

**ТЕХНОЛОГИЧЕСКО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ  
към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Java базирана система работеща на Андроид мобилно устройство,  
която има за цел да улесни ориентирането в затворени пространства, част  
1.

Дипломант:  
*Деян Геновски*

Научен ръководител:  
*ас. Кирил Митов*

**С О Ф И Я**  
**2 0 1 2**



## УВОД

В днешно време системите за ориентиране в отворени пространства, за определяне на мястото, на което се намираш в момента и на заобикалящите ни неща са много добре развити.

Те използват глобалната позиционираща система и с помощта на спътници определят местоположението на устройството на земята. Използвайки карти те визуализират на екрана на устройството моментната позиция. С развитието на тези системи се предоставиха и услуги като показване на магазини, ресторанти и други точки, които могат да представляват интерес на ползвателя. Основата цел на подобни системи е да улеснят потребителите при ориентиране в непозната обстановка. Ориентирането в големи затворени пространства (търговски центрове, болници и т.н) също представлява трудност.

Прилагането на устройства използващи глобалната позиционна система в затворени пространства е трудно, тъй като те използват сигнал изпратен от сателити, а този сигнал трудно прониква през бетонни материали.

Това налага използването на различна технология. Един от вариантите е използването на безжични точки за достъп и получената от тях сила на сигнала за определяне на позицията на устройството. Това има редица предимства. Безжичните точки за достъп са относително евтини и малки по размери устройства и могат лесно да бъдат интегрирани незабелязано в сградата. Освен това повечето мобилни телефони в днешно време имат модули за приемане на подобен сигнал и това елиминира нуждата от закупуване на специални устройства за реализиране на подобна технология. И тази технология като всяка друга има недостатъци като например нуждата от изграждане на специална система от безжични точки за достъп. Подобна система може да изчисли позицията на устройството с грешка от 2-3 метра, което е напълно достатъчно за да се ориентиращ къде се намираш в момента.

Целта на този проект е да се изгради система, с която да може да се определя сградата, в която се намираш, да се определи позицията на устройството. Да се предостави метод за автоматично обновление и изтегляне на карта и друга информация свързана с обекта без нуждата потребителя да задава каквато и да било

информация. Да се предостави списък от обекти вътре в сградата, които да се визуализират заедно с позицията върху екрана на устройството. За всеки обект да може да се предостави допълнителна информация.

## **ПЪРВА ГЛАВА**

# **МЕТОДИ И ТЕХНОЛОГИИ ЗА РЕАЛИЗИРАНЕ НА ПОЗИЦИОНИРАЩА СИСТЕМА В ЗАТВОРЕНИ ПРОСТРАНСТВА И ПРЕГЛЕД НА СЪЩЕСТВУВАЩИ СИСТЕМИ**

### ***1.1 Основни принципи на работа на позициониращите системи***

Основния принцип на работа на всички позициониращи системи е, че винаги има множество устройства изпращащи определени данни, които биват разчитани от приемник и с помощта на различни технологии се определя разстоянието от изпращащото до приемащото устройство. Знаейки координатите на трансмитера и разстоянието до него, приемника изчислява позицията, на която се намира. Има различни технологии както за предаване на сигнал, така и за определяне на разстоянието между приемник и предавател.

### ***1.2 Видове позициониращи системи според технологията на предавателя и приемника***

#### **1.2.1 Използване на инфрачервени вълни**

При този вид системи в сградата се поставят множество предаватели, които изпращат през определен интервал от време под формата на инфрачервени лъчи сигнал, който съдържа уникален номер идентифициращ всеки един предавател. Приемника улавя тези сигнали и с помощта на информацията, която има относно положението на всеки един излъчвател определя собствената си позиция.

Недостатък на технологията използваща инфрачервени лъчи е, че при нея е абсолютно задължителна пряката видимост между приемника и предавателя, в противен случай предаването на данни е невъзможно. Освен това тази система би била скъпа за реализиране тъй като на пазара не се предлага прост предавател на уникален инфрачервен сигнал, а трябва да се изработи нестандартен такъв.

### **1.2.2 Използване на RFID четци и предаватели**

Всяка RFID карта притежава уникален номер и поставяйки я на определени места в дадена сграда при прочитането и с четеща и използвайки информация за позицията на RFID картата може да се определи позицията на четеща.

Основен недостатък на подобна система е, че поради технологията която използват пасивната RFID картата и четеща, разстоянието между тях не може да бъде по-голямо от няколко сантиметра.

Този проблем може да бъде решен чрез използване на активен RFID предавател, който може да предава данни на няколко метра.

### **1.2.3 Използване на безжична технология стандарт IEEE 802.1\***

#### **1.2.3.1 IEEE 802.11a/b/g**

Стандартите 802.11a/b/g по-известни като Wi-Fi предлагат редица предимства при използването има за позициониране в сграда. Устройствата са относително евтини, могат да предават на достатъчно големи разстояния без допълнителни преработки, дори и без наличие на пряка видимост между приемника и предавателя.

Принципа на работа на тези устройства е такъв, че безжичната точка за достъп изпраща през определен период от време сигнал наречен Beacon, с който уведомява останалите устройства за присъствието си. В този сигнал се съдържат няколко параметъра, като един от тях е BSSID, уникален номер идентифициращ всяка точка за достъп. Голямо предимство е, че едно клиентско устройство(приемник) може да прочете множество такива сигнали едновременно(реално това не се случва едновременно, но за пренебрежително малко време).

След приемане на beacon пакета, клиентското устройство може да измери силата на приетия сигнал и да го използва за изчисляване на разстоянието до безжичната точка за достъп. Използвайки разстоянието до безжичната точка за достъп и координатите ѝ, приемника може да изчисли своето местоположение.

### **1.2.3.2 IEEE 802.15 Bluetooth**

Повечето от изредените предимства в 1.2.3.1, с изключение на няколко, важат и за технологията Bluetooth.

Проблемът при използване на Bluetooth е, че почти не съществуват самостоятелни устройства, които да работят като Bluetooth безжични точки за достъп. Което означава, че или трябва да се използва устройство, което притежава Bluetooth технология за комуникация, но основното му предназначение не е за безжична точка за достъп. Или използване на множество Bluetooth адаптери свързани към персонален компютър. И в двата случая използването на тази технология би било скъпо и освен това ако се използва персонален компютър, това означава при проблем с него(само едно устройство), се срина цялата система.

### **1.2.4 Начини за изчисляване на разстоянието между приемник и предавател**

#### **1.2.4.1 Изчисляване на разстоянието чрез силата на получения сигнал**

Някои технологии като Bluetooth и Wi-fi позволяват приемника да измери силата на получения сигнал. С помощта на специален алгоритъм, за който са важни видът на антените и силата, с която предава трансмитера, се изчислява разстоянието между приемника и предавателя.

#### **1.2.4.2 Изчисляване на разстоянието чрез измерване на времето за получаване на сигнала**

При някои технологии има възможност при изпращането на данните предавателя да маркира точното време, когато пакетът е бил изпратен. Приемникът от своя страна изчислява времето, което е отнело на информацията да пристигне. Имайки се в предвид времето, за което дадения сигнал изминава единица разстояние се изчислява разстоянието между приемника и предавателя. При този метод се изисква някакъв вид първоначална синхронизация на времето между двете устройства.

*Информацията относно начините за позициониране е описана в [1]*

## ***1.3 Съществуващи системи за позициониране в затворени пространства***

### **1.3.1 Skyhook Wireless**

Skyhook Wireless е компания намираща се в Бостън, която разработва система за позициониране на устройства посредством безжични точки за достъп. Системата работи с точност около 10-20 метра. Предоставя система за позициониране подобна на GPS, но работеща в градски условия и в затворени пространства. Има възможност системата да бъде интегрирана с GPS, като се получава хибридна позиционираща система. Системата събира данни за безжични точки за достъп чрез техника наречена "wardriving". Техниката се състои в обикаляна с кола и използване на мобилно устройство, което сканира за безжични мрежи и за всяка засечена безжична мрежа записва уникалният и BSSID адрес и местоположението, на което е засечена. Голямо предимство на системата е, че работи в градски условия и в закрити помещения, където използването на GPS е трудно и понякога дори невъзможно. Недостатък е това, че точността на позиционирането с подобен тип система и с подобна техника на записване на местоположението на безжичните точки за достъп не може да бъде по малко от 10-15 метра.

### **1.3.2 Google Maps Indoors**

Google Maps Indoors е достъпно за мобилни телефони работещи с операционната система Android и имащи инсталирана версия на Google Maps след 6.0. Системата се използва в закрити пространства, като спомага за лесното ориентиране в големи сгради. Има възможност за добавяне на планове на многоетажна сграда и показване на местоположението на устройството върху картата. Подробности относно технологията, която се използва не са официално представени.

### **1.3.3 Navizon I.T.S**

Системата разработвана от компанията Navizon, наричаща се Indoor Triangulation System, представлява система за следене на мобилни устройства намиращи се в сгради. Използва безжични точки за достъп, чието местоположение предварително се конфигурира заедно с картите на сградата. Системата също може да се използва за



навигиране в големи постройки или за бързо откриване на някои от приятелите ти. Състои се в две части, една работеща на мобилното приложение и администраторска част, в която се добавят безжични точки за достъп и се редактират карти. В администраторския режим има възможност за наблюдаване на устройствата намиращи се в сградата и тяхното местоположение.

## **ВТОРА ГЛАВА**

# **ПРОЕКТИРАНЕ НА СТРУКТУРАТА НА ПОЗИЦИОНИРАЩА СИСТЕМА ЗА ЗАТВОРЕНИ ПРОСТРАНСТВА**

### ***2.1 Функционални изисквания към позиционираща система за затворени пространства***

1. Да се изчислява позицията на устройство с максимална грешка 2-3 метра
2. Автоматично да се определя сградата, в която се намира устройството.
3. Предварително начертаните карти и точки представляващи интерес да се съхраняват на сървър и да се зареждат автоматично на клиентското устройство.
4. Да се прави проверка за обновления на картите на сървъра и автоматично да се свалят на потребителското устройство.

### ***2.2 Избор на технология за позициониране и устройство, на което да се имплементира***

Поради предимствата изброени в 1.2.3.1 беше избрана технология за изчисляване на позиция основаваща се на стандарта IEEE 802.11a/b/g (Wi-Fi).

Освен множеството предимства на самата технология като цяло, под внимание трябваше да се вземе и устройството върху, което трябва да се имплементира. Повечето съвременни мобилни телефони притежават модул за приемане на Wi-Fi сигнали, което значително улеснява създаването на подобна система. По-този начин отпада изискването за създаване на специално устройство. На потребителя ще бъде дадена възможност да използва мобилния си телефон като навигатор. Отсъствието на специално устройство драстично намалява стойността на подобна система.

### ***2.3 Избор на мобилно устройство и среда за разработка***

#### **2.3.1 Избор на мобилно устройство**

За мобилно устройство беше избран телефон работещ с операционната система Android.

Причините са няколко:

1. Android е бързо разпространяваща се платформа, която постоянно бива обновявана
2. На пазара има мобилни телефони с тази операционна система на много достъпни цени
3. Мобилната операционна система Android предоставя удобно API за командване на вградените в мобилното устройство wi-fi модул. Wifi API на Android предоставя абстракция, с която могат да се вземат данни от безжичния модул както и да се задават команди на същия без програмиста да се интересува от конкретния модул и от неговите специфики.

### **2.3.2 Избор на среда за разработка**

Android SDK е безплатен и е на разположение, както за „Windows”, така и за „Linux”. Основния език за разработка на „Android” приложения е „Java” с работна среда „Eclipse”. Наличието на базови познания за този език за програмиране и работната среда „Eclipse” също повлия на избора на платформа.

## ***2.4 Създаване на алгоритъм за изчисляване на позицията на мобилния телефон***

За еднозначното изчисляване на позицията на телефона са нужни координатите на три безжични точки за достъп и разстоянието между тях и телефона.

Позицията се изчислява посредством системата показана на фиг. 2.1

**Фиг 2.1** Система от уравнения за изчисляване на позицията

$$d1 = \sqrt{(m1 - x)^2 + (n1 - y)^2}$$

$$d2 = \sqrt{(m2 - x)^2 + (n2 - y)^2}$$

$$d3 = \sqrt{(m3 - x)^2 + (n3 - y)^2}$$

Параметрите на системата от фиг. 2.1 са както следва:

d1 е разстоянието до първата безжична точка за достъп и съответно d2 и d3 до втората и третата.

(m1,n1) са координатите на първата безжична точка за достъп, (m2,n2) съответно на втората и (m3,n3) на третата.

(x,y) са търсената точка (позицията на устройството).

Уравненията на системата от фиг. 2.1 са взети от така наречената „distance formula”, коя служи за изчисляване на разстоянието между две точки, чиито координати са известни. „Distance formula” е описана в [2].

При изчисляване на позицията се използва „Distance Formula” в обратна посока, при известни разстояния между три точки и търсената и известни координати на трите точки се намира четвъртата неизвестна.

Тъй като разстоянията до безжичните точки за достъп, които ползваме не са точно измерени не можем да вкараме и трите уравнения в система, защото тя няма да има решение. След решаване на видоизменената система показана на фиг. 2.2 се получават две точки.

**Фиг. 2.2** Система при пресмятането, на която се получават две точки

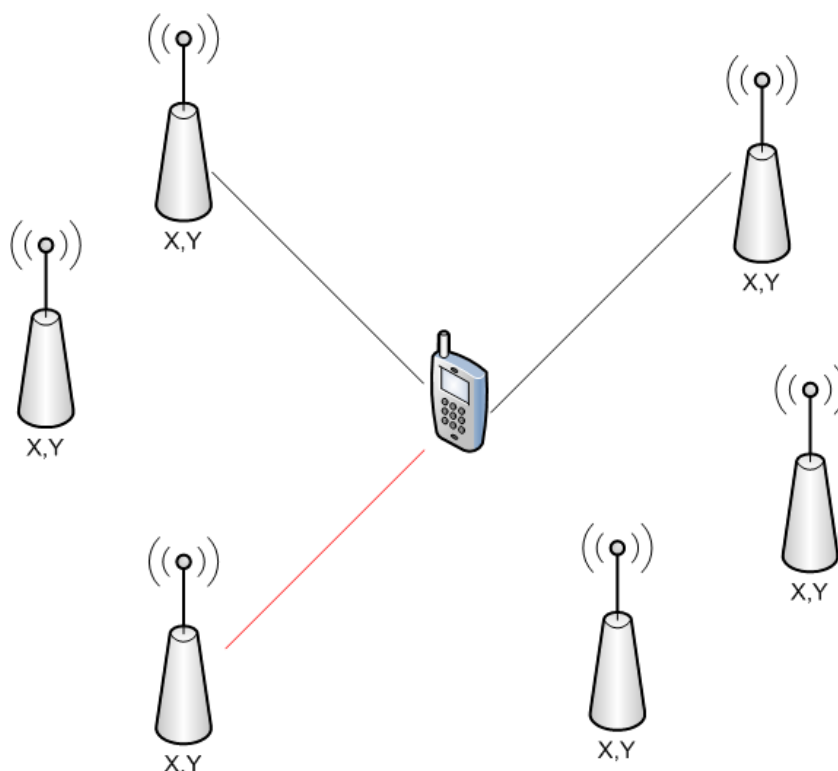
$$d1 = \sqrt{(m1 - x)^2 + (n1 - y)^2}$$

$$d2 = \sqrt{(m2 - x)^2 + (n2 - y)^2}$$

Само една от получените от системата показана фиг. 2.2 точки е вярна. Трета точка, чиито координати са известни се използва за контролна. Изчислява се разстоянието между получените от системата на фиг. 2.2 точки и контролната. Разстоянието, което е по-близо до разстоянието измерено между устройството и контролната точка е разстоянието до правилната точка.

На фиг. 2.3 е показана принципната схема на работа на позициониращата система. Избират се три случайни безжични точки за достъп, изчислява се разстоянието между всяка една от тях и устройството и след това с помощта на тези разстояния и координатите на безжичните точки се изчислява позицията на мобилния телефон.

**Фиг. 2.3** Принципна схема на работа на позициониращата система

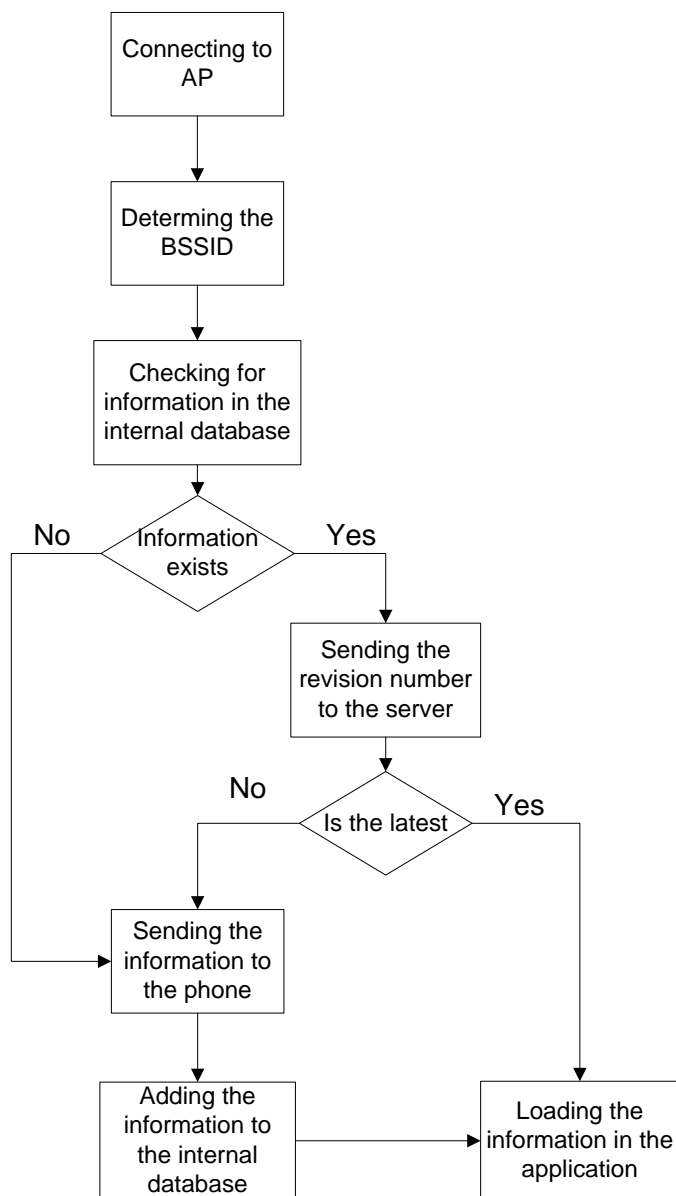


## ***2.5 Определяне на сградата, в която се намираме и вземане на последните данни за нея***

Приложението е направено по такъв начин, че при пускането му първо се определя в коя сграда се намираме и се търсят данни за нея в базата данни на телефона. Ако данните са налични се прави проверка с сървър, дали тези данни са актуални и ако не са, се изискват последните такива от сървър. След получаването им се записват в базата данни на телефона и се зареждат в приложението. Ако данните са налични в вътрешната база данни и са актуални се прави заявка към нея, полученият отговор се обработва и се зарежда в приложението.

Блоковата схема показана на фиг.2.4 онагледява начина на работа на система за определяне на сградата и обновление.

**Фиг. 2.4** Определяне на сградата и изтегляне на последната информация за нея



В базата данни на телефона за всяка една точка за достъп има записан BSSID адрес, който е уникален. Освен това има и идентификационен номер на сградата, в която се намира. Това позволява след свързване към него, лесно да се определи сградата.

## 2.6 Модел на данните използван за съхранение и пренасяне на информацията между клиента и сървъра

На фиг. 2.4 е показана принципния модел на данните, който се използва при съхранението на данните на сървъра и на клиента, както и при комуникацията между тях. Тъй като не всички бази от данни използвани в приложението са релационни, моделът на данните не е релационен.

Фиг. 2.4 Модел на данни

Building	
PK	<u>ID</u>
	rev floorNumber administrator name description

WifiAp	
PK	<u>ID</u>
	BSSID POWER X Y buildingID

Shop	
PK	<u>ID</u>
	shopName shopDescr shopFloor X Y picture buildingID

Map	
PK	<u>ID</u>
	SVG FLOOR HEIGHT WIDTH buildingID

## ТРЕТА ГЛАВА

# ПРОГРАМНА РЕАЛИЗАЦИЯ НА ПОЗИЦИОНИРАЩА СИСТЕМА ИЗПОЛЗВАЩА WI-FI

### *3.1 Android API за работа с модула за безжична връзка*

Работата с wifi модула в мобилната операционна система Android се извършва посредством клас наречен `WifiManager` намиращ се в пакета `android.net.wifi`. Инстанция от този клас се взима чрез извикването на метода `Context.getSystemService(Context.WIFI_SERVICE)`, който от своя страна връща инстанция на `WifiManager`. Класът `WifiManager` предоставя редица методи като някои от тях са :

```
boolean setWifiEnabled(boolean enabled)
boolean startScan()
List<ScanResults> getScanResults()
int addNetwork(WifiConfiguration config)
```

Класът `ScanResults` съдържащ се в списъка върнат от метода `getScanResults()`, съдържа множество полета, като две от тях са `BSSID`(уникален идентифициращ адрес) и `RSSI`(силата на получени сигнал).

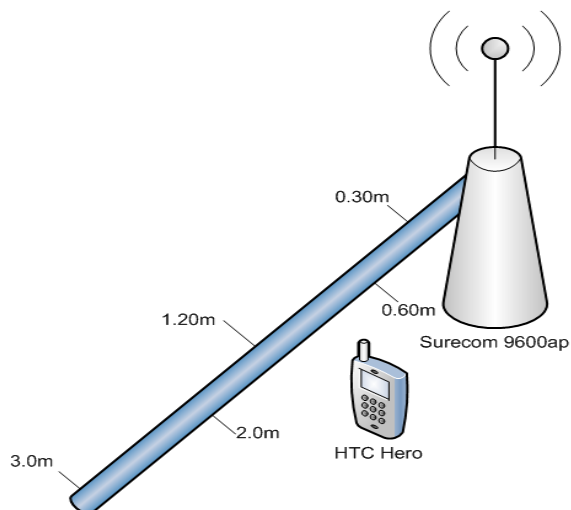
### *3.2 Експерименти*

#### **3.2.1 Експериментална постановка**

Експерименталната постановка е показана на фиг. 3.1



**Фиг. 3.1** Експериментална постановка



Използваните устройства изобразени на фиг. 3.1 са Surecom 9600AP(firmware DLINK 2100AP) като безжична точка за достъп и мобилен телефон HTC hero (OS: Android 2.1 Custom firmware:heroine 1.0).

**Цел на експеримента:** Да се установи как се разпространява безжичният сигнал с честота 2.4ghz в въздушна среда и до колко ще бъде възможно използването на rssi(received signal strength) получен от андроид апи за измерването на разстоянието между устройствата.

### 3.2.2 Провеждане на експеримента и събиране на данни

Експериментът беше извършен при следните конфигурации на безжичната точка за достъп:

1. Стандартни настройки и поставена 7dbm кръгова антена.
2. Стандартни настройки и отсъствие на антена (само медния конектор на устройството).

Резултатът от първият експеримент (съпоставка разстоянието в метри -> силата на сигнала) е изобразен на фиг. 3.2

**Фиг 3.2** Таблица с резултати 1

Разстояние	Сила на сигнала
0.30m	-36dbm
0.60m	-42dbm
1.20m	-46dbm
2.0m	-47dbm
3.0m	-50dbm

Резултатът от вторият експеримент (съпоставка разстоянието в метри -> силата на сигнала) е изобразен на фиг. 3.2

**Фиг. 3.3** Таблица с резултати 2

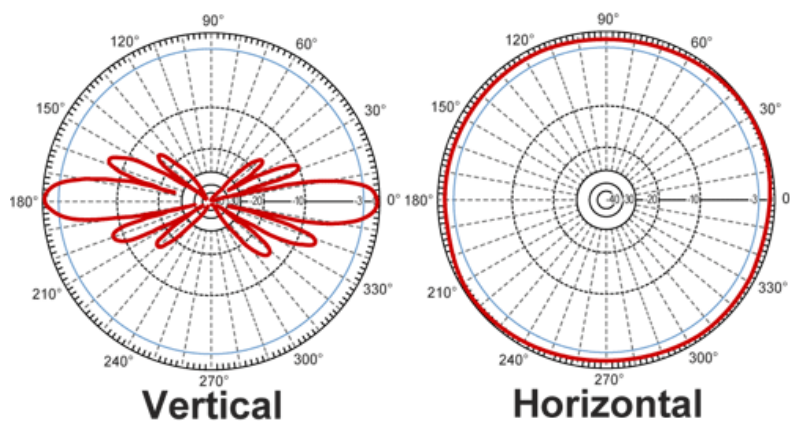
Разстояние	Сила на сигнала
0.30m	-40dbm
0.60m	-44dbm
1.20m	-50dbm
2.0m	-55dbm
3.0m	-62dbm

### **3.2.3 Анализ на събраните по време на експеримента данни.**

#### **Установени проблеми и решения**

И при двата експеримента сканирането ставаше през интервал 200ms, при което се наблюдаваше отсъствието на безжичната точка за достъп от списъка с резултати при някои от сканиранията. Проблемът беше високата стойност по подразбиране на beacon interval[3]. Това е интервалът, през който всеки access point изпраща така нареченият beacon frame, с който устройството известява останалите устройства за присъствието си. След намаляване на стойността му до минималната възможна, проблемът изчезна. Сред обстоен преглед на резултати от първата постановка установихме, че от 0.30m до 1.20m силата на сигнала намалява относително пропорционално на увеличаването на разстоянието, но в диапазона от 1.20 до 4,0m тя остава почти непроменена. Друг установен проблем при използването на кръговата антена е, че силата на сигнала силно се влияе от положението на телефона по вертикала спрямо безжичната точка за достъп. Този проблем се корени в спектралните характеристики на кръговата антена изобразени на фиг.3.

**Фиг. 3.3** Спектрална характеристика на кръгова антена



Поради отсъствието на антена по време на втората постановка, неточностите породени от нелинейните ѝ характеристики изчезнаха. Силата на сигнала намаляваше относително пропорционално на разстоянието в целия диапазон от 0.30м до 8.0м. При разстояние от 8.0м силата на сигнала е около -80dbm, стойности по-малки от тази се считат за невалидни

### ***3.4 Запазване на резултатите от сканиранията***

За всяка една безжична точка за достъп използвана от системата за изчисляване на позиция има създаден обект, който съдържа: местоположението, BSSID адреса, коефициент за мощността на радио предавателя и опашка с стойностите за нивото на сигнала. Нуждата от опашка за нивата на сигнала е обяснена в 3.4.1

```
public class WifiAP {  
    .....  
    private final String BSSID;  
    private final double POWER; // Used to identify different  
    wireless radios  
    private Queue<Integer> rssi;  
    /*  
     * The coordinates of the ap  
     */  
    private final double x;  
    private final double y;  
    .....  
}
```

Препратка към всеки един WifiAp обект се държи в клас наречен WifiApMap. Всеки WifiAp се идентифицира с уникалния BSSID адрес на безжичната точка за достъп.

```
public class WifiApMap {  
    private LinkedHashMap<String, WifiAP> map;  
    .....  
}
```

Обновяването на нивото на сигнала на AP съдържащи се в WifiApMap става чрез подаване на списъка с резултати върнат от WifiManager.getScanResults();

```
public void refresh(List<ScanResult> update){  
    for (ScanResult scanResult : update) {  
        WifiAP current = map.get(scanResult.BSSID);  
        if(current!=null){  
            current.updateRssi(scanResult.level);  
        }  
    }  
}
```

### ***3.4 Проверка и извличане на валидни данни от сканиранията на ефира***

При провеждането на експериментите установихме, че понякога се получават скокове в нивото на сигнала, които повлияват зле на определянето на позицията. Причините бяха много и разнообразни като например: рязка промяна в положението на телефона, държането му с две ръце и редица други физически фактори.

#### **3.4.1 Взимане на средната стойност от няколко нива на сигнала**

За да се предотвратят резки скокове в стойностите , реалната сила на сигнала се взима след 5 последователни измервания, които се съхраняват в опашка и при всяко ново постъпило обновление на силата на сигнала се изважда първата въведена в опашката с нива на сигнала стойност и се поставя новата. Горното е валидно при положение, че в опашката има записани минималния брой стойности.

```
private void actualRssiUpdate(int level) {
```

```

        if (rssi.size() >= ScanNumber) {
            rssi.poll();
        }
        rssi.offer(level);
    }

```

### 3.4.2 Откриване на невалидни сигнали и маркирането им като такива

Описаното в 3.3.1 частично реши проблема, но тъй като сканирането се осъществява през 200ms отново имаше скокове.

За да се постигне още по-плавна промяна в нивото на сигнала се въведе клас наречен InvalidSignalAnalyzer. Неговата идея е, при постъпване на сигнал с ниво, което е по-голямо или по-малко с 20% от текущото средно, този сигнал не бива въведен веднага в опашката от която се взима средната стойност.

```

    int highLimit = getAverageRssi() * (1 + PERCENTAGE / 100);
    int lowLimit = getAverageRssi() * (1 - PERCENTAGE / 100);
    if (level > highLimit || level < lowLimit) {
        analyzer.addInvalidValue(level);
        analyzer.tryUpdate(rssi);
    } else {
        actualRssiUpdate(level);
    }

```

Засеченият като невалиден сигнал се предава на анализатора, който го маркира с time stamp.

```

protected void addInvalidValue(int rssi){
    validate();
    holder.add(new
        InvalidSignalHolder(System.currentTimeMillis(),
            rssi));
}

```

При всеки ъпдейт на нивото на сигнала на даден access point се извиква метода tryUpdate на InvalidSignalAnalyzer и ако в анализатора има повече от една стойност маркирана като невалидна и последната такава е добавена в списъка преди по-малко

от 1.5s то това означава, че тези стойности са валидни и те се добавят в списъка с валидни стойности.

```
/*
 * validate() Validates the values stored in the holder by
 * checking the time stamp of the last stored value. If the
 * interval after which * the values are considered invalid has
 * expired the analyzer is being wiped.
 */
protected boolean validate() {
    if (holder.size() <= 0)
        return false;
    InvalidSignalHolder last = holder.get(holder.size() - 1);
    if ((last.getTimestamp() + interval) <
        System.currentTimeMillis()) {
        holder = new ArrayList<InvalidSignalHolder>();
        return false;
    }
    return true;
}
/*
 * This Method tries to transfer the invalid values to the list
 * of valid ones. This method will accomplish the transfer if
 * more than one value was considered invalid.
 */
public void tryUpdate(Queue<Integer> rssi) {
    if (holder.size() >= 2) {
        for (int i = 0; i < holder.size(); i++) {
            rssi.poll();
        }
        for (int i = 0; i < holder.size(); i++) {
            rssi.offer((holder.get(i).getRssi()));
        }
        holder.clear();
    }
}
```

### ***3.5 Изчисляване на разстоянието между безжичната точка за достъп и мобилния телефон***

След проведените експерименти (3.2) се достигна до извод с помощта, с помощта, на който се превръща нивото на сигнала в разстояние, при 0.30m между ар и мобилния телефон силата на сигнала е -32dbm. При удвояване на това разстояние сигнала намалява с около 6dbm. Използвайки тази зависимост стигнахме до формулата показана на фиг. 3.4

Фиг. 3.4 Формула за преобразуване от сила на сигнала към разстояние  
*(средната стойност на силата на сигнала - 32)/6*

Формулата от фиг. 3.4 е валидна за мощност на радио предавателя на безжичната точка за достъп от 60mW.

За нестандартни мощности на радио предавателя е въведена променлива, която в стандартния случай е равна на 1, но трябва да се промени ако се използват нестандартни предаватели.

```
public double getDistance() {  
    if (isValid())  
        return (((-getAverageRssi()) - 30) / (6.0) * POWER);  
    return -1;  
}
```

Функцията isValid() проверява дали последното обновление на силата на сигнала на безжичната точка за достъп е било преди по-малко от 4s и дали в списъка с нива на сигнала има достатъчно стойности.

```
public boolean isValid(){
    if (rssi.size() < ScanNumber) {
        return false;
    }
    boolean valid =
    (lastUpdate+validInterval)>System.currentTimeMillis();
    if(!valid){
        analyzer.validate();
        rssi.clear();
    }
    return valid;
}
```

### ***3.6 Изчисляване на позицията на телефона***

#### **3.6.1 Взимане на нужната информация**

Както е описано в 2.4 за еднозначното изчисляване на позицията на телефона са нужни координатите на три безжични точки за достъп и разстоянието между тях и телефона. Тези три точки се вземат на случаен избор, но трябва да са валидни по смисъла на метода WifiAp.isValid().



```

public void getPosition(Point point){
    List<WifiAP> values = new ArrayList<WifiAP>(map.values());
    Collections.shuffle(values);
    List<Point> temp = new ArrayList<Point>();
    for (WifiAP wifiAP : values) {
        Point p = wifiAP.getDistanceAsPoint();
        if(p.getDistance()!=-1)
            temp.add(p);
    }
    if(temp.size()<3){
        point.setX(-1);
        point.setY(-1);
        return;
    }
    Point p1 = temp.get(0);
    Point p2 = temp.get(1);
    Point p3 = temp.get(2);
    try {

        Point returned = calc.evaluate(p1,p2,p3);
        point.setX(returned.getX());
        point.setY(returned.getY());
        return;

    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    point.setX(-1);
    point.setY(-1);
    return;
}

```

### 3.6.2 Извършване на пресмятането

За изчисляването на позицията се използва WolframAlpha Api [4] като данните се пращат до сървърите на WolframAlpha, те пресмятат резултата и го връщат обратно. За да се използват услугите предоставяни от WolframAlpha Api е нужно да се регистрира приложение на страницата на WolframAlpha. След като се регистрирате ви се предоставя уникален за вашето приложение ключ с помощта, на който ви се разрешава достъпът до услугите.

Данните се предоставят на сървъра в следния формат:

```
public Point evaluate(Point p1, Point p2, Point control){  
    double d1 = p1.getDistance();  
    double x1 = p1.getX();  
    double y1 = p1.getY();  
  
    double d2 = p2.getDistance();  
    double x2 = p2.getX();  
    double y2 = p2.getY();  
  
    String input = "solve " + d1 * d1 + "=( " + x1 + "-x)^2 +  
    ( " + y1+ "-y)^2, " + d2 * d2 + "=( " + x2 + "-x)^2 + ( " + y2 + "-  
    y)^2";
```

WolframAlpha връща низ, който може да представен по няколко начина:

1.  $x \sim \sim$

2.  $x =$

3  $x = a/b$

Този низ трябва да бъде парснат до double.

```
if (res.contains("~~")) {
    startX = res.indexOf("~~") + 3;
    endX = res.indexOf("and");
    startY = res.indexOf("~~", endX) + 3;
    endY = res.length();
} else {
    startX = res.indexOf("x = ") + 4;
    endX = res.indexOf("and");
    startY = res.indexOf("y = ") + 4;
    endY = res.length();
}
x = parseFraction(
    res.substring(startX, endX),
    -1, true);
y = parseFraction(
    res.substring(startY, endY),
    -1, true);
pointContainer.add(new Point(x, y));
```

Проверката с контролната точка става чрез обхождане на списъка pointContainer, който съдържат върнатите като резултат точките.

```
Point p = null;
double delta = 0;
for (Point point : pointContainer) {
    if (p == null) {
        p = point;
        delta = Math.abs(control.getDistance()
            - p.estimateDistance(control));
    } else {
        double tempDelta =
            Math.abs(control.getDistance()
                -
                point.estimateDistance(control));
```

```

        if (tempDelta < delta) {
            p = point;
        }

    }

}

```

### ***3.7 Имплементация на алгоритъма за определяне на сградата и вземане на данните за нея (фиг.2.4)***

Всички операции свързани с определяне на сградата, в която се намираме, с четене и записване в базата на телефона, с вземане на данни от сървъра се извършват в отделна нишка, чрез използване на класа : android.os.AsyncTask. Това позволява по време на тези операции да се изведе на потребителя съобщение за изчакване и нишката на потребителския интерфейс да остане свободна.

#### **3.7.1 Свързване към безжичната точка за достъп**

В базата данни на телефона за всяка една точка за достъп има записан BSSID адрес, който е уникален. Освен това има и идентификационен номер на сградата, в която се намира. Това позволява след свързване към него, лесно да се определи сградата.

При стартиране на приложението автоматично се добавя в списъка с безжични точки на достъп на телефона, запис с специален SSID адрес и парола и се разрешава свързване само към тази мрежа. При спиране на приложението записът за безжичната мрежа се изтрива от списъка на телефона, чрез което се елиминира възможността за свързване към тази мрежа, когато приложението не работи.

```

public boolean connect() {
    addNetwork();
    return manager.enableNetwork(networkId, true);
}

```

```

public boolean disconnect() {
    manager.disconnect();
    manager.removeNetwork(networkId);
    return manager.disableNetwork(networkId);
}

```

### 3.7.2 Определяне на BSSID адреса на безжичната точка за достъп

Първата стъпка при определянето BSSID адреса на безжичната точка за достъп е изчакване на мобилния телефон да се свърже с нея.

След като това премине успешно се извиква метода на `android.net.wifi.WifiManager` - `getConnectionInfo()`; Този метод връща като резултат `android.net.wifi.WifiInfo`, в които са записани данните за BSSID адреса на безжичната точка за достъп към която сме свързани в момента.

### 3.7.3 Проверка за наличие на информация за сградата във вътрешната база данни

#### 3.7.3.1 Работа със вътрешната база данни

Създадени са няколко класа в пакета: `org.elsys.wps.database`. Класовете са: `ApListDataSource`, `BuildingDataSource`, `ShopListDataSource`, `MapDataSource`. Те предоставят методи, с помощта, на които може лесно да се вземе или добави информация от и във вътрешната SQLite база данни, която притежава всяко едно Android приложение.

Класът `ApListDataSource` предоставя метод за вземане на информация за всички безжични точки за достъп от една сграда, чрез идентификационен номер на сградата или чрез BSSID адрес на някоя от безжичните точки за достъп в сградата. Предоставя се и метод, чрез който лесно да се добавят нови безжични точки за достъп.

Класът `BuildingDataSource` предоставя метод за извличане на данните за сграда по нейния идентификационен номер и метод за добавяне на нова сграда.

Класовете `ShopListDataSource` и `MapDataSource` предоставят метод за взимане на информация за магазини/картите чрез идентификационния номер на сградата, в която се намират и метод за добавяне на нови магазини/карти в базата данни.

### **3.7.3.2 Определяне на сградата в която се намира устройството**

Извиква се метода `getBuildingIdFromBssid` на класа `ApListDataSource`, на който се подава BSSID адреса като параметър. Този метод връща идентификационния номер на сградата, в която се намира безжичната точка за достъп към която сме свързани в момента ако той върне стойност различна от -1, това означава, че данните за сградата са налични. С помощта на този идентификационен номер се взима номера на версията на данните за тази сграда.

### **3.7.4 Изпращане на заявка към сървъра**

Какъвто и да е резултата от 3.7.3.2 приложението прави заявка към сървъра.

Ако не са намерени данни за дадената сграда в заявката, която се праща към сървъра на параметъра отговарящ за версията на данните се записва -1. Това кара сървъра директно да изпрати данните на мобилното приложение.

В противен случай се прави сравнение и ако на сървъра има по-нова версия на данните за сградата те се пращат към мобилното приложение.

### **3.7.5 Подготовка за зареждане на данните за сградата в приложението**

Ако данните за сградата не са били налични във вътрешната база данни на приложението или пък версията на данните върху сървъра е била по-нова получената от сървъра информация се записва върху телефона.

## ***3.8 Комуникация Телефон<->Сървър***

Комуникацията между телефона и сървъра се извършва чрез HTTP заявки и отговори, като в заявката от страната на телефона се изпраща към сървъра параметри с помощта на query string от следния вид :

`http://wpsgwt.appspot.com/update?bssid=00:16:01:af:e5:73&rev=-1`

Ако в базата данни на телефона няма данни за дадената сграда, на параметъра „rev” се задава стойност -1, което кара сървъра да изпрати данните.

### 3.8.1 Кодиране на данните

За да бъдат прехвърлени данните от сървъра към мобилния телефон, те трябва да бъдат трансформирани, в подходящ формат за предаване през HTTP протокола. За целта се използват JSON обекти. С помощта на тях всяко едно поле от класа се кодира в формат ключ:стойност. Това конвертиране се извършва автоматично посредством външна библиотека наречена Jackson JSON Processor. Голямо предимство на тази библиотека е, че няма никакви изисквания към други външни библиотеки, нуждае се само от JRE. Тази библиотека предоставя клас `org.codehaus.jackson.map.ObjectMapper`. В този клас има методи с помощта, на които един обект може да се преобразува в JSON обект и обратно.

```
mapper.writeValue(destination, value);  
mapper.readValue(source, valueType.class);
```

Source и destination, съответно източник и местоназначение, могат да бъдат от тип String, поток, файл, масив от байтове и т.н.

## 3.9 Сървърно приложение

Сървърното приложение работи на сървъра на Google : Appengine, и използва Java сървлет за обработка на заявките и изпращане на данните.

### 3.9.1 Обработка на заявката и изпращане на отговор

Извличането на параметрите от заявката и изпращането на отговора стават в класа `org.elsys.wps.server.AndroidCommunication.RequestHandler`, който наследява `HttpServlet`.

Тъй като мобилното приложение изпраща `HttpGet` заявка към сървърното приложение, обработката на тази заявка става в метода `doGet` на класа `RequestHandler`.

Първата стъпка е извличане на параметрите подадени с заявката.

```
String bssid = request.getParameter("bssid");  
String sRev = request.getParameter("rev");
```

Ако някои от параметрите отсъства или е невалиден се изпраща HTTP Bad Request

```
response.sendError(response.SC_BAD_REQUEST,  
                    "Illegal Parameters");
```

Ако подадените параметри са валидни се сравнява версията на данните подадена с заявката.

Ако тя е същата като тази намираща се на сървъра се изпраща съобщение, че данните са актуални

```
response.getOutputStream().println("Result is up to date!");
```

Ако намиращата се на сървъра версия е по-нова, данните се пакетират в клас наречен BuildingTransferable, който съдържа всичката информация за една сграда и се изпращат като отговор на мобилното приложение.

### 3.9.2 Съхранение на данните за сградите на сървърното приложение

Създаден е клас org.elsys.wps.server.AndroidCommunication.UpdateService, който предоставя методи за извличане на данни от базата по различни критерии нужни за определяне на точната информация, която трябва да бъде изпратена.

Съществува метод за взимане на идентификационния номер на сграда чрез BSSID на някоя от безжичните точки за достъп намиращи се в нея, метод за взимане на данните за цялата сграда от BSSID, метод за взимане на всички безжични точки за достъп в сграда, метод за взимане на данните за сграда по идентификационния и номер или собственик.

```
public int getBuildingID(String bssid);  
public BuildingEntity getBuildingByBssid(String bssid);  
public List<WifiApEntity> getWifiApsByBssid(String bssid);  
public List<WifiApEntity> getWifiApsByBuildingId(int id);  
public List<BuildingEntity> getBuildingsByOwner(String owner);
```



## Четвърта глава. Ръководство за потребителя

### 4.1 Сваляне и инсталация

След стартиране на приложението Google Play, което е вградено в мобилна операционна система Android, избираме търсене и въвеждаме WifiPosition.

Избираме приложението, което се казва WifiPosition, иконата му се вижда в горния десен ъгъл на фиг. 4.0

След като изберем иконата на приложението се появява екрана показан на фиг. 4.0



Фиг.4.0

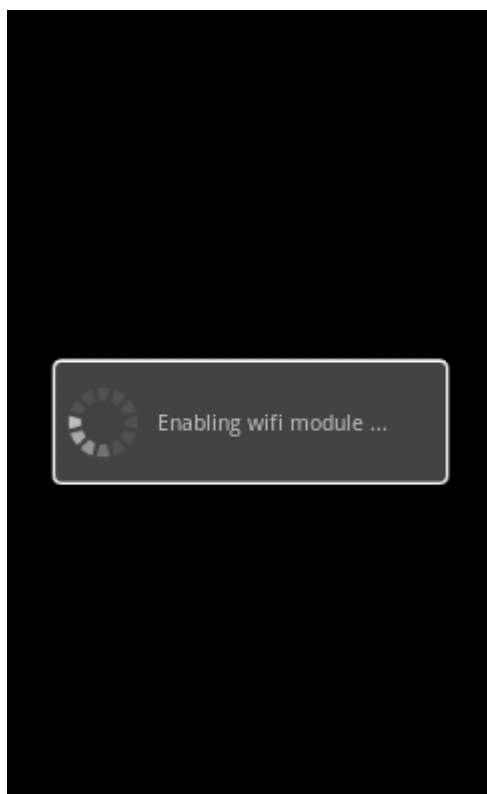
От този екран избираме "Download", след това избираме, че сме съгласни да дадем правата нужни на приложението

## 4.2 Подготвителни операции

При стартиране на приложението трябва да се извършат няколко подготвителни операции преди да е възможна работата с позициониращата система.

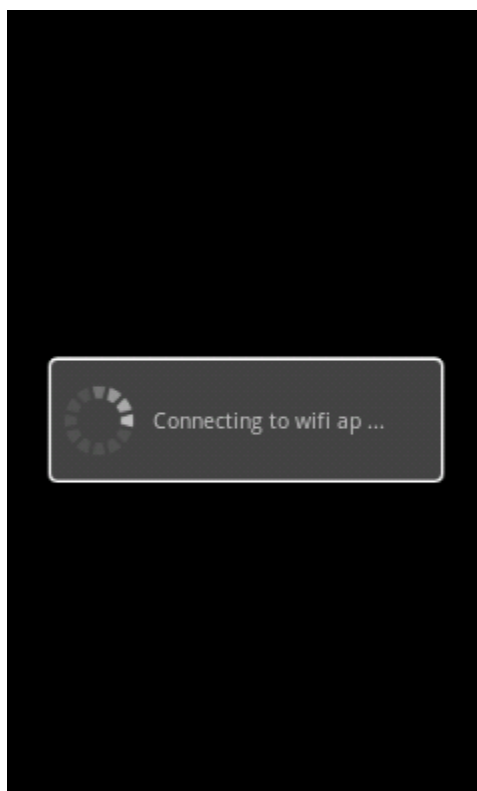
Първата операция е разрешаване на модула за безжична комуникация. При извършването на тази операция се показва екран изобразен на фиг. 4.1

**Фиг. 4.1** Разрешаване на модула за безжична връзка

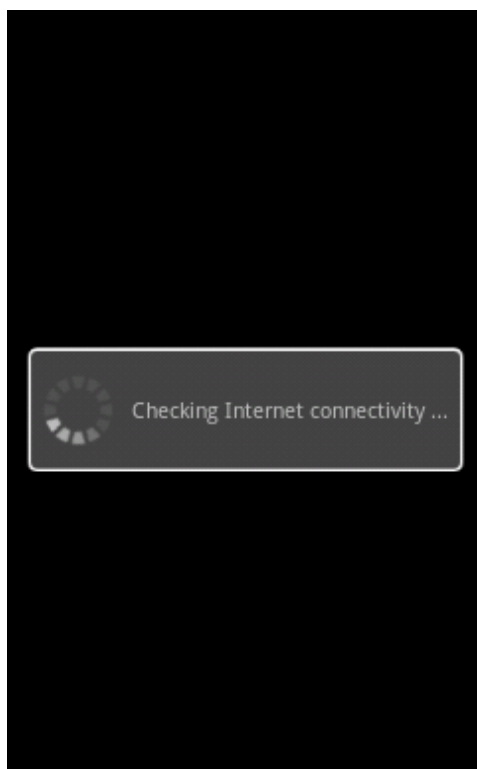


Втората операция която трябва да се извърши е свързване към безжичната точка за достъп за да се определи сградата, в която се намира устройството и за да се свали нужната информация. Екрана, който ще се появи е изобразен на фиг. 4.2

**Фиг. 4.2** Свързване към безжичната точка за достъп



Следващата операция е проверка на свързаността към интернет. Това е нужно за могат успешно да се свалят данните за сградата от сървъра. Екрана, които ще се покаже е изобразен на фиг. 4.3

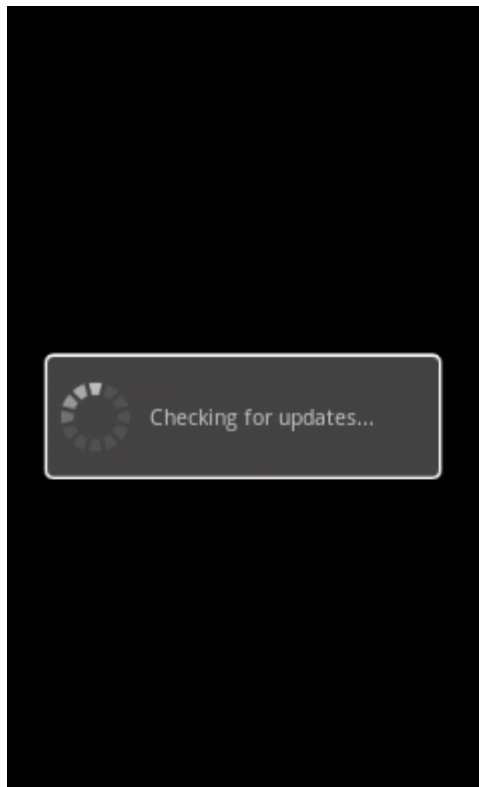


**Фиг. 4.3** Проверка на свързаността към интерне

Ако данните за сградата са налични във вътрешната база данни на телефона, ще се изпълни операция целяща проверка за обновления на информацията за сградата.

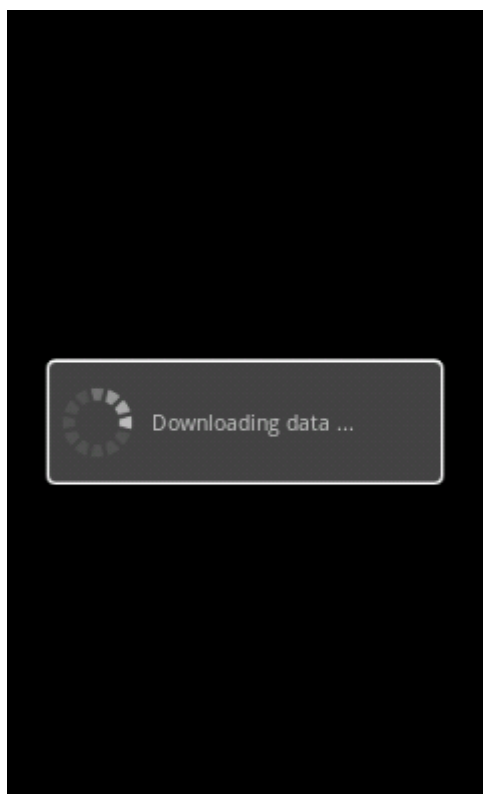
Екранът, който ще се появи е изобразен на фигура 4.4

**Фиг. 4.4** Проверка за обновления



Ако във вътрешната база данни на телефона не са налични данни за конкретната сграда, в която в момента се намира устройството, ще се изпрати заявка към сървъра за изпращане на тези данни. По време на изчакването за получаване на данните ще се появи екран изобразен на фиг. 4.5

**Фиг. 4.5** Сваляне на данни за сградата



#### **4.2.1 Възможни проблеми по време на изпълнение на операциите от 4.1**

Всяка една операция трябва да отнеме не повече от няколко секунди. Ако някоя от операциите не завърши след повече от 30 секунди, това означава, че има проблем с телефона, със сървърите или с изградената инфраструктура в сградата.

##### **4.2.1.1 Проблеми, причини, решения**

Ако приложението се задържи повече от 30 секунди на операцията за разрешаване на модула за безжична връзка(фиг. 4.1) , вероятната причина е или друго приложение забраняващо непрестанно модула за безжична връзка или хардуерен проблем с устройството.

**Решение:** Опитайте да рестартирате телефона и след това да стартирате приложението.

Ако приложението се задържи повече от 30 секунди на операцията за свързване към безжичната точка за достъп(фиг. 4.2), вероятна причина е, че се намирате в сграда, в която няма изградена инфраструктура за използване на *Wifi Position*. Ако обаче сте сигурни, че се намирате в сграда, в която има изградена инфраструктура за нашата

система, то причината за неуспешното преминаване на стъпката е проблем с инфраструктурата. В такъв случай моля свържете се с нас и докладвайте за проблема за да можем бързо да го отстраним.

Ако приложението се задържи повече от 30 секунди на операцията за проверка на свързаността към интернет (фиг. 4.3), то вероятно имате зададен ръчно IP адрес, моля уверете се, че в настройките на модула за безжична връзка отметката върху „Use static IP” или „Използвай статичен IP адрес” е премахната. Ако отметката върху посоченото поле е премахната, но все още не успявате да преминете тази стъпка, моля свържете се с нас, тъй като вероятно има проблем с нашата инфраструктура.

Ако приложението се задържи на стъпка „Проверка за обновления” (фиг 4.4) или „Сваляне на данни” (фиг. 4.5), това означава, че има проблем с нашите сървъри. Моля докладвайте за проблема за да спомогнете за бързото му разрешаване.

### ***4.3 Главно меню на приложението***

Ако всички подготвителни операции са преминали успешно, на вашия дисплей ще се появи първоначалния екран на приложението, от който може да започне работата(фиг. 4.6). На екрана има 4 бутона



Фиг. 4.6

#### 4.3.1 Бутон „Show map”

Първият бутон е “Show Map”. Той отваря екрана с картите на сградата(фиг. 4.7).  
(снимка без ръка)

Този екран дава възможност на потребителите да разглеждат картите на етажите на сградата, в която се намират в момента. В горната част на екрана се намира изгледа на картата на текущия етаж. Върху картата на текущия етаж се намират логата на различните магазини, намиращи се на същия етаж, позиционирани според реалното им местоположение. Потребителите също виждат на картата текущото си местоположение, представено като червено-черна точка.Потребителите имат възможност да се местят по картата, чрез движение с един пръст по екрана.



Фиг.4.7

Потребителите могат да мащабират текущата карта с два пръста, чрез така наречения „pinch zoom”(фиг.4.8). За да се случи това е нужно екрана да бъде докоснат с два пръста и съответно те да се движат един към друг за намаляване на картата и да се раздалечат за увеличаване на картата.





Фиг. 4.8

Под картата на текущия етаж потребителите разполагат с компоненти за контрол над картата. За смяна на текущия етаж потребителя може да избере бутона „Change floors”, както и да натисне бутона „Menu” на устройството си. Това ще покаже менюто с избор на етажите на сградата. При избиране на етаж, различен от текущия, картата автоматично се подменя с тази, избрана от потребителя, и на нея се изчертават логотата на обектите, намиращи се на този етаж.

Чрез бутоните „-” и „+” в долния десен ъгъл на екрана потребителите имат възможност да мащабират картата. Мащабирането, чрез бутоните, се осъществява в центъра на текущата видима част на картата.

#### 4.3.2 Бутон „Shop list”

Вторият бутон в менюто е „Shop list”. При избирането му се отваря екран (фиг. 4.9), даващ възможност на потребителите да разгледат всички обекти в сградата, както и да получат повече информация относно тях.



Shop: Coca cola Floor: 1  
yummy



Фиг.4.9

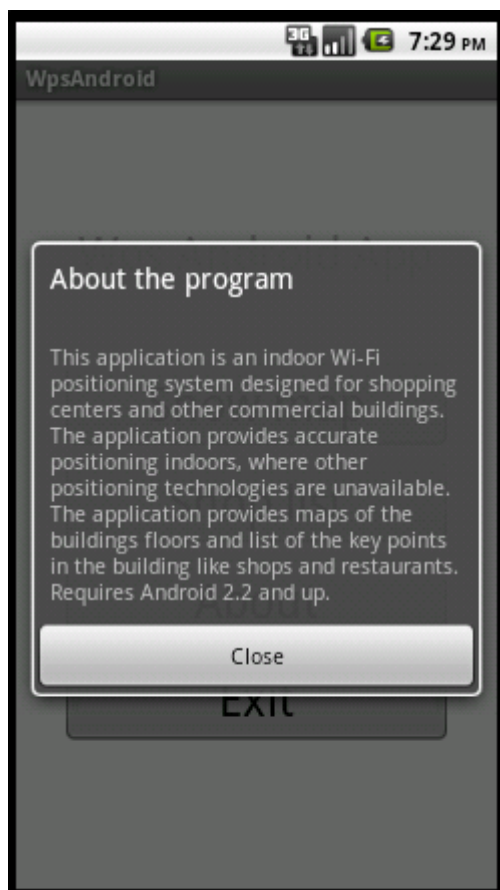
В горната част на екрана се намира логото на текущо избрания магазин. Под него се намират съответно името, етаж и описанието на магазина. В най-долната част на екрана се намира компонент с логата на всички магазини. При избиране на някое от логата магазинът, чието лого е избрано, става текущ магазин. Логата в този компонент могат да бъдат превъртани с движение на един пръст върху тях(фиг. 4.10).



фиг. 4.10

### 4.3.3 Бутон „About”

Третият бутон в менюто е „About”. При избирането му се отваря екран с информация относно приложението(фиг. 4.11).



фиг.4.11

### 4.3.4 Бутон „Exit”

Четвъртият бутон в менюто е бутон „Exit”. Чрез него потребителите могат да затворят приложението.

## **Заключение**

### ***5.1 Постижения в дипломната работа***

В дипломната работа е постигнато позициониране на закрито, посредством безжични точки за достъп, разположени в сградата. Разработено е мобилно приложение, което може да визуализира карти на етажите на сградата и да показва текущото местоположение на мобилното устройство върху нея. Разработен е сървър, от който мобилното приложение да приема информация. Разработено е администраторско web-приложение, чрез което удобно да бъдат редактирани данните за сградите от администратор

### ***5.2 Виждане за усъвършенстване на разработката***

За бъдещо усъвършенстване на разработката е възможно да се подобри потребителския интерфейс, като се направи по-красив. Да има възможност за автоматично идентифициране на етажа, на който се намира устройството.

## **Използвана литература**

1. Comparison of Wireless Indoor Positioning Technologies,  
[http://www.productivet.com/docs-2/Wireless\\_Comparison.pdf](http://www.productivet.com/docs-2/Wireless_Comparison.pdf)
2. "Distance Formula", [http://en.wikipedia.org/wiki/Distance\\_formula#Geometry](http://en.wikipedia.org/wiki/Distance_formula#Geometry)
3. Информация за beacon interval, [http://en.wikipedia.org/wiki/Beacon\\_frame](http://en.wikipedia.org/wiki/Beacon_frame)
4. WolframAlpha Api, <http://products.wolframalpha.com/developers/>

# Съдържание

УВОД .....	1
ПЪРВА ГЛАВА.....	5
МЕТОДИ И ТЕХНОЛОГИИ ЗА РЕАЛИЗИРАНЕ НА ПОЗИЦИОНИРАЩА СИСТЕМА В ЗАТВОРЕНИ ПРОСТРАНСТВА И ПРЕГЛЕД НА СЪЩЕСТВУВАЩИ СИСТЕМИ .....	5
1.1 Основни принципи на работа на позициониращите системи.....	5
1.2 Видове позициониращи системи според технологията на предавателя и приемника.....	5
1.2.1 Използване на инфрачервени вълни.....	5
1.2.2 Използване на RFID четци и предаватели .....	6
1.2.3 Използване на безжична технология стандарт IEEE 802.1* .....	6
1.2.4 Начини за изчисляване на разстоянието между приемник и предавател ...	7
1.3 Съществуващи системи за позициониране в затворени пространства.....	8
1.3.1 Skyhook Wireless.....	8
1.3.2 Google Maps Indoors .....	8
1.3.3 Navizon I.T.S .....	8
ВТОРА ГЛАВА.....	10
ПРОЕКТИРАНЕ НА СТРУКТУРАТА НА ПОЗИЦИОНИРАЩА СИСТЕМА ЗА ЗАТВОРЕНИ ПРОСТРАНСТВА .....	10
2.1 Функционални изисквания към позиционираща система за затворени пространства .....	10
2.2 Избор на технология за позициониране и устройство, на което да се имплементира .....	10
2.3 Избор на мобилно устройство и среда за разработка.....	10
2.3.1 Избор на мобилно устройство .....	10
2.3.2 Избор на среда за разработка.....	11
2.4 Създаване на алгоритъм за изчисляване на позицията на мобилния телефон .....	11
2.5 Определяне на сградата, в която се намираме и вземане на последните данни за нея.....	13
2.6 Модел на данните използван за съхранение и пренасяне на информацията между клиента и сървъра .....	15
ТРЕТА ГЛАВА.....	16
ПРОГРАМНА РЕАЛИЗАЦИЯ НА ПОЗИЦИОНИРАЩА СИСТЕМА ИЗПОЛЗВАЩА WI-FI .....	16
3.1 Android API за работа с модула за безжична връзка.....	16
3.2 Експерименти.....	16
3.2.1 Експериментална постановка .....	16
3.2.2 Провеждане на експеримента и събиране на данни.....	17

3.2.3 Анализ на събраните по време на експеримента данни. Установени проблеми и решения.....	18
3.4 Запазване на резултатите от сканиранията .....	19
3.4 Проверка и извличане на валидни данни от сканиранията на ефира .....	20
3.4.1 Взимане на средната стойност от няколко нива на сигнала .....	20
3.4.2 Откриване на невалидни сигнали и маркирането им като такива .....	21
3.5 Изчисляване на разстоянието между безжичната точка за достъп и мобилния телефон.....	23
3.6 Изчисляване на позицията на телефона .....	24
3.6.1 Взимане на нужната информация .....	24
3.6.2 Извършване на пресмятането .....	26
3.7 Имплементация на алгоритъма за определяне на сградата и вземане на данните за нея (фиг.2.4) .....	28
3.7.1 Съвързване към безжичната точка за достъп .....	28
3.7.2 Определяне на BSSID адреса на безжичната точка за достъп .....	29
3.7.3 Проверка за наличие на информация за сградата във вътрешната база данни .....	29
3.7.4 Изпращане на заявка към сървър .....	30
3.7.5 Подготовка за зареждане на данните за сградата в приложението .....	30
3.8 Комуникация Телефон<->Сървър .....	30
3.8.1 Кодиране на данните .....	31
3.9 Сървърно приложение.....	31
3.9.1 Обработка на заявката и изпращане на отговор .....	31
3.9.2 Съхранение на данните за сградите на сървърното приложение.....	32
Четвърта глава. Ръководство за потребителя.....	33
4.1 Сваляне и инсталация .....	33
4.2 Подготвителни операции.....	34
4.2.1 Възможни проблеми по време на изпълнение на операциите от 4.1 .....	37
4.2.1.1 Проблеми, причини, решения .....	37
4.3 Главно меню на приложението .....	38
4.3.1 Бутон „Show map” .....	39
4.3.2 Бутон „Shop list” .....	41
4.3.3 Бутон „About” .....	43
4.3.4 Бутон „Exit” .....	43
Заклучение .....	44
5.1 Постижения в дипломната работа.....	44

5.2 Виждане за усъвършенстване на разработката .....	44
Използвана литература.....	45