

Web applications
Nick Ridder
535504

Contents

Introduction	2
Rest API specification	3
Wireframes	10
Index/Main page	10
About	11
Contact	12
Projects	13
Add project	14
Specific project	15
Use case diagrams	16
Sequence diagrams	20

Introduction

This is the functional design document for a personal website made for the subject Web Applications. This document outlines the API, wireframes, use case diagrams and sequence diagrams.

Rest API specification

The api specification is made using the tool provided in the api specification template. This, however, outputted a YAML file. The best way I found to include this is to use code blocks. The original YAML file is also present. I apologize for the inconvenience this might cause.

```
openapi: 3.0.0
info:
  title: My API
  description: API for managing articles, programmers, and projects
  version: 1.0.0
servers:
  - url: http://localhost:3000/api
paths:
  /articles/{id}:
    get:
      summary: Get an article by ID
      parameters:
        - in: path
          name: id
          required: true
          description: ID of the article to retrieve
          schema:
            type: string
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Article'
    put:
      summary: Update an article by ID
      parameters:
        - in: path
          name: id
          required: true
          description: ID of the article to update
          schema:
            type: string
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/UpdateArticle'
      responses:
        '200':
          description: OK
        '404':
```

```
      description: Article not found
delete:
  summary: Delete an article by ID
  parameters:
    - in: path
      name: id
      required: true
      description: ID of the article to delete
      schema:
        type: string
  responses:
    '200':
      description: OK
    '404':
      description: Article not found
/articles:
  post:
    summary: Add a new article
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/NewArticle'
    responses:
      '201':
        description: Created
/programmers/{id}:
  get:
    summary: Get a programmer by ID
    parameters:
      - in: path
        name: id
        required: true
        description: ID of the programmer to retrieve
        schema:
          type: string
    responses:
      '200':
        description: OK
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Programmer'
  put:
    summary: Update a programmer by ID
    parameters:
      - in: path
        name: id
        required: true
        description: ID of the programmer to update
        schema:
          type: string
    requestBody:
      required: true
      content:
```

```
    application/json:
      schema:
        $ref: '#/components/schemas/UpdateProgrammer'
  responses:
    '200':
      description: OK
    '404':
      description: Programmer not found
delete:
  summary: Delete a programmer by ID
  parameters:
    - in: path
      name: id
      required: true
      description: ID of the programmer to delete
      schema:
        type: string
  responses:
    '200':
      description: OK
    '404':
      description: Programmer not found
/programmers:
  post:
    summary: Add a new programmer
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/NewProgrammer'
    responses:
      '201':
        description: Created
/projects:
  get:
    summary: Get all projects
    responses:
      '200':
        description: OK
        content:
          application/json:
            schema:
              type: array
              items:
                $ref: '#/components/schemas/Project'
  post:
    summary: Add a new project
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/NewProject'
    responses:
      '201':
```

```
        description: Created
/projects/{id}:
  get:
    summary: Get a project by ID
    parameters:
      - in: path
        name: id
        required: true
        description: ID of the project to retrieve
        schema:
          type: string
    responses:
      '200':
        description: OK
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Project'
      '404':
        description: Project not found
  put:
    summary: Update a project by ID
    parameters:
      - in: path
        name: id
        required: true
        description: ID of the project to update
        schema:
          type: string
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/UpdateProject'
    responses:
      '200':
        description: OK
      '404':
        description: Project not found
  delete:
    summary: Delete a project by ID
    parameters:
      - in: path
        name: id
        required: true
        description: ID of the project to delete
        schema:
          type: string
    responses:
      '200':
        description: OK
      '404':
        description: Project not found
components:
  schemas:
```

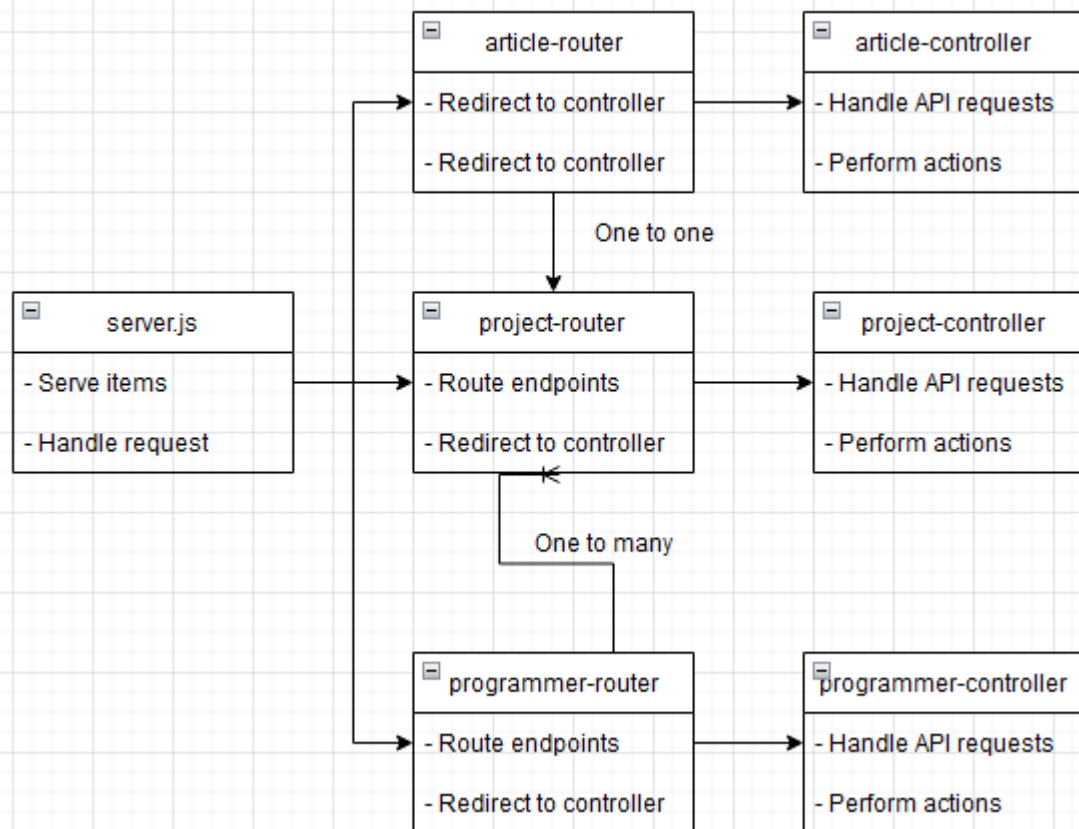
```
Article:
  type: object
  properties:
    id:
      type: string
      description: Unique identifier of the article
    projectID:
      type: string
      description: ID of the project associated with the article
    paragraph:
      type: string
      description: Content of the article
  required:
    - id
    - projectID
    - paragraph
NewArticle:
  type: object
  properties:
    projectID:
      type: string
      description: ID of the project associated with the article
    paragraph:
      type: string
      description: Content of the article
  required:
    - projectID
    - paragraph
UpdateArticle:
  type: object
  properties:
    projectID:
      type: string
      description: ID of the project associated with the article
    paragraph:
      type: string
      description: Content of the article
  required:
    - projectID
    - paragraph
Programmer:
  type: object
  properties:
    id:
      type: string
      description: Unique identifier of the programmer
    name:
      type: string
      description: Name of the programmer
    age:
      type: integer
      description: Age of the programmer
  required:
    - id
    - name
    - age
```

```
NewProgrammer:
  type: object
  properties:
    name:
      type: string
      description: Name of the programmer
    age:
      type: integer
      description: Age of the programmer
  required:
    - name
    - age
UpdateProgrammer:
  type: object
  properties:
    name:
      type: string
      description: Name of the programmer
    age:
      type: integer
      description: Age of the programmer
  required:
    - name
    - age
Project:
  type: object
  properties:
    id:
      type: string
      description: Unique identifier of the project
    name:
      type: string
      description: Name of the project
    languages:
      type: array
      items:
        type: string
      description: List of programming languages used in the project
    skills:
      type: array
      items:
        type: string
      description: List of skills required for the project
    programmerID:
      type: string
      description: ID of the programmer associated with the project
  required:
    - id
    - name
    - languages
    - skills
    - programmerID
NewProject:
  type: object
  properties:
    name:
```



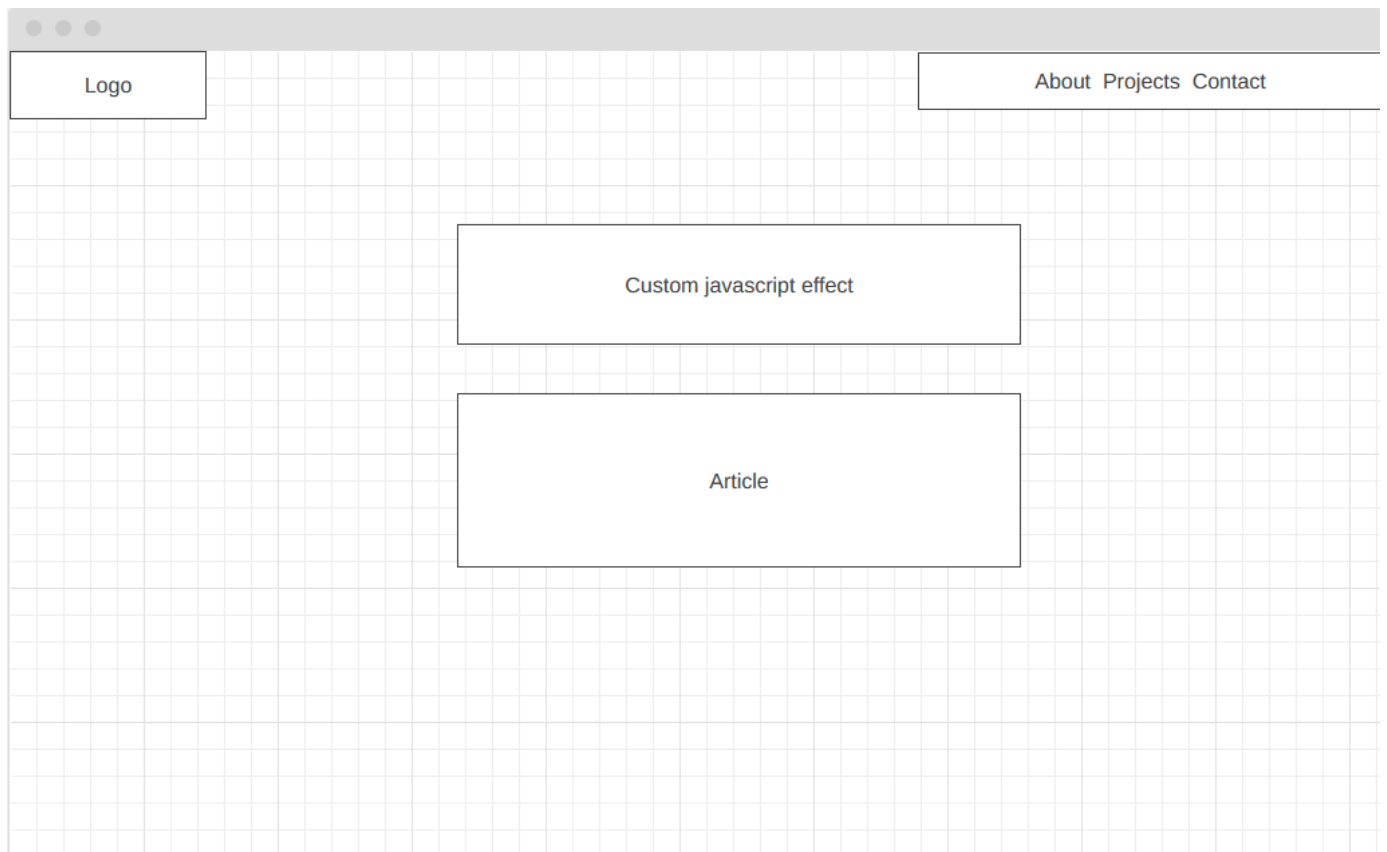
```
    type: string
    description: Name of the project
languages:
  type: array
  items:
    type: string
  description: List of programming languages used in the project
skills:
  type: array
  items:
    type: string
  description: List of skills required for the project
programmerID:
  type: string
  description: ID of the programmer associated with the project
required:
  - name
  - languages
  - skills
  - programmerID
UpdateProject:
  type: object
  properties:
    name:
      type: string
      description: Name of the project
    languages:
      type: array
      items:
        type: string
      description: List of programming languages used in the project
    skills:
      type: array
      items:
        type: string
      description: List of skills required for the project
    programmerID:
      type: string
      description: ID of the programmer associated with the project
  required:
    - name
    - languages
    - skills
    - programmerID
```

Class diagram



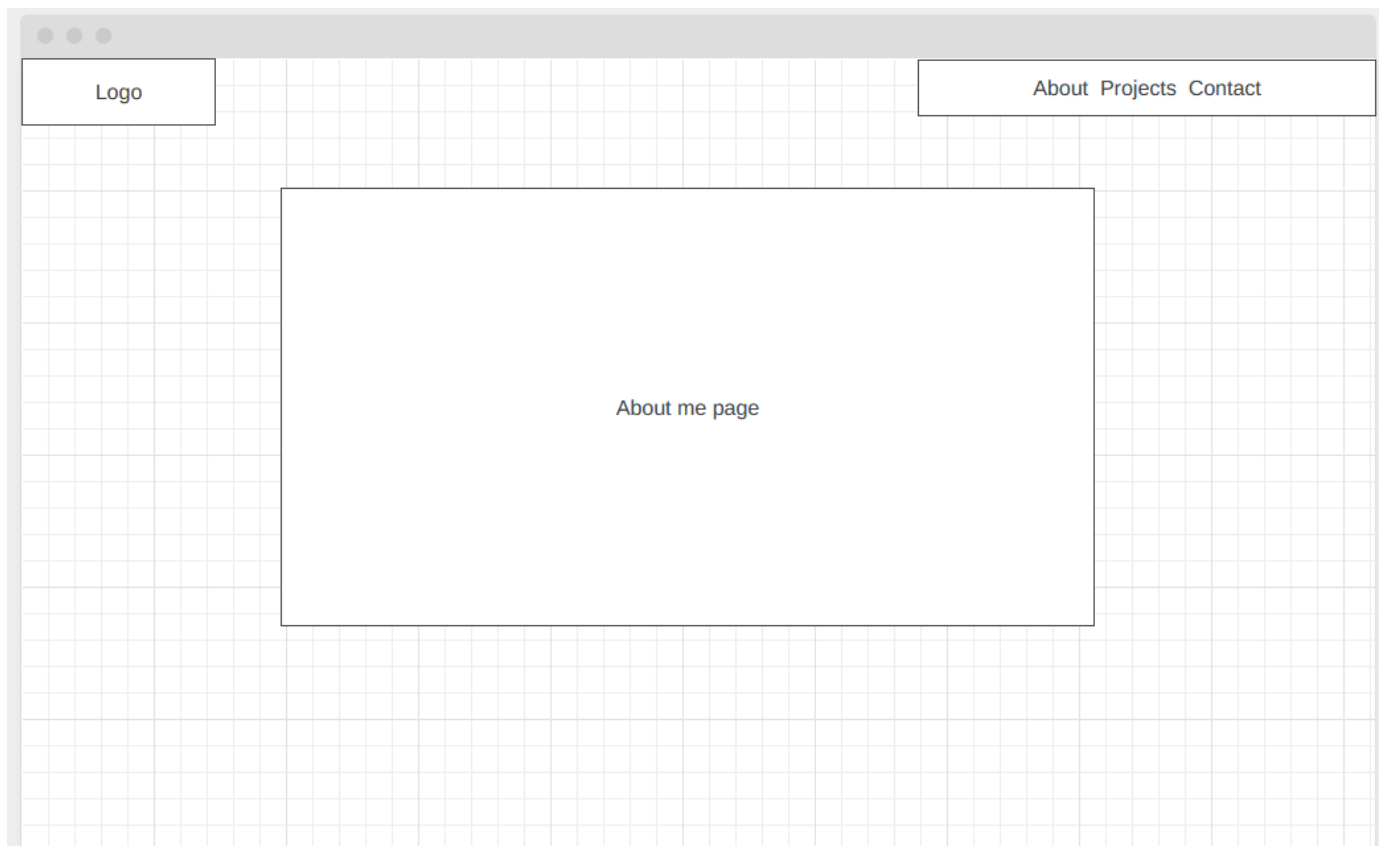
Wireframes

Index/Main page



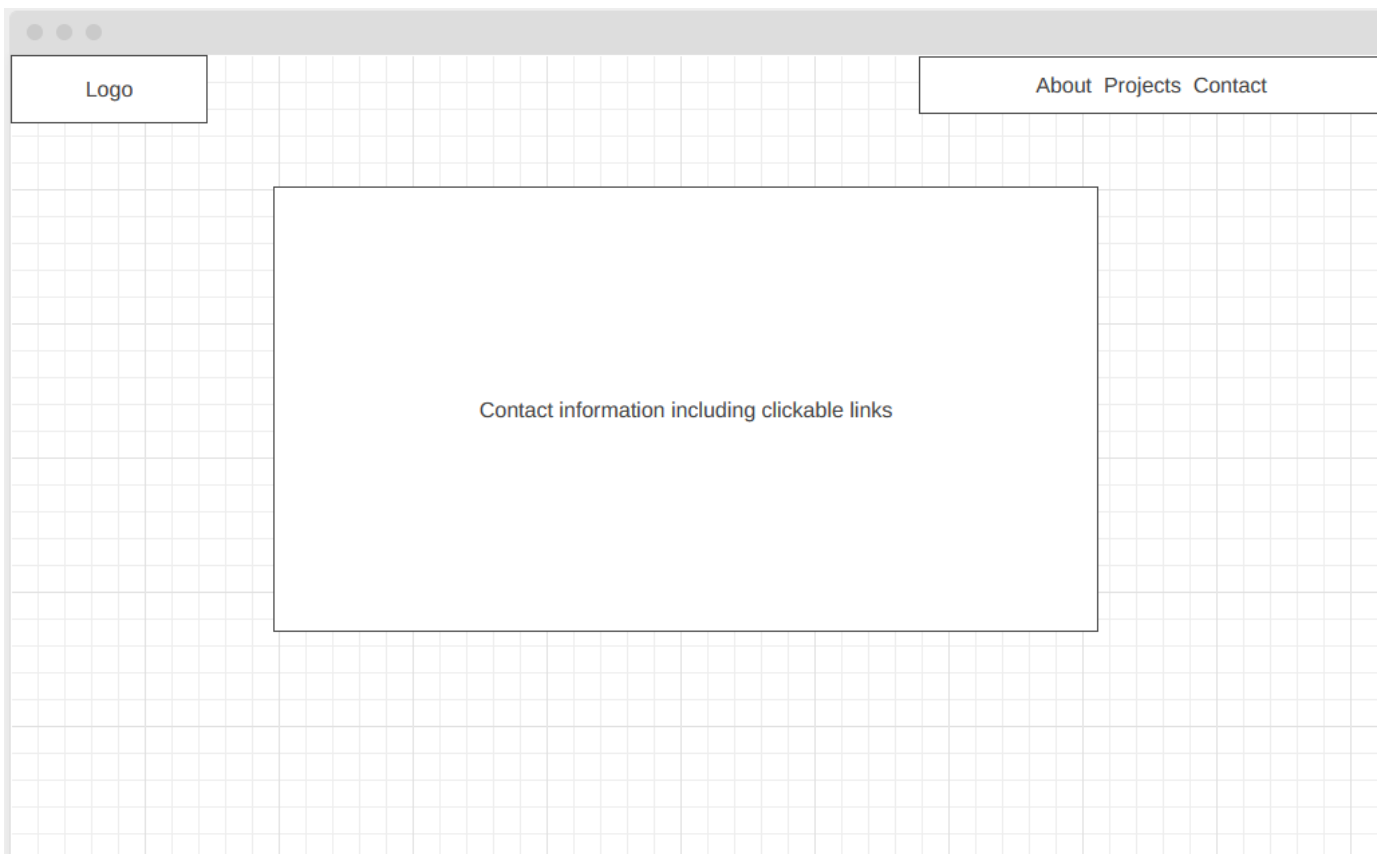
The index page, the javascript effect ended up being a typing effect. Top right is for navigation, and the logo takes the user back to this main page. The article explains a bit about myself.

About



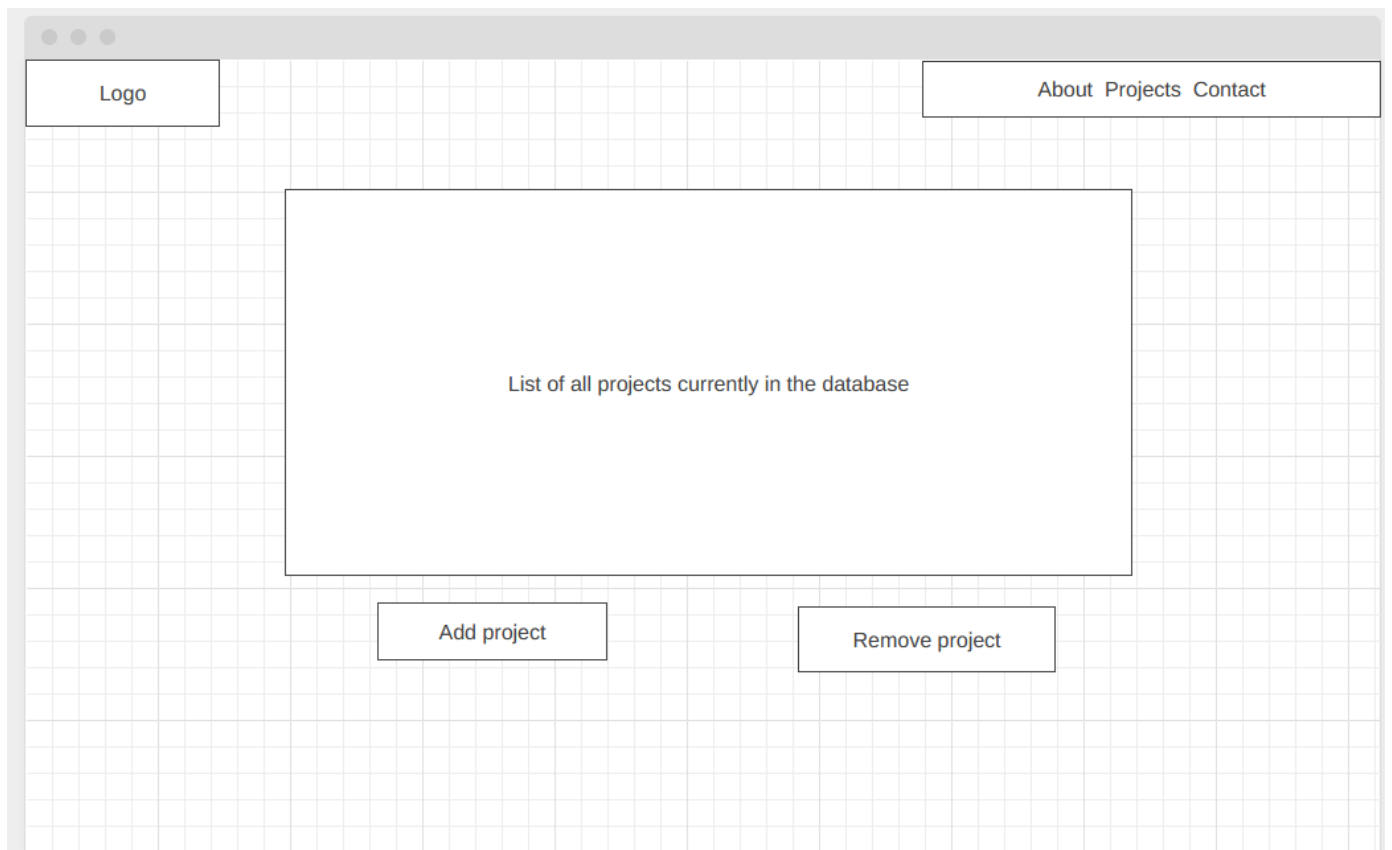
This is a simple about me page, the top part is the same navigation as the main page. The about me part is only text.

Contact



This is a simple contact page, where the user can find places to contact me. These places for contact are hyperlinks that can be clicked taking the user directly to my profile.

Projects



This page has a list of all projects present in the database, loaded using javascript. The user can click on a button to add or remove a project, these buttons take the user to a different page where these actions can be performed

Add project

Logo

About Projects Contact

Add a project

Project name: user input field

Project languages

User input field +

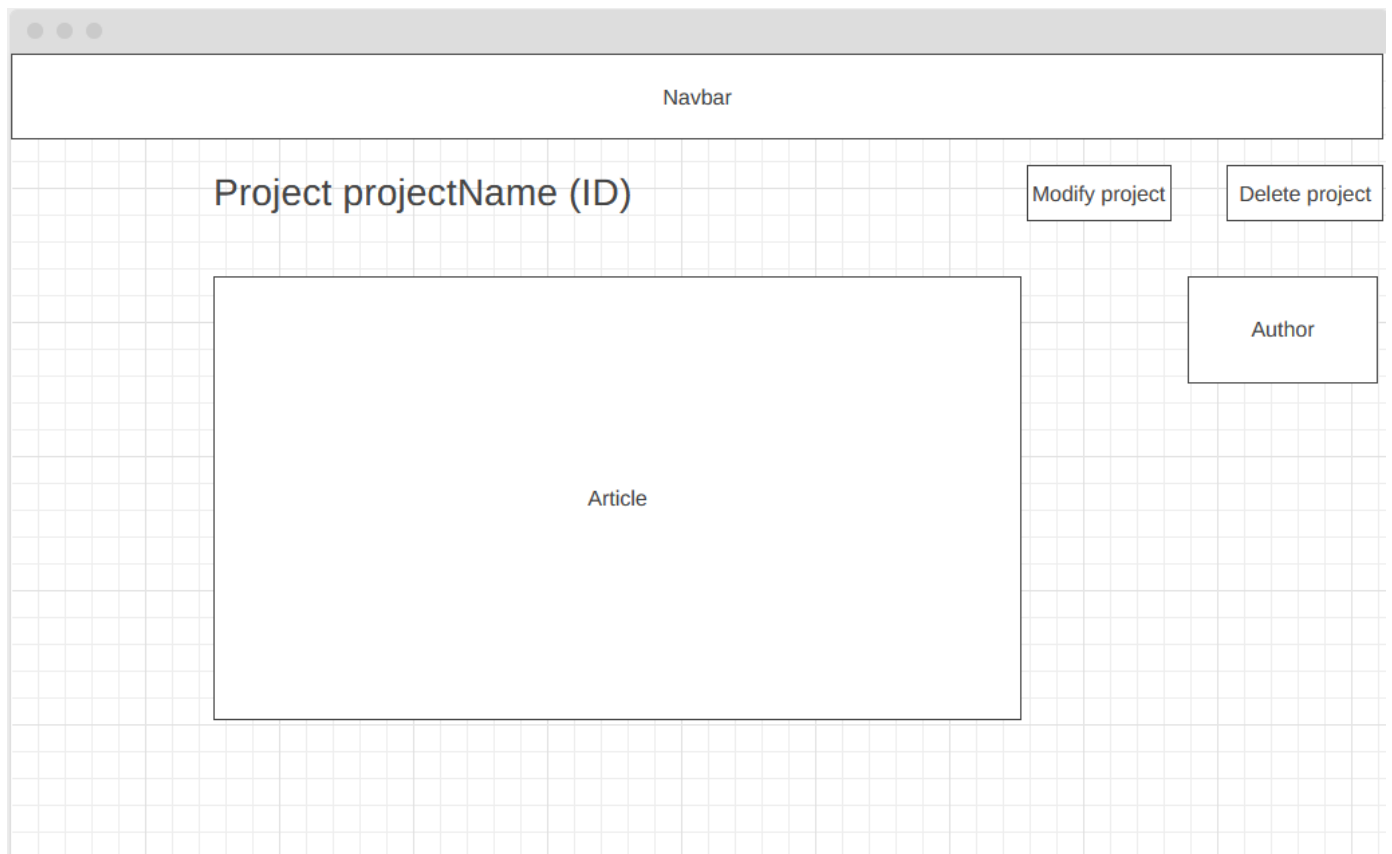
Project skills

User input field +

Add project button

This page allows the user to input data for a new project. The + buttons next to both the language and skill input is there to add an extra input field, to allow for more skills to be added. When one of these boxes is clicked and an extra field is there a - button is created, when this - button is pressed, the extra field will disappear. When the user presses the add project button, the project gets added if the input is valid, finally the user is forwarded to the projects page.

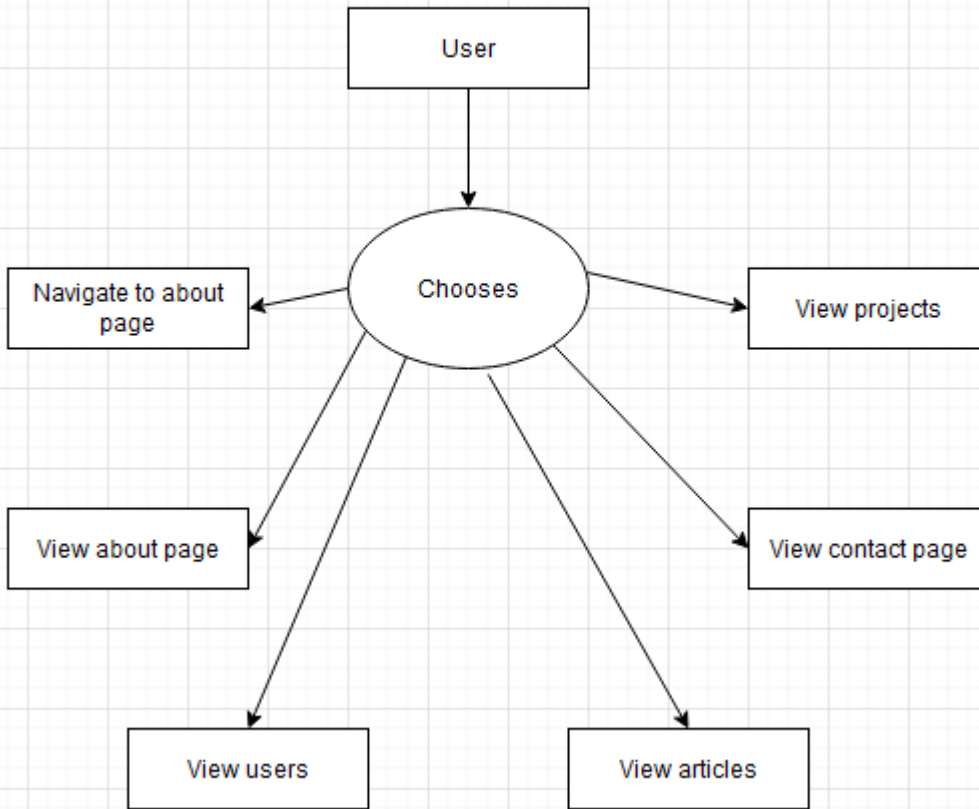
Specific project



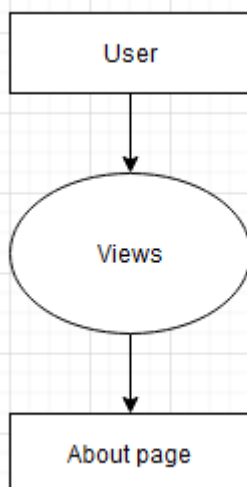
This page allows the user to delete a project. The dropdown menu is filled with the names of all projects present. When the user presses on delete, the selected project will be deleted from the database and then the user is forwarded to the projects page.

Use case diagrams

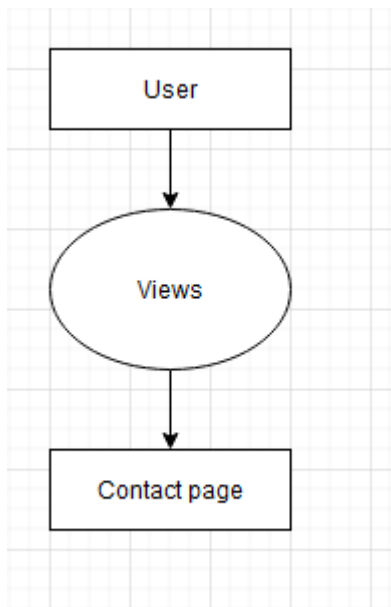
Index/Main page



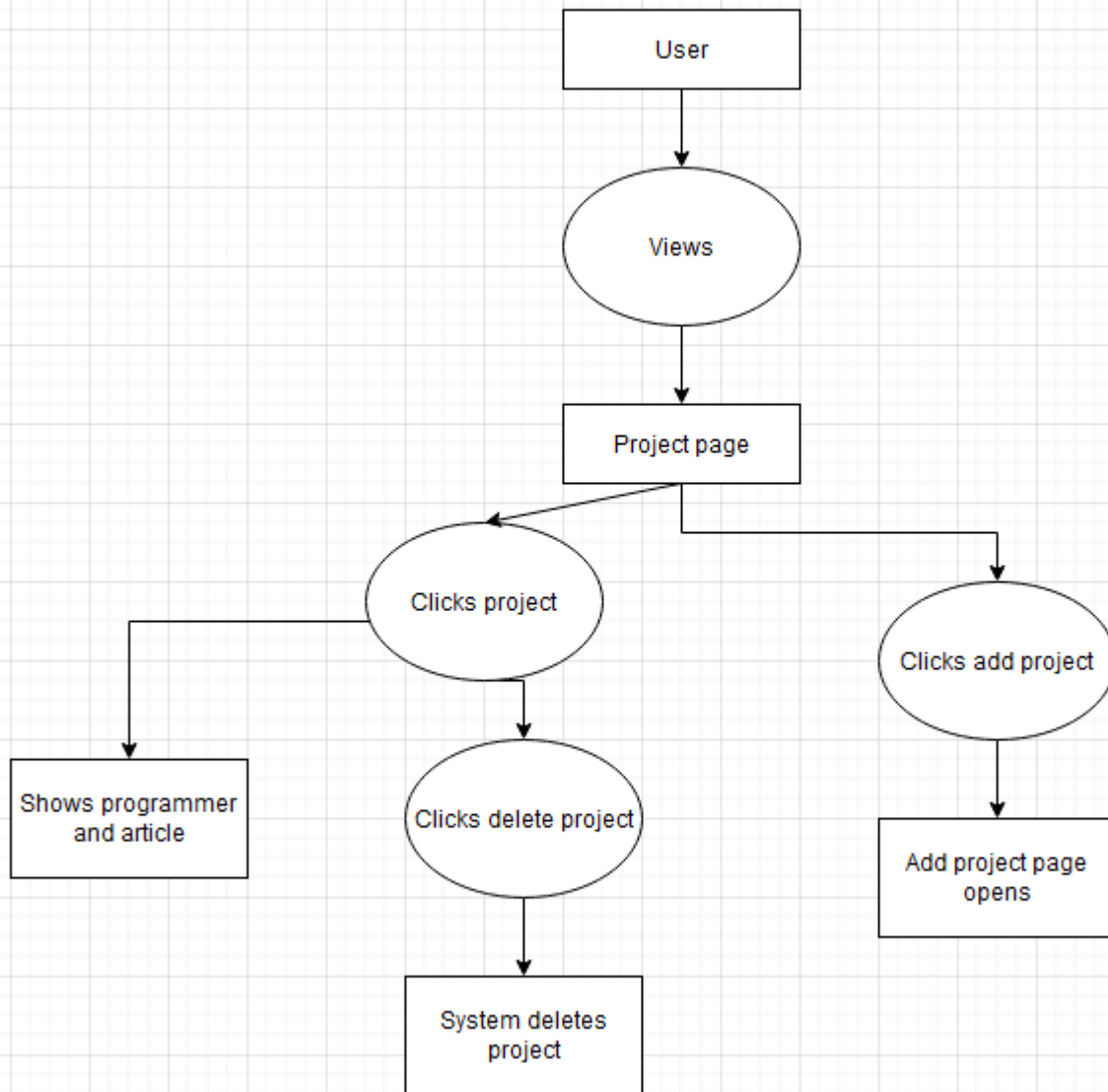
About page



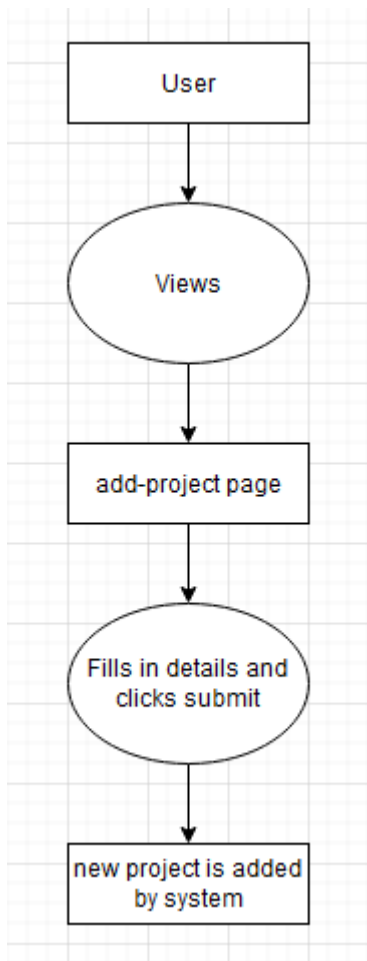
Contact page



Projects & project-details page

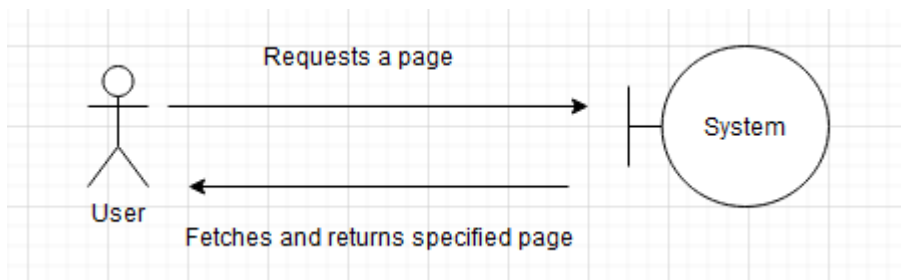


Add project page

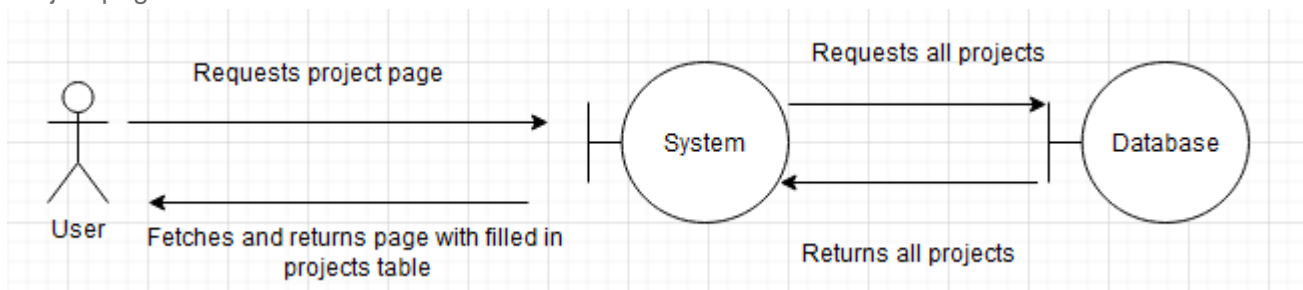


Sequence diagrams

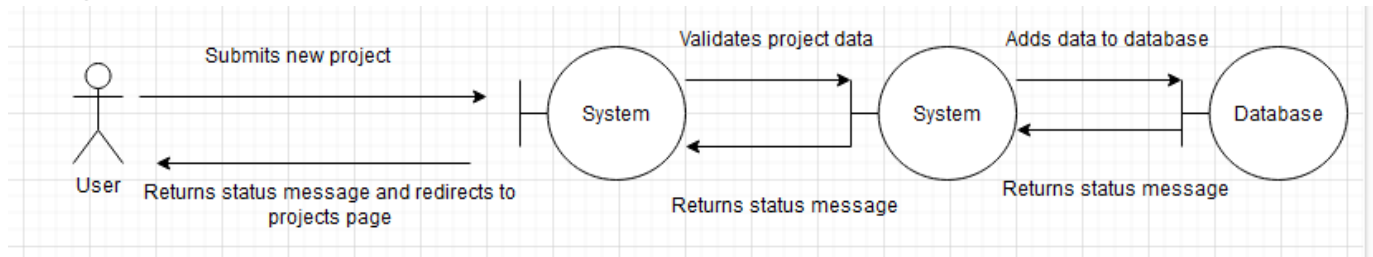
The basic function of the system:



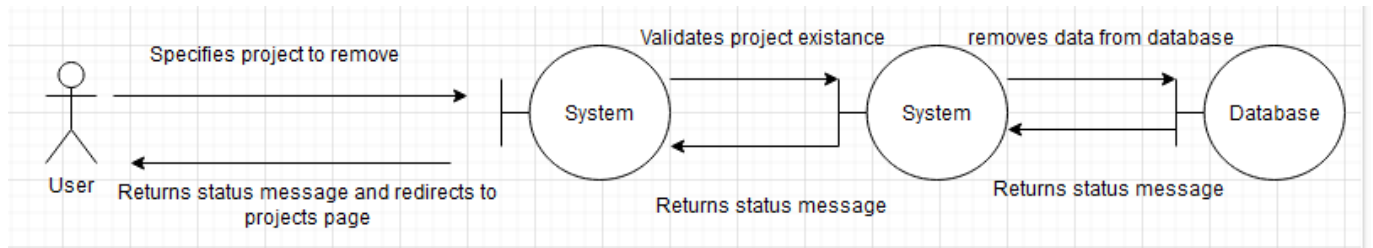
Project page:



Adding a project:



Removing a project:



Projects specification page:

