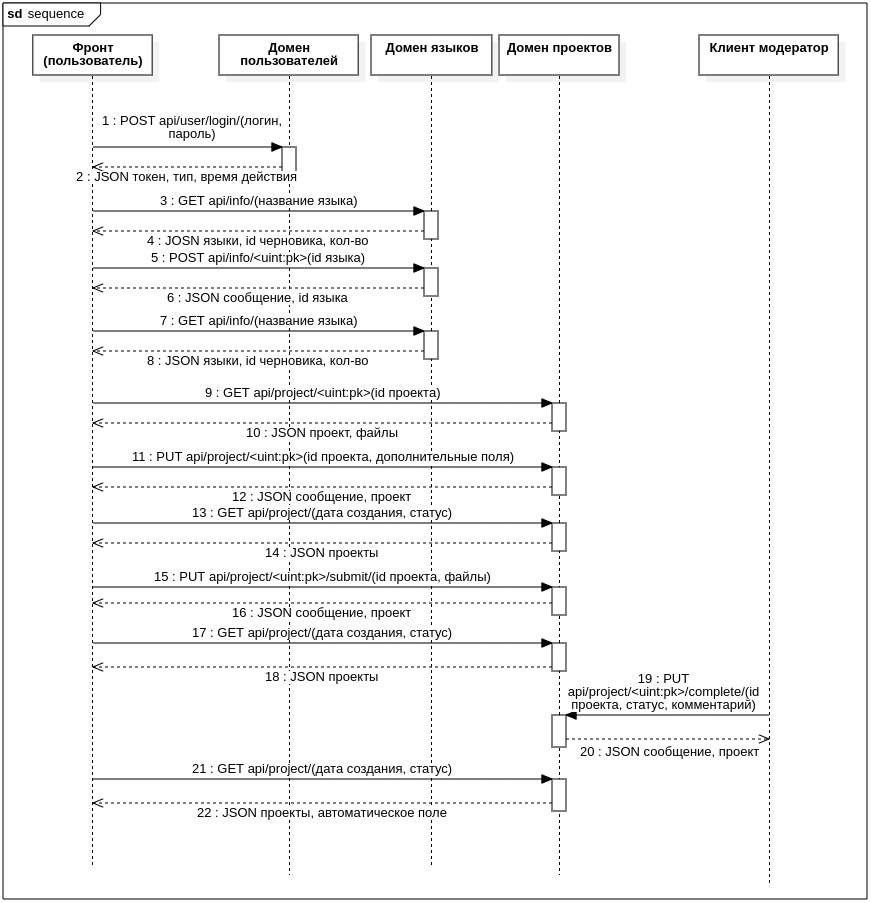


Рисунок 3 - Диаграмма последовательности

**ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

| Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  Научный руководитель |
| --- | --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_А.И. Канев  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Проверка кода студентов**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

\_

(количество листов)

| ИСПОЛНИТЕЛЬ: |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Цыпышев Тимофей Александрович |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва - 2024

1. **Введение**

Цель разработки заключается в создании универсальной платформы, которая обеспечивает удобный и эффективный процесс проверки, анализа и оптимизации студенческого кода преподавателями.

1. **Назначение разработки**

Назначение системы заключается в создании платформы для организации эффективной проверки и анализа студенческого кода. Студенты могут создавать заявки, в которых указывают, на каком языке программирования написан код, задачу и требования для проверки. Преподаватели получают доступ к списку заявок, проводят проверку и добавляют комментарии или рекомендации. После анализа преподаватель отмечает заявку как выполненную, а студент может просмотреть подробный отчет с результатами.

1. **Стадии и этапы разработки**
   1. Создание MVP и базового дизайна на основе github.com;
   2. Реализовать хранение данных в БД (Postgres);
   3. Создать веб-сервис в бэкенде на Golang (DRF);
   4. Реализовать авторизацию и хранение сессий в Redis;
   5. Разработать базовый SPA на React для гостя;
   6. Внедрение адаптивности, разработка Tauri приложения;
   7. Интерфейс студента на React с менеджером состояний Redux Toolkit;
   8. Реализовать интерфейс преподавателя в React;
   9. Реализовать нативное приложение;
   10. Развернуть веб-приложение React на Github Pages;
   11. Подготовка документацию к системе (РПЗ, ТЗ, набор диаграмм);
   12. Создать репозиторий Git для системы.
2. **Требования к функциональным характеристикам**
   1. Список HTTP методов
      1. GET Список языков
      2. GET Информация об одном языке
      3. POST Добавление нового языка
      4. PUT Изменение информации о языке
      5. POST Добавление или изменение изображения для языка
      6. DELETE Удаление языка
      7. POST Добавление языка в заявку-черновик
      8. GET Список проектов пользователя
      9. GET Информация о конкретном проекте
      10. PUT Изменение полей проекта
      11. PUT Формирование заявки студентом
      12. PUT Завершение или отклонение заявки преподавателем
      13. DELETE Удаление проекта
      14. DELETE Удаление файла из проекта
      15. PUT Изменение информации о файле
      16. POST Регистрация нового пользователя
      17. PUT Изменение информации о пользователе
      18. POST Аутентификация пользователя
      19. POST Деавторизация пользователя
   2. Меню
      1. Главная - перенаправляет на страницу 4.5
      2. Список языков программирования – перенаправляет на страницу 4.6 (вызывается метод 4.1.1)
      3. Список проектов – перенаправляет на страницу 4.9
      4. Редактирование языков программирования – перенаправляет на страницу 4.11, доступно только преподавателям
      5. Регистрация – перенаправляет на страницу 4.3 появляется только для гостей
      6. Личный кабинет – перенаправляет на страницу 4.10, только аутентифицированным пользователям
      7. Войти – перенаправляет на страницу 4.4, только для гостей, появляется только для гостей
      8. Выйти – перенаправляет на страницу 4.6 (вызывается метод 4.1.19), доступно только аутентифицированным пользователям
   3. Регистрация
      1. Доступно только гостям
      2. Отображает форму регистрации
         1. Поле имени пользователя
         2. Поле почты
         3. Поле короткого имени
         4. Поле пароля
      3. Действия
         1. Регистрация пользователя – (вызывается метод 4.1.16)
         2. Вернуться к аутентификации – перенаправляет на страницу 4.4
   4. Аутентификация
      1. Доступно только гостям
      2. Отображает форму аутентификации
         1. Поле короткого имени
         2. Поле пароля
      3. Действия
         1. Войти – (вызывается метод 4.1.18)
         2. Регистрация – перенаправляет на страницу 4.3
   5. Личный кабинет
      1. Доступно пользователю
      2. Действия
         1. Изменить данные пользователя – (вызывается метод 4.1.17)
   6. Главная
      1. Доступна всем
      2. Отображается статическая информация
         1. Назначение сервиса
         2. Контакты для связи
   7. Список языков программирования
      1. Доступна всем
      2. Отображаются элементы карточек с языками программирования, вызывается метод 4.1.1
         1. Название языков программирования
         2. Краткое описание
      3. Действия
         1. Поиск – перенаправляет на страницу 4.7, (используется метод 4.1.1), с фильтрующем параметром
         2. Подробнее – перенаправляет на страницу 4.8 (используется метод 4.1.2)
         3. Добавить в проект – добавляет язык в проект-черновик, (вызывается метод 4.1.7), только пользователи.
         4. Кнопка корзины – перенаправляет на страницу 4.9, только пользователи.
   8. Подробнее о языке программирования
      1. Доступна всем
      2. Отображается подробная информация выбранного языка, (вызывается метод 4.1.2)
   9. Проект
      1. Доступно только пользователям
      2. Отображает текущий проект-черновик пользователя, (метод 4.1.9)
         1. Список выбранных языков
         2. Дата создания проекта
         3. Дата последнего сохранения проекта
         4. Поле ввода кода
         5. Комментарии к каждому файлу
      3. Действия, доступны только в случае, если статус «черновик»
         1. Сохранить – сохраняет текущий проект-черновик, (вызывается метод 4.1.11)
         2. Удалить – удаляет проект-черновик, (вызывается метод 4.1.13)
   10. Личный кабинет
       1. Доступно пользователю
       2. Отображается список проектов (метод 4.1.8)
          1. Только проекты, созданные данным студентом
          2. Только проекты, к которым данный преподаватель закреплён как проверяющий
       3. Действия
          1. Фильтрация – фильтрует проекты по дате создания или статусу, вызывается (метод 4.1.8)
          2. Принять – сохраняет положительное решение по проекту, выполняется метод 4.1.12, доступно только преподавателю
          3. Отклонить – сохраняет отрицательное решение по проекту, вызывается метод 4.1.12, доступно только преподавателю
          4. Посмотреть подробную информацию о проекте – перенаправляет на страницу 4.9 (вызывается метод 4.1.9)
   11. Список языков программирования
       1. Доступно преподавателю
       2. Отображаются все языки программирования (вызывается метод 4.1.1)
       3. Действия
          1. Удалить – удаляет язык (вызывается метод 4.1.6)
          2. Редактирование/создание – переход на страницу 4.12
   12. Редактирование/создание языка программирования
       1. Доступно преподавателю
       2. Отображается информация об изменяемом/добавляем языке (вызывается метод 4.1.3)
          1. Название
          2. Короткое описание
          3. Подробное описание
          4. Автор языка
          5. Год выпуска
          6. Актуальная версия
          7. Список особенностей языка
       3. Действия
          1. Сохранить – если добавляет новый язык, то вызывается метод, 4.1.3, иначе если изменяется существующий, то вызывается метод 4.1.4
          2. Картинка – добавляет/изменяет картинку языка программирования (вызывается метод 4.1.5)
   13. 404
       1. Доступно всем
       2. Отображается в случае отсутствия ресурса
   14. 403
       1. Доступно всем
       2. Отображается в случае запрета на использование ресурса
   15. 401
       1. Доступно только неавторизованным пользователям
       2. Отображается в отсутствия/недостатка прав на использования ресурса
3. **Требования к составу и параметрам технических средств**
   1. Сервер
      1. Процессор минимум 2-ядерный с частотой от 2 ГГц.
      2. Оперативная память от 4 Гб.
      3. Место на жестком диске от 2 Гб.
   2. Клиентская часть
      1. Процессор с частотой от 2 ГГц.
      2. Оперативная память от 4 Гб.
4. **Требования к информационной и программной совместимости**
   1. Серверная часть
      1. Операционная система Linux (версия ядра 5.10 и выше)
      2. Minio (RELEASE.2022-10-15T19-57-03Z и выше)
      3. Redis (6.2.16 и выше)
      4. Golang (1.21.13 и выше)
      5. СУБД PostgreSQL (12.2 и выше)
      6. Node JS
      7. Docker
   2. Веб-браузер (любой из)
      1. Chrome (119.0.6045 и выше)
      2. Opera (105.0.4970.16 и выше)
      3. Mozilla Firefox (121.0 и выше)

**ПРИЛОЖЕНИЕ Б СПИСОК HTTP МЕТОДОВ**

Таблица 1 – HTTP методы разрабатываемого веб-сервиса

| **№** | **Метод** | **URL** | **Описание** | **Входные данные** | **Выходные данные** |
| --- | --- | --- | --- | --- | --- |
| 4.1.1 | GET | /api/info | Список языков | langname=string | {  count: int,  draftID: int,  langs: [  {  ID: uint,  Name: string,  ShortDescription: string,  Description: string,  ImgLink: string,  Author: string,  Year: string,  Version: string,  List: {string: string},  Status: bool  }  ]  } |
| 4.1.2 | GET | /api/info/{pk} | Информация об одном языке | pk:uint | {  info: {  ID: uint,  Name: string,  ShortDescription: string,  Description: string,  ImgLink: string,  Author: string,  Year: string,  Version: string,  List: {string: string},  Status: bool  }  } |
| 4.1.3 | POST | /api/info | Добавление нового языка | {  Name: string  ShortDescription: string  Description: string  Author: string  Year: string  Version: string  List: {string:string}  } | {  message: string,  serviceID: int  } |
| 4.1.4 | PUT | /api/info/{pk} | Изменение информации о языке | pk:uint  {  Name: string  ShortDescription: string  Description: string  Author: string  Year: string  Version: string  List: {string:string}  } | {  message: string,  service: {  ID: uint,  Name: string,  ShortDescription: string,  Description: string,  ImgLink: string,  Author: string,  Year: string,  Version: string,  List: {string: string},  Status: bool  }  } |
| 4.1.5 | POST | /api/info/{pk} | Добавление или изменение изображения для языка | pk:uint  {  file: Image } | {  message: string,  service: {  ID: uint,  Name: string,  ShortDescription: string,  Description: string,  ImgLink: string,  Author: string,  Year: string,  Version: string,  List: {string: string},  Status: bool  }  } |
| 4.1.6 | DELETE | /api/info/{pk} | Удаление языка | pk:uint | {  message: string,  status: bool  } |
| 4.1.7 | POST | /api/info/add-service | Добавление языка в заявку-черновик | {  IDLang: uint  } | {  message: string,  projectID: int  } |
| 4.1.8 | GET | /api/project | Список проектов пользователя | start\_date=string  end\_date=string  status=string | {  projects: [  {  ID: uint,  UserID: string,  User: {  id: string,  name: string,  email: string,  login: string,  role: string  },  CreationTime: string,  FormationTime: string,  CompletionTime: string,  Status: string,  ModeratorID: string,  Moderator: {  id: string,  name: string,  email: string,  login: string,  role: string  },  ModeratorComment: string,  Count: int  }  ]  } |
| 4.1.9 | GET | /api/project/{pk} | Информация о конкретном проекте | pk:uint | {  files: [  {  ID: uint,  LangID: uint,  Lang: {  ID: uint,  Name: string,  ShortDescription: string,  Description: string,  ImgLink: string,  Author: string,  Year: string,  Version: string,  List: {string: string},  Status: bool  },  ProjectID: uint,  Code: string,  FileName: string,  FileSize: int,  Comment: string,  AutoCheck: int  }  ],  project: {  ID: uint,  UserID: string,  CreationTime: string,  FormationTime: string,  CompletionTime: string,  Status: string,  ModeratorID: string,  ModeratorComment: string,  Count: int  }  } |
| 4.1.10 | PUT | /api/project/{pk} | Изменение полей проекта | pk:uint  {  status: string  comment: string  } | {  message: string,  project: {  ID: uint,  UserID: string,  CreationTime: string,  FormationTime: string,  CompletionTime: string,  Status: string,  ModeratorID: string,  ModeratorComment: string,  Count: int  }  } |
| 4.1.11 | PUT | /api/project/{pk}/submit | Формирование заявки студентом | pk:uint  {  FileCodes: {uint:string}  } | {  message: string,  project: {  ID: uint,  UserID: string,  CreationTime: string,  FormationTime: string,  CompletionTime: string,  Status: string,  ModeratorID: string,  ModeratorComment: string,  Count: int  }  } |
| 4.1.12 | PUT | /api/project/{pk}/complete | Завершение или отклонение заявки преподавателем | pk:uint  {  status: string  comment: string  } | {  message: string,  project: {  ID: uint,  UserID: string,  CreationTime: string,  FormationTime: string,  CompletionTime: string,  Status: string,  ModeratorID: string,  ModeratorComment: string,  Count: int  }  } |
| 4.1.13 | DELETE | /api/project/{pk}/complete | Удаление проекта | pk:uint  {  FileCodes: {uint:string}  } | {  message: string,  status: bool  } |
| 4.1.14 | DELETE | /api/file/delete | Удаление файла из проекта | {  IDProject: uint  IDLang: uint  } | {  message: string,  status: bool  } |
| 4.1.15 | PUT | /api/file/update | Изменение информации о файле | {  IDProject: uint  IDLang: uint  Code: string  FileName: string  Comment: string  } | {  file: {  ID: uint,  LangID: uint,  ProjectID: uint,  Code: string,  FileName: string,  FileSize: int,  Comment: string,  AutoCheck: int  },  message: string  } |
| 4.1.16 | POST | /api/user/register | Регистрация нового пользователя | {  Name: string  Email: string  Login: string  Password: string  } | {  message: string,  userID: string  } |
| 4.1.17 | PUT | /api/user/update | Изменение информации о пользователе | {  Name: string  Email: string  Password: string  } | {  message: string,  user: {  id: string,  name: string,  email: string,  login: string,  role: string  }  } |
| 4.1.18 | POST | /api/user/update | Аутентификация пользователя | {  Login: string  Password: string  } | {  expires\_in: int,  access\_token: string,  token\_type: string  } |
| 4.1.19 | POST | /api/user/logout | Деавторизация пользователя |  | {  message: string,  status: bool  } |