Документация на програмния код

SuperVet

Автор: Стела Сусанина 501222047 49а

Дата: 2025/04/23

Operator notes:

All instructional text is displayed in blue font colour in the template. These should be deleted prior to the document being distributed for review and approval.

Update text within square brackets which is placeholder text with relevant text.

# История на техническата документация

**Автор**

| Фак. номер | Име | Контакт (ел. поща) |
| --- | --- | --- |
| 501222047 | Стела Сусанина | stelasusanina@gmail.com |

**История на версиите**

| Версия | Дата | Автор | Описание |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# 

# Съдържание

[1 Въведение 4](#_Toc447998552)

[2 Детайлен дизайн 4](#_Toc447998553)

[2.1 Диаграма на класовете 4](#_Toc447998554)

[2.2 Диаграма на последователностите 4](#_Toc447998555)

[2.3 Примерен потребителски интерфейс 4](#_Toc447998556)

[3 Тестване 4](#_Toc447998557)

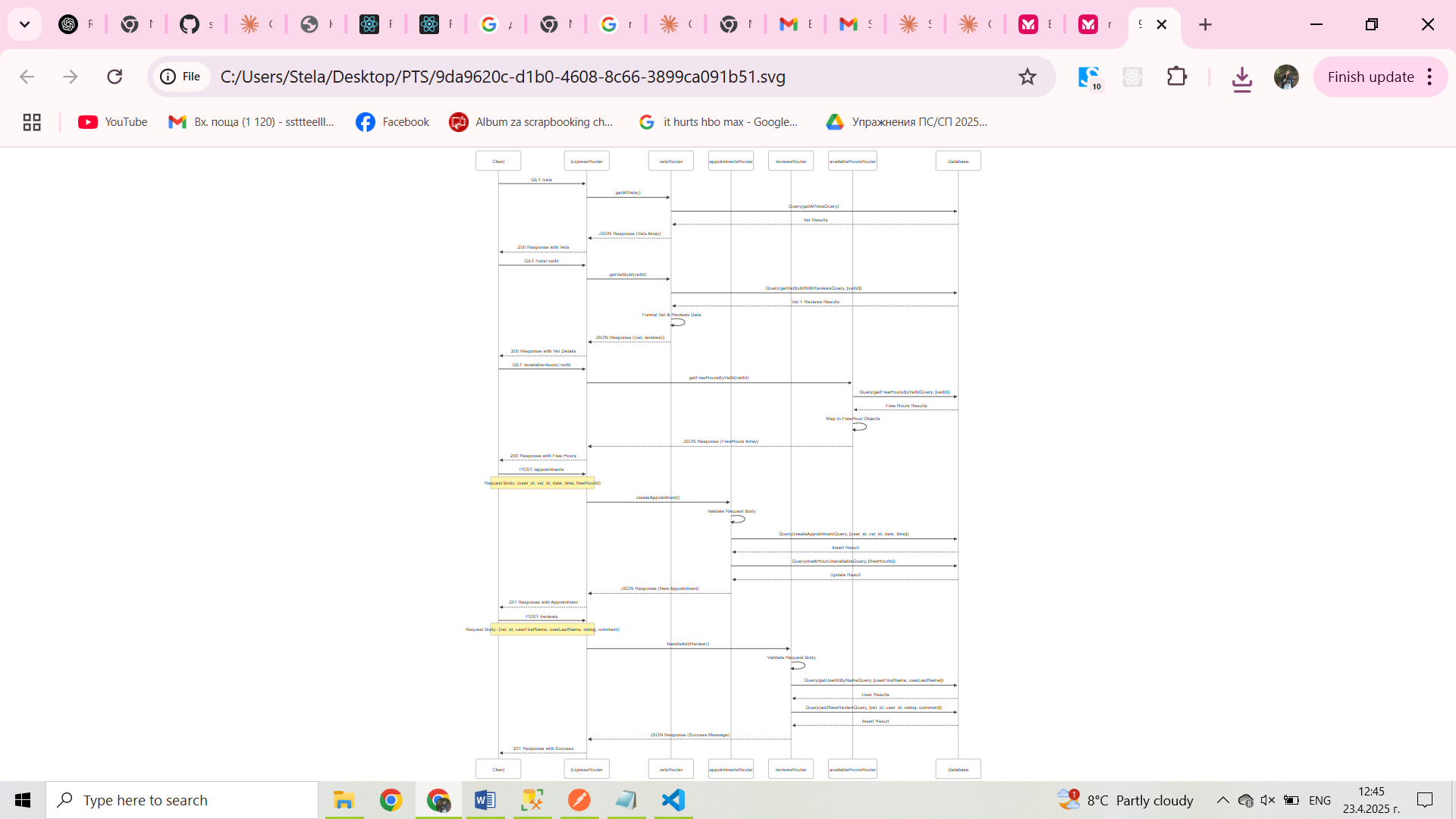
# Въведение

|  |  |  |  |
| --- | --- | --- | --- |
| 4 | Аз като регистриран потребител | Искам да мога да записвам часове при налични ветеринари | Така, че да планирам посещенията за домашния си любимец |
| 5 | Аз като регистриран потребител | Искам да мога да оставям отзиви и оценки за ветеринарите | Така, че други потребители да знаят за качеството на услугите им |

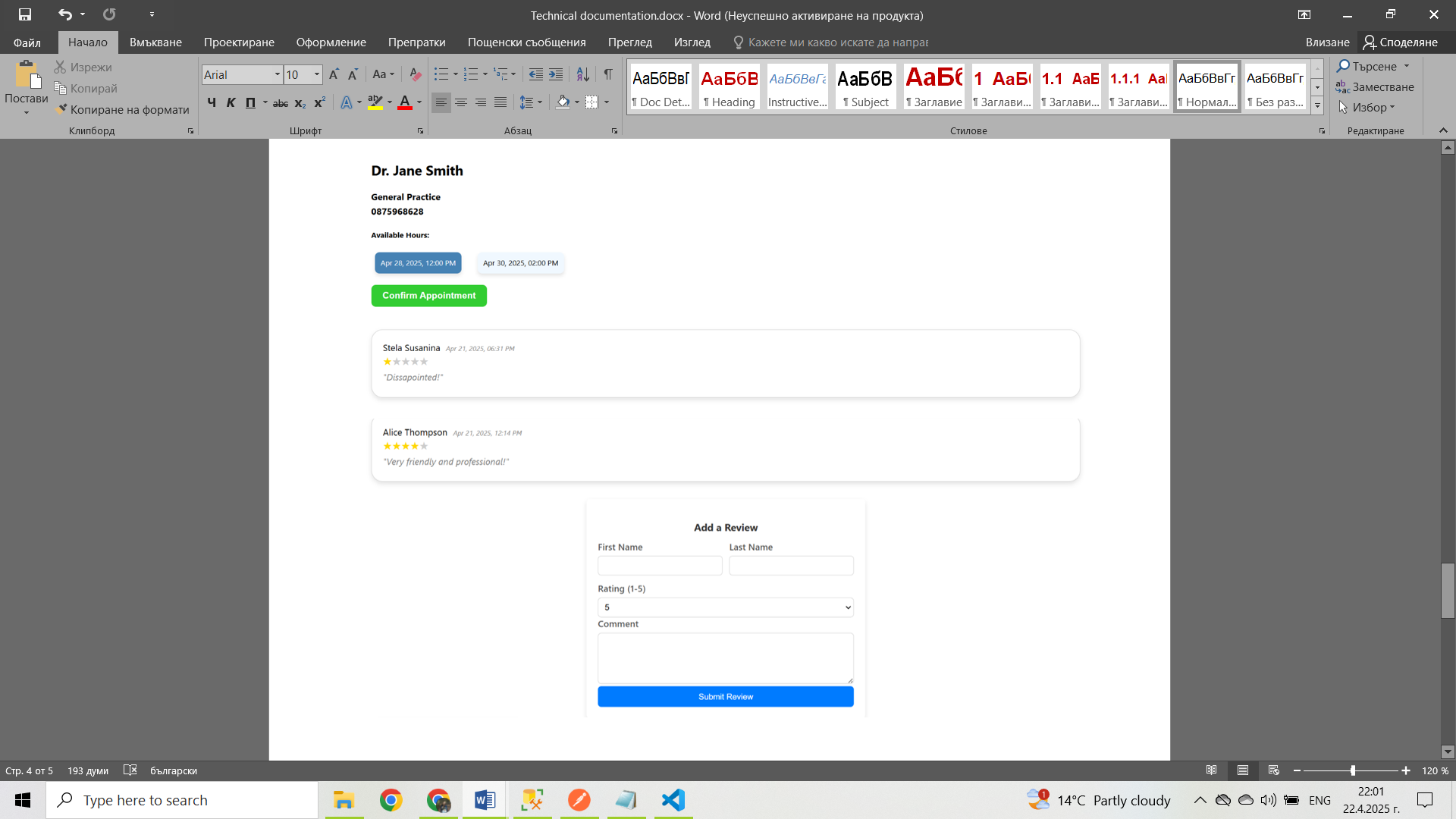
# Детайлен дизайн

## Диаграма на класовете

## Диаграма на последователностите



## Примерен потребителски интерфейс



# Тестване

### 4. "Аз като регистриран потребител искам да мога да записвам часове при налични ветеринари, така че да планирам посещенията за домашния си любимец."

### Критерии за допустимост:

1. Потребителят може да избира ветеринар за посещение.
2. Потребителят трябва да може да избере свободен час за ветеринаря.
3. Потребителят трябва да има възможност да потвърди и създаде час за посещение.
4. Часовете на ветеринарите трябва да бъдат отбелязани като заети след създаване на запис.
5. Потребителят трябва да получи потвърждение след успешен запис на час.
6. Ако възникне грешка при създаване на записа, потребителят трябва да бъде информиран с подходящо съобщение за грешка.

**5. Като** регистриран потребител **искам** да мога да оставям отзиви и оценки за ветеринарите, **така че** другите потребители да знаят за качеството на техните услуги.

**Критерии за допустимост:**

1. Потребителят трябва да може да подава отзив за ветеринар.
2. Отзивът трябва да съдържа следните полета:
   * Първо име на потребителя (userFirstName)
   * Фамилия на потребителя (userLastName)
   * Оценка (rating)
   * Коментар (comment)
3. Системата трябва да валидира дали всички задължителни полета са попълнени от потребителя.
4. Системата трябва да проверява дали потребителят съществува в базата данни въз основа на неговото първо и фамилно име.
5. Ако потребителят съществува, системата трябва да добави отзива в базата данни и да го свърже с потребителя и ветеринара.
6. Ако потребителят не съществува, системата трябва да върне съобщение за грешка.

**Примерни входове и изходи при тестване**

### **Тестове за налични часове**

1. **Тест: трябва да върне 404, когато няма намерени свободни часове**  
   **Вход:** Заявка с vetId (напр. "1"). Базата данни връща празен масив.  
   **Очакван изход:**

{

"message": "No free hours found for this vet"

}

**Статус:** 404

1. **Тест: трябва да върне свободни часове, когато бъдат намерени**  
   **Вход:** Заявка с vetId (напр. "1"). Базата данни връща списък със свободни часове:

**Очакван изход:**

[

{ "id": 1, "vet\_id": 1, "date\_time": "2025-05-01T10:00:00" },

{ "id": 2, "vet\_id": 1, "date\_time": "2025-05-01T11:00:00" },

{ "id": 3, "vet\_id": 1, "date\_time": "2025-05-01T14:00:00" }

]

**Статус:** 200

### **Тестове за ветеринарите (Vet Routes)**

1. **Тест: трябва да върне 404, когато не бъдат намерени ветеринари**  
   **Вход:** Заявка за getAllVets. Базата данни връща празен масив:

**Очакван изход:**

{

"message": "No vets found"

}

**Статус:** 404

1. **Тест: трябва да върне данни за ветеринарите, когато бъдат намерени**  
   **Вход:** Заявка за getAllVets. Базата данни връща списък с данни за ветеринарите:

**Очакван изход:**

[

{ "id": 1, "name": "Dr. John", "specialization": "Surgery" },

{ "id": 2, "name": "Dr. Alice", "specialization": "Dentistry" }

]

**Статус:** 200

1. **Тест: трябва да върне 404, когато ветеринарят не бъде намерен**  
   **Вход:** Заявка за getVetById с vetId "999" (предполага се, че не съществува).  
   **Очакван изход:**

{

"message": "Vet not found"

}

**Статус:** 404

1. **Тест: трябва да върне ветеринар с отзиви, когато бъде намерен**  
   **Вход:** Заявка за getVetById с vetId "1". Базата данни връща данни за ветеринаря и отзиви:

**Очакван изход:**

{

"id": 1,

"name": "Dr. John",

"reviews": ["Great vet!", "Very caring"]

}

**Статус:** 200

1. **Тест: трябва да върне ветеринар, който няма отзиви**  
   **Вход:** Заявка за getVetById с vetId "2". Базата данни връща ветеринаря без отзиви.  
   **Очакван изход:**

{

"id": 2,

"name": "Dr. Alice",

"reviews": []

}

**Статус:** 200

### **Тестове за запазване на часове**

1. **Тест: трябва да създаде запис за час и да отбележи часа като зает успешно**  
   **Вход:** Валидна заявка за създаване на запис:

{

"user\_id": 1,

"vet\_id": 2,

"date\_time": "2025-05-01T14:00:00",

"freeHourId": 5

}

**Очакван изход:**

{

"message": "Appointment created and hour marked unavailable successfully",

"appointment":

{ "id": 10, "user\_id": 1, "vet\_id": 2, "date\_time": "2025- 05-01T14:00:00" }

}

**Статус:** 201

### **Тестове за отзиви**

1. **Тест: трябва да върне 400, когато липсват задължителни полета**  
   **Вход:** Заявка за добавяне на отзив без задължителното поле user\_id:

{

"vet\_id": 1,

"review": "Great vet!"

}

**Очакван изход:**

{

"error": "Missing required fields"

}

**Статус:** 400

1. **Тест: трябва да върне 404, ако потребителят не е намерен**  
   **Вход:** Заявка за добавяне на отзив за не съществуващ потребител.  
   **Очакван изход:**

{

"error": "User not found"

}

**Статус:** 404

1. **Тест: трябва да върне 500, ако добавянето на отзив не успее**  
   **Вход:** Заявка за добавяне на отзив, но запитването към базата данни не успява.  
   **Очакван изход:**

{

"error": "Failed to add review"

}

**Статус:** 500

1. **Тест: трябва да върне 201 при успешно добавяне на отзив**  
   **Вход:** Заявка за добавяне на отзив:

**Очакван изход:**

{

"message": "Review added successfully",

"review": {

"user\_id": 1,

"vet\_id": 2,

"review": "Very professional"

}

}

**Статус:** 201

Requirement traceability matrix

| **ID** | **Изискване** | **Тип** | **Свързани модули** | **Тестове** | **Статус** |
| --- | --- | --- | --- | --- | --- |
| R1 | Потребителят може да прегледа списък с ветеринари | Функционално | /vets, база данни vets | getAllVets returns list, getAllVets returns 404 | Завършено |
| R2 | Потребителят може да избере ветеринар и да види информацията му | Функционално | /vets/:id | getVetById returns vet, getVetById returns 404 | Завършено |
| R3 | Потребителят може да види свободни часове на ветеринар | Функционално | /available-hours/:vetId | returns list, returns 404, passes correct vetId | Завършено |
| R4 | Потребителят може да запази час | Функционално | /appointments | creates appointment, missing fields, db connection error | Завършено |
| R5 | Часовете се маркират като заети след запазване | Функционално | update hours table, appointments | marks hour unavailable, marking hour fails | Завършено |
| R6 | Потребителят може да напише отзив за ветеринар | Функционално | /reviews | review success, missing fields, user not found, db error | Завършено |