

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 în 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.

Data azi,

Semnătură student

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Titlul complet al lucrării*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) 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
1.1	Tema proiectului	9
1.2	Abordări	9
1.3	Contribuții	10
1.4	Schimb titlu cu ceva extensibilitate.....	10
2	Tehnologii.....	11
2.1	Android.....	11
2.1.1	Versiunile compatibile cu aplicația.....	11
2.2	Android Studio/SDK.....	12
2.2.1	Uneltele de dezvoltare pentru platforma Android	12
2.2.2	Mediul de dezvoltare (IDE)	12
2.3	Bluetooth Low Energy	13
2.4	Beacon.....	13
2.4.1	Descriere	13
2.4.2	Rolul in Aplicație.....	13
2.4.3	Metoda de implementare in aplicație	14
2.5	SQLite/Room Persistence Library	14
2.5.1	Rolul in aplicație	14
2.5.2	Implementare	14
2.6	AsyncTask.....	15
2.7	Rolul in aplicație	15
2.8	Implementare.....	15
3	Obiectivul	16
3.1	Idei conceptuale.....	16
3.2	Dinamic	16
3.3	Simplitate	16
4	Descrierea problemei	17
4.1	Necesitate	17

4.2	Alte implementări.....	17
5	Descrierea sumara a soluției	19
6	Arhitectura proiectului.....	20
6.1	Structura aplicației.....	20
6.1.1	Pachete (Directoare).....	20
6.1.2	Clase și interfețe.....	21
6.1.3	Module și bibliotecile importate	22
6.1	Implementarea soluției	22
6.1.1	MainActivity – Ecranul principal al aplicației.....	23
6.1.2	Fire de execuție asincrone.....	24
6.1.3	Baza de date	24
6.1.4	Interfețe de comunicare.....	25
6.1.5	SearchBeaconFragment	26
6.1.6	DrawActivity și MapView	26
6.1.7	Algoritmul de drum minim	27
7	Interacțiunea cu utilizatorul	28
7.1	Ecranul Principal	28
7.2	Ecranul de căutare	29
7.3	Harta și afișarea drumului	29
8	Testarea și rezultate	31
8.1	Unit Testing.....	31
8.2	Testarea manuala.....	31
9	Viitoare îmbunătățiri.....	32
10	Concluzie	33

Motivație

Datorită pasiunii mele pentru platformele mobile și 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 simplă comunicare cu acesta observând datele și informațiile pe care acestea le aduc.

Așadar Folosind tehnologia *Bluetooth Low Energy* ce oferă atât dispozitivelor android cât și a celor ce emit semnale o metoda de comunicare cu cât mai puțină energie am decis să creez o aplicație ce comunica cu acestea. Cu ajutorul dispozitivelor relativ nou introduse numite *Bacon* mi-am propus să realizez o aplicație care să ajute noii studenți, persoane interesate și alți vizitatori să se deplaseze mai ușor în cadrul facultății de informatică.

Observând funcționalitatea *Beacon*-urilor în alte domenii, cum ar fi în cadrul aeroporturilor, marketing-ului și chiar în domeniul educației, ce mai interesantă aplicație din punctul meu de vedere fiind cea care identifică prezența studenților la cursuri și le oferă acestora informații cu privire la cursul curent am decis să mă folosesc de acestea pentru a crea o hartă, secționată pe etaje care comunică în timp real poziția curentă a utilizatorului precum și un drum optim spre locație aleasă de utilizator .

Fără de cel puțin o aplicație care pur și simplu oferă informații legate de obiectul atașat de dispozitiv cum ar fi în cadrul medical pentru identificarea unică a unui pacient oferind medicului informații despre istoricul medical sau în domeniul asigurărilor preluarea de informații despre oferta și în multe alte domenii aplicația mea oferă informații în timp real și comunicare mult mai strânsă cu aceasta.

Motivul pentru care am folosit ca și platformă de lucru dispozitivele mobile, lăsând deoparte pasiunea mea în această direcție, a fost nevoia de o comună dinamică cu dispozitivele *Bluetooth* dar și prin popularitatea acestora astfel aplicația se adresează unei marje demografice mai mari. Utilizarea prin intermediul dispozitivelor mobile este mult mai intuitivă și mai la îndemână decât serviciile web.

1 Introducere

Tehnologia *Bluetooth* cu energie slabă sau *Bluetooth* Intelligent este pretutindeni. Având un impact neglijabil asupra consumului de energie aceasta tehnologie a fost aplicata în multe domenii, fie ca vorbim de cel medical , educațional sau comercial această tehnologie s-a dovedit extrem de folositoare îmbunătățindu-ne as putea spune viața si modul în care interacționam mediul din jur.

1.1 Tema proiectului

Tema lucrării s-a ivit odată cu problema orientării în cadrul corpului facultății de informatica. 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 ne fiind capabil sa găsesc în timp util anumite săli si prin interacționarea cu studenți străini cărora a trebuit sa le ofer indicații.

1.2 Abordări

Abordări similare au fost introduse în domeniul medial unde medicii își pot găsi pacienții si sa primească informații în mod instant cu privire la starea actuala a acestuia, în 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 în la intrare cate un *bacon ce vor indica evident identificarea unica a acestora*.

Relația dintre dispozitiv si sala în 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 eficienței, după cum știm comunicarea cu un server presupunând conexiune constanta la internet si riscul de erori crescut.

1.3 Contribuții

Cel mai important lucru ce îl aduce nou aplicația este mediul dinamic pe care li 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 lucrează in mod constant cu dispozitivele din jur, oferind informații despre locația ta curenta, sălile pe lângă care ai trecut in parcursul 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.

1.4 Schimb titlu cu ceva extensibilitate

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.

2 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) si echipamentul de dezvoltare
- **Bluetooth LE** ²: tehnologie standard fără fir pentru interschimbare de date
- **Beacon** : dispozitiv ce folosește tehnologia BLE³ pentru a transmite informații
- **SQLite/Room Persistence Library** ⁴: Baza de date implementata
- **AsyncTask** – *Sarcini asincrone ce lucrează pe fire diferite de execuție*

2.1 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. Fiind într-un mediu de dezvoltare deschisa, Google a reușit sa dezvolte specializate si in alte arii având acum sisteme precum Android TV , *Wear* ⁵Os pentru ceasuri, Android Auto pentru mașini si multe alte dispozitive inteligente.

2.1.1 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 (Pie⁶) ultima versiune recomandata pentru dezvoltare fiind Android 5 (Lollipop) deoarece atunci s-a trecut arhitectura procesoarelor x64. Aplicația având ca Target minim dispozitive android cu minim android 5 si versiunea de unelte (SDK) 28 deoarece acestea dispun de tehnologia Bluetooth inteligent si sunt capabile sa proceseze conexiuni intru-un timp optim.

¹ Mediu de dezvoltare

² Tehnologie wireless cu consum mic de energie

³ Bluetooth Low Energy

⁴ Biblioteca implementata de compania Google ce abstractizează motorul de baze de date SQLite

⁵ Sintagma ce exprima dispozitive ce pot fi purtate ca si accesorii

⁶ Denumirea ultimei versiuni de android

Android este cel mai răspândit sistem de operare ce se dezvoltă prin intermediul acoperind peste 70% din totalul de dispozitive mobile active astfel cererea de dezvoltatori pentru această platformă fiind foarte mare iar aplicațiile concepute pentru această platformă având popularitate crescută.

2.2 Android Studio/SDK

2.2.1 Uneltele de dezvoltare pentru platforma Android

Android Studio reprezintă gruparea de unelte oferite de Google pentru dezvoltarea de aplicații pe sistemul de operare android. Acesta poate fi scris în mai multe limbaje de programare, Java fiind principalul limbaj, însă dezvoltarea tinde să incline spre noul limbaj implementat, Kotlin ce spera să elimine multe din problemele întâmpinate în Java, prin sintaxa modernă și numărul mare de dezvoltatori ce lucrează la îmbunătățirea lui. Ca și alternativă se poate utiliza și limbajul C++ , acesta este folosit în special pentru operații ce necesită o atenție sporită asupra performanței.

Am decis astfel să folosesc ca și limbaj de programare Java pentru a avea o mai bună stabilitate, suportul oferit acestui limbaj fiind mult mai îndelungat.

2.2.2 Mediul de dezvoltare (IDE)

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

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

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

2.3 Bluetooth Low Energy

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

După mai multe etape de dezvoltarea a acestei tehnologii a fost introdus și conceputul de Energy slaba (Bluetooth Low Energy) ce are ca țintă minimizare consumului de energie și este folosită împreună dispozitive inteligente ce nu necesită transmiterea unei cantități mari de informație printre care se numără și Beacon.

Aceasta este implementată pe dispozitivele cu sistem de operare Android cu ajutorul bibliotecilor oferite de Google și actualizate în mod constant conform protocolului de implementare oferind astfel o mai mare dezvoltare a dispozitivelor Bluetooth.

2.4 Beacon

2.4.1 Descriere

Beacon este un dispozitiv vizibil constant ce acționează pe o rază mică de acțiune și are ca scop transmiterea de informații de dimensiuni relativ mici. Cu ajutorul tehnologiei Bluetooth Low Energy este capabil să transmită informații pe o durată îndelungată de timp (chiar până la un an) și cu un consum extrem de mic de energii, fiind necesară o simplă baterie ce poate fi ulterior schimbată. Acesta își are utilitatea în foarte multe domenii cum ar fi transportul, publicitar și multe altele. Cei de la Apple au fost primii care au introdus primul protocol pentru acestea în comerț, urmați de cei de la Google și Radius Network.

2.4.2 Rolul în Aplicație

Aplicația se folosește de acestea pentru a determina locația curentă a utilizatorului relativ cu distanța față de unul din dispozitive. Dispozitivele se pot achiziționa la un preț destul de mic în comparație cu utilitatea lor și reprezentând astfel o investiție ieftină.

2.4.3 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 sursa deschisa si este capabila sa funcționeze cu aproape toate dispozitivele din gama beacon. Comunicare a fost realizata stric sub documentația oferita de aceștia ce oferă posibilitatea de a modifica si a interpreta intru-un mod lizibil si modern datele transmise de dispozitive.

2.5 SQLite/Room Persistence Library

Librărie dezvoltata deschis 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 unei memorie cache a aplicației oferind utilizatorului informații fără a fi nevoit ca acesta sa fie conectat la rețea.

2.5.1 Rolul in aplicație

Aplicația se folosește de aceasta librărie pentru a stoca informațiile respective fiecărui dispozitiv in parte legătura dintre acestea.

Baza de date este generata construita local si generata pentru fiecare dispozitiv in parte odată cu instalarea aplicației. Spațiul Consumat 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, eliminând anumite griji ale dezvoltatorului. Metodele oferite sunt intuitive folosindu-se adnotări ce generează cod in funcție de parametrii oferiți.

2.5.2 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.

2.6 AsyncTask

Sarcinile asincrone, independente de firul principal de execuție, au fost create în 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.

2.7 Rolul în aplicație

În aplicație, aceasta tehnologie a fost folosită pentru executarea de interogări pe baza de date, acestea întrerupând interfața grafică din modelare, neobservabile pentru accesare rară, însă care, la apariția anumitor probleme, lipsa informației, accesare concurentă și altele ele pot bloca și chiar opri aplicația din funcționare. Ele sunt folosite mai rar în practică datorită complexității însă își au punctele lor forte când vine vorba de a executa o secvență de cod independent de context.

2.8 Implementare

Implementarea s-a realizat cu ajutorul interfeței AsyncTask ⁷ oferite de Java și adaptate de Google pentru nevoile sistemului de operare android.

⁷ AsyncTask – Fire de execuție ce se pot executa în mod independent de firul principal

3 Obiectivul

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

3.1 Idei conceptuale

În prima fază, odată cu introducerea bazei de date aplicația a fost concepută pentru a oferi informații despre sala de care utilizatorul este cel mai apropiat, utilizatorul primea informații în mod constant odată ce se deplasa de la o sală la alta. Mai apoi utilizatorul poate să își vadă locația sa curentă prin asocierea dispozitivului unei săli cu coordonatele pe o schiță a etajului curent.

3.2 Dinamic

Oferind o listă cu sălile din corpul facultății, utilizatorul poate să selecteze destinația la care acesta vrea să ajungă, iar apoi să îi ofere drumul prin intermediul schiței etajului curent și eventual la etajul de destinație în care să fie aprinse în dreptul fiecărei săli pe lângă care utilizatorul trebuie să treacă.

Odată ce utilizatorul va trece pe lângă o anumită sală, aceasta va fi marcată ca și vizitată iar dacă utilizatorul va trece din nou pe lângă acea sală va fi notificat de faptul că se abate de la direcția proprie. Harta se va schimba dinamic odată ce se va detecta un dispozitiv asociat cu un alt etaj decât al celui precedent.

3.3 Simplitate

Utilizatorul își va putea schimba destinația în orice moment, apăsând butonul de back al telefonului și selectând o altă sală, fiind nevoit însă să se apropie de o sală de la care aplicația să poată seta drumul.

4 Descrierea problemei

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

4.1 Necesitate

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

Fie că sunt noi veniți sau pur și simplu nefamiliarizați cu structura facultății, orientare reprezintă o problemă ce poate consuma resurse de timp în mod inutil. Necesitatea unei platforme care să ofere o imagine de ansamblu asupra facultății fiind susținută de noii studenți care sunt nevoiți să ceară indicații persoanelor familiarizate cu clădirea sau chiar să piardă timp prețios din cadrul unui curs datorită acestei probleme.

4.2 Alte implementări

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

Prima și cea mai interesantă aplicație care este în dezvoltare în România este aplicația *Smart Shopping* oferită de firma internațională de comerț *Carrefour*⁸. Aceasta are o rețea vastă de dispozitive beacon în aproape peste 28 de hipermarketuri din țară prin care lanțul de magazine oferă consumatorilor o aplicație simplă, intuitivă și distractivă pentru orientarea în hipermarket de la o zonă la alta. Aplicația automatizează informații despre distribuția produselor în interiorul magazinului, informându-i pe consumatori cu privire la oferte atașate produselor și

⁸ Carrefour - <https://www.magazinulprogresiv.ro/carrefour-romania-pus-dispozitive-beacon-toate-hypermarketurile>

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 disponibilă doar pe dispozitivele mobile oferite de ei

O alta aplicație interesantă este oferită de cei de la OSRAM din Germania, o firmă de energie electrică in care departamentul de cercetare și inovare in domeniul digital a realizat un proiect numit beaconsmind⁹ ce oferă o platformă 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 platformă atât clienților dar și angajaților firmei, acestea fiind notificate 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 numită fiule¹⁰on . Aceasta le permite studenților sa înregistreze prezențele la ore intru-un mod cat mai optim, astfel acestea pot sa își verifice dacă au numărul de prezențe. Atât studenți cat și profesorii s-au declarat mulțumiți de aceasta tehnologie, scutindu-i de timp pierdut pentru realizarea unei foi de prezență sau pentru consultarea ei, toate aceste informații fiind transmise dispozitivului personal mobil și bazei de date interne a facultății.

⁹ Beaconsmind - <https://beaconsmind.com/>

¹⁰ iFullert - <https://blog.cubeacon.com/ibeacon-implementation-in-cal-state-university-for-attendance-tracking.html>

5 Descrierea sumara a soluției

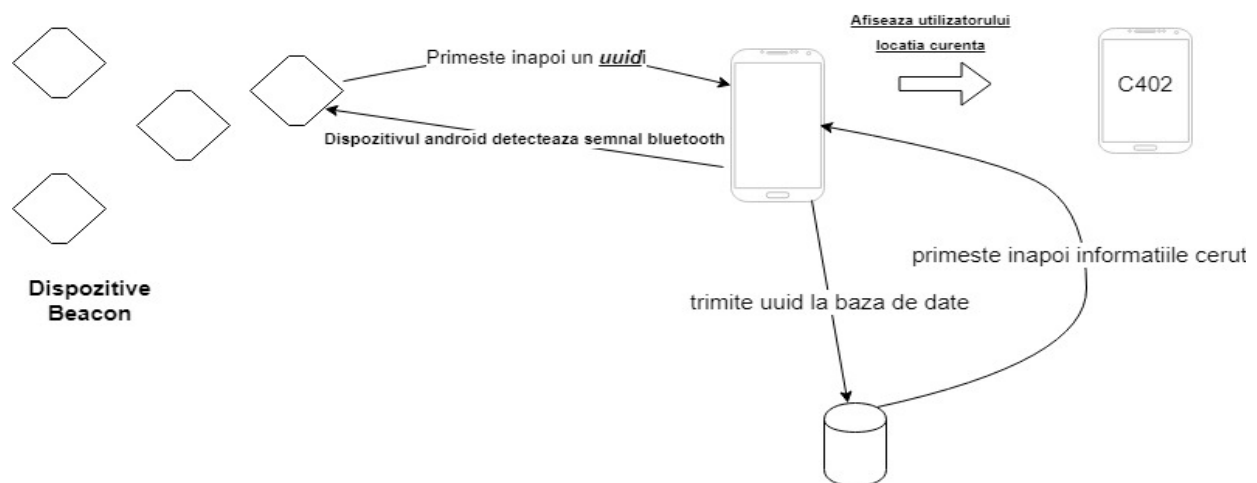
Punctul central este aplicația mobilă, aceasta se ocupa de tot workflow-ul aplicației exceptând dispozitivele externe.

Workflow-ul principal al aplicației este cel în care utilizatorul deschide aplicația și caută o sală de curs anume cu presupunea ca permisiunile aplicației au fost acceptate . Acesta trebuie să se apropie de o sală din apropiere pentru a putea identifica locația curentă. Pe parcursul momentului în care utilizatorul se deplasează dispozitivul android va căuta în permanentă locația curentă a utilizatorului, acesta va trebui să se afle în prima fază la o distanță de minim jumătate de metru.

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

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

Odată selectată destinația utilizatorul va fi trimis la o nouă activitate ce conține imaginea respectivă etajului curent și un drum semnalizat prin puncte până la destinație ce își schimbă culoarea odată ce utilizator va trece pe lângă ele, dispozitivul android căutând în continuare dispozitivele beacon din jurul său. Harta se va schimba odată cu vizitarea unui beacon respectiv următorului etaj.



6 Arhitectura proiectului

6.1 Structura aplicației

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

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

6.1.1 Pachete (Directoare)

Ca și prezentare generală codul este structurat în întregime pe pachete, fiecare dintre acestea au denumiri specifice rolului componentelor care se află în interiorul lor.

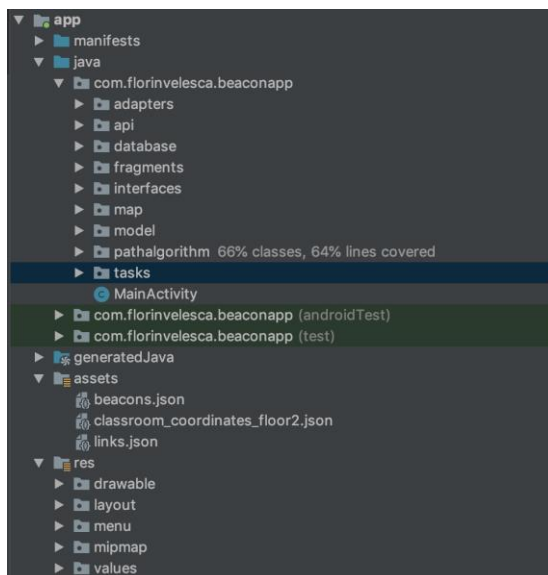


Figura. 1 – Structura pachetelor

Fiecare pachet respecta un anumit pattern pentru denumiri indicând exact rolul claselor și interfețelor pe care le conține

Pachetul de resurse respectă de asemenea aceste reguli. Fiecare schema de interfață vizuală găsind-se într-un pachet specific.

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

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

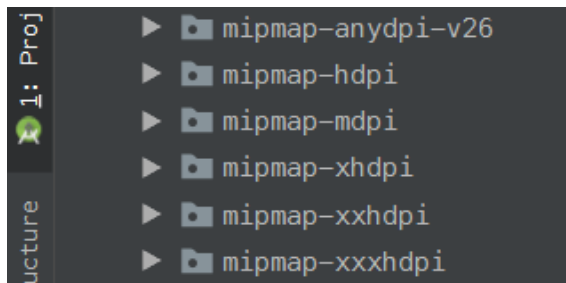
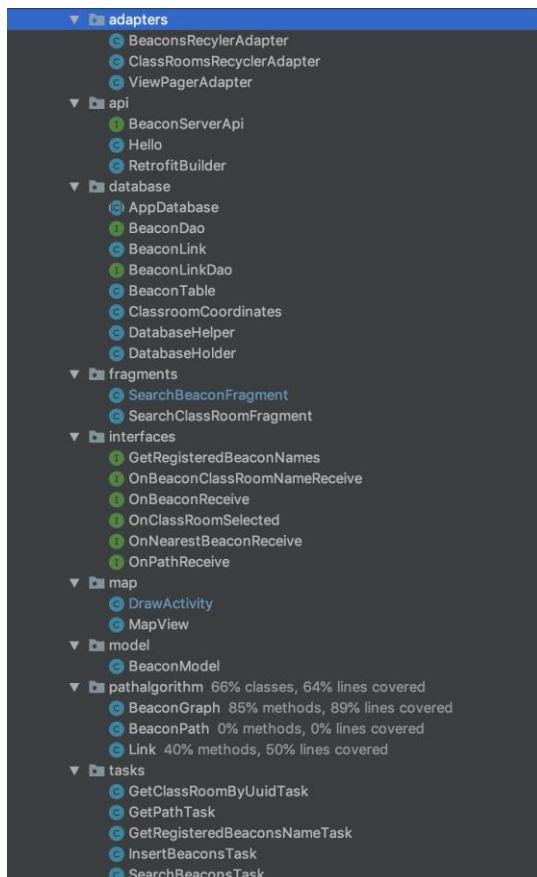


Fig. 1 – structura pachetelor de imagini

Imaginile sunt împărțite pentru diferite dimensiuni de ecran, afișarea acestora făcându-se in funcție de rezolutia dispozitivului mobil

6.1.2 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.

6.1.3 Module și bibliotecile importate

Marea majoritate a bibliotecilor necesare fac parte din gama largă 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 minime de android setate ca ținta de dezvoltare cu bibliotecile importate.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.florinvelesca.beaconapp"
        minSdkVersion 21
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:support-v4:28.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.google.android.gms:play-services:12.0.1'
    implementation 'org.altbeacon:android-beacon-library:2.16.1'
    implementation 'com.android.support:recyclerview-v7:28.0.0'
    implementation 'com.android.support:design:28.0.0'
    implementation 'com.squareup.retrofit2:retrofit:2.5.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.5.0'
    implementation 'android.arch.persistence.room:runtime:1.1.1'
    annotationProcessor 'android.arch.persistence.room:compiler:1.1.1'
    debugImplementation 'com.amitshekhar.android:debug-db:1.0.4'
    implementation group: 'androidx.annotation', name: 'annotation', version: '1.1.0-rc01'
    testImplementation 'com.google.truth:truth:0.45'
}
```

Imaginea. 1

6.1 Implementarea soluției

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

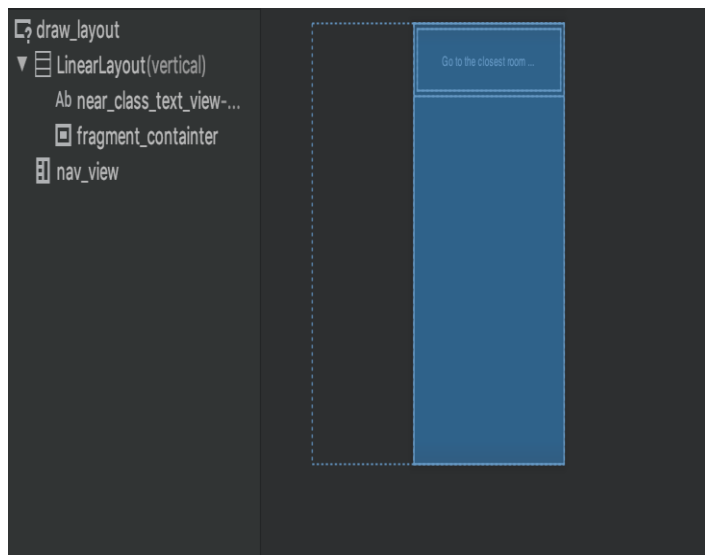
După cum se poate observa în Imaginea.1 bibliotecile au fost importate cu ajutorul motorului *Gradle*. Aici a fost setat ca și Target de versiune minimă Android 5.0 deoarece acesta

suporta majoritatea bibliotecilor cel mai important lucru fiind suportul pentru tehnologia BLE¹¹ ce pe versiuni mai mici este inexistentă .

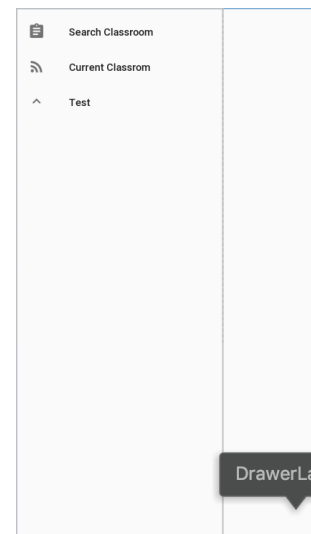
Majoritatea bibliotecilor folosite fac parte din Android SDK, acestea au fost folosite pentru proiectarea interfeței grafice folosind biblioteca *AppCompatActivity* și a mediului de lucru cu sistemul de operare Android.

6.1.1 MainActivity – Ecranul principal al aplicației

Principalele ecrane (Activități ¹²) ale aplicației 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 zisă . In locul cadrul 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.



Imaginea. *Error! Main Document Only.*



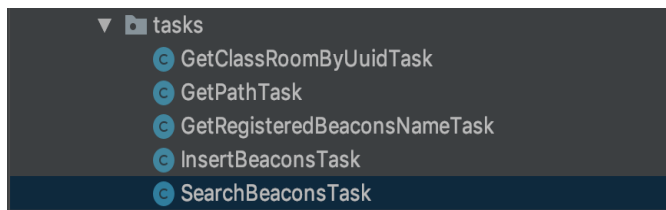
¹¹ BLE- Bluetooth Low energy

¹² Activitate – Ecranul ce conține componentele de interfața grafică.

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

6.1.2 Fire de execuție asincrone

După cum se poate vedea în Imaginea. 3 aplicația se folosește tehnologia oferită de `AsyncTask` pentru efectuarea mai multor operații ce necesită atenție sporită din partea puterii de procesare a dispozitivului.



Imaginea. 2

GetClassRoomByUuid,
GetRegisteredBeaconName, *InsertBeacons*
apelează interogări către baza de date
GetPath – calculează drumul minim
SearchBeacons – pornește căutarea
dispozitivelor Bluetooth

6.1.3 Baza de date

Baza de date a fost implementată cu ajutorul bibliotecii *Room Persistence Api* (3). Aceasta oferă suport pentru a crea o bază de date *SQLite* locală .

Pentru implementare a fost nevoie de aplicarea șablonului de proiectare Singleton, după cum se observă în Imaginea. 4 pentru ca aceasta să nu poată fi instanțiată de alte fire de execuție aflate în stadiul de rulare.

```
public class DatabaseHolder {  
    private static AppDatabase database;  
  
    public static AppDatabase getDatabase(Context context) {  
        if (database == null) {  
            database = Room.databaseBuilder(context.getApplicationContext(), AppDatabase.class, "my_database").build();  
        }  
        return database;  
    }  
}
```

Imaginea. 3

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

BeaconLink – folosit pentru determinarea drumului minim, acesta simulează o listă simplă înlănțuită iar tuplele acestora vor servi mai apoi ca noduri într-un graf.

```
@Entity(indices = {@Index(value = {"uuid", "minor"}, unique = true)})
public class BeaconTable implements Serializable {

    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "id")
    private int id;

    @ColumnInfo(name = "uuid")
    private String uuid;

    @ColumnInfo(name = "minor")
    private String minor;

    @ColumnInfo(name = "classRoomName")
    private String classRoomName;
```

```
@Entity
public class BeaconLink {
    @PrimaryKey(autoGenerate = true)
    private int id;
    private int beacon1Id;
    private int beacon2Id;
    private int distance;
```

Interogările au fost făcute prin intermediul metodelor implementate pentru fiecare tabelă în parte, biblioteca accesând baza de date prin intermediul anotărilor ce generează codul necesar accesării și folosind parametri metodelor ca și elemente în interogarea respective după cum se poate vedea în Imaginea. 5 .

```
@Dao
public interface BeaconLinkDao {

    @Query("SELECT * FROM beaconlink")
    List<BeaconLink> getAllBeaconLinks();

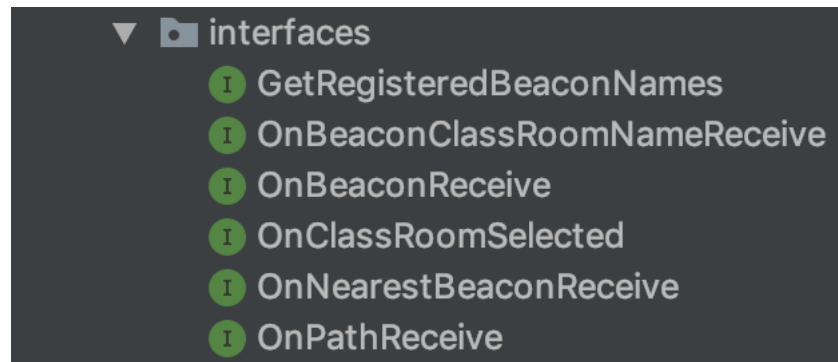
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void insertLink(BeaconLink link);
}
```

Imaginea. 4

6.1.4 Interfețe de comunicare

Clasele aplicației comunică între ele prin intermediul unor interfețe de mijloc de a avea un rol bine stabilit și specific prin denumirea lor după cum se poate observa în Imaginea. 6. Prin

intermediul lor se transmit date între clasele ce extind pornesc operații aflate pe fire diferite de execuție.



Imaginea. 5

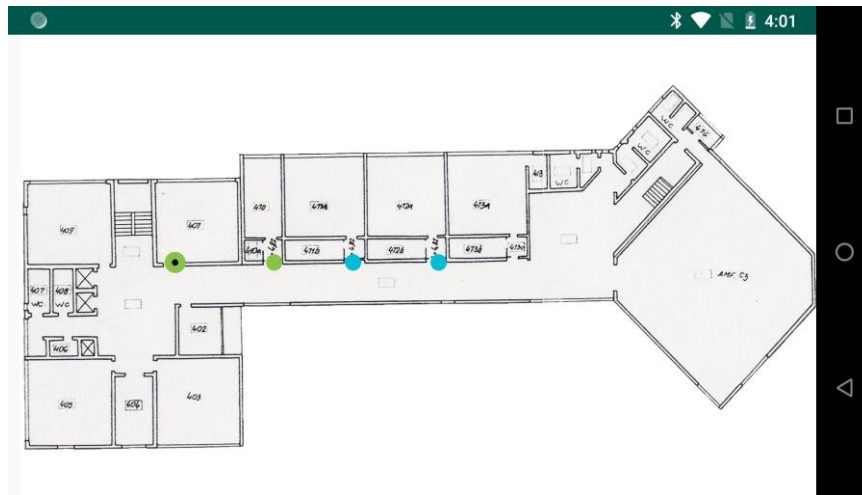
6.1.5 SearchBeaconFragment

Interfața grafică ce poate fi accesată din meniul disponibil în MainActivity unde utilizatorul poate selecta destinația aleasă, după cum se poate vedea în Imaginea. 6 se va genera o listă cu sălile disponibile, acestea vor fi extrase cu ajutorul clasei GetClassRoomByNameTask și vor fi afișate folosind ca un layout personalizat pentru element în interiorul unei liste de tip RecyclerView folosit pentru liste de dimensiuni mari ce vor fi generat progresiv din motive evidente de performanță, această listă a fost implementată în interiorul unui SearchView pentru a putea folosi funcționalitatea de căutare după nume.

6.1.6 DrawActivity și MapView

MapView extinde clasa abstractă ImageView, aceasta gestionează drumul afișat utilizatorului prin desenarea punctelor ce indică de asemenea prin schimbarea aspectului locația curentă a utilizatorului, și drumul parcurs până la un moment dat.

DrawActivity afișează harta de la etajul curent, după cum se poate vedea în Imaginea. 6 și gestionează MapView indicând locația curentă a utilizatorului prin punerea în 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. 6

6.1.7 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 curentă 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.

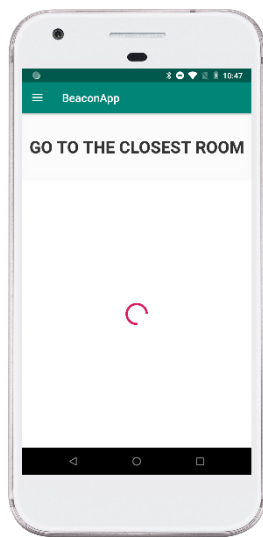
Toate aceste operații sunt computeate pe un fir de execuție propriu lansat din *DrawActivity* minimizând riscul de întrerupere al afișării componentelor grafice.

7 Interacțiunea cu utilizatorul

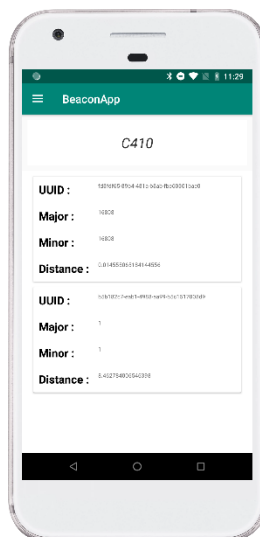
Interfața aplicației este cat se poate de trivială pentru a fi cat mai ușor de folosit de către utilizator. Scopul aplicației este de a ușura viața utilizatorului deci implementarea de funcționalități sau componente grafice inutile ar fi fost un minus din acest punct de vedere.

7.1 Ecranul Principal

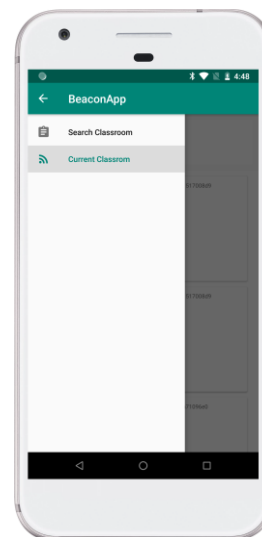
Ecranul principal are ca prin scop dezvăluirea locației curente a utilizatorului. După cum se observa în Imaginea . 7 utilizator va întâmpina în prima fază o bară de progres fapt ce indică lansarea în execuție algoritmului de căutare a dispozitivelor, precum și un text care sugerează utilizatorului să se deplaseze către cea mai apropiată încăpere.



Imaginea. 7



Imaginea. 8



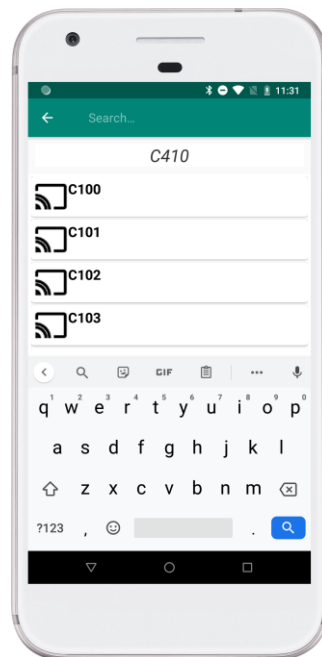
Imaginea. 9

Odată ce utilizatorul a ajuns la una dintre săli aceasta va fi afișată în chenarul de sus , aceasta informație se va modifica în mod dinamic în cazul în care utilizatorul își schimbă poziția după cum se poate observa în Imaginea. 8. Activitatea mai oferă de asemenea și informații cu privire la dispozitivele mai îndepărtate , utilizatorul având posibilitatea de a afla distanța până la acestea.

Ecranul principal mai dispune și de un meniu in Imaginea. 9, aflat la îndemână in coltul din stânga sus, de aici putem naviga între ecranul principal și cel de căutare a unei săli destinație.

7.2 Ecranul de căutare

Aici avem la dispoziție lista cu sălile disponibile Imaginea. 10 . 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.

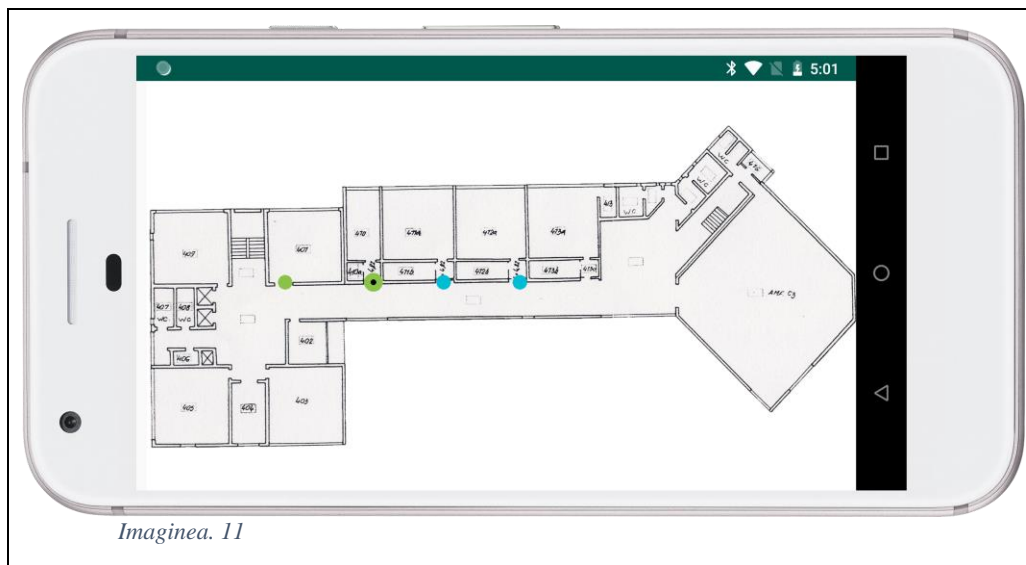


Imaginea. 10

7.3 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. 11 , locația curenta a utilizatorului este evidențiată 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 albastră.



Imaginea. 11

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

8 Testarea și rezultate

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

8.1 Unit Testing

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

8.2 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.

9 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 într-un mod modular este dispusa la îmbunătățiri, partea cea mai complicat, 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.

10 Concluzie

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

Aplicația are ca și scop principal să ofere studenților o posibilitate de orientare în interiorul corpului facultății de informatică.

Folosind bibliotecile de bază oferite de Google și biblioteca oficială a celor de la AltBeacon aplicația reușește să recepteze dispozitivele aflate în raza de acțiune a tehnologiei Bluetooth și să colecteze informații ce sunt mai apoi procesate într-o bază de date locală. Mai apoi oferă utilizatorului locația sa curentă și posibilitatea de a putea selecta o destinație din sălile stocate pentru a putea găsi cu ușurință sala respectivă. Toate aceste funcționalități sunt oferite în timp real.

Interacțiunea utilizatorului cu aplicația a fost concepută să fie cât mai simplă din punct de vedere al funcționalității nefiind necesară o documentație anterioară despre modul în care se poate folosi aplicația.

Managementul de date este făcut cât se poate de eficient, folosind tehnologia de Bluetooth Low Energy și utilizare de fire de execuție asincrone ce facilitează totodată performanța aplicației.

11 Bibliografie

1. **Android developers docs.** <https://developer.android.com/docs>.
2. **AltBeacon.** <https://altbeacon.org/>.
3. **Room Persistence API.**
<https://developer.android.com/topic/libraries/architecture/room>.