**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

****

LUCRARE DE LICENȚĂ

**FiiBeacon**

**propusă de**

***Florin Veleșca***

**Sesiunea:** *Iulie, 2019*

**Coordonator științific**

Conf. Dr. Anca Vitcu

**UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

FiiBeacon

*Florin Veleșca*

**Sesiunea:** *Iulie, 2019*

**Coordonator științific**

*Conf. Dr. Anca Vitcu*

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Data \_\_\_\_\_\_\_\_\_\_\_\_ Semnătura \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a) ………………………………………………………………………………………

domiciliul în …………………………………………………………………………………………………..

născut(ă) la data de ………………..…., identificat prin CNP ………….……………..………………..., absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de ………………………. specializarea …………………………………………………………, promoția …………………………., declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licență cu titlul: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_elaborată sub îndrumarea dl. / d-na \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice in vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, ………………………… Semnătură student …………………………

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*FiiBeacon*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de testetc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *data*

Absolvent *Prenume Nume*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(semnătura în original)

Cuprins

[1 Introducere 9](#_Toc12805956)

[1.1 Tema proiectului 9](#_Toc12805957)

[1.2 Abordări 9](#_Toc12805958)

[1.3 Contribuții 10](#_Toc12805959)

[1.4 Capabilități de dezvoltare ulterioara 10](#_Toc12805960)

[2 Tehnologii 11](#_Toc12805961)

[2.1 Android (1) 11](#_Toc12805962)

[2.1.1 Versiunile compatibile cu aplicația 11](#_Toc12805963)

[2.2 Android Studio/SDK 12](#_Toc12805964)

[2.2.1 Uneltele de dezvoltare pentru platforma Android 12](#_Toc12805965)

[2.2.2 Mediul de dezvoltare (IDE) 12](#_Toc12805966)

[2.3 Bluetooth Low Energy 13](#_Toc12805967)

[2.4 Beacon 13](#_Toc12805968)

[2.4.1 Descriere 13](#_Toc12805969)

[2.4.2 Rolul in Aplicație 13](#_Toc12805970)

[2.4.3 Metoda de implementare in aplicație 14](#_Toc12805971)

[2.5 SQLite/Room Persistence Library 14](#_Toc12805972)

[2.5.1 Rolul in aplicație 14](#_Toc12805973)

[2.5.2 Implementare 14](#_Toc12805974)

[2.6 AsyncTask 15](#_Toc12805975)

[2.7 Rolul in aplicație 15](#_Toc12805976)

[2.8 Implementare 15](#_Toc12805977)

[3 Obiectivul 16](#_Toc12805978)

[3.1 Idei conceptuale 16](#_Toc12805979)

[3.2 Dinamic 16](#_Toc12805980)

[3.3 Simplitate 16](#_Toc12805981)

[4 Descrierea problemei 17](#_Toc12805982)

[4.1 Necesitate 17](#_Toc12805983)

[4.2 Alte implementări 17](#_Toc12805984)

[5 Descrierea sumara a soluției 19](#_Toc12805985)

[6 Arhitectura proiectului 20](#_Toc12805986)

[6.1 Structura aplicației 20](#_Toc12805987)

[6.1.1 Pachete ( Directoare ) 20](#_Toc12805988)

[6.1.2 Clase și interfețe 21](#_Toc12805989)

[6.1.3 Module și bibliotecile importate 22](#_Toc12805990)

[6.1 Implementarea soluției 22](#_Toc12805991)

[6.1.1 MainActivity – Ecranul principal al aplicației 23](#_Toc12805992)

[6.1.2 Fire de execuție asincrone 24](#_Toc12805993)

[6.1.3 Baza de date 24](#_Toc12805994)

[6.1.4 Interfețe de comunicare 25](#_Toc12805995)

[6.1.5 SearchBeaconFragment 26](#_Toc12805996)

[6.1.6 DrawActivity și MapView 26](#_Toc12805997)

[6.1.7 Algoritmul de drum minim 27](#_Toc12805998)

[7 Interacțiunea cu utilizatorul 28](#_Toc12805999)

[7.1 Ecranul Principal 28](#_Toc12806000)

[7.2 Ecranul de căutare 29](#_Toc12806001)

[7.3 Harta și afișarea drumului 29](#_Toc12806002)

[8 Testarea și rezultate 31](#_Toc12806003)

[8.1 Unit Testing 31](#_Toc12806004)

[8.2 Testarea manuala 31](#_Toc12806005)

[9 Viitoare îmbunătățiri 32](#_Toc12806006)

[10 Concluzie 33](#_Toc12806007)

[11 Bibliografie 34](#_Toc12806008)

Motivație

Datorită pasiunii mele pentru platformele mobile si studiului pe care l-am făcut pe parcursul facultății și în cadrul internship-ului în care am avut prilejul să acumulez cunoștințe legate de comunicarea cu dispozitive Bluetooth și modul prin care acestea pot comunica cu dispozitive de platforma android mi-am propus să lucrez, având la îndemână un dispozitiv Bluetooth, la o simpla comunicarea cu acesta observând datele si informațiile pe care acestea le aduc.

Folosind tehnologia *Bluetooth Low Energy* ce oferă atât dispozitivelor android cat si a celor ce emit semnale o metoda de comunicare cu cat mai putina energie am decis creez o aplicație care comunica cu acestea. Cu ajutorul dispozitivelor ce emit semnale BLE, numite Beacon, mi-am propus sa realizez o aplicație care sa ajute noii studenți, persoanele interesate si alți vizitatori sa se orienteze mai ușor in cadrul facultății de informatica.

Observând funcționalitatea *Beacon*-urilor in alte domenii, cum ar fi in cadrul aeroporturilor, marketing-ului si chiar in domeniul educației, ce mai interesanta aplicație din punctul meu de vedere fiind cea care identifica prezenta studenților la cursuri si le oferă acestora informații cu privire la cursul curent. Am decis sa mă folosesc de acestea pentru a crea o harta secționate pe etaje ce comunica in timp real poziția curenta a utilizatorului precum si un drum optim spre locația aleasa de utilizator.

Față de celelalte aplicații care pur si simplu oferă informații legate de obiectul atașat de dispozitiv, cum ar fi in cadrul medical pentru identificarea unica a unui pacient oferind medicului informații despre istoricul medica sau in domeniul asigurărilor preluarea de informații despre oferta si in multe alte domenii, aplicația mea oferă informații in timp real si comunicare mult mai strânsă cu aceasta.

Motivul pentru care am folosit ca si platforma de lucru dispozitivele mobile, lăsând deoparte pasiunea mea in aceasta direcție, a fost nevoia de a comunică dinamic cu dispozitivele *Bluetooth* dar si prin popularitatea acestora astfel aplicația se adresează unui număr de utilizatori mai mare. Utilizarea prin intermediul dispozitivelor mobile este mult mai intuitiva si mai la îndemână decât serviciile web.

# Introducere

Tehnologia BLE (Bluetooth low energy) este pretutindeni, in telefoane, brățări fitness, smartwatch-uri, dispozitive medicale . Având un impact mic asupra consumului de energie, aceasta tehnologie a fost aplicata in multe domenii, fie ca vorbim de cel medical , educațional sau comercial această tehnologie s-a dovedit extrem de folositoare îmbunătățindu-ne viața si modul in care interacționam mediul din jur.

## Tema proiectului

Proiectul are ca scop gestionarea problemei găsirii punctelor de interes din cadrul facultății. Pentru studenții noi veniți sau cei din străinătate informațiile cu privire la localizarea anumitor săli de curs erau foarte incerte, lucru ce pot spune ca l-am observat din propria experiență odată cu venirea mea la primele cursuri nereușind sa găsesc in timp util anumite săli si prin interacționarea cu studenți străini cărora a trebuit sa le ofer indicații.

## Abordări

Abordări similare au fost introduse in domeniul medical unde medicii își pot găsi pacienții si sa primească informații in mod instant cu privire la starea actuala a acestuia, in domeniul transportului, calcularea prețului rutei parcurse, a timpului si multe alte informații. Anumite aeroporturi oferind posibilitatea de a transmite informații despre locația ta curenta sau a bagajului de cală. Sunt numeroase abordări ale acestei tehnologii, limitele fiind foarte largi.

Așadar am decis sa încerc o abordare pentru facultate, ideea este ca fiecare sala , fie ca vorbim de laboratoare , săli de curs, birouri ale profesorilor sau a personalului toate vor conține la intrare cate un *beacon* ce va ajuta la identificarea unica a acestora*.*

Relația dintre dispozitiv si sala in sine va fi trecuta într-o baza de date, totodată acest lucru va avea si un impact financiar benefic pentru o ulterioara implementare deoarece se vor putea achiziționa dispozitive simple ieftine ce transmit doar informații despre identificarea sa unica si distanta relativa pana la acesta, nefiind nevoie de dispozitive scumpe ce au nevoie de o memorie mai larga si un acumulator mai puternic pentru a susține aceasta abilitate.

Baza de date va fi locala, pe fiecare dispozitiv si generata in momentul instalării aplicației, motivul pentru care am ales aceasta abordare a fost sfatul primit de la un profesor al universității cu privire la numărul relativ mic al dispozitivelor, comunicarea cu un server sau *cloud* fiind costisitoare si din punct de vedere financiar dar si din punct de vedere al eficientei, după cum știm comunicarea cu un server presupunând conexiune constanta la internet si riscul de erori crescut.

## Contribuții

Cel mai important lucru pe care îl aduce nou aplicația este mediul dinamic pe care îl creează , fata de celelalte implementări ce folosesc un număr mai redus de dispozitive oferind informații doar despre un loc precis cum ar fi terminalul unui aeroport la care mă aflu sau oferta prezentata de un anumit magazin, aceasta comunica in mod constant cu dispozitivele din jur, oferind informații despre locația ta curenta, sălile pe lângă care ai trecut precum si avertizări in timp real in cazul in care mergi in direcția greșita. Interacțiunea dintre utilizator si aplicație fiind mult mai activa spre deosebire de afișarea anumitor notificări.

## Capabilități de dezvoltare ulterioara

Aplicația a fost structurata astfel încât sa poată fi extinsa si sa ii fie adăugate funcționalități noi cat mai ușor posibil, clasele si interfețele proiectului sunt structurate pe pachete ce au denumiri corespunzătoare si intuitive astfel facilitând lizibilitatea codului.

Modul de implementare folosind principii de cuplaj slab si coeziune ridicata ajuta de asemenea la extindere, clasele având roluri corespunzătoare si specifice.

# Tehnologii

În cadrul acestui proiect s-au folosit următoarele tehnologii :

* ***Android*** : Sistemul de operare peste care funcționează aplicația
* **Android Studio / SDK** : mediul de lucru (*IDE*[[1]](#footnote-1)) si echipamentul de dezvoltare
* ***Bluetooth LE*** [[2]](#footnote-2): tehnologie standard fără fir pentru interschimbare de date
* ***Beacon*** : dispozitiv ce folosește tehnologia BLE***[[3]](#footnote-3)*** pentru a transmite informații
* ***SQLite/Room Persistence Library*** [[4]](#footnote-4): Baza de date implementata
* ***AsyncTask – Sarcini asincrone ce lucrează pe fire diferite de execuție***

## Android (1)

Android este un sistem de operare dezvoltat de Google bazat pe o versiune adaptata de Linux si proiectat pentru dispozitivele mobile cu ecran tactil. Android-ul a fost adaptat de Google pentru a fi folosit si pe alte tipuri de dispozitive cum ar fi TV(Android Tv), smartwatch (Wear OS) si mașini (Android Auto).

### Versiunile compatibile cu aplicația

De la prima versiune de sistem de operare scoasa pe piața pe un dispozitiv in 2008 de către Google s-a ajuns in 2018 la versiunea stabila 9 (Pie[[5]](#footnote-5)) ultima versiune recomandata pentru dezvoltare fiind Android 5 (Lollipop) deoarece atunci s-a trecut arhitectura procesoarelor x64. Aplicația a fost dezvoltata pentru telefoanele mobile cu minim android 5, deoarece aceasta este versiunea minima recomandata si care suporta BLE.

Android este cel mai răspândit sistem de operare, ce acopera peste 70% din totalul de dispozitive mobile active astfel cererea de dezvoltatori pentru aceasta platforma fiind foarte mare iar aplicațiile concepute pentru aceasta platforma având popularitate crescuta.

## Android Studio/SDK

### Uneltele de dezvoltare pentru platforma Android

Android SDK reprezintă gruparea de unelte oferite de Google pentru dezvoltarea de aplicații pe sistemul de operare android. Acesta poate fi scris in mai multe limbaje de programare, Java fiind principalul limbaj, însă dezvoltarea tinde sa incline spre noul limbaj implementat ,Kotlin ce spera sa elimine multe din problemele întâmpinate in Java, prin sintaxa moderna si numărul mare de dezvoltatori ce lucrează la îmbunătățirea lui. Ca si alternativa se poate utiliza si limbajul C++ , acesta este folosit in special pentru operații ce necesita o atenție sporita asupra performatei.

Am decis astfel sa folosesc ca si limbaj de programare Java pentru a avea o mai buna stabilitate, suportul oferit acestui limbaj fiind mult mai îndelungat.

### Mediul de dezvoltare (IDE)

Android Studio este mediul de dezvoltare suportat oficial pentru dezvoltarea de aplicații pe sistemul de operare Android disponibil pe majoritatea sistemelor de operare.

Ca si caracteristici Android Studio dispune suportul de construcție *Gradle* ce ajuta la importarea de biblioteci ajutătoare, posibilitatea de a crea mașini virtuale necesare dezvoltării precum și un editor puternic si modern pentru crearea componentelor de interfața.

Android Studio a fost utilizat pentru proiectarea întregului proiect , oferind toate uneltele necesare pentru dezvoltare.

## Bluetooth Low Energy

Bluetooth este cunoscutul standard de tehnologie fără fir ce oferă posibilitatea transmiterii de informații într-un mod securizat simplu si rapid.

După mai multe etape de dezvoltarea a acestei tehnologii a fost introdus si conceputul de BLE (Bluetooth Low Energy) ce are ca ținta minimizare consumului de energie si este folosita împreună dispozitive inteligente ce nu necesita transmiterea unei cantități mari de informație printre care se numără si Beacon.

Aceasta este implementata pe dispozitivele cu sistem de operare Android cu ajutorul bibliotecilor oferite de Google si actualizate in mod constant conform protocolului de implementare.

## Beacon

### Descriere

Beacon este un dispozitiv vizibil constant ce acționează pe o raza mica de acțiune si are ca scop transmiterea de informații de dimensiuni relativ mici. Cu ajutorul tehnologiei Bluetooth Low Energy este capabil sa transmită informații pe o durata îndelungata de timp (chiar pana la un an) si cu un consum de energie extrem de min, fiind necesara o simpla baterie ce poate fi ulterior schimbata. Acesta își are utilitatea in foarte multe domenii cum ar fi transportul, publicitar si multe altele. Cei de la Apple au fost primii care au introdus primul protocol pentru acestea in comerț, urmați de cei de la Google si Radius Network

### Rolul in Aplicație

Aplicația se folosește de acestea pentru a determina locația curenta a utilizatorului relativ cu distanta fata de unul din dispozitive. Dispozitivele se pot achiziționa la un preț destul de mic in comparație cu utilitatea lor si reprezentând astfel o investiție ieftina.

### Metoda de implementare in aplicație

Deși cei de la Google au lansat pe piața protocol EddyStone , aplicația se folosește de librăria oferita de cei de la Radius Network deoarece spre deosebire de Google ce oferă un mod specific de implementare , librăria lor are o licenta ce permite folosirea si modificarea si este capabila sa funcționeze cu aproape toate dispozitivele din gama beacon. Comunicare a fost realizata strict sub documentația oferita de aceștia ce oferă posibilitatea de a modifica si a interpreta într-un mod lizibil si modern datele transmise de dispozitive.

## SQLite/Room Persistence Library

Librărie dezvoltata de Google ce oferă un strat de abstractizare peste SQLite pentru a permite accesul la baza de date într-un mod mai robust. Aceasta librărie permite crearea unui cache local pentru aplicației oferind utilizatorului informații fără a fi nevoit ca acesta sa fie conectat la rețea.

### Rolul in aplicație

Aplicația se folosește de aceasta librărie pentru a stoca informațiile oferite de dispozitivele Bluetooth precum si maparea acestora cu sălile in care sunt distribuite.

Baza de date este generata local si generata pentru fiecare dispozitiv in parte odată cu instalarea aplicației. Spațiul ocupat este foarte mic relativ la spațiul de stocare disponibil pe dispozitivele Android mai noi, iar tabelele vor avea un număr mic de intrări, fiind necesar doar pentru a stoca informațiile unui număr relativ mic de dispozitive.

Librăria oferă o serie de clase si interfețe moderne si bine puse la punct, făcând treaba utilizatorului mai ușoară. Metodele oferite sunt intuitive folosindu-se adnotări ce generează cod in funcție de parametrii oferiți.

### Implementare

Pentru implementarea ei a fost nevoie de folosirea de design pattern Singleton si utilizarea de sarcini asincrone (Async Task) deoarece in mod implicit operațiile pe baza de date nu sunt permise in firul principal de execuție.

## AsyncTask

Sarcinile asincrone, independente de firul principal de execuție, au fost create in nevoia executării de sarcini cu un mare impact computațional pe altfel fire de execuție decât cel principal, executarea acestora într-un singur fir de execuție întrerupând activitatea altor procese.

## Rolul in aplicație

În aplicație, aceasta tehnologie a fost folosita pentru executarea de interogări pe baza de date, aceste întrerupând interfața grafica din modelare, neobservabile pentru accesare rara, însă care, la apariția anumitor probleme , lipsa informației, accesare concurenta si altele ele pot bloca si chiar opri aplicația din funcționare. Ele sunt folosite mai rar in practica datorită complexității însă își au punctele lor forte când vine vorba de a executa o secvența de cod independent de context.

## Implementare

Implementarea s-a realizat cu ajutorul interfeței AsyncTask [[6]](#footnote-6) si urmând documentația oferită de Google

# Obiectivul

Cu acest proiect mi-am setat ca și obiectiv îmbunătățirea orientării in cadrul facultății de informatica și sa creez o aplicație care sa vina in ajutorul celor care nu sunt familiarizați cu structura acesteia.

## Idei conceptuale

In prima faza, odată cu introducerea bazei de date, aplicația a fost conceputa pentru a oferi informații despre sala de care utilizatorul este cel mai apropiat, utilizatorul primea informații in mod constant odată ce se deplasa de la o sala la alta. Mai apoi utilizatorul poate sa își vadă locația sa curenta prin asocierea dispozitivului unei săli cu coordonatele pe o schița a etajului curent.

## Dinamic

Oferind o lista cu sălile din corpul facultății , utilizator poate sa selecteze destinația la care acesta vrea sa ajungă, iar apoi sa ii ofere drumul prin intermediul schiței etajului curent și eventual la etajul de destinație in care sa fie aprins cate un punct în dreptul fiecărei săli pe lângă care utilizatorul trebuie sa treacă .

Odată ce utilizatorul va trece pe lângă o anumita sala, aceasta va fi marcata ca și vizitata iar daca utilizatorul va trece din nou pe lângă acea sala va fi notificat de faptul ca se abate de la direcția propriu zisa. Harta se va schimba dinamic odată ce se va detecta un dispozitiv asociat cu un alt etaj decât al celui precedent

## Simplitate

Utilizatorul își va putea schimba destinația in orice moment, apăsând butonul de back al telefonului și selectând o alta sala, fiind nevoit însă sa se aporie de o sala de la care aplicație sa poate seta drumul

# Descrierea problemei

Problema pe care încerc sa o rezolv odată cu realizarea acestui proiect este lipsa unei platforme care sa ajute studenții sa se orienteze mai bine in cadrul facultății, găsirea anumitor săli sau a unui laborator fiind o problema pentru cei nefamiliarizați cu clădirea facultății.

## Necesitate

In mod tradițional, studenții sunt nevoiți sa își găsească singur sălile de curs și de asemenea sa învețe singuri modul optim pentru a ajunge la acestea, facultatea nu ofera in mod explicit o informare cu privire la acest lucru.

Fie ca sunt noi veniți sau pur și simplu nefamiliarizați cu structura facultății, orientare reprezintă o problema ce poate consuma resurse de timp in mod inutil. Necesitatea unei platforme care sa ofere o imagine de ansamblu asupra facultății fiind susținuta de noii studenți care sunt nevoiți sa ceara indicații persoanelor familiarizate cu clădirea sau chiar sa piardă timp prețios din cadrul unui curs datorita acestei probleme.

## Alte implementări

Din cercetările făcute din articole postate pe diferite pagini de blog sau științifice am reușit sa selectez câteva aplicații interesante ce implementează aceasta tehnologie într-un mod inteligent și inovator și reușesc totodată ușureze viața utilizatorilor.

Prima și cea mai interesantă aplicație care este in dezvoltare in România este aplicația *Smart Shopping* oferita de firma internaționala de comerț *Carrefour[[7]](#footnote-7)*. Aceasta are o rețea vastă de dispozitive beacon in aproape peste 28 de hipermarketuri din tara prin care lanțul de magazine oferă consumatorilor o aplicație simpla, intuitiva și distractiva pentru orientarea in hipermarket de la o zona la alta. Aplicația automatizează informații despre distribuirea produselor in interiorul magazinului, informându-i pe consumatori cu privire la oferte atașate produselor și servicii oferite de fiecare departament in parte, acestea au înregistrat o creștere uimitoare in ceea ce privește folosirea aplicației, care la doua luni după introducerea sa , numărul total de utilizatori crescând cu 400% și cu 700% in următoarele șapte luni. Aplicația este disponibila doar pe dispozitivele mobile oferite de ei

O alta aplicație interesanta este oferita de cei de la OSRAM din Germania, o firma de energie electrica in care departamentul de cercetare și inovare in domeniul digital a realizat un proiect numit beaconsmind[[8]](#footnote-8) ce oferă o platforma pentru firmele de vânzări din toate domeniile oferind informații despre anumite oferte și colectând informații pentru a putea furniza ofertele intru-un mod cat mai optim și mai specific preferințelor fiecărui utilizator. Aplicația oferă o platforma atât clienților dar și angajaților firmei, acestea fiind notificat odată ce un client fidel a intrat in magazinul respectiv.

Institutul de tehnologie german din domeniul vânzărilor a oferit și un premiu acestei aplicații.

Un proiect, ce de asemenea, se adresează sistemului de învățământ este cel implementat la universitatea de stat din California Fullerton numita fiuleț[[9]](#footnote-9)on . Aceasta le permite studenților sa înregistreze prezentele la ore intru-un mod cat mai optim, astfel acestea pot sa își verifice daca au numărul de prezente. Atât studenți cat și profesorii s-au declarat mulțumiți de aceasta tehnologie, scutindu-i de timp pierdut pentru realizarea unei foi de prezenta sau pentru consultarea ei, toate aceste informații fiind transmise dispozitivului personal mobil și bazei de date interne a facultății.

# Descrierea sumara a soluției

Punctul central este aplicația mobila, aceasta se ocupa de tot workflow-ul aplicației exceptând dispozitivele externe.

|  |
| --- |
| C:\Users\bogdan.velesca\AppData\Local\Microsoft\Windows\INetCache\Content.Word\MidWorkFlowDiagram.jpg |

Workflow-ul principal al aplicației este cel in care utilizatorul deschide aplicația și caută o sala de curs anume cu presupunea ca permisiunile aplicației au fost acceptate . Acesta trebuie sa se apropie de o sala din apropiere pentru ai putea identifica locația curenta. Pe parcursul momentului in care utilizatorul se deplasează dispozitivul android va caută in permanenta locația curenta a utilizatorului, acesta va trebui sa se afle in prima faza la o distanta de minim jumătate de metru.

Odată detectat primul beacon , se va face o interogare asupra bazei de date ce se afla in memoria telefonului având ca și cheie unica identificatorul transmis de dispozitivul Bluetooth.

Baza de date întoarce la cerere numele salii la care dispozitivul este atașat care se va afișa pe ecran. Mai apoi utilizatorul are posibilitatea de a-și alege dintr-o lista cu numele sălilor degerata din baza de date destinația.

Odată selectata destinația utilizatorul va fi trimis la o noua activitate ce conține imaginea respectiva etajului curent și un drum semnalizat prin puncte pana la destinație ce își for schimba culoarea odată ce utilizator va trece pe lângă ele, dispozitivul android căutând in continuare dispozitivele beacon din jurul sau. Harta se va schimba odată cu vizitarea unui beacon respectiv următorului etaj.

# Arhitectura proiectului

## Structura aplicației

Proiectul este format doar din aplicația mobila, el funcționează fără conectarea la o rețea, dispozitivele la care se conectează nu necesita o implementare, aplicația se poate conecta la orice model de Beacon , fie el IBacon, AltBeacon (2) sau EddyStone.

Deoarece aplicația are ca și scop extinderea sa aceasta a primit o atenție sporita in ceea ce privește structura codului, a claselor și pachetelor.

### Pachete ( Directoare )

Ca și prezentare generala codul este structurat in întregime pe pachete, fiecare dintre acestea au denumiri specifice rolului componentelor care se afla in interiorul lor.

|  |  |
| --- | --- |
| Figura. 1 – Structura pachetelor | Fiecare pachet respecta un anumit pattern pentru denumiri indicând exact rolul claselor și interfețelor pe care le conține  Pachetul de resurse respecta de asemenea aceste ergului. Fiecare schema de interfața vizuala găsind-se într-un pachet specific. |

In ceea ce privește informațiile stocate pe dispozitiv , aplicația conține baza de date pentru stocare dispozitivelor și informațiilor ce trebuie folosite la generarea ei, Imaginile cu schițele etajelor, motivul pentru care acestea sunt stocate pe dispozitivul mobil fiind eficienta in afișarea acestora, numărul mic al acestora nu necesita o stocare pe o mașina din exterior.

De asemenea florerul de *Assets* ce conține coordonatele dispozitivelor, pentru a putea oferi in timp cat mai scurt locația curenta.

|  |  |
| --- | --- |
| Fig. 1 – structura pachetelor de imagini | Imaginile sunt împărțite pentru diferite dimensiuni de ecran, afișarea acestora făcând-se in funcție de rezolutia dispozitivului mobil |

### Clase și interfețe

Denumirile claselor și interfețelor urmează modelul de scrie *Camel case* fiecare având o întrebuințare specifica.

|  |  |
| --- | --- |
|  | Fiecare clasa are își propune un scop precis, acestea au fost construite astfel încât sa poată fi extinse fără nevoia de schimbare de cod.  Clasele ce se ocupa de baza de date sunt independente de rostul codului, interogările și alte operații care se pot face pe acestea fiind făcute prin intermediul tehnologiei AsyncTask  Interfețele ce gestionează datele dintre clasele aplicației sunt grupate și denumirea lor exprima efectul așteptat. |

### Module și bibliotecile importate

Marea majoritate a bibliotecilor necesare fac parte din gama larga oferite de cei de la Google și sunt importate și gestionate cu ajutorul motorului *Gradle* acesta gestionând totodată versiunile ale acestora notificând dezvoltatorul la fiecare schimbare dar și compatibilitatea versiunii minimei de android setate ca ținta de dezvoltare cu bibliotecile importate.

|  |
| --- |
| Imaginea. 1 - Biblioteci |

## Implementarea soluției

Implementarea soluției a fost realizata in mediul de programare Android in limbajul de programare Java cu ajutorul echipamentului de dezvoltare Android Studio.

După cum se poate observa in Imaginea.1 bibliotecile au fost importate cu ajutorul motorului *Gradle*. Aici a fost setat ca și Target de versiune minima Android 5.0 deoarece acesta suporta majoritatea bibliotecilor cel mai important lucru fiind suportul pentru tehnologia BLE[[10]](#footnote-10) ce pe versiuni mai mici este inexistenta .

Majoritatea bibliotecilor folosite fac parte din Android SDK, acestea au fost folosite pentru proiectarea interfeței grafice folosind biblioteca *AppCompactActivity* și a mediului de lucru cu sistemul de operare Android.

### MainActivity – Ecranul principal al aplicației

Principalele ecrane (Activități [[11]](#footnote-11)) ale aplicație sunt MainActivity și DrawActivity. Prima dintre acestea , anume MainActivity gestionează ecranul principal, primul pe care utilizatorul îl vede când deschide aplicația. Acesta conține cum este prezentat în Imaginea. 2 un simplu chenar de text unde va fi afișat numele salii cele mai apropiate. Înainte de toate acesta va afișa doar o bara de progres bar , acesta se va încarcă odată cu începerea căutării dispozitivelor din jur după care se va afișa interfața propriu zisa . In locul chenarului de text , pana la găsirea primului dispozitiv, se va afișa un mesaj ce indica utilizatorului sa se aporie de o sala sau un laborator din apropiere. Activitatea dispune și de o bara de navigații unde se pot selecta anumite acțiuni implementata personalizat având o resursa grafica separata.

Pornirea procesului de căutare se va face prin intermediul unui fir de execuție adiacent implementate in interiorul clasei ce extinde interfața AsyncTask in scopul creșterii performanței aplicației .

### Fire de execuție asincrone

|  |  |
| --- | --- |
| C:\Users\bogdan.velesca\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screen Shot 2019-06-27 at 12.58.24 AM.PNG  Imaginea. 2 – Fire de executie | *GetClassRoomByUuid, GetRegisteredBeaconName, InsertBeacons* apelează interogări către baza de date  *GetPath* – calculează drumul minim  *SearchBeacons* – pornește căutarea dispozitivelor Bluetooth |

După cum se poate vedea in Imaginea. 2 aplicația se folosește tehnologia oferita de AsyncTask pentru efectuarea mai multor operații ce necesita atenție sporita din parte puterii de procesare a dispozitivului.

### Baza de date

Baza de date a fost implementata cu ajutorul bibliotecii *Room Peristence Api* (3)*.* Aceasta oferă suport pentru a crea o baza de date *SQLite* locala .

Pentru implementare a fost nevoie de aplicarea șablonului de proiectare Singleton, după cum se observa in Imaginea. 3 pentru ca aceasta sa nu poată fi instanțiala de alte fire de execuție aflate in stadiul de rulare.

|  |
| --- |
| C:\Users\bogdan.velesca\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screen Shot 2019-06-27 at 2.02.10 AM.PNG  Imaginea. 3 – Singleton DataBase |

Tabelele bazei de date sunt reprezentate de BeaconTable ce are ca și atribute un **id -** folosit pentru interogări ca și cheie unica , uuid – codul de identificare al dispozitivului , **floor -** etajul căruia ii este atribuit.

BeaconLink – folosit pentru determinarea drumului minim, acesta simulează o lista simplu înlănțuita iar tuplele acesteia vor servi mai apoi ca noduri într-un graf.

|  |  |
| --- | --- |
| Imaginea. 4 – Beacon Table | Imaginea. 5 – Beacon Link |

Interogările au fost fac prin intermediul metodelor implementate pentru fiecare tabela in parte, biblioteca accesând baza de date prin intermediul adnotărilor ce generează codul necesar accesării și folosind parametri metodelor ca si elemente in interogarea respective după cum se poate vedea in Imaginea. 6 .

|  |
| --- |
| Imaginea. 6 - interogari |

### Interfețe de comunicare

Clasele aplicației comunica intre ele prin intermediul unor interfețe de mijloc de au un rol bine stabilit și specific prin denumirea lor după cum se poate observa in Imaginea. 6. Prin intermediul lor se transmit date intre clasele ce extind pornesc operații aflate pe fire diferite de execuție.

|  |
| --- |
| Imaginea. 7 - interfețe |

### SearchBeaconFragment

Interfața grafica ce poate fi accesata din meniul disponibil in MainActivity unde utilizatorul poate selecta destinația aleasa, după cum se poate vedea in Imaginea. 6 se va genera o lista cu sălile disponibile, acestea vor fi extrase cu ajutorul clasei GetClassRoomByNameTask și vor fi afișate folosind ca un layout personalizat pentru element in interiorul unei liste de tip RecyclerView folosit pentru liste de dimensiuni mari ce vor fi generat progresiv din motive evidente de performanta , aceasta lista a fost implementata in interiorul unui SearchView pentru a putea folosi funcționalitatea de căutare după nume.

### DrawActivity și MapView

MapView extinde clasa abstract ImageView , aceasta gestionează drumul afișat utilizatorului prin desenarea punctelor ce indica de asemenea prin schimbarea aspectului locația curenta a utilizatorului, și drumul parcurs pana la un moment dat.

DrawActivity afișează harta de la etajul curent, după cum se poate vedea in Imaginea. 6 și gestionează MapView indicând locația curenta a utilizatorului prin punerea in rulare a firului de execuție ce caută dispozitivele Bluetooth din raza de acțiune și returnează prin intermediul interfeței de mijloc *OnNearBeaconReceive* numele dispozitivului cel mai apropiat prin interogarea bazei de date.

|  |
| --- |
| Imaginea. 8 - Harta |

### Algoritmul de drum minim

Implementat sub forma unui graf și gestionat de clasa *BeaconPath,* acesta extrage din tabela BeaconLink toate intrările și construiește un drum minimum având ca parametri de intrare locația curenta și destinația.

Algoritmul implementat este unul simplu, el se folosește de informațiile oferite de tabela BeaconLink pentru a genera toate legăturile dintre dispozitive căutând fiecare drum posibil din acea locație și oferind mai apoi cel mai scurt drum posibil, evitarea implementării unui algoritm diferit vinde prin pricina faptul ca legăturile dintre dispozitive sunt triviale, ele fiind de cele mai multe ori legate in serie.

Toate aceste operații sunt computate pe un fir de execuție propriu lansat din DrawActivity minimizând riscul de întrerupere al afișării componentelor grafice.

# Interacțiunea cu utilizatorul

Interfața aplicației este cat se poate de triviala pentru a fi cat mai ușor de folosit de către utilizator. Scopul aplicației este de a ușura viața utilizatorului deci implementarea de funcționalități sau componente grafice inutile ar fi fost un minus din acest punct de vedere.

## Ecranul Principal

Ecranul principal are ca prin scop dezvăluirea locației curente a utilizatorului. După cum se observa in Imaginea. 9 utilizator va întâmpina in prima faza o bara de progres fapt ce indica lansarea in execuție algoritmului de căutare a dispozitivelor, precum și un text care sugerează utilizatorului sa se deplaseze către cea mai apropiata încăpere.

|  |  |  |
| --- | --- | --- |
| C:\Users\bogdan.velesca\AppData\Local\Microsoft\Windows\INetCache\Content.Word\closeRoom.png  Imaginea. 9 – ecran inițial | C:\Users\bogdan.velesca\AppData\Local\Microsoft\Windows\INetCache\Content.Word\NowroomFrame.png  Imaginea. 10 –ecran principal | Imaginea. 11 - meniu |

u

Odată ce utilizatorul a ajuns la una dintre săli aceasta va fi afișata in chenarul de sus , aceasta informație se va modifica in mod dinamic in cazul in care utilizatorul își schimba poziția după cum se poate observa în Imaginea. 10. Activitatea mai oferă de asemenea și informații cu privire la dispozitivele mai îndepărtate , utilizatorul având posibilitatea de a afla distanta pana la acestea.

Ecranul principal mai dispune și de un meniu in Imaginea. 11, aflat la îndemână in coltul din stânga sus, de aici putem naviga intre ecranul principal și cel de căutare a unei săli destinație.

## Ecranul de căutare

Aici avem la dispoziție lista cu sălile disponibile Imaginea. 12. Acesta mai dispune și de un motor de filtrare al căutărilor, utilizatorul având posibilitatea de a filtra lista pentru o mai ușoara interacțiune cu aplicația și serviciile oferite de aceasta.

|  |
| --- |
| Imaginea. 12 – Lista destinatiiolr |

## Harta și afișarea drumului

Odată selectata o destinație , utilizatorul va fi trimis către noul ecran ce va conține harta și drumul pe care acesta trebuie sa îl urmeze printr-o serie de puncte colorate diferit in dreptul fiecărei săli de curs sau laborator. Indicațiile sunt sugestive , după cum putem observa in Imaginea. 13, locația curenta a utilizatorului este evidențiata printr-un punct de culoare verde și de dimensiuni mai mari decât restul componentelor. Drumul parcurs este afișat prin puncte de culoare verde dar mai mici ca dimensiune decât cel al locației curente, iar drumul ce urmează a fi parcurs prin puncte de culoare albastra.

|  |
| --- |
| Imaginea. 13 - Harta |

Utilizatorul poate in orice moment sa își schimbe destinația apăsând butonul *Back* altelefonului si intricând-se in ecranul principal.

# Testarea și rezultate

Testarea rezultatelor a fost făcută prin Unit Testing și prin testare manuala.

## Unit Testing

A fost realizata in special pentru algoritmul de căutare al celui mai scurt drum posibil, evitând apariția erorilor.

## Testarea manuala

A fost realizata prin achiziționarea unor dispozitive din gama beacon, plasate la o distanta de cel puțin un metru am fost capabil sa observ apariția multor erori. Una din cele mai preponderente a fost cea de calcul a distantei, aceasta fiind mai mult estimata din puterea semnalului emis de acestea. Un dezavantaj aici a fost numărul mic de dispozitive pe care am putut sa îl achiziționez, fiind nevoit sa simulez aceste dispozitive cu alte telefoane mobile de la diferiți colegi de facultate. Un alt dezavantaj a fost prețul unui dispozitiv de calitate , fiind nevoit sa achiziționez de la firme mici, am constatat ca dispozitivele aveau același identificator și am fost nevoit sa improvizez unicitatea lor in baza de date.

Un mare suport l-am primit din partea companiei la care am aplicat pentru internship vara aceasta, acestea având de gând sa implementeze de asemenea un sistem asemănător in cadrul clădirii firmei , mi-au oferit o un număr mai mare de dispozitive cu care sa lucrez și astfel am putut testa un număr mai mare de cazuri de eșec al acestora.

In final, datorita acestui tip de testare am reușit sa rezolv o mare parte din erorile ce pot apărea utilizând aceste dispozitive.

# Viitoare îmbunătățiri

Orice proiect , in orice stadiu de dezvoltare s-ar afla este disponibil la îmbunătățiri și cum am spus in repetate rânduri , aplicația de fata nu face excepție. Imaginea aplicației poate fi îmbunătățita , prin adăugarea de animații și alte componente de grafica ce dau un aspect și o interacțiune mult mai plăcută cu aplicația.

Din punct de vedere al funcționalității aplicația , fiind făcută întru-un mod modular este dispusa la îmbunătățiri, partea cea mai complicata, aceea de conectare și colectare de informații de la dispozitive fiind deja implementata corect și complet totul tine de imaginație.

Odată cu trecerea pe lângă anumite săli aplicația ar putea afișa informații mai utile cum ar fi orarul salii respective, sau prezenta și ora întoarcerii a unui profesor in cabinetul sau, totodată mai pot fi afișate și informații legate evenimente sau lucruri care au fost uitate in acea sala și unde anume se pot găsi.

Cabinetele Secretarilor pot transmite de asemenea informații utile, cu privire la semnare de contracte sau ridicarea de documente precum și schimbări in orarul acestora.

O alta îmbunătățire o reprezintă extensia, aplicația fiind momentan limitata la corpul facultății de informatica ea poate fi implantata in întreaga universitate, de asemenea oferind informații utile chiar de la intrarea in diferite clădiri.

Lucrurile care se pot face sunt multe și țin doar de imaginație , așadar când vine vorba de capabilitățile aplicației nu exista loc de limitări, ideea de aplicație putând fi extinsa și pe alte platforme mobile cu sistem diferit de operare lucru ce mi-a fost propus odată cu terminarea internshipului.

# Concluzie

La finalul terminării acestui proiect am reușit sa realizez o aplicație stabila pe platforma Android ce poate comunica și extrage informații utile de la dispozitive Bluetooth.

Aplicația are ca și scop principal sa ofere studenților o posibilitate de orientare in interiorul corpului facultății de informatica.

Folosind bibliotecile de baza oferite de Google și biblioteca oficiala a celor de la AltBeacon aplicația reușește sa recepteze dispozitivele aflate in raza de acțiune a tehnologiei Bluetooth și sa colecteze informații ce sunt mai apoi procesate într-o baza de date locala. Mai apoi oferă utilizatorului locația sa curenta și posibilitatea de a putea selecta o destinație din sălile stocate pentru a putea găsi cu ușurința sala respectiva. Toate aceste funcționalități sunt oferite in timp real.

Interacțiunea utilizatorului cu aplicația a fost conceputa sa fie cat mai simplista din punct de vedere al funcționalității nefiind necesara o documentație anterioara despre modul in care se poate folosi aplicația.

Managementul de date este făcut cat se poate de eficient, folosind tehnologia de Bluetooth Low Energy si utilizare de fire de execuție asincrone ce facilitează totodată performata aplicației.

Față de alte aplicații ce folosesc aceasta tehnologie, acest proiect creează o legătura mai strânsă cu dispozitivele Bluetooth oferind informații atât despre locația curenta a utilizatorului cat si despre locația dispozitivelor ce au fost sau urmează a fi vizitate

Aplicația este conceputa ca si un start pentru introducerea tehnologiilor in instituțiile romanești. Deși pot părea bizare ca si concept, ele își au o buna aplicare in viața si pot oferi un ajutor importat in rezolvarea anumitor probleme.

# Bibliografie

1. **Android developers docs. [Interactiv] https://developer.android.com/docs.**

**2. AltBeacon. [Interactiv] https://altbeacon.org/.**

**3. Room Persistance API. [Interactiv] https://developer.android.com/topic/libraries/architecture/room.**

1. Mediu de dezvoltare [↑](#footnote-ref-1)
2. Tehnologie wireless cu consum mic de energie [↑](#footnote-ref-2)
3. Bluetooth Low Energy [↑](#footnote-ref-3)
4. Biblioteca implementata de compania Google ce abstractizează motorul de baze de date SQLite [↑](#footnote-ref-4)
5. Denumirea ultimei versiuni de android [↑](#footnote-ref-5)
6. AsyncTask – Fire de execuție ce se pot executa in mod independent de firul principal [↑](#footnote-ref-6)
7. Carrefour - <https://www.magazinulprogresiv.ro/carrefour-romania-pus-dispozitive-beacon-toate-hypermarketurile> [↑](#footnote-ref-7)
8. Beaconsmind - <https://beaconsmind.com/> [↑](#footnote-ref-8)
9. iFullert - <https://blog.cubeacon.com/ibeacon-implementation-in-cal-state-university-for-attendance-tracking.html> [↑](#footnote-ref-9)
10. |  |  |
    | --- | --- |
    | C:\Users\bogdan.velesca\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screen Shot 2019-06-27 at 1.02.41 AM.PNG  Imaginea. 2 | C:\Users\bogdan.velesca\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screen Shot 2019-06-27 at 1.03.37 AM.PNG |

    BLE- Bluetooth Low energy [↑](#footnote-ref-10)
11. Activitate – Ecranul ce conține componentele de interfața grafica. [↑](#footnote-ref-11)