Saityno taikomųjų programų projektavimas Praktikos užduotis

Turinys

[1. Pavadinimas. 2](#_Toc433361461)

[2. Užduotis. 2](#_Toc433361462)

[3. Sistemos paskirtis ir tikslas. 2](#_Toc433361463)

[4. Vartotojai. 2](#_Toc433361464)

[5. Apribojimai sistemai. 2](#_Toc433361465)

[6. Funkciniai reikalavimai. PĮ yra keliami šie funkciniai reikalavimai: 2](#_Toc433361466)

[7. Nefunkciniai reikalavimai. PĮ yra keliami šie nefunkciniai reikalavimai: 3](#_Toc433361467)

[8. Vartotojo dokumentacija. Sistemos kūrėjas turi pateikti tokią dokumentaciją: 3](#_Toc433361468)

[9. Papildomi reikalavimai. PĮ kūrimo procesas turi vykti etapais: 3](#_Toc433361469)

[10. Api dokumentacija. 3](#_Toc433361470)

[1. GET /api 4](#_Toc433361471)

[2. GET /api/detail/:id 4](#_Toc433361472)

[3. GET /api/rooms/:id 5](#_Toc433361473)

[4. GET /api/rooms/:id 6](#_Toc433361474)

[11. Srautų diagrama. 6](#_Toc433361475)

# Pavadinimas.

Viesbučio kambarių rezervacija.

# Užduotis.

Sukurti programinę įrangą, skirtą rezervuoti viešbučio kambarius. Funkcionalumą turi sudaryti viešbučių sąrašas, viešbučio miesto muziejų sąrašas, viešbučio kambarių paieška pagal filtrą, kambario puslapis, tekstiniai puslapiai, duomenų pasiekimas per REST API. Sistema turi tvarkingą dizainą.

# Sistemos paskirtis ir tikslas.

Rezervuoti viešbučio kambarį.

# Vartotojai.

Visi, kurie turi interneto prieigą.

# Apribojimai sistemai.

* 1. Programinės įrangos viena iš sudedamų daliu truri būti web servisas.
  2. Sistemos frontend funkcionalumas bent dalinai turi būti padengtas E2E testais.
  3. Web servisas turi būti suprogramuotas su PHP ir Symfony, klientinė dalis su HTML (bootstrap).
  4. Sistema turės tureti RESP API ir leisti atlikti rezervaciją.

# Funkciniai reikalavimai. PĮ yra keliami šie funkciniai reikalavimai:

* 1. Analizuojamo internetinio tinklapio adreso įvedimas naudojant grafinę vartotojo sąsają.
  2. Pilnas “Restfull” serviso palaikymas: <https://en.wikipedia.org/wiki/Representational_state_transfer>

# Nefunkciniai reikalavimai. PĮ yra keliami šie nefunkciniai reikalavimai:

* 1. PĮ turi būti realizuota grafinė vartotojo sąsaja pradiniams duomenims įvesti.
  2. Vartotojo sąsaja anglų kalba.
  3. Vartotojo sąsaja turi būti paprasta, intuityvi ir funkcionali.
  4. Programos reakcijos laikas turi būti priimtinas vartotojui. Jeigu reikalingas ilgesnis laikas atsakui pateikti (sudėtingi skaičiavimai), apie tai turi būti pranešama vartotojui ir pagal galimybę prognozuojamas atsakymo pateikimo laikas.
  5. PĮ turi turėti dokumentaciją.

# Vartotojo dokumentacija. Sistemos kūrėjas turi pateikti tokią dokumentaciją:

* 1. Ataskaitą.
  2. Vartotojo vadovą.
  3. Diegimo instrukciją.
  4. Diegimo kompaktinę plokštelę

# Papildomi reikalavimai. PĮ kūrimo procesas turi vykti etapais:

* 1. Užduoties analizė ir sprendimo metodo tyrimas.
  2. UML diagramų sudarymas.
  3. Programinės įrangos architektūros diagrama(web servisai, kaip jie tarpusavį bendrauja).
  4. Algoritmo realizacija.
  5. Grafinės vartotojo sąsajos realizacija.
  6. Algoritmo ir grafinės dalies apjungimas į sistemą.
  7. Projekto dokumentacijos rengimas.

# Api dokumentacija.

The API consists of the following methods:

|  |  |  |
| --- | --- | --- |
| **Method** | **URL** | **Action** |
| GET | [/api](https://github.com/andsci/hotels/wiki/api) | Retrieves all hotels |
| GET | [/api/detail/:id](https://github.com/andsci/hotels/wiki/api-detail-:id) | Retrieves all hotel rooms |
| GET | [/api/rooms/:id](https://github.com/andsci/hotels/wiki/api-rooms-:id) | Retrieves room information |
| POST | [/api/rooms](https://github.com/andsci/hotels/wiki/api-rooms) | Add new reservation to room |

## GET /api

Retrieves all hotels.

##### Example Request

GET

<http://hotel.dev.lt/api>

##### Parameters

Takes no parameters

##### Example Result

[{"id":"1","title":"Tonyresort","image":"\/src\/Frontend\/Files\/hotels\/images\/source\/download(6).jpg"},{"id":"3","title":"Kaunas","image":"\/src\/Frontend\/Files\/hotels\/images\/source\/36207618.jpg"},{"id":"4","title":"Gaia Village","image":"\/src\/Frontend\/Files\/hotels\/images\/source\/211050\_xl.jpg"},{"id":"5","title":"Europa Royale","image":"\/src\/Frontend\/Files\/hotels\/images\/source\/big\_4ac32e645761a.jpg"}]

## GET /api/detail/:id

Retrieves all hotel rooms.

##### Example Request

GET

<http://hotel.dev.lt/api/detail/1?start=2015/10/27&end=2015/10/29>

##### Parameters

|  |  |
| --- | --- |
| id | Hotel id |
| start | Arrival to hotel date |
| end | Departure from hotel date |

##### Example Result

[{"id":"2","title":"Vip","price":"180","image":"\/src\/Frontend\/Files\/rooms\/images\/source\/vip\_ton.jpg","count":"2","available\_rooms":"0"},{"id":"3","title":"Standart","price":"100","image":"\/src\/Frontend\/Files\/rooms\/images\/source\/standart\_ton.jpg","count":"6","available\_rooms":"4"},{"id":"4","title":"Family","price":"200","image":"\/src\/Frontend\/Files\/rooms\/images\/source\/family\_ton.jpg","count":"4","available\_rooms":"4"}]

## GET /api/rooms/:id

Retrieves all rooms information.

##### Example Request

GET

<http://hotel.dev.lt/api/rooms/1>

##### Parameters

|  |  |
| --- | --- |
| id | Room id |

##### Example Result

{"id":"1","title":"Vip","price":"180","image":"\/src\/Frontend\/Files\/rooms\/images\/source\/vip\_ton.jpg","count":"2"}

## GET /api/rooms/:id

Retrieves all rooms information.

##### Example Request

GET

<http://hotel.dev.lt/api/rooms/1>

##### Parameters

|  |  |
| --- | --- |
| id | Room id |

##### Example Result

{"id":"1","title":"Vip","price":"180","image":"\/src\/Frontend\/Files\/rooms\/images\/source\/vip\_ton.jpg","count":"2"}

# Srautų diagrama.

# C:\Users\Andrius\Downloads\flowchart.png

# Architektūros diagrama