**Софтуерни изисквания**

**за**

**QueueSizeMatters**

**Автори:**

**Бойко Цветанов**

**Лиляна Видева**

**Милен Дончев**

**Силвия Янакиева**

**Христо Христов**

**СЪДЪРЖАНИЕ**

1. **[Въведение………………………………………………………](#kx37f6jwv1vt)**[1.1 Цел](#44a7q6iblxb9)[1.2 Документални конвенции](#8y8u0pbu0n4)  
   [1.3 Целева аудитория на документацията](#bji514s73lep)  
   [1.4 Обхват на продукта](#wjnv02vjzjrg)  
   [1.5 Препратки](#ojlcspe2kz0o)

1. **[Кратко Описание………………………………………………](#4yqbpquvmomp)**[2.1 Перспектива на продукта](#8sxx0ex6hbx)  
   [2.2 Основна функционалност](#3y0z8lmzam)  
   [2.3 Потребителски класове и характеристики](#1gl0y38vgp1w)  
   [2.4 Работна среда](#mtftvltvk4mr)  
   [2.5 Ограничения на дизайна и имплементацията](#j0mwckipecvk)  
   [2.6 Потребителска документация](#8zft97jqgruj)  
   [2.7 Предположения и зависимости](#vhvg6xdn69xx)

1. **[Външни изисквания на интерфейса](#y35rklcdo6i)**[3.1 Потребителски интерфейси](#xbhkvgn2pegn)  
   [3.2 Хардуерни интерфейси](#fye0nzs61ccv)  
   [3.3 Софтуерни интерфейси](#k42w2wdrrpm5)  
   [3.4 Комуникативни интерфейси](#9l8h63pyq4pj)

1. **[Свойства на системата](#hui38alojnw)****[4.1 Регистриране на обектите](#3x0m38zc87x8)**  
   [4.1.1 Описание и приоритет](#n4s3rnl16p9z)  
   [4.1.2 Последователност от действия](#darmtzt8mxxh)   
   [4.1.3 Функционални изисквания](#yqpcmzwqhs2a)  
   **[4.2 Броене на хората на дадена опашка](#qj0wjkqtzq0c)**  
   [4.2.1 Описание и приоритет](#4726qvbtc3ta)  
   [4.2.2 Последователност от действия](#1skvayllld0r)  
   [4.2.3 Функционални изисквания](#ycj7ek21sq20)  
   **[4.3 Търсене на обект](#wgkjzbcpkzza)**  
   [4.3.1 Описание и приоритет](#oro5kpl3vq4n)  
   [4.3.2 Последователност от действия](#3j1ri8xgkry)  
   [4.3.3 Функционални изисквания](#9jux3ywwmkp3)  
   **[4.4 Проверка на дължината на опашката на даден обект](#a8h2md6xkotx)**  
   [4.4.1 Описание и приоритет](#o33tx3qt3g0)  
   [4.4.2 Последователност от действия](#xk6vr63i8pbs)  
   [4.4.3 Функционални изисквания](#yoaidz4r1p8e)  
   **[4.5 Отбелязване на “любими” обекти](#b01jioix15ax)**  
   [4.5.1 Описание и приоритет](#41g4m5jxwhix)  
   [4.5.2 Последователност от действия](#ux2rvhkaqf3m)  
   [4.5.3 Функционални изисквания](#1nqcehr3ypsr)  
   **[4.6 Статистическа извадка на данните](#h0su51igyod9)**  
   [4.6.1 Описание и приоритет](#uh58865a3b33)  
   [4.6.2 Последователност от действия](#szaz1kn7lft5)  
   [4.6.3 Функционални изисквания](#50o0qr9c8t52)  
   **[4.7 Показване на обектите по критерий](#et8a6pnamncf)**  
   [4.7.1 Описание и приоритет](#2ts9mn5edvaw)  
   [4.7.2 Последователност от действия](#930mur1nydne)  
   [4.7.3 Функционални изисквания](#190dhuxdws23)

1. **[Нефункционални изисквания](#j1f445vdztpl)**[5.1 Бързодействие](#jnezu628so2h)  
   [5.2 Сигурност](#bag7wxs2m7dx)  
   [5.3 Защита](#oxdl5ojplrh)  
   [5.4 Качествени атрибути](#2hbh25tv0aov)  
   [5.5 Бизнес правила](#kepo0gvzwg3e)
2. [**Други изисквания**](#6uk2bht87p9y)[Притурка А: Технологичен речник](#f5yi7bytcs1)  
   [Притурка Б: Аналитични модели](#efhkmm9ecs2p)  
   [Притурка В: Списък с препратки за доопределяне](#z38doza48d7f)

**QueueSizeMatters**

*Размерът на опашката има значение*

# **[1. Въведение](#n3ysqevpt5qh)**

## **[1.1 Цел](#pviq9lrwr0a5)**

Този документ има за цел подробно да опише софтуерните изисквания за изготвяне на мобилното приложение– QueueSizeMatters.

## **[1.2 Документални конвенции](#2no70qm0bhhq)**

подчертан текст = хипервръзка

## **[1.3 Целева аудитория на документацията](#ln94gmyjo301)**

Този документ служи като основа и ръководство в процеса на планиране и дизайн на приложението. Изискваната функционалност e описана подробно в него.

Предназначен е за project мениджъри, разработчици и заинтересовани бизнес клиенти.

## **[1.4 Обхват на продукта](#x5a2jyoscvab)**

Целта на QueueSizeMatters е да предоставя информация за опашките от клиенти на дадени търговски/ граждански обекти в реално време. Обектите могат да бъдат спирки от градския транспорт, търговски вериги, светофари, стадиони и т.н.

Проектът е предназначен за приложение в градски условия.

# **[2.Кратко Описание](#f4wx02xy3te8)**

## **[2.1 Перспектива на продукта](#4nddw0f0vw1n)**

QueueSizeMatters е неразработван досега проект. Не е част от други системи, работата му не зависи от външни продукти, освен устройствата за преброяване на хора.

## **[2.2 Основна функционалност](#fl8k45xka3kh)**

Основни задачи на продукта са:

* Събиране и обработване информация за дължина на опашка, посредством датчици за преброяване на хора.
* Пренос и съхранение на придобитите данни от датчиците към базата данни на сървъра.
* Обработка на постъпилата информация.
* Разпространение на резултатите сред потребителите в реално време, посредством удобен интерфейс.  
  [(диаграма)](#ee0hmo809f2f)

## **[2.3 Потребителски класове и характеристики](#8d0w7b9ca26n)**

Основния предвиден клас потребители на отброяващата система са хора, притежатели на мобилни устройства, без ограничения относно размери на доход, професия, пол, възраст. За да могат да използват приложението, техните устройства трябва да отговарят на съответстващите му изисквания за работа. Предвидено е данните,събирани от системата, да бъдат използвани за съставяне на статистически извадки.

## **[2.4 Работна среда](#l3d3k3klhjtt)**

Продуктът е предназначен за употреба чрез мобилни устройства(Android и iOS на по- късен етап), като евентуално средата може бъде достъпна и чрез уеб сайт.

## **[2.5 Ограничения на дизайна и имплементацията](#a2ul66z4jts2)**

Приложението има нужда от връзка с интернет и GPS. Паметта на устройството няма да бъде повлияна от приложението – данни няма да бъдат съхранявани на него (освен за кеш и запазени любими локации)

## **[2.6 Потребителска документация](#cmpixar3cv4w)**

Ще има въведение(tutorial) при първо стартиране на приложението. Потребителят ще придоби умениятя, нужни да използва всички функционалности на приложението, за 20 минути. Също ще има и помощ(help option) в самото приложение(евентуално и в web сайта).

## **[2.7 Предположения и зависимости](#v4f27e3nve5)**

Поддръжка на сензорите за броене на хората при повреда или подмяна.

Всеки собственик на търговски или друг обект сам преценява дали да използва приложението и да си закупи датчиците за отчитане на хората.

Предвидена е обработка на голям брой заявки към сървъра.

# **[3.Външни изисквания на интерфейса](#3d3nk1k0y2y6)**

## **[3.1 Потребителски интерфейси](#9e9vuwdujt2)**

**3.1.1 Начална страница**



Началната страница представлява основните менюта на приложението.

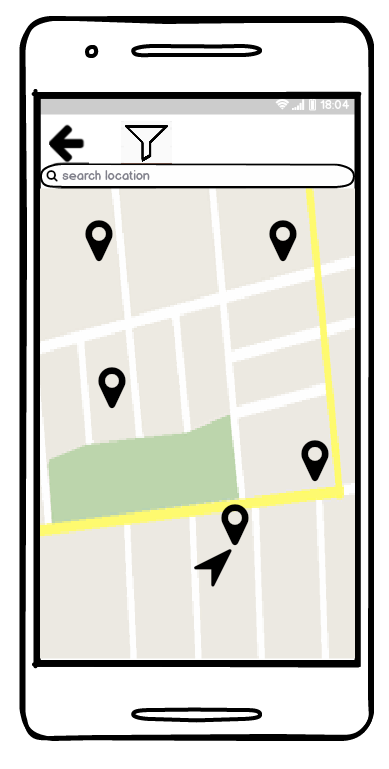
При натискане на “търсене”, се отива на изгледа с картата и search полето. Описано е подробно по-надолу.

При натискане на “любими”, се отива на изгледа с данните за любими и текущите им детайли.

При натискане на “случаен обект”, се отива на случаен регистриран обект на картата и детайли за обекта (брой души, средно време и т.н).

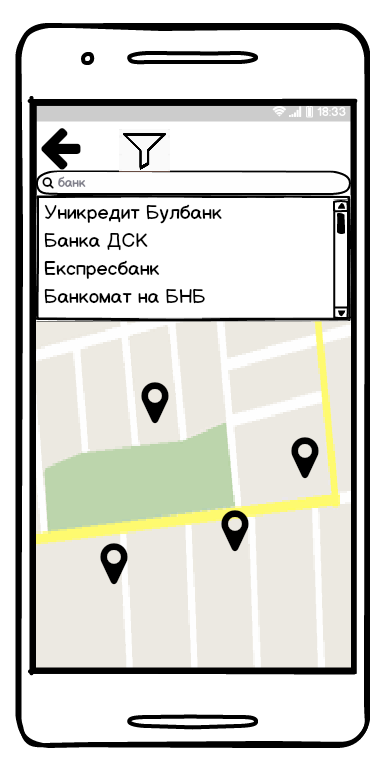
Отделно ще има и поле за реклама, която ще представлява линк в интернет.

**3.1.2 Търсене на обекти**



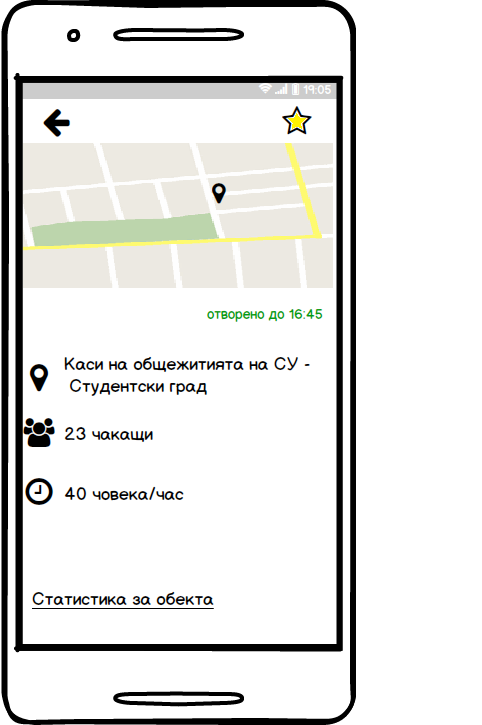
В полето “search location” потребителят ще има възможност да потърси желаното от него местоположение по име. Това може да направи и директно върху картата, където обектите, чиито данни са достъпни за приложението са изрично обозначени.

Бутонът за филтриране помага за по-лесното отсяване на търсените локации, а чрез стрелката (бутонът назад) потребителят се връща обратно към началната страница на приложението.



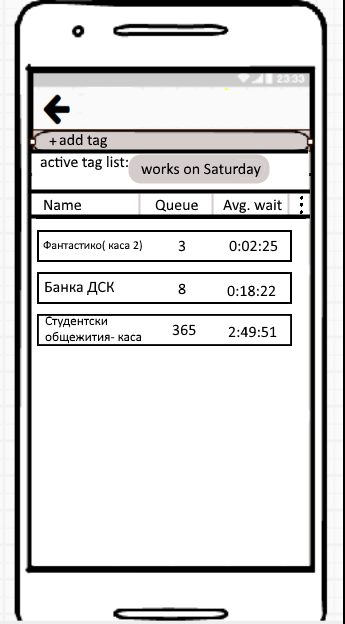
При търсене на обект по име се появява списък с предложения,които биха могли да помогнат на потребителя да намери мястото по-лесно, като може междувременно да се ориентира и по картата.

**3.1.3 Преглед на обект**



Този изглед дава информация за местоположението на обекта на картата, броя на хората на опашката и скоростта, с която се движи опашката. Под картата с обозначения обект има информация за работното време на обекта - дали е отворен и до колко часа или кога отваря. Най-долу на екрана има хипервръзка “Статистика за обекта”. При натискане на тази връзка се отваря изглед със статистическа извадка за обекта.

**3.1.4 Филтриране и сортиране на резултатите**



При натискане на бутона с филтъра( намира се в прозореца за търсене на обект) се появява екрана за показване по параметри/ етикети.

В полето за въвеждане се добавят етикети, а под него са изобразени всички избрани етикети. При натискане върху вече наличен етикет, той се премахва от филтрирането.

Name е задължителен параметър. Останалите параметри са опционални- чрез натискане на триеточието най- вдясно на реда се появява меню за избор на параметри( и приоритети). След затваряне на това меню списъкът се сортира по обозначения начин.

## **[3](#7yncn4e0c10j)**[**.2 Хардуерни интерфейси**](#7yncn4e0c10j)

Проектът изисква набиране на данни за дължината на дадена опашка чрез външна система за преброяване на хора, която използва като хардуерен продукт датчици. Те доставят информация за натовареността на дадено място в цифри, като за тази цел използват протоколите HTTP и Simple Sensor Interface Protocol.

## **[3](#xrcg9bgyows5)**[**.3 Софтуерни интерфейси**](#xrcg9bgyows5)

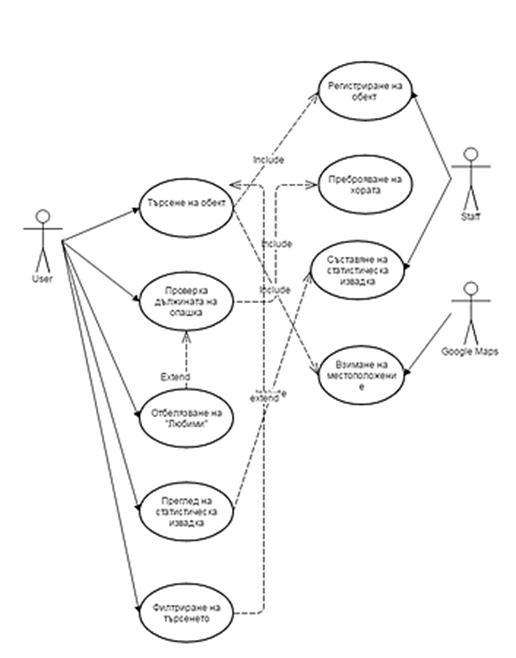
С цел успешното реализиране на проекта е необходимо използването на приложението Google Maps. Неговата роля е върху географска карта да бъдат обозначени обектите, които предоставят информация за дължината на опашката от хора там, като за тази цел то използва Google Data Structure Protocol.

## **[3](#brggc8d1ut60)**[**.4 Комуникационни интерфейси**](#brggc8d1ut60)

Идеята на проекта включва активна комуникация между датчиците и системата, която включва доставка на актуални данни за дължината на опашката на дадено местоположение посредством сигнали. За тази цел се налага използването на протокола HTTP.

# **[4. Свойства на системата](#8ezdqpc2ebbj)**

Тази диаграма описва основните функции на приложението. Работата и изискванията за всяка функция е подробно описана в следващите подточки.



## **[4.1 Регистриране на обектите](#5byxxoywuy7n)**

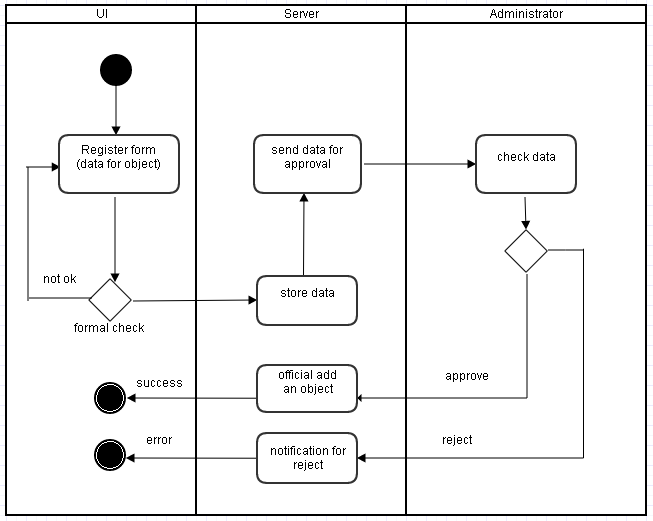
### **[4.1.1](#4j89sked6d42)** [**Описание и приоритет**](#4j89sked6d42)

Бизнес представител на приложението сключва договор със собственика. Ако има система за преброяване(банки, общини, т.н) се използва тя, иначе технически екип инсталира нова и я обвързва с устройството, отговорно за изпращането на информацията до сървъра. Необходимите условия за да бъде осъществено регистрирането са: предоставяне на адрес (може и локация от maps), заплащане на хардуера, отговорен за преброяването, съгласяване с общите условия на проекта (например без злоупотреби с цел увеличаване на клиенти) и предоставяне на достоверна информация от страна на управителя или собственика на обекта. Приоритетът му е второстепен, защото зависи от функционалността за преброяване на хората.

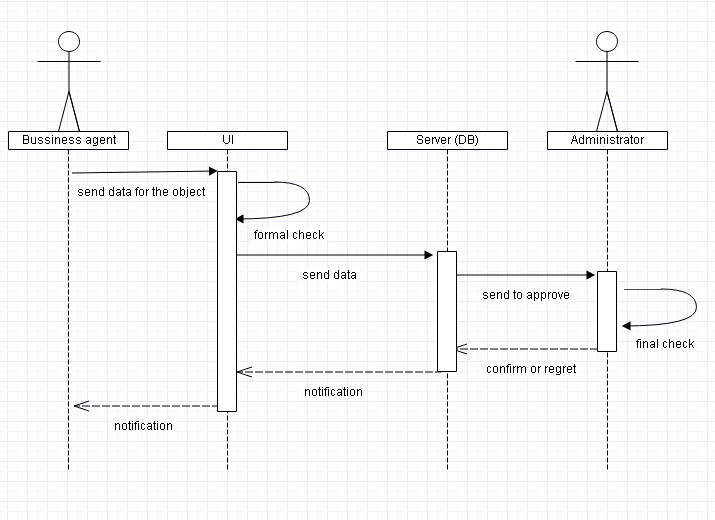
### **[4.1.2](#d9utqezgykhy)** [**Последователност**](#d9utqezgykhy) **от действия**

В приложението ще има форма за регистриране на обект, където ще се попълнят данните за фирмата, те ще се изпратят към базата данни на сървъра със статус „в изчакване”. След това администратор на системата може да одобри регистрацията или да я отхвърли. И в двата варианта, служителят който е изпратил заявката получава нотификация по email, sms или в самото приложение, че заявката е съответно одобрена или отхвърлена. След одобрение, обектът е официално признат и наличен в системата.

Процесът би изглеждал така:



По-подробно представяне на процесите:



### **[4.1.3](#1zba216xjku6)** [**Функционални изисквания**](#1zba216xjku6)

4.1.3.1: Обектът да притежава устройство за преброяване.

4.1.3.2: Изпращане на данните за регистрация на обекта към сървъра.

4.1.3.3: Осведомяване на клиента за това дали регистрацията е била успешна

## **[4.2 Броене на хората на дадена опашка](#dqbvmxexbc45)**

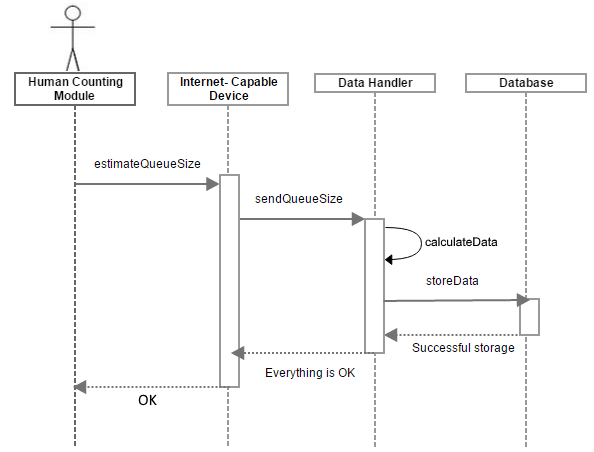
### **[4.2.1](#783zhjcdff1f)** [**Описание и приоритет**](#783zhjcdff1f)

Приложението придобива представа за размера на опашката от хора чрез външна и независима от този проект система за преброяване на хора. Чрез това решение се преброява всеки човек, който е на опашката, независимо дали той има приложението на телефона си или не. Всички останали функционалности на приложението зависят от тази и за това тя е с най- висок приоритет.

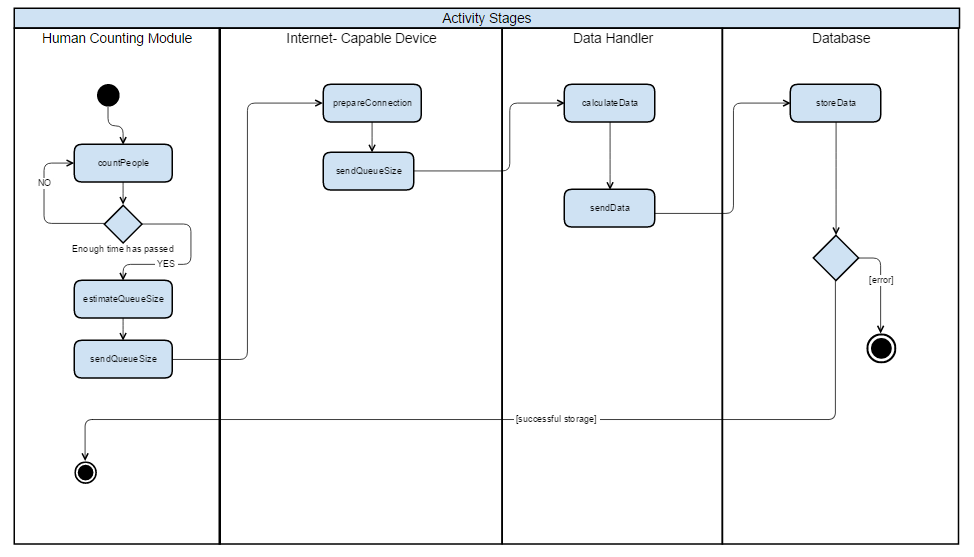
### **[4.2.2](#b7ig97idud8w)** [**Последователност**](#b7ig97idud8w) **от действия**

Преброяването на опашката се извършва от външна система за преброяване на хора, като методът, който тя използва, не ни интересува, стига да е прецизен. Internet- capable устройство събира данните от системата и ги изпраща на сървъра през 30 секунди( а не непрекъснато) с цел спестяване на ресурси, след което данните се обработват от DataHandler, който след това ги складира в базата данни.

Начинът на събиране на информацията е описан в следната диаграма:



По- подробна визуализация на горния процес:



### **[4.2.3](#fl91b5ueo26m)** [**Функционални изисквания**](#fl91b5ueo26m)

4.2.3.1: Преброяване на дължината на опашката (при налична адекватна СПХ)

4.2.3.2: Изпращане на средните данни към сървъра и складирането им

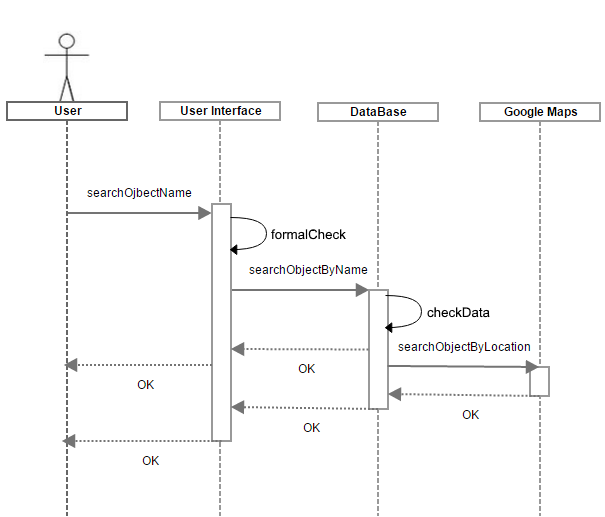
## **[4.3 Търсене на обекти](#e10e2nht4udw)**

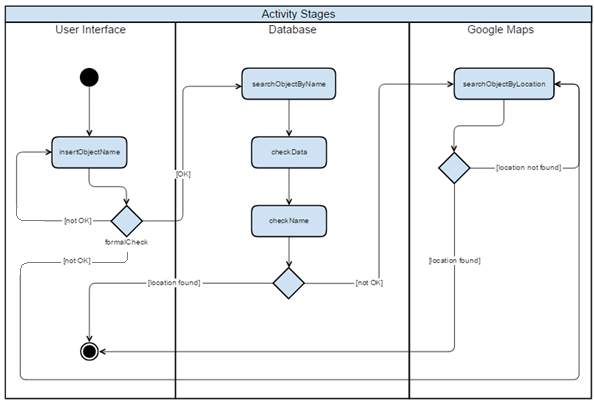
### **[4.3.1 Описание и приоритет](#5hcfoinniwq2)**

Това свойство е свързано с локализирането на места, на които всеки потребител може да види колко е дължината на опашката от хора. То е от изключителна важност за целта на приложението, тъй като самата идея на проекта е да се следи своеобразно трафика в дадени точки на населеното място.

### **[4.3.2 Последователност](#2aupjkj6y8f) от действия**

Първи вариант, по който могат да се търсят обекти е като в диалогов прозорец потребителят въвежда името на искания обект. Системата от своя страна приема информацията, обработва я, проверява дали желаната локация е част от базата й данни и съответно извежда подходящо съобщение. Ако мястото е открито, то в приложението се отваря прозорец със стойностите за дължината на опашката от хора в конкретния момент. Ако не е, на потребителя се предоставя възможност да потърси обекта директно върху картата, което е и вторият възможен начин за определяне на обект. При него лицето, ползващо приложението, има възможност да намери желаното място като го потърси върху картата. В зависимост от това дали успее да го локализира, съответно ще получи данни за него или не. Пресъздаването на процеса би изглеждало така:



По-подробно илюстриране на процесите:

### **[4.3.3 Функционални изисквания](#i0ee9o3ea2xz)**

4.3.3.1 Съхранение на информация за обектите

4.3.3.2 Търсене на обект по име

4.3.3.3 Визуализация на обектите върху Google Maps карта

4.3.3.4 Извеждане на стойностите за дължината на опашката в даден обект

## **[4.4 Проверка на дължина на опашката на даден обект](#3oyxfvnmwpg1)**

### **[4.4.1 Описание и Приоритет](#ulsqlmvugk6y)**

Проверката за дължина на опашката е главната функционалност. Тя е основна идея на системата и е с висок приоритет. В пряка зависимост е от предшестващите я операции (наличие на обекта в базата и успешното му намиране).

### **[4.4.2 Последователност от действия](#de35geh9hzlz)**

Проверката за дължина на опашката започва след като потребителят е приключил с търсенето на избрания от него обект.

С натискане веднъж той избира най-подходящия обект от списъка, върнат от търсенето

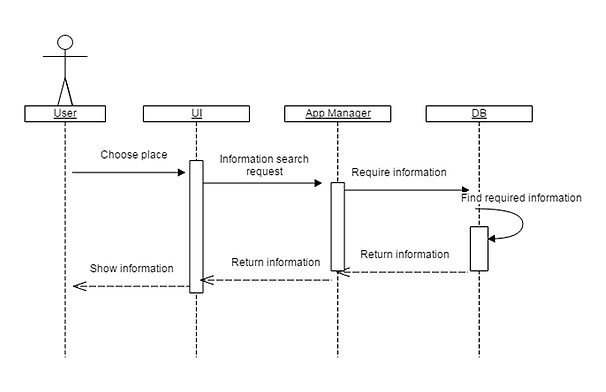
Отваря се нов изглед на приложението, в който е указана дължината на опашката

на съответното място, скоростта, с която се придвижва опашката (обслужени хора за единица време (например 1 час)), както и местоположението му на картата

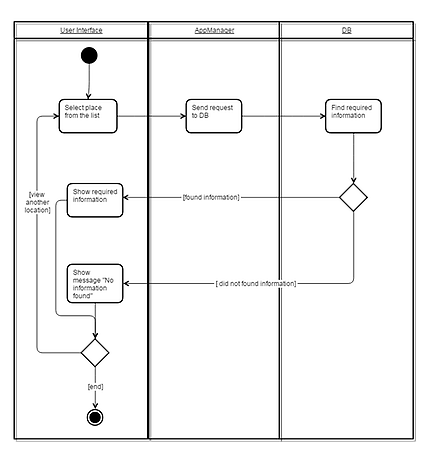
С натискане на бутона "Назад" в горния ляв ъгъл на екрана, потребителят може да се

върне обратно в списъка с намерени обекти и да избере нов.

Успешния преглед на информация протича по следния начин:



### В общия случай, дейностите по прегледа са следните:



### **[4.4.3 Функционални изисквания](#e8u58rybfabn)**

4.4.3.1Наличие на информация за обектите

4.4.3.2 Успешно завършено търсене на обект

4.4.3.3 Обектът да е регистриран

## **[4.5 Отбелязване на „любими“ обекти](#8p5erl4640xp)**

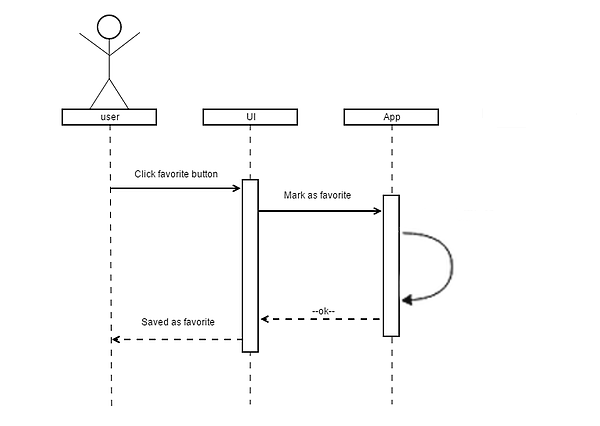
## **[4.5.1 Описание и Приоритет](#z51vipllwvp7)**

Отбелязването на “любими” обекти е второстепенна функционалност. Направена е с цел удобство на потребителя и е с нисък приоритет за цялата система.

### **[4.5.2 Последователност от действия](#dd0iwil98pan)**

### Отбелязването на “любими” обекти ще става чрез натискане на бутон с формата на звездичка. С натискане на бутона дадения обект се запазва като любим в данните на приложението (в паметта на устройството). Има графа “любими обекти”, където потребителят може да види по всяко време какви опашки има на обектите, които е маркирал като “любими”.

Отбелязването на “любим” обект протича по следния начин:



### **[4.5.3 Функционални изисквания](#i0hd6dwwyak7)**

4.5.3.1. Обектът трябва да е регистриран в системата ([виж 4.1](#3x0m38zc87x8))

4.5.3.2. В графа “любими обекти” потребителят може да види по всяко време какви опашки има на обектите, които обича да посещава.

## **[4.6 Статистическа извадка на данните](#lpn6zyhfaeuh)**

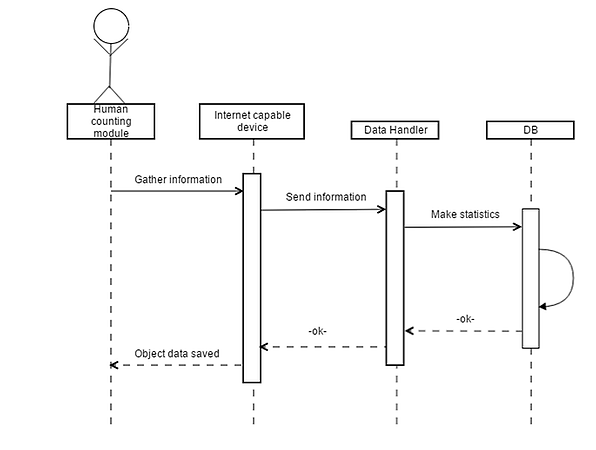
### **[4.6.1 Описание и Приоритет](#79awtq6dbvlu)**

Статистическата извадка на данните ще се записва в историята на обекта. Така може да се напише математически модел, по който да се направи прогноза за опашката на обекта в даден час на деня.Тази функционалност е със среден приоритет, защото подпомага главната идея на системата.

### **[4.6.2 Последователност](#hnxzjsic1r6o)** **от действия**

Статистическата извадка на данните се извършва всеки ден за всеки обект. Данните за дължините на опашките се записват за всеки час от деня в Data Handler-а, където след края на работното време се смята средната дължина на опашката за деня и се записва в датабазата. Чрез добре направен математически модел по тези статистически извадки, се прави прогноза за дължината на опашките в съответния час през всеки ден от седмицата.Тези данни са достъпни за потребителя.

Статистическата извадка на данните се извършва по следния начин:



### **[4.6.3 Функционални изисквания](#cp6tux9gic5r)**

4.6.3.1. Почасово запазване на информацията за съответния обект.

4.6.3.2. Данните да са достъпни за потребителя.

4.6.3.3. Всеки обект да има своя статистика.

## **[4.7 Показване на обектите по критерий](#f76zpd1bnpym)**

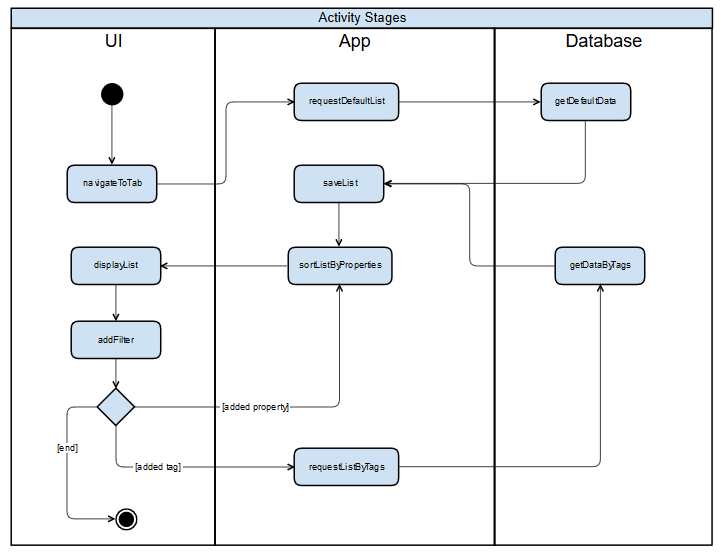
### **[4.7.1 Описание и Приоритет](#47f81h69f4y7)**

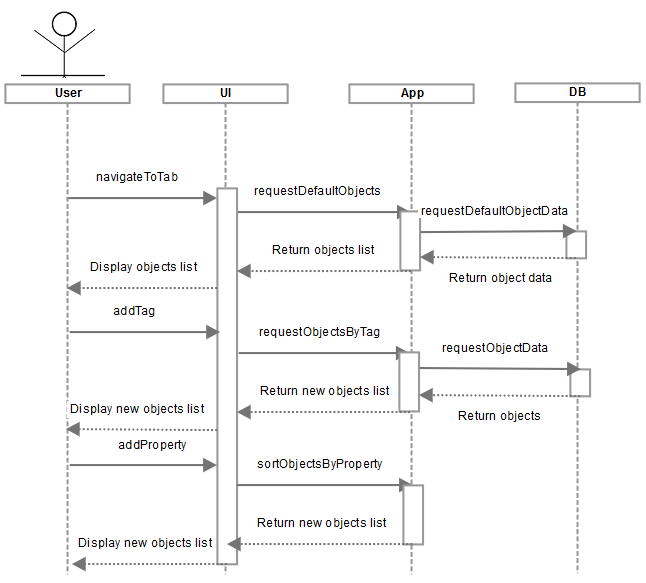
При превключване на секция( на английски: tab) се отваря списък с всички регистрирани обекти, които могат да се подредят според избраните от следните критерии: популярност, посещаемост, разстояние от текущото местоположение на устройството, време за чакане на опашка. Също така могат да се показват само обекти от определен тип, например: хранителни вериги, банки, бензиностанции, дискотеки, специални обекти( напр. опашка на каса за общежития).

### **[4.7.2 Последователност](#degt1ybq3mum) от действия**

Потребителят навигира приложението до екрана, на което обектите могат да бъдат сортирани по изброените отгоре критерии. На този екран първоначално ще бъдат показани обекти по подразбиране( обекти, които са платили за реклама, задължително ще бъдат вмъкнати в този списък), след което потребителят ще може да модифицира този списък спрямо етикети и свойства, които задава. Показаните резултати ще имат част от данните си изобразени на екрана.

Процесът е изобразен на следващата диаграма:





### **[4.7.3 Функционални изисквания](#5td5nql9krnx)**

4.7.3.1: Зареждане на списък по подразбиране

4.7.3.2: Показване на обекти спрямо избраните етикети

4.7.3.3: Сортиране на обекти спрямо избраните свойства

4.7.3.4: Изобразяване на някои от данните на показаните обекти

# **[5. Нефункционални изисквания](#5ozhh7d91nvv)**

## **[5.1 Бързодействие](#ghtgqpkk4qys)**

Бързодействието зависи от натовареността на сървъра. Колкото повече заявки приема, толкова по-бавно ги обработва. Също така трябва и приложението да е бързо, тоест потребителят да не чака повече от 3 секунди докато му се отвори дадено меню или използва някоя функционалност.

Ефикасността зависи и от скоростта на комуникацията между отделните компоненти (системи).

## **[5.2 Сигурност](#vzyol3usbwjt)**

Когато интернет достъпът е под 50Mbps и GPS сигналът не достига 1500MHz, може да се получи забавяне при локализирането на конкретен обект. За да може да се минимизират максимално тези неудобства, при стартиране на приложението ще се проверява връзката с онлайн мрежата и, ако тя не е задоволителна – то няма да се отвори.

[**5.3 Защита**](#yhmchyvkwqz3)

Данните за обектите биват запазени в бази от данни. Базите се намират на сървър, защитен с firewall. Паролата за достъп до данните е известна само на системния администратор. Сървърът трябва да е достатъчно бърз и надежден, за да обработва голямо количество заявки без забавяне.

Потокът на данни от преброявящите устройства на обектите също е защитен(криптира се). Прехвърлянето на информация се осъществява на всеки 15 секунди, а не при всяко преминаване на човек(от съображения за сигурност и по-добра работа на приложението).

## **[5.4 Качествени атрибути](#87dsb6azldgi)**

Продуктът ще е лесен за поддръжка. На системния администратор ще му се налага да работи до 20 часа всяка седмица в месеца.  
  
Продуктът ще е предназначен предимно за смартфони, затова първоначалния продукт ще бъде написан за Android (защото е най-използван). След това ще се направят версии и за iOS и Windows. Ще има и немобилна версия за Linux, Mac OS и Windows (съвместима за всеки Windows по-нов от или Windows 7)  
  
Продуктът ще е лесен и бърз за използване. При първото пускане на приложението, ще има бърз урок, който да преведе потребителя през всички функционалности на приложението. Потребителят ще е усвоил необходимите умения и знания да използва цялата функционалност до 20 минути след началото на използване на приложението,

## **[5.5 Бизнес правила](#unwhz2bd952z)**

Обикновените потребители имат достъп до текущата натовареност на дадените обекти, както и до статистическите данни свързани с тези от тях, за които е налична такава информация.

Записването и обозначаването на обекти се осъществява от бизнес- представител на приложението, който сключва договор със собствениците на желаещите да се включат в системата обекти.

# **[6.Други изисквания](#pa815a7ng3qu)**

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

[**Притурка А: Технологичен речник**](#sgjubehsc4o5)

* СПХ - Система за преброяване на хора( на английски: Human Counting Module), тя винаги е външна за системата.
* Internet - Capable Device- устройство, което да може да предава данни през Интернет.
* Сървър - компютър, стартиращ сървърен софтуер и предоставящ една или повече услуги (като например хост) на други компютри в същата мрежа. В повечето случаи хардуерните изисквания към този компютър са по-високи от изискванията към хардуера на стандартния настолен компютър, който не функционира като сървър. В допълнение, може да има специални изисквания за архивиране и др.
* Датабаза (DB) - организирана колекция от данни. Тя съдържа схеми,таблици, заявки, отчети, възгледи, както и други обекти. Използва се за събиране, обработка и използване на информация.
* Firewall(Защитна стена) - специализиран хардуер или софтуер, който проверява мрежовия трафик, преминаващ през него, и разрешава или забранява достъпа по определени правила.

**[Притурка Б: Аналитични модели](#73pc8cik3rwh)**

