

# Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

Проект по дисциплината

“Fullstack Application Development with Node.js + Express.js + React.js”



Проект на:

Мария Пламенова Паскова фн. 25311, ИИОЗ

05.07.2017 г .

# Съдържание

Факултет по математика и информатика	1
Проект по дисциплината	1
“Fullstack Application Development with Node.js + Express.js + React.js”	1
<b>Анализ на потребностите</b>	<b>4</b>
Основни цели и задачи	4
Целева група	5
Анализ на нуждите	6
Изискване към системата	6
<b>Проектиране на системата</b>	<b>7</b>
Use Case UML диаграма	7
Кратко описание на потребителските случаи (Use cases)	8
Use Cases	10
Добавяне на дестинация	10
Preconditions	10
Main Success Scenario	10
Extensions	10
Регистрация на потребител	11
Main Success Scenario	11
Extensions	11
Вход като потребител	12
Preconditions	12
Main Success Scenario	12
Extensions	13
Activity диаграми	14
Други диаграми	14
SPA Frontend	15
API Resources	15
<b>Описание на реализацията</b>	<b>16</b>

Избор и обосновка на избраните технологии	16
Описание на реализацията на основните функции	17
Инсталация, поддръжка, потребителска документация	17
<b>Инсталиране и конфигуриране</b>	<b>17</b>
Setup	17
<b>Оценяване</b>	<b>18</b>
Описание на планираните методи за оценка на разработения интерфейс	18
Описание на проведените експерименти за оценка на интерфейса на системата	18
Анализ на проведените експерименти и основни заключения относно качеството на интерфейса	19
Идеи за бъдещо развитие и подобряване на интерфейса	19
<b>Материали</b>	<b>20</b>
Презентации - <a href="https://github.com/iproduct/course-node-express-react/wiki">https://github.com/iproduct/course-node-express-react/wiki</a>	20
GitHub - <a href="https://github.com/mimipaskova/NodeJS_React_Hike">https://github.com/mimipaskova/NodeJS_React_Hike</a>	20
GitHub issues - <a href="https://github.com/mimipaskova/NodeJS_React_Hike/issues">https://github.com/mimipaskova/NodeJS_React_Hike/issues</a>	20
React - React router - <a href="https://facebook.github.io/react/">https://facebook.github.io/react/</a> ; <a href="https://github.com/ReactTraining/react-router">https://github.com/ReactTraining/react-router</a>	20
Axios - <a href="https://github.com/mzabriskie/axios">https://github.com/mzabriskie/axios</a>	20
Material-ui - <a href="http://www.material-ui.com/#/">http://www.material-ui.com/#/</a>	20
Express.js - <a href="https://expressjs.com/">https://expressjs.com/</a>	20
Mongo - <a href="https://www.mongodb.com/">https://www.mongodb.com/</a>	20
Mongoose - <a href="http://mongoosejs.com/">http://mongoosejs.com/</a>	20
NodeJS - <a href="https://nodejs.org/en/">https://nodejs.org/en/</a>	20
Passport - <a href="http://passportjs.org/">http://passportjs.org/</a>	20
Create React App - <a href="https://github.com/facebookincubator/create-react-app">https://github.com/facebookincubator/create-react-app</a>	20

# 1. Анализ на потребностите

## 1.1. Основни цели и задачи

В днешни дни, множеството хора са по цял ден пред компютъра, в офиса или в срещи и все по малко им остава време за отдих.

За да сме здрави е много важно да поддържаме добра физическа форма. Много от хората в големия град смятат, че ходенето на фитнес, в който въздух е почти толкова чист, колкото покрай големите булеварди, е достатъчен, за да са здрави.

Всъщност е нужно много повече. Човек трябва да излиза сред природата, да се натоварва физически в естествена среда. Това му гарантира освен добра физическа способност, добро здраве и отпочинала нервна система. Доказано е, че разходките сред природата действат успокояващо на нервите и спомагат на човек да бъде по работоспособен.

И тъй като всичко е свързано с компютри, ние смятаме, че най-лесният начин, по който можем да накараме хората да пътуват и да се разхождат сред природата е да им разкажем истории на хората.

Системата „**Take A Hike**” се състои от статии за преходи ( в планината или сред туристически забележителности) в които освен разказа, има възможност за качване на снимки или клипове. Друга основна функционалност е регистрация и вход за потребители. Всеки потребител има възможност да добавя нови статии към профила си като задължително се отбелязва от къде до къде е било пътешествието или разходката върху интерактивна карта на България в страницата. И ако искат статиите да бъдат още по-интересни е хубаво да добавят снимки към интерактивната галерия. Освен това всеки потребител може да разглежда статиите на другите хора и да дава мнението си за тях.

Системата ще улесни и ще мотивира хората да обикалят повече из интересните кътчета на България.

Друго важно нещо, което системата поддържа е разширена търсачка. Тя ще позволява да се търсят статии по основни критерии:

- Име на потребител
- Забележителност, място
- Ключови думи на темата

Нещо интересно за търсачката е, че има възможност да се търси като се нарисова по картата къде е приблизително мястото, за което търсите статия.

Задачите, които се решават от тази система са леснота, бързина, информираност на потребителите. Една основна задача, която до сега е струвала на потребителите часове в търсене на мнение от познат за даден маршрут/място, чрез системата ще става за минути. Достоверността на историите ще е много висока именно, защото потребителите ще са пряко участващи в изграждането на истории. По този начин всеки желаещ да разбере нещо повече за дадено място(от достоверни източници) ще има пряк интерес да добавя коректни свои впечатления. Друг аспект на приложението е, че ще помага при намирането на място за почивка на хора, които нямат богат опит в туристическите дестинации.

Системата се състои от множество истории ( публикации, дестинации ) на хора, които са посетили дадена дестинация и са споделили собствения си опит от посещението.

## 1.2. Целева група

Системата е насочена към следните персони:

- **Нерегистриран потребител**
  - **регистрация в системата** – Всеки потребител може да се регистрира в системата при въвеждане на следната информация:
    - **Потребителско име**
    - **Email** – всяка регистрация трябва да е с уникален Email.
    - **Парола** – Трябва да е поне 6 символа
- **Потребители** – те са основната група, която таргетира проекта. Тук влизат няколко подроли - Млади хора, Туристи, Всеки човек, който има желание за разходка.  
На тях им е предоставена редица функционалности:
  - **създаване на потребителски профил** – всеки потребител може да създаде свой собствен профил, който да съдържа информацията, от която се нуждае системата, за да е възможно да се споделя история.
  - **настройки на профила** – потребителите могат да редактират съдържанието на своя профил.
  - **разглеждане на историите в системата** – основната функция на системата е да предостави на потребителите удобен начин за разглеждане на споделени истории. Целта е

даден потребител да може бързо да се ориентира в историите в системата и да получи информацията, от която се интересува.

- **създаване на собствена история**– потребителят има възможност да напише история, в която може да включи и клип или снимка от самата разходка.
- **търсене на дестинация**- търсенето може да се осъществява чрез въведени ключови думи.

### 1.3. Анализ на нуждите

- Подробен списък с проблеми на потребителите, които ще бъдат решени от приложението **Take A Nike**:
  - **Създаване на потребителски профил** – приложението дава възможност за създаване на потребителски профили, вход в такъв профил.
  - **Добавяне на туристически обекти**– потребителят има възможност да добавя дестинации, които е посетил и/или има мнение за тях, което иска да сподели. Освен самите дестинации(като име и коментар към тях) има възможност за добавянето на снимка. Друга интересна функционалност е възможността да се асоциира новодобавената дестинация с конкретно местоположение на географска карта.
  - **Разглеждане на туристически обекти** - след като има вече добавени обекти те са общодостъпни и всеки регистриран потребител може да ги разглежда. По този начин потребителят ще може да си изгражда мнение за даден обект на базата на мненията и преживяванията на други хора. Това ще помогне за правилната ориентация на даден потребител за това къде би се чувствал(би си изкарал) най-добре;
  - **Търсене** - потребителят по лесен и удобен начин може да търси желана за дестинация. Това се случва чрез обикновено въвеждане на символен низ и търсенето по ключовите думи, съдържащи се в низа.
- Изброяване на всички нови и допълнителни възможности, които системата **Take A Nike** предоставя след първи прототип:
  - При редактиране на профила на даден потребител, има бутон "Запази промените", който ще извършва запазването на промените в базата данни.
  - В секция "Моите истории" е добавена опция за редакция на история.

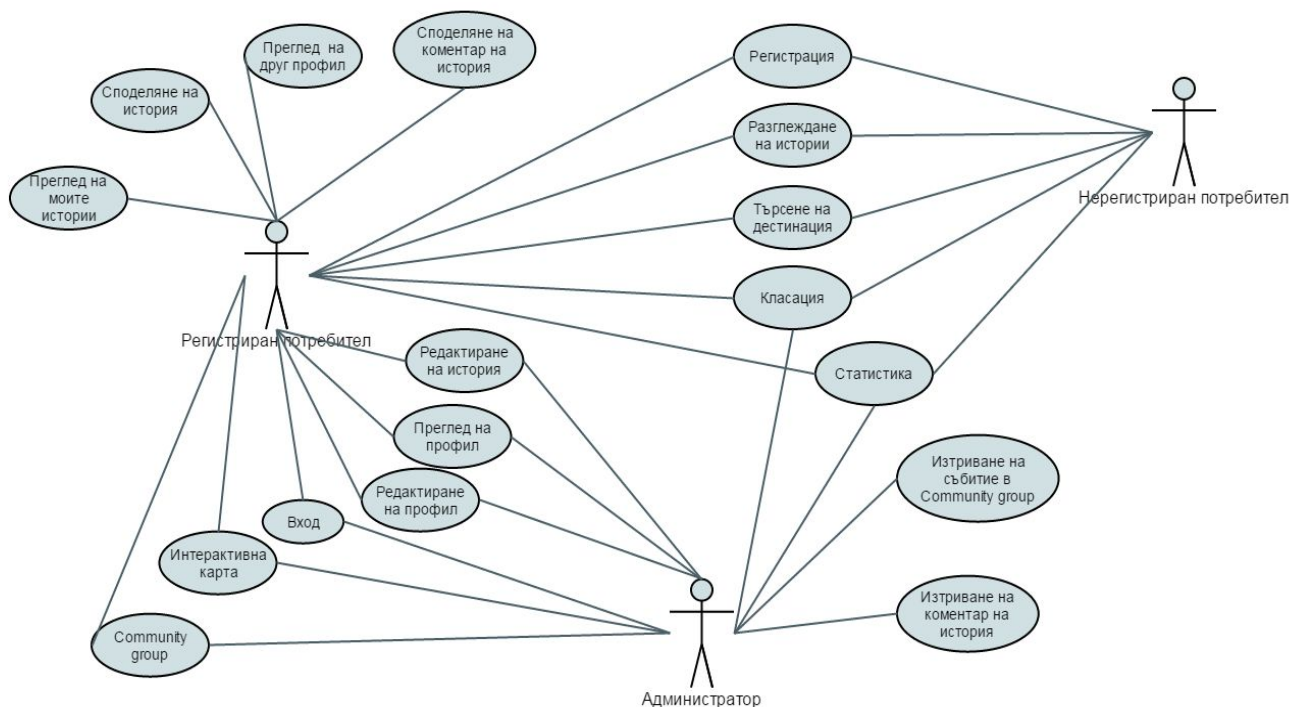
### 1.4. Изискване към системата

- **Функционални изисквания**

- o Регистрация на потребител(Създаване на потребителски профил)
  - o Вход на потребител
  - o Добавяне
    - на туристически обекти
    - на снимки към туристически обекти
    - на координати към туристическите обекти
  - o Търсене на желан туристически обект
    - чрез ключови думи
  - o Добавяне на публикация за пътуване
  - o Редактиране на "Моите истории"
- **Нефункционални изисквания**
    - o Availability-системата е достъпна денонощно(90%).
    - o Изисквания за безопасност
      - Използването на системата трябва да не носи риск за физическата безопасност на потребителите.
    - o Performance-бързодействие на системата време на response не повече от 0.5 сек. Load Balancer за разпределяне на request-ите.
    - o Cross browser support-е с добро качество и е запазена функционалност на следните версии браузъри:
      - [Internet Explorer](#) (версия 9.0 или по-висока);
      - [Mozilla FireFox](#) (версия 27 или по-висока);
      - [Google Chrome](#) (версия 33 или по-висока);
      - [Opera](#) (версия 20 или по-висока).

## 2. Проектиране на системата

### 2.1. Use Case UML диаграма



ф и г у р а 1: Use Case д и а г р а м а з а  
а к т њ о р и т е в с и с т е м а т а

## 2.2. Кратко описание на потребителските случаи (Use cases)

Име на потребителския случай	Кратко описание (Brief Descriptions)	Кратко описание на актьорите (Actor Brief Descriptions)
1.1. Регистрация на потребител	Потребителят задава своите лични данни (име, фамилия, потребителско име, пол,	Потребител желаещ да се регистрира Система-валидираща, приемаща и записваща данните на



	рождена дата). Системата записва неговите данни.	потребителя
<b>1.2. Вход на потребител</b>	Потребителят може да влезе със своето име или Email и парола в системата.	Потребител, който влиза в системата.
<b>1.3. Разглеждане на профил</b>	Потребител или администратор, който е влезнал в системата има възможност да променя личните си данни и да качва своя снимка.	
<b>1.4. Преглед на посетени дестинации</b>	Потребителят разглежда каталога с туристически дестинации посредством "Карта на дестинациите", интегрирана в нашата web система. Друг начин да направи това е като разгледа всички статии с имената им.	Потребител-лице желаещо да се запознае с подходяща за него дестинация  Карта на дестинациите-интерактивна карта на района съдържаща посетени туристически дестинации.
<b>1.5. Добавяне на посетени туристически дестинации и мнения</b>	Регистриран потребител има право да добавя свои посетени дестинации и мнение за тях. Чрез "Карта на дестинациите" той избира местоположението на обекта, дата на посещението и история на посещението.	Потребител - добавя статия
<b>1.6. Добавяне на снимка към галерията</b>	Потребителят има възможността за добавяне на снимки към своята галерия. Системата му предоставя подходяща форма за добавяне снимка. Потребителят въвежда желаните данни.	Потребителят-лице добавящо снимки към галерията си.  Системата-предоставяща входа(форма за снимка) и изхода(обновена, добавена нова снимка)
<b>1.7. Търсене на желана дестинации</b>	Регистрираният потребител има възможност да търси желана дестинация чрез въвеждане на ключови думи в търсачката.	Потребител - търси дестинация чрез търсачка

## 2.3. Use Cases

### 2.3.1. Добавяне на дестинация

Регистриран потребител има възможност за добавяне на посетена дестинация.

#### *Preconditions*

Регистрация на  
потребител (UC-3)  
Вход като потребител  
(UC-4)

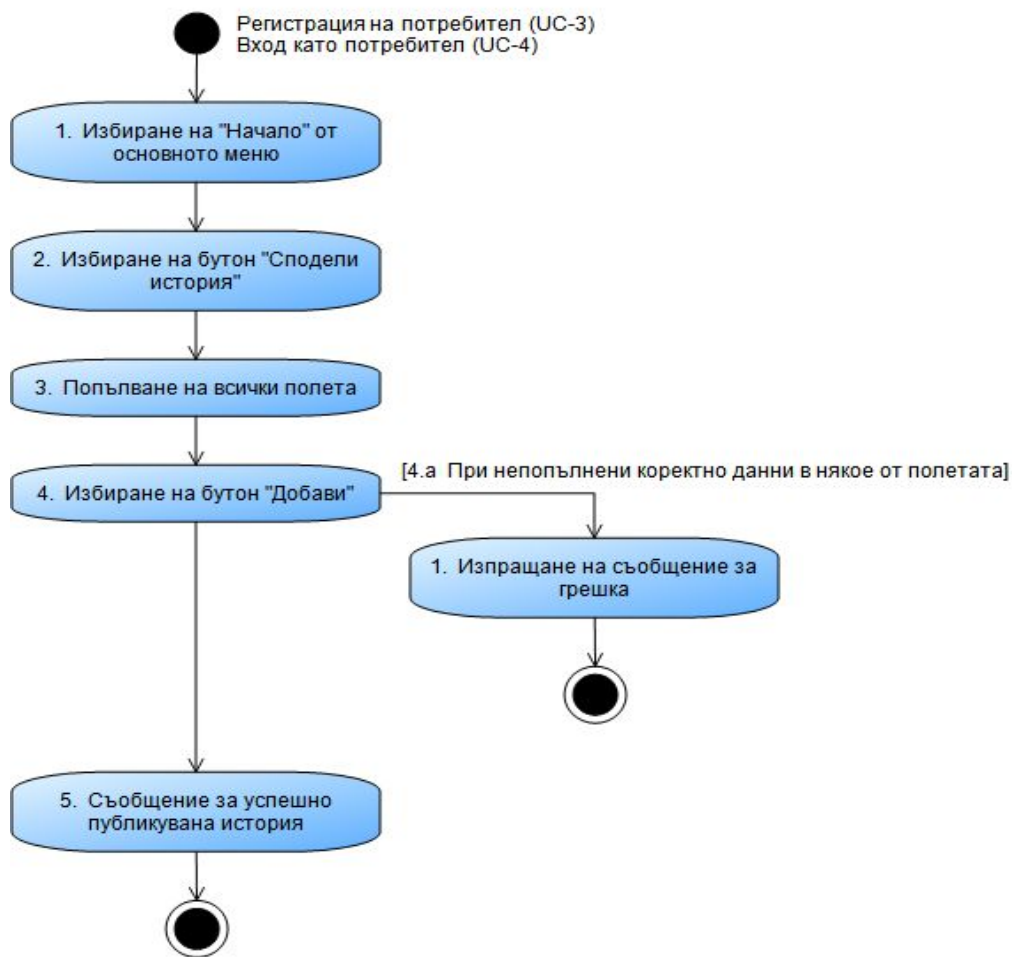
#### *Main Success Scenario*

1. Избиране на "Начало" от основното меню
2. Избиране на бутон "Add story"
3. Попълване на всички полета
4. Избиране на бутон "Add"
5. Съобщение за успешно публикувана история

#### *Extensions*

- 4.a При не попълнени коректно данни в някое от полетата
1. Изпращане на съобщение за грешка
  2. the case ends

## UC-1 Добавяне на дестинация



### 2.3.2. Регистрация на потребител

Регистриране в системата като потребител чрез избрани потребителско име и парола

#### Main Success Scenario

1. Потребител избира от главното меню "реgistрай се"
2. Потребител трябва да попълни следните полета - име, Email, парола, потвърждаване на парола, имена.
3. Съобщение за успешна регистрация
- 4.

#### Extensions

- 2.a Несъвпадение на парола
  1. Системата изкарва съобщение за грешка,

което дава подробна информация

2. Потребител е върнат в начален екран

3. The use case ends

2.b Невалиден email

1. Системата изкарва съобщение за грешка, което дава подробна информация

2. Потребител е върнат в начален екран

3. The use case ends

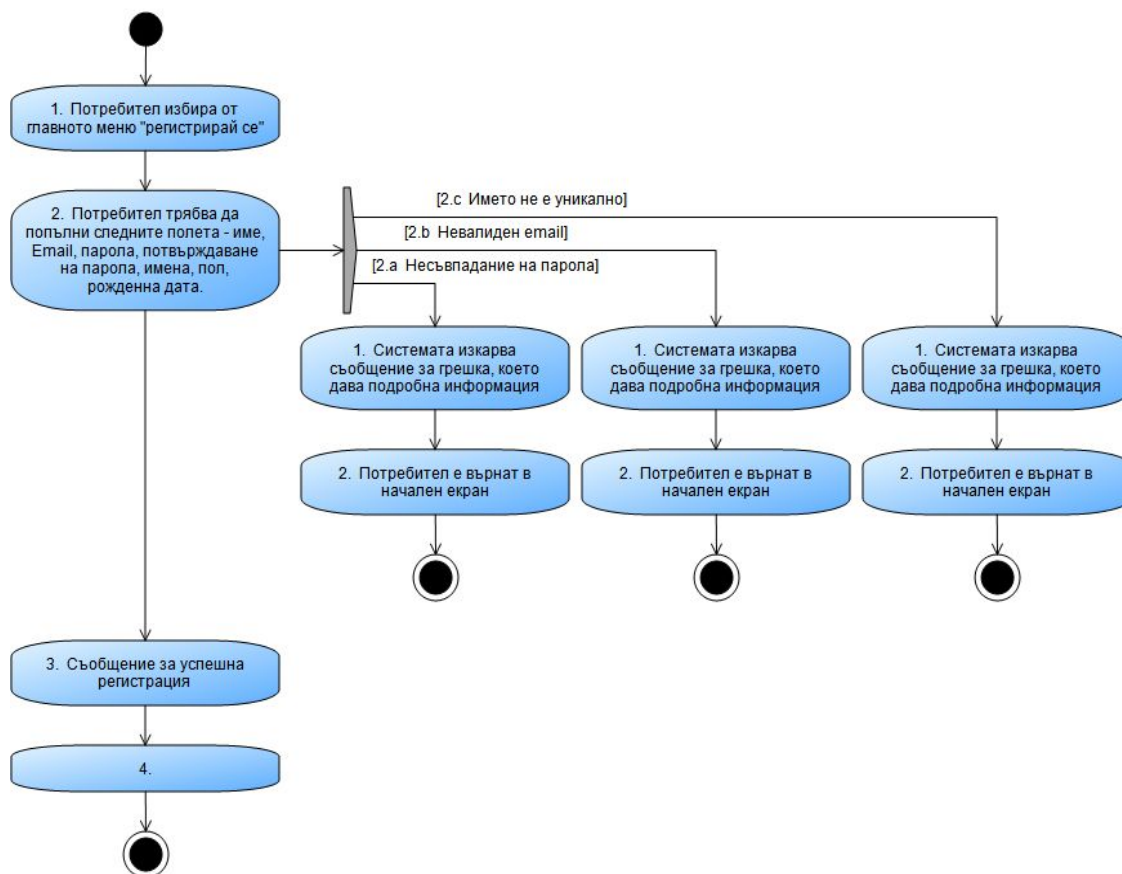
2.c Името не е уникално

1. Системата изкарва съобщение за грешка, което дава подробна информация

2. Потребител е върнат в начален екран

3. The use case ends

### UC-3 Регистрация на потребител



### 2.3.3. Вход като потребител

Влизане в системата като потребител чрез потребителско име и парола.

### *Preconditions*

Регистрация на  
потребител (UC-3)

### *Main Success Scenario*

1. Потребител избира от главното меню "Начало"
2. Потребител трябва да попълни два полета - име и парола
3. Потребител е върнат в начален екран
4. Потребител влиза в профила си

### *Extensions*

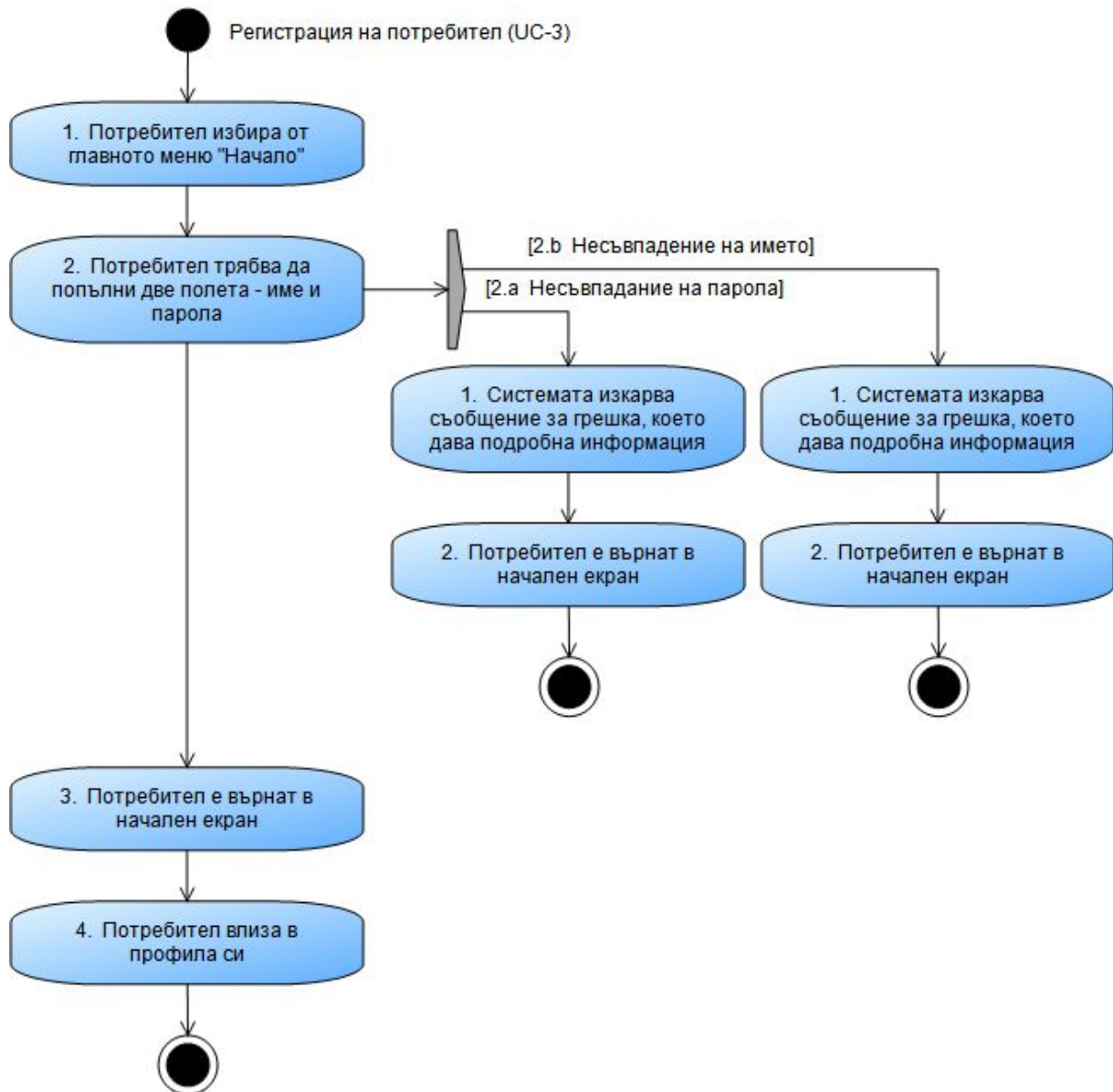
#### 2.a Несъвпадение на парола

1. Системата изкарва съобщение за грешка, което дава подробна информация
2. Потребител е върнат в начален екран
3. The use case ends

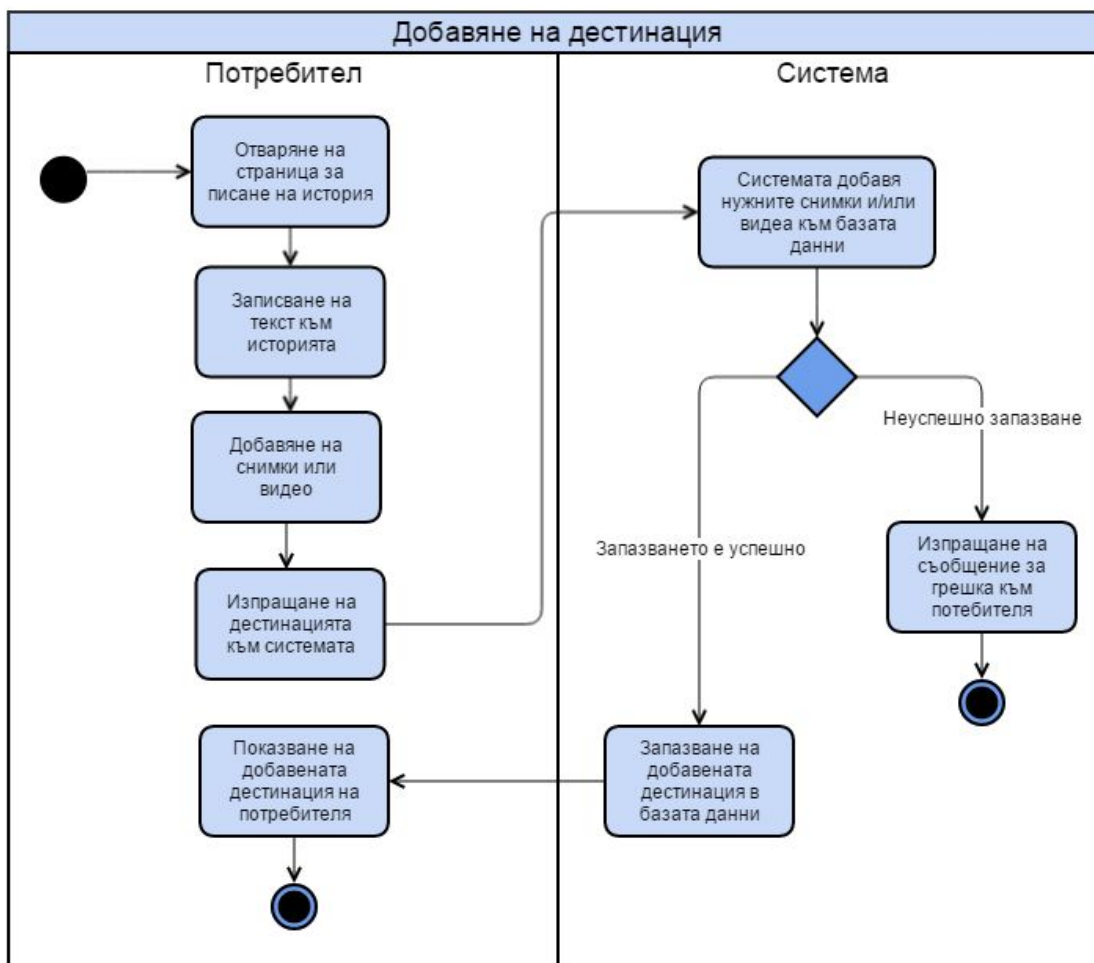
#### 2.b Несъвпадение на името

1. Системата изкарва съобщение за грешка, което дава подробна информация
2. Потребител е върнат в начален екран
3. The use case ends

## UC-4 Вход като потребител

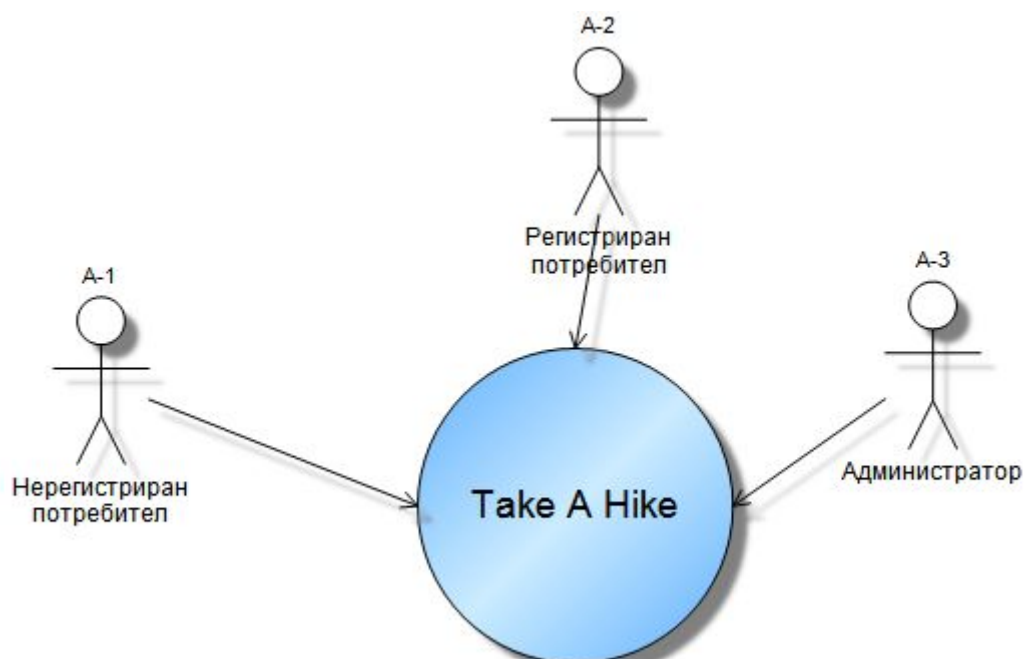


## 2.4. Activity диаграми



**Ф и г у р а 2: Activity д и а г р а м а з а  
д о б а в я н е н а д е с т и н а ц и я**

## 2.5. Други диаграми



ф и г у р а 5: К о н т е к с т н а д и а г р а м а н а  
с и с т е м а т а

## 2.6. SPA Frontend

Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
1.1. Начална страница	Начална сттаница на приложението с форма за влизане.	/
1.2. Feed	Показва страница с последни истории	/feed
1.3. Профил	Показва профила на потребителя, който се е вписал	/profile
1.4. Моите истории	Показва страница с всички истории на даден потребител	/myStories

## 2.7. API Resources



1. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
1.1. Истории	GET заявка, която дава всички истории хронологично POST заявка - добавя история	/api/story?userId=<userId>
4.2 История	PUT заявка - променя текуща история GET - дава история	/api/story/{storyId}
1.2. Потребител	GET данни за потребител спрямо неговото ID.	/api/user/{userId}
	GET, PUT промяна на потребител	/api/me
1.3.	PUT, DELETE - за потребител (админ да редактира/затрива потребител)	/api/user/{userId}

## 3. Описание на реализацията

### 3.1. Избор и обосновка на избраните технологии

Технологиите, които бяха използвани в проекта са:

- React - React router - <https://facebook.github.io/react/> ;  
<https://github.com/ReactTraining/react-router>
- Axios - <https://github.com/mzabriskie/axios>
- Material-ui - <http://www.material-ui.com/#/>
- Express.js - <https://expressjs.com/>
- Mongo - <https://www.mongodb.com/>
- Mongoose - <http://mongoosejs.com/>
- NodeJS - <https://nodejs.org/en/>
- Passport - <http://passportjs.org/>
- Create React App -  
<https://github.com/facebookincubator/create-react-app>
  - o webpack with webpack-dev-server, html-webpack-plugin and style-loader
  - o Babel with ES6 and extensions used by Facebook (JSX, object spread, class properties)
  - o Autoprefixer
  - o ESLint

## 3.2. Описание на реализацията на основните функции

След като избрах част от технологиите за използване, трябваше да избира каква система за управление на версиите на проекта ще използвам. Решихм, че ще използваме Git, като имам repository в github.

Структурата на кода е следната:

Папки server и client:

- Server
  - controllers - login, story - requests
  - models - db connector, user, story - db schemas
  - authentication.js - passport.js
  - server.js - пуска се node server
- Client/src
  - components - main part
  - pages - welcome - first page
  - app.js - BrowserRouter
  - index.js - ReactDOM.render()

## 3.3. Инсталация, поддръжка, потребителска документация

Системата е налична online. Нужен е един от браузърите:

- [Internet Explorer](#) (версия 9.0 или по-висока);
- [Mozilla FireFox](#) (версия 27 или по-висока);
- [Google Chrome](#) (версия 33 или по-висока);
- [Opera](#) (версия 20 или по-висока).

Потребителят няма нужда от помощ, за да се ориентира в системата, тъй като тя е направена по аналог на други системи, с които потребителят е свикнал

## 4. Инсталиране и конфигуриране

Инсталирането може да бъде намерено в README файла на следната страница  
[https://github.com/mimipaskova/NodeJS\\_React\\_Hike](https://github.com/mimipaskova/NodeJS_React_Hike)

## Setup

---

You need nodejs, npm, mongo

- clone the repo
- `npm install` in client and server folders
- start mongod
- `npm start` in client and server folders
- now the app is running on port 3000 - <http://localhost:3000/>

## 5. Оценяване

### 5.1. Описание на планираните методи за оценка на разработения интерфейс

Системата се тества на няколко етапа. Първият етап е тестване от QA екип от поне двама души. Целта е всеки независимо да тества потребителските случаи и техните изключения и след това да сверят мнението си. Тестваше се User Experience на потребителите.

Вторият етап от тестване предстои. Той ще се състои в пускане на системата в среда с повече на брой тестови потребители ( между 50 - 100) от различни групи (алфа и бета тестване).

### 5.2. Описание на проведените експерименти за оценка на интерфейса на системата

Самата идея за създаване на такъв софтуер възникна при наблюдение на няколко семейства, които обичат да посещават различни дестинации в България.

Процесите, през които преминаваше всяко семейство бяха мудни и ги стресираща излишно.

Едно от наблюденията се състоеше в следното:

Семейството решаваше къде да отиде. След като си избираше район, в който иска да посети някаква дестинация, някой член от семейството търсеше в интернет, в форуми и звънеше на познати, за да се посъветва за точното място, където може да се направи преход/посещение на туристическа забележителност. След неколкодневни проучвания, семейството обсъждаше цялата информация, която е научил всеки член от семейството и взема решение къде точно да се отседне и какъв да е маршрута на дестинацията.

Смятам, че този проблем може да се реши чрез системата „Take A Hike“, тъй като тя предоставя бързо и лесно цялата информация нужна на човек при желание на посещение на дадена дестинация.

След създаването на системата, показах я на тези семейства като ги остави сами да се ориентират в нея. На всеки член от семейството бяха нужни под 10 минути за да се ориентира в системата и да разбере основната и функционалност.

### **5.3. Анализ на проведените експерименти и основни заключения относно качеството на интерфейса**

При показването на системата на потенциални потребители, те дадоха няколко забележки по някои от функционалностите. Всички те бяха поправени с цел по-високо качество на софтуера.

### **5.4. Идеи за бъдещо развитие и подобряване на интерфейса**

- Част от бъдещото развитие на интерфейса е интерактивна карта.
- Въвеждане на възможност за комуникация между потребители чрез съобщения и чат канали
- Създаване на „съвети“ (помощ) за системата, което да е интерактивно на стъпки. Всяка стъпка показва даден бутон и описва функционалностите му
- Добавяне на още полета при регистрация с цел анализ на потребителите на системата

- Анализ на всяка страница за поведението на потребителите, за да става все по-интересна системата за тях
- Създаване на отделна секция за дестинации извън България
- Мобилно приложение, в което след засичане с GPS на прехода/разходката да е възможно, да се публикува история

## 6. Материали

### 6.1. Презентации -

<https://github.com/iproduct/course-node-express-react/wiki>

### 6.2. GitHub - [https://github.com/mimipaskova/NodeJS\\_React\\_Hike](https://github.com/mimipaskova/NodeJS_React_Hike)

### 6.3. GitHub issues -

[https://github.com/mimipaskova/NodeJS\\_React\\_Hike/issues](https://github.com/mimipaskova/NodeJS_React_Hike/issues)

### 6.4. React - React router - <https://facebook.github.io/react/> ;

<https://github.com/ReactTraining/react-router>

### 6.5. Axios - <https://github.com/mzabriskie/axios>

### 6.6. Material-ui - <http://www.material-ui.com/#/>

### 6.7. Express.js - <https://expressjs.com/>

### 6.8. Mongo - <https://www.mongodb.com/>

### 6.9. Mongoose - <http://mongoosejs.com/>

### 6.10. NodeJS - <https://nodejs.org/en/>

### 6.11. Passport - <http://passportjs.org/>

### 6.12. Create React App -

<https://github.com/facebookincubator/create-react-app>