

Национальный исследовательский университет ИТМО, г. Санкт-Петербург

Факультет ПиИКТ



Бизнес-логика программных систем

Лабораторная работа № 1

Вариант: 1502

Работу выполнили:

Соловьев Артемий Александрович

Велюс Арина Костас

Группа: № Р33151

Преподаватель:

Кривоносов Егор Дмитриевич

г. Санкт-Петербург

2024

Задание:

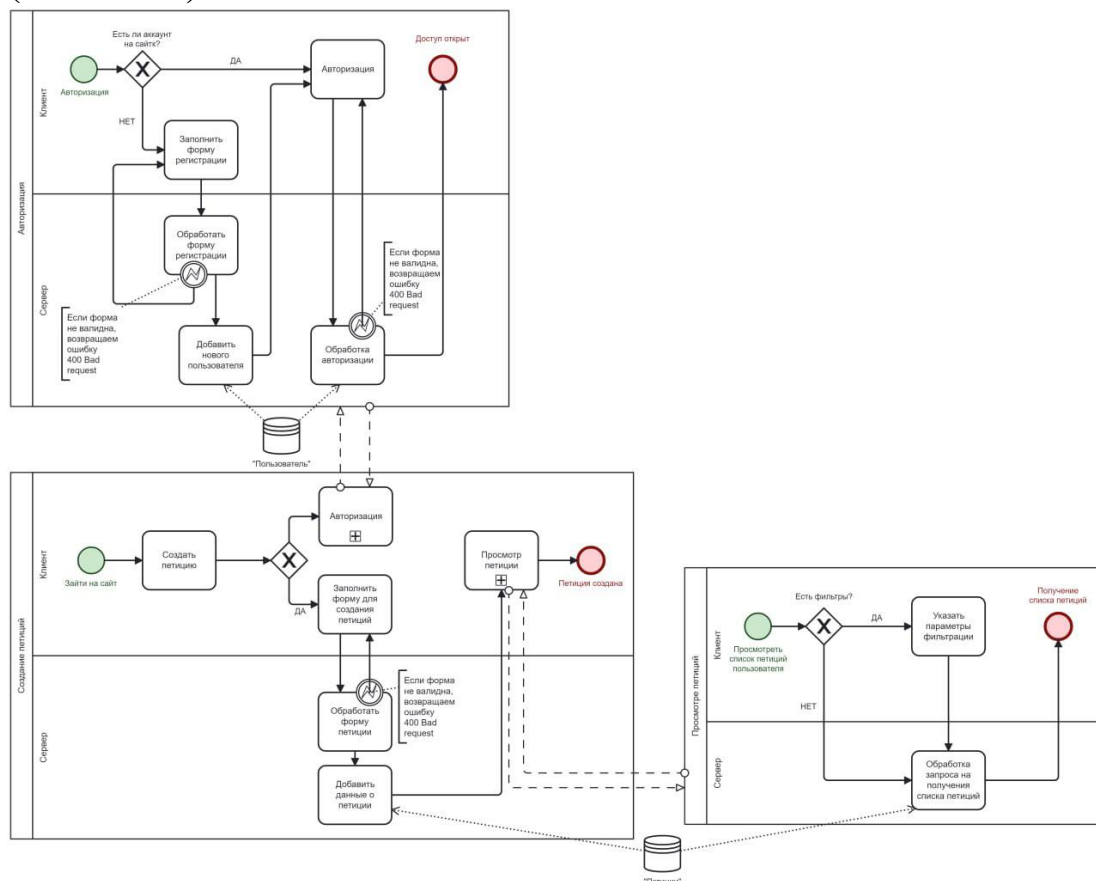
Вариант №1502: Change.org - <https://www.change.org/>

Описать бизнес-процесс в соответствии с нотацией BPMN 2.0, после чего реализовать его в виде приложения на базе Spring Boot.

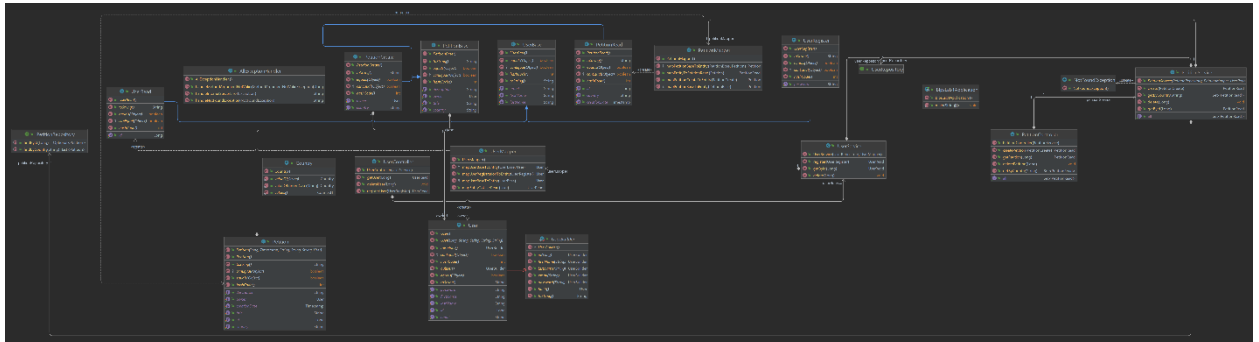
Порядок выполнения работы:

1. Выбрать один из бизнес-процессов, реализуемых сайтом из варианта задания.
2. Утвердить выбранный бизнес-процесс у преподавателя.
3. Специфицировать модель реализуемого бизнес-процесса в соответствии с требованиями BPMN 2.0.
4. Разработать приложение на базе Spring Boot, реализующее описанный на предыдущем шаге бизнес-процесс. Приложение должно использовать СУБД PostgreSQL для хранения данных, для всех публичных интерфейсов должны быть разработаны REST API.
5. Разработать набор curl-скриптов, либо набор запросов для REST клиента Insomnia для тестирования публичных интерфейсов разработанного программного модуля. Запросы Insomnia оформить в виде файла экспорта.
6. Развернуть разработанное приложение на сервере **helios**.

Модель потока управления для автоматизируемого бизнес-процесса (BPMN 2.0):



UML-диаграммы классов и пакетов разработанного приложения:



Спецификация REST API для всех публичных интерфейсов разработанного приложения:

Petition:

/petition:

post:

summary: Create Petition

description: Creates a new petition.

requestBody:

required: true

content:

application/json:

schema:

\$ref: '#/components/schemas/PetitionCreate'

responses:

'201':

description: Created

content:

application/json:

schema:

\$ref: '#/components/schemas/PetitionRead'

get:

summary: Get All Petitions

description: Returns a list of all petitions.

responses:

'200':

description: OK

content:

application/json:

schema:

type: array

items:

\$ref: '#/components/schemas/PetitionRead'

/petition/{id}:

get:

summary: Get Petition by ID

description: Returns information about a petition by its ID.

parameters:

- in: path

name: id
 required: true
 description: ID of the petition to retrieve
 schema:
 type: integer
 format: int64

responses:
 '200':
 description: OK
 content:
 application/json:
 schema:
 \$ref: '#/components/schemas/PetitionRead'

delete:
 summary: Delete Petition by ID
 description: Deletes a petition by its ID.
 parameters:
 - in: path
 name: id
 required: true
 description: ID of the petition to delete
 schema:
 type: integer
 format: int64

responses:
 '204':
 description: No Content

/petition/country/{country}:
 get:
 summary: Get Petitions by Country
 description: Returns a list of petitions by country.
 parameters:
 - in: path
 name: country
 required: true
 description: Name of the country to filter by
 schema:
 type: string

responses:
 '200':
 description: OK
 content:
 application/json:
 schema:
 type: array
 items:
 \$ref: '#/components/schemas/PetitionRead'

components:
 schemas:
 PetitionCreate:
 type: object
 properties:

```

    // define properties for PetitionCreate schema
PetitionRead:
  type: object
  properties:
    // define properties for PetitionRead schema
User:
  /user:
    post:
      summary: Register User
      description: Registers a new user.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/UserRegister'
      responses:
        '201':
          description: Created
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/UserRead'
  /user/{userId}:
    get:
      summary: Get User by ID
      description: Returns information about a user by their ID.
      parameters:
        - in: path
          name: userId
          required: true
          description: ID of the user to retrieve
          schema:
            type: integer
            format: int64
      responses:
        '200':
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/UserRead'
    delete:
      summary: Delete User by ID
      description: Deletes a user by their ID.
      parameters:
        - in: path
          name: userId
          required: true
          description: ID of the user to delete
          schema:
            type: integer

```

```
    format: int64
  responses:
    '204':
      description: No Content
  components:
    schemas:
      UserRegister:
        type: object
        properties:
          // define properties for UserRegister schema
      UserRead:
        type: object
        properties:
          // define properties for UserRead schema
```

Исходный код:

Ссылка на GitHub: https://github.com/Fer0ty/BLPS_lab1

Вывод:

В процессе выполнения лабораторной работы мы выбрали бизнес-процесс и разработали его модель, используя нотацию BPMN 2.0. После было разработано приложение на базе Spring Boot , что позволило нам улучшить свои навыки. Также мы поработали с СУБД PostgreSQL и созданием REST API.