**sofia-university-logo-small Софийски университет „Св. Климент Охридски” sofia-university-logo-small**

**sofia-university-logo-small Софийски университет „Св. Климент Охридски” sofia-university-logo-small**

**Факултет по математика и информатика**

**Документация**

към курсов проект по

*Разработка на клиент-сървър приложения с Node.js + Express.js + React.js*

на тема

**Web Interactive Map**

Изготвил: Денис Ангелов Ангелов, фак. № 25 222

Преподавател: ас. Траян Илиев

Съдържание

[Въведение 3](#_Toc486888949)

[Потребители на системата 3](#_Toc486888950)

[Функционални изисквания 3](#_Toc486888951)

[Основни потребителски случаи 3](#_Toc486888952)

[Основни изгледи 4](#_Toc486888953)

[Server Resource API 4](#_Toc486888954)

[Нефункционални изисквания 5](#_Toc486888955)

[Описание на използваните технологии и библиотечни модули 6](#_Toc486888956)

[Кратка потребителска документация за работа със системата 6](#_Toc486888957)

[Основни екрани в системата 7](#_Toc486888958)

[Начална страница 7](#_Toc486888959)

[Вход в системата 7](#_Toc486888960)

[Регистрация 7](#_Toc486888961)

[Карта 7](#_Toc486888962)

[Маркери 7](#_Toc486888963)

[Изгледи 8](#_Toc486888964)

[Бъдещо развитие на системата 8](#_Toc486888965)

[Използвана литература 8](#_Toc486888966)

# Въведение

В днешно време проблемите със замърсяването на околната среда са от изключително значение за хората. Едно лесно решение за справяне с една малка част от проблема- вредните газове от автомобилите е използването на моторни превозни средства в населените места да се ограничи и замени с използването на велосипеди. В нашата страна обаче трудностите пред велоактивистите са големи. Именно с тази цел е разработена системата Web Interactive Map- за улеснение на велосипедистите и справяне с ежедневните им проблеми, с които се сблъскват по време на каране.

Системата не просто показва карта с маршрути на потребителите. Тя предоставя възможност за сигнализиране за нередности по трасетата, даване на информация на други велосипедисти за състоянието на пътя в дадена точка, както и за визуализация/симулация на даден маршрут. Потребителите на системата могат да измерят разстоянието от точка до точка и да видят състоянието на пътя, по който ще се движат ако други потребители са въвели информация за него. Към всеки един маркер може да бъде прикачен файл, който нагледно да предоставя информация на потребителите за състоянието на трасето.

# Потребители на системата

Системата може да бъде ползвана анонимно - без необходимост от регистрация или след направена такава, изискваща валиден имейл, потребителско име и парола. Вторият описан тип потребители имат достъп до разширени функционалности на системата.

Администраторът на системата ще е потребителят с най-много права - той ще има възможност да редактира и изтрива маркери на всички потребители в системата, да редактира или изтрива (при необходимост) профили на потребители

# Функционални изисквания

## Основни потребителски случаи

|  |  |  |
| --- | --- | --- |
| Потребителски случай | Описание | Системни потребители |
| Вход в системата | Регистриран потребител може да се впише в системата, въвеждайки валидно потребителско име и парола, с които се е регистрирал | Регистриран потребител |
| Регистрация | Анонимен потребител може да си направи регистрация в системата, въвеждайки имена, валиден имейл и парола | Анонимен потребител |
| Редакция на профил | След вход в системата, регистриран потребител може да смени информацията в профила си | Регистриран потребител |
| Намиране на местоположение | Всеки потребител може да намери текущото си местоположение и то да се визуализира на картата с точност до 23м. | Всички потребители |
| Изчисляване на разстояние/площ | По задени точки, всеки потребител може да изчисли на картата разстояние между две и повече точки или площта по зададени точкови критерии | Всички потребители |
| Създаване на маркер | Регистриран потребител може да създава маркер, посочвайки местоположението му на картата и описвайки детайлно неговата идея. Системата предоставя възможност за прикачване на файл към всеки маркер | Регистриран потребител |
| Редакция на маркер | Всеки потребител има право да редактира само маркерите, които сам е създал. Администраторът има право да редактира маркерите на всички потребители | Регистриран потребител, Администратор |
| Разглеждане на маркери | Всеки потребител може да вижда, както своите, така и маркерите на други потребители в системата, но без да ги редактира | Всички потребители |
| Симулация | Регистриран потребител може да симулира избран маршрут | Регистриран потребител |

## Основни изгледи

|  |  |  |
| --- | --- | --- |
| Име на изгледа | Описание | URI |
| Home | Начална страница | / |
| Log in | Страница за вписване в системата. Предоставя възможност на потребителя да въведе потребителско име и парола, за да влезе в системата. | */login* |
| Sign up | Форма за регистрация в системата. Анонимният потребител може да направи профил. | */signup* |
| Profile | Предоставя възможност на регистриран потребител да редактира своя профил | */profile* |
| Map | Предоставя възможност за достъп до картата на системата, съдържаща информация за вело маршрути и описаните маркери по тях. Има възможност за различен изглед на картата и за измерване на разстояние между точки- засичане дължина на маршрут. | */map* |
| About | Съдържа информация за проекта и неговия създател. | */about* |

## Server Resource API

|  |  |  |
| --- | --- | --- |
| Наименование на ресурса | Описание | URI |
| 1. Users | GET Информация за всички потребители и техните данни  POST въвежда потребител с формата на съдържанието  (Content-Type: application/json) :  { name: string,  email: string,  username: string,  password: string,  confirmPassword: string  } | */api/users* |
| 1. User | GET информация за потребител с id= userID  DELETE изтрива потребител с id= userID  PUT update на потребител с формат на съдържанието (Content-Type: application/json):  { name: string|required,  email: string|required,  username: string|required,  password: string|required,  confirmPassword: string|required,  role: string  } | */api/users/{userId}* |
| 1. Test User | GET информация за потребител с username “test” | /api/users?username=test |
| 1. Markers | GET Информация за всички потребители и техните данни  POST въвежда маркер в формата на съдържанието (Content-Type: application/json)  { user\_id: integer,  title: string|required,  text: string|required,  coordinates: string|required,  date: string|required  image: string  } | */api/markers* |
| 1. Marker | GET информация за потребител с id= userID  DELETE изтрива потребител с id= userID  PUT update на потребител с формат на съдържанието (Content-Type: application/json)  {  user\_id: integer,  title: string|required,  text: string|required,  coordinates: string|required,  date: string|required  image: string  } | */api/markers/{markerId}* |

# Нефункционални изисквания

## Изисквания към производителността

* Системата осигурява следните времена за реакция при взаимодействие на потребителите с нея (като се изключи забавянето от Интернет свързаността):
  + Достъп до приложението < 1 секунда;
  + Достъп ресурси от базата < 1 секунда;
  + Картна функционалност < 1 секунда.
* Системата е проектирана и съответно разработена, така че да се запазва консистентността на данните при нарушена работоспособност
* Системата осигурява цялостност на данните при многопотребителски режим на работа

## Изисквания за сигурността

* Системата извършва проверка на въвежданите от потребителите данни, като например формат, задължителност и др.

## Изисквания към дизайна и имплементацията

* Системата е платформено независима, базирана на отворени технологии и да не изисква закупуване на допълнителни лицензи
* Архитектурата на системата е многослойна, позволяваща скалируемост и гъвкавост
* Потребителския интерфейс е responsive – системата може да се използва на различни мобилни устройства

# Описание на използваните технологии и библиотечни модули

## Front-end

Потребителският интерфейс е разработен с [React.js](https://facebook.github.io/react/), с помощта на [React Router](https://github.com/reacttraining/react-router) за рутиране между страниците и [Redux](https://github.com/reactjs/redux) за управление на state-a. Responsive design-a е постигнат чрез използване на Bootstrap.

Основни библиотеки:

* react
* react-dom
* react-redux
* redux-thunk
* react-router
* react-router-redux
* react-router-dom { Link }
* react-router-bootstrap { LinkContainer }
* react-bootstrap
* react-fontawesome

За реализация на картната функционалност е използвана високопроизводителната и динамична библиотека с отворен код за разработване на картографски услуги [OpenLayers](http://openlayers.org/). Информацията за слоевете и географските данни са предоставени от [OpenStreetMap](https://www.openstreetmap.org/) и [Thunderforest Platform](http://thunderforest.com).

## Back-end

[Express.js](https://expressjs.com/)

[Node.js](https://nodejs.org/en/)

## Database

[sqlite3](https://github.com/mapbox/node-sqlite3)

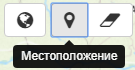
## Документация

# Описание на използваните технологии и библиотечни модули

# Кратка потребителска документация за работа със системата

1. Потребителят влиза в системата
   1. Регистриран потребител - влиза посредством направената си регистрация - въвежда потребителско име и парола
   2. Нерегистриран потребител
      1. Използва системата като анонимен потребител, без да си прави регистрация
      2. Преминава на стъпка 2.
2. (*Опционална. Само за нерегистрирани потребители*) Регистрация на потребител
   1. Потребителят въвежда поредица от данни:
      1. Име
      2. Валиден имейл
      3. Потребителско име
      4. Парола
      5. Потвърждение на паролата
   2. Потвърждава действието си

Стъпката се изпълнява еднократно за всеки потребител.

1. Дейности с картата
   1. Намиране на местоположение:

Всеки потребител може да намери

текущото си местоположение с

точност до 30м.

* 1. Измерване на разстояние между точки
  2. Измерване на площ
  3. Създаване на маркер (само за регистрирани потребители)
     1. Избиране на точно местоположение от картата, на което да бъде поставен маркерът
     2. Задаване на име на маркерът
     3. Въвеждане на описателен текст
     4. (Опционално) Качване на файл
     5. Потвърждение на действието
  4. Изтриване на съществуващ маркер (само за регистрирани потребители)

3.5.1. Избиране на маркер на текущия вписан потребител, който ще бъде редактиран/изтриван. Действието се извършва от десния показващ се помощен панел.

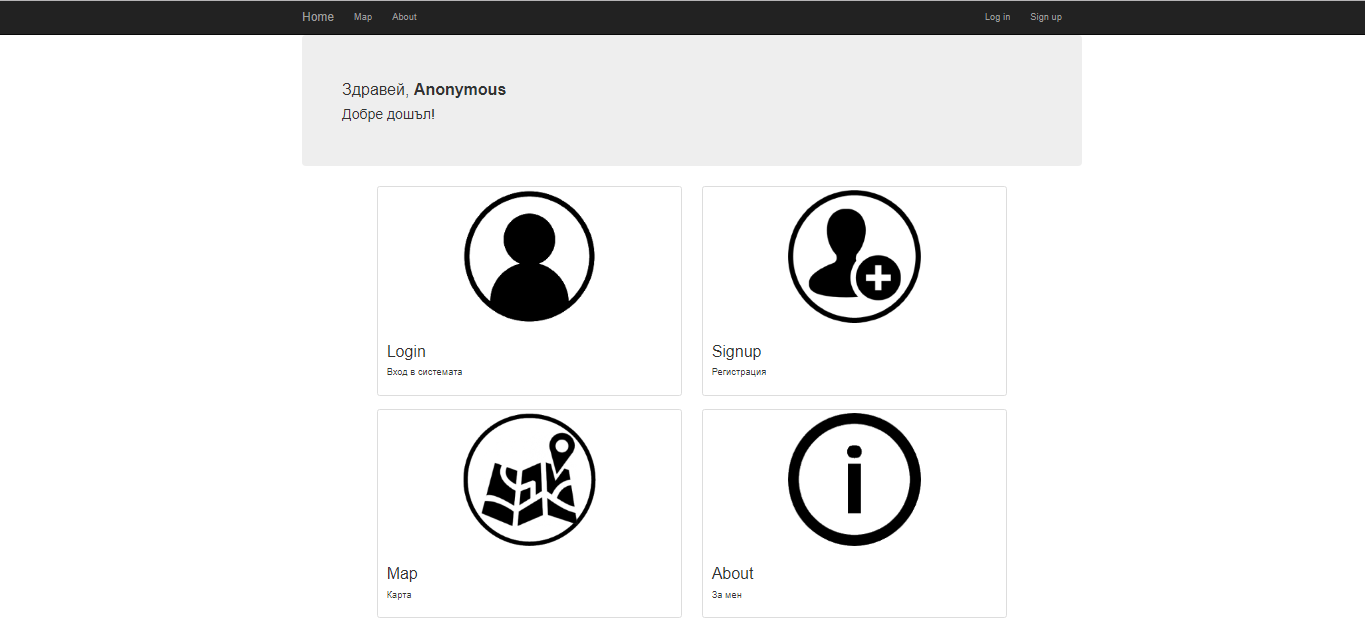
3.5.2. Редактиране/ изтриване на данните

3.6. Симулация– виж [Бъдещо развитие на системата](#_Бъдещо_развитие_на)

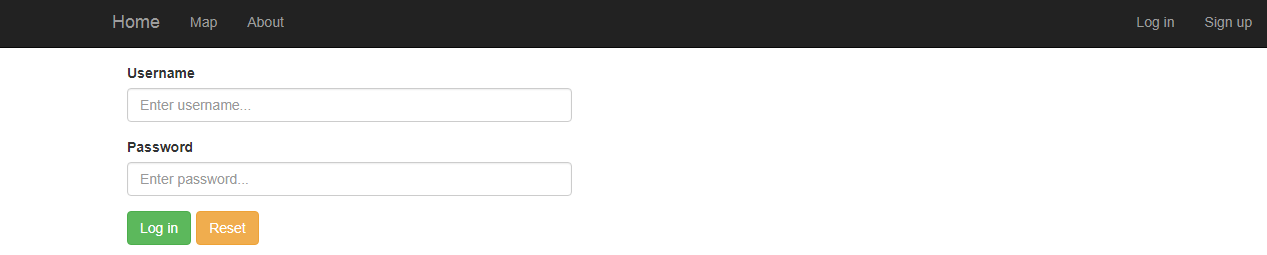
1. Преглеждане на информация за системата и нейния създател.

# Основни екрани в системата

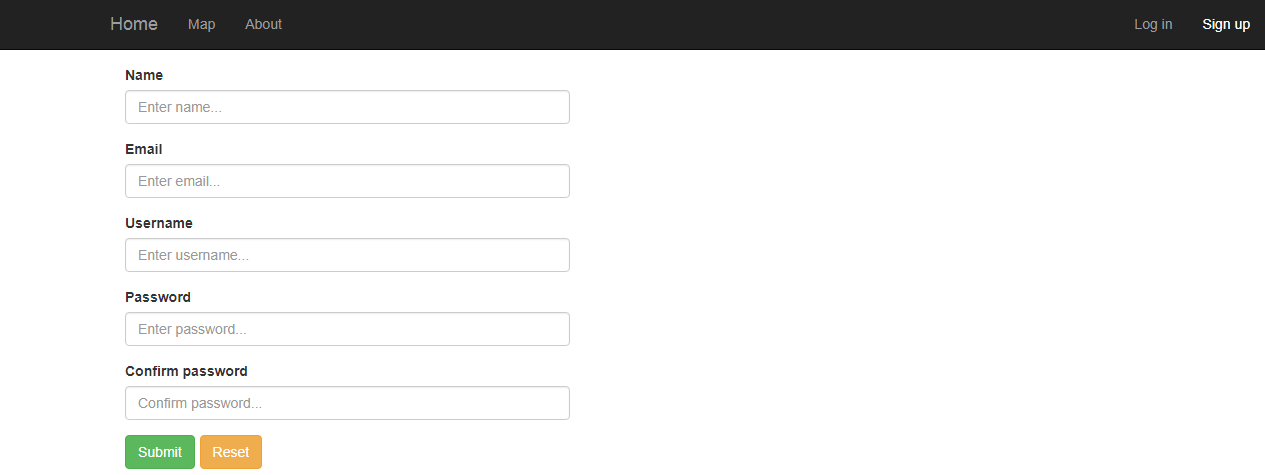
## Начална страница



## Вход в системата



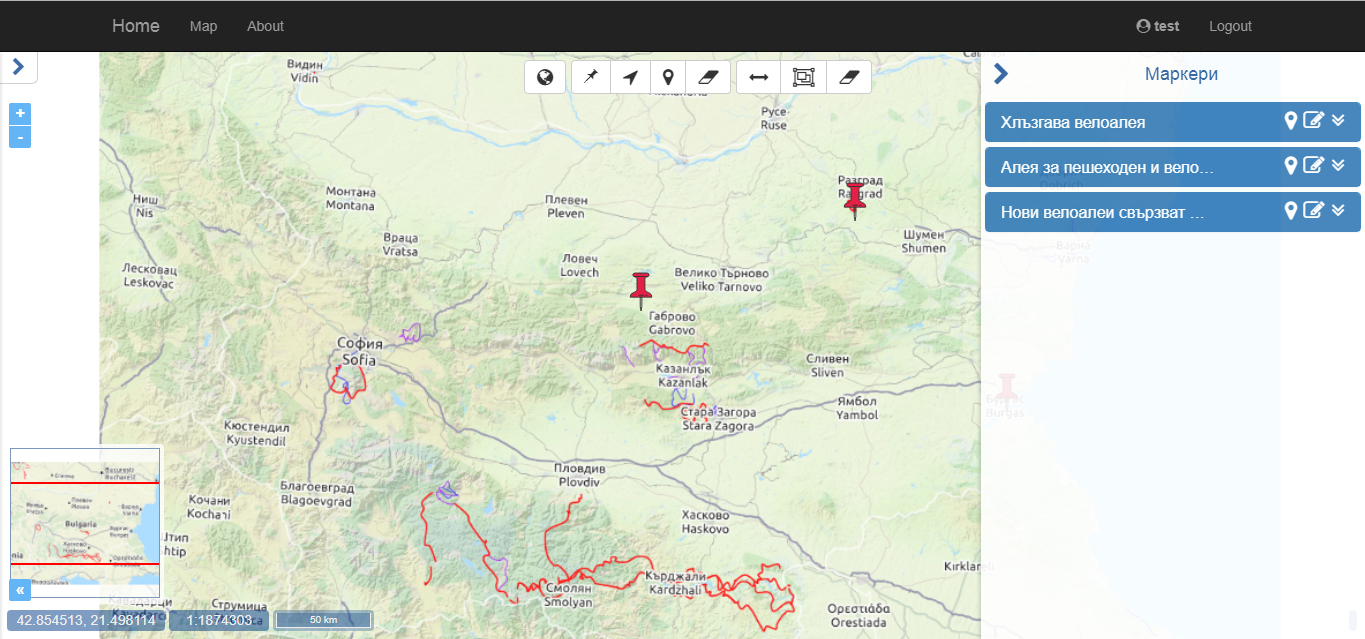
## Регистрация



## Карта

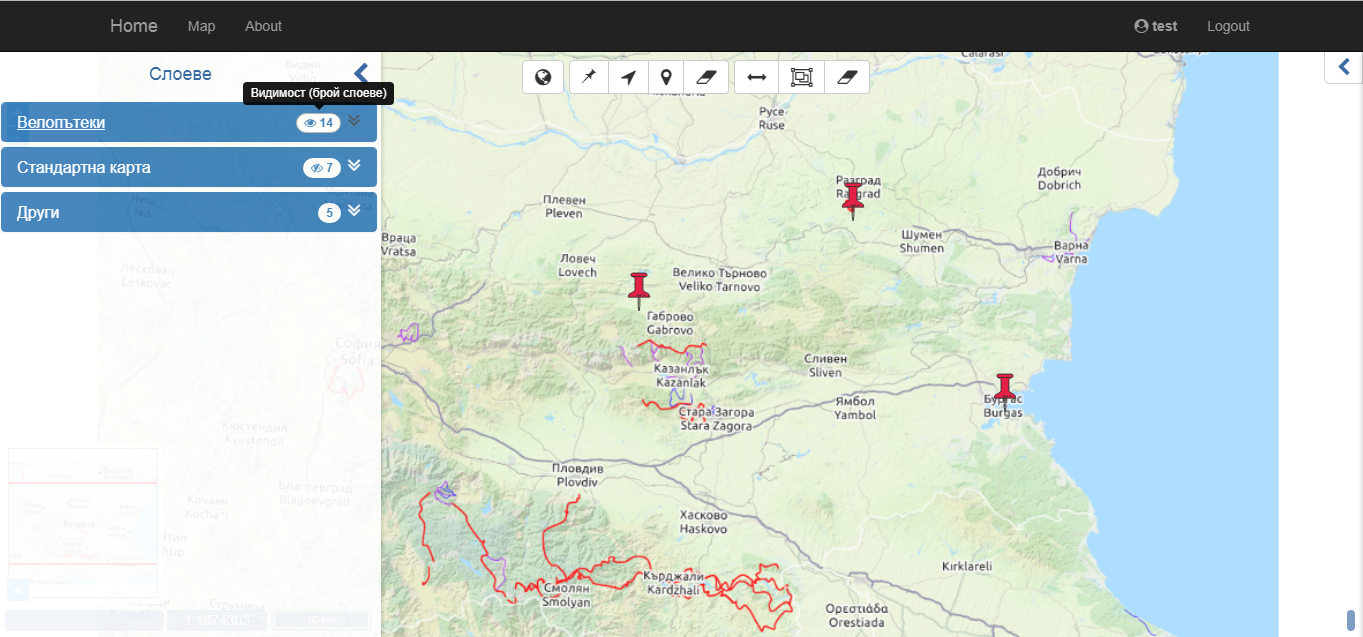
### Маркери

Изгледът е наличен само за регистрирани потребители. Те могат да редактират създадени от себе си маркери. Имат бърз достъп до списък от всички създадени маркери през десния помощен панел „Маркери“. Избор на определен маркер разширява информацията за него като се показва час и дата на създаване на маркера, пълна информация за него и прикачения към него файл (при наличие на такъв). Избор на иконата за местоположение показва на картата точното местоположение на маркера в по-малък мащаб.



### Изгледи

Достъпни за всички потребители от левия помощен панел. Избор на иконата  дава възможност за смяна на изгледа- стандартна карта/велопътеки/стил на картата.



# Бъдещо развитие на системата

Бъдещо развитие на системата ще се състои в реализиране на възможност за симулация на даден маршрут. Чрез нея потребителят ще може да задава начална и крайна точка на своя маршрут и системата анимирано ще му показва маршрута, взимайки предвид поставените от него и/или други потребители маркери и обръщайки им внимание при симулацията.

# Използвана литература