



Universitatea POLITEHNICĂ din București
Facultatea de Automatică și Calculatoare
Departamentul Automatică și Informatică Industrială



Noi Servicii Integrate pentru Orientare și Accesibilitate pentru un Campus Universitar Inteligent cu Locații Multiple

Teză de Doctorat susținută în cadrul Școlii Doctorale de Automatică și
Calculatoare Universitatea Politehnică București pentru îndeplinirea
parțială a cerințelor de obținere a titlului de doctor

Drd.Ing. Ioan DAMIAN
București 2023

ABSTRACT

Problema orientării și accesibilității este întâlnită în spațiile publice largi. Această teză tratează această problemă din perspectiva navigării pietonale, ceea ce vine în ajutorul noi-veniților, persoanelor cu nevoi speciale, și protejarea împotriva riscurilor pandemiilor. O investigație amplă a stadiului actual demonstrează că există diferite soluții pentru această problemă, cu o posibilă direcție spre conceptul de campus inteligent prin intermediul hărților și instrumentelor de ghidare, implementate cu tehnologii diverse. Cu toate acestea, fiecare abordare încearcă să rezolve o altă problemă particulară, ceea ce face dificil procesul de centralizare pentru utilizator. De aceea, obiectivele cercetării propuse în această teză sunt adăugarea contribuțiilor personale referitoare la servicii care îmbunătățesc experiența utilizatorului final, evitarea situațiilor de risc prin strategii facile de integrare, folosirea informațiilor generate de comunitate și datele locale sau externe, dezvoltarea configurațiilor pentru validare. Metoda de cercetare folosită pentru această teză implică scenariile pentru determinarea experienței utilizatorului, reprezentarea spațiului public, arhitectura orientată pe servicii, servicii de navigare multi-criteriale prin resurse denumite *policies* și dezvoltarea unei platforme pentru validare. Contribuțiile originale de cercetare se referă la dezvoltarea unui model inspirat din grafuri ponderate ce suportă navigarea pietonală multi-criterială, integrarea nevoilor reale ale utilizatorilor cu ajutorul resurselor de tip *policies*, operații generale și arhitectură detaliată pentru o platformă ce suportă serviciile de rutare. Resursele propuse de tip *policies* actualizează modelul de bază de graf în funcție de nevoile utilizatorilor, aceste contribuții fac ca sistemul să fie determinat de utilizator. Algoritmii propuși în această teză pentru propusele *policies* au fost implementați și validați. Aceștia se axează pe voturile comunității, evitarea zonelor aglomerate, evitarea zonelor poluate, evitarea condițiilor meteo nefavorabile și evitarea porțiunilor de drum cu risc de tip locomotor. Serviciile corespunzătoare ce oferă experiența utilizatorului final legată de orientare și accesibilitate sunt implementate într-o platformă nouă. Utilizatorul invocă algoritmul de rutare expus de această platformă pentru a obține informații de orientare. Mai mult, utilizatorului îi sunt oferite recomandări de navigare generate sub formă textuală și vizuală, bazate pe ruta rezultată din executarea algoritmilor de generare de drum. Platforma nouă este implementată și validată prin scenarii de test supuse la condiții bazate pe cerințe reale. Aceste cerințe sunt deduse din cercetarea planurilor clădirilor reale și din cercetarea numerelor reale de participanți la diferite universități. Rezultatele obținute din testele de performanță demonstrează că platforma este viabilă și că poate servi între 1000 și 5000 de utilizatori virtuali (numere comparabile cu scenariile reale) cu timpul mediu de răspuns între 500ms și 4500ms și sub 4% procentaj de răspunsuri cu erori.

MULȚUMIRI

Mulțmirile mele merg către Coordonatorul meu Științific Profesor Anca Daniela Ioniță, pentru profesionalism, cunoaștere vastă, răbdare și tact. Sfaturile dumneaei au avut o influență enormă asupra gândirii mele și studiilor mele doctorale.

Aș vrea să mulțumesc comisiei mele care m-a ghidat în ultimii ani, pentru implicarea neîncetată și criticile constructive, Profesor Mariana Mocanu, Profesor Silvia Anton, și Profesor Adriana Olteanu.

Aș vrea să le mulțumesc și colegilor mei Florin Lăcătușu și Marian Lăcătușu pentru punctele lor de vedere unice, și pentru conversațiile și colaborările noastre.

Nu în cele din urma, aș vrea să le mulțumesc mamei mele, tatălui meu, mătușii mele Ana și surorilor mele, Ecaterina și Elena, pentru toată răbdarea pe care au avut-o cu mine și pentru încurajările continue.

Cuprins

Lista de Tabele	6
Lista de Figuri	6
1 Introducere	7
1.1 Context și Motivație	7
2 Analiza Stadiului Actual	9
2.1 Planificarea drumului	9
2.2 Mobilitate și Accesibilitate în Spațiul Public	9
2.3 Evitarea Aglomerației	10
2.4 Tendințe pentru Interfața cu Utilizatorul	11
2.5 Lecții învățate	11
3 Obiectivele Cercetării	12
3.1 Servicii pentru Accesibilitate Optimă în Spațiul Public	12
3.2 Servicii pentru Reducerea Riscurilor de Pandemie	13
3.3 Rutarea Pietonilor Generată de Date și Comunitate	13
3.4 Configurare și Validare pentru un Campus Universitar	14
4 Metoda de Cercetare	14
4.1 Analiza scenariilor	15
4.2 Reprezentarea Spațiului Public	15
4.3 Design Orientat pe Servicii	16
4.4 Rutarea Multi-policies pentru Pietoni	17
4.5 Validare pentru un Campus Universitar	19
5 Contribuțiile Științifice referitoare la Orientare și Accesibilitate	19
5.1 Generarea de Drum Bazată pe Graf Ponderat Multicriterial	20
5.2 Modelul de Bază a Grafului pentru Spațiul Public	21
5.2.1 Scenariile de Navigare pentru Spațiile Publice	22
5.3 Resurse Policies pentru Rutare	23
5.3.1 Algoritmul Multi-Policy de Rutare a Pietonilor	24
5.3.2 Algoritmul de Feedback de la Comunitate	26
5.3.3 Algoritmul de Evitarea Aglomerației	27
5.3.4 Algoritmul de Evitarea Poluare	29
5.3.5 Algoritmul de Evitare a Condițiilor Meteo Nefavorabile	31

5.3.6	Algoritmul de Evitare a Riscurilor Locomotorice	32
5.4	Operațiile Generale pentru Platformă	33
5.4.1	Design pentru Faza de Bootstrap	34
5.4.2	Fazele platformei. Exemplu de Campus Universitar	34
5.5	Servicii Integrate pentru Orientare și Accesibilitate	35
5.5.1	Arhitectura Software	35
5.5.2	Servicii RESTful	37
5.5.3	Sistemul de Recomandare	40
6	Evaluarea și Validarea Serviciilor Integrate de Orientare și Accesibilitate	43
6.1	Metoda de Evaluare	43
6.1.1	Evaluarea Performanței	43
6.2	Rezultate	44
6.2.1	Exemple pentru policy locomotor pentru un campus universitar	44
6.2.2	Rezultatele de Performanță	45
6.3	Discuții	47
6.3.1	Interpretarea Rezultatelor	47
6.3.2	Cerințe de Încărcare Comparative cu Realitatea	47
7	Concluzii	48
7.1	Rezumatul Contribuțiilor Originale	49
7.2	Lista Publicațiilor	49
7.3	Perspective de Viitor	50
	Referințe	51

Lista de Tabele

Tabel 1. Descrierea Tehnică a Serviciul de Încărcare a Hărților Vizuale pentru Spațiul Public

Tabel 2. Descrierea Tehnică a Serviciului de Încărcare a Meta-datelor pentru Spațiul Public

Tabel 3. Descrierea Tehnică a Serviciului de Încărcare a Informațiilor Extra

Tabel 4. Descrierea Tehnică a Serviciului de Rutare

Tabel 5. Descrierea Tehnică a Serviciului de Direcții

Tabel 6. Descrierea Tehnică a Serviciului de Hărți

Tabel 7. Descrierea Tehnică a Serviciului Managementul Resurselor Graf

Table 8. Descrierea Tehnică a Serviciului Managementul Policy

Lista de Figuri

Figura 1. Sistemul de rutare pietonală.

Figura 2. Utilizatorul e nevoit să invoce servicii pentru fiecare nevoie.

Figura 3. Model centralizat de graf cu politici particulare.

Figura 4. Graf general ponderat pentru rutarea multicriterială.

Figura 5. Modelul Graf cu resurse *polities* al clădirilor.

Figura 6. Cladire Ipotetică și Modelul Graf respective.

Figura 7. Operațiile Administratorului și Utilizatorului cu platforma.

Figura 8. Sistemul ca Arhitectură Orientată Obiect.

Figura 9. Interfața cu Utilizatorul Implementată cu detalii de navigare.

Figura 10. Harta Visuală Implementată cu rută în spațiul public real.

Figura 11. Drumul generat dintre ED117 și ED012. Comparație între 0-policy și policy locomotor.

Figura 12. Timpul mediu de răspuns pentru 0-policy vs 1-policy vs 2-policy.

Figura 13. Ratele de eșec pentru 0-policy vs 1-policy vs 2-policy.

1 Introducere

1.1 Context și Motivație

Campusurile universitare sunt spații vaste și necunoscute pentru un nou student, diferite față de spațiile de liceu [1]. Este complex și sentimentul de a fi pentru prima dată într-un spațiu vast și complex ca un campus universitar, în special atunci când ai nevoie de indicații precise și predictibile pentru a ajunge într-o anumită încăpere dintr-o anumită clădire. Este foarte important posibilitatea orientării în aceste circumstanțe pentru că oferă un sentiment de apartenență și siguranță pentru planificarea activităților. În prezent avem multe dispozitive inteligente care pot oferi informații complexe și de acuratețe, și totuși ne putem pierde în locurile noi sau nu ne putem planifica drumul în lumea reală. Conceptul unui campus inteligent ar trebui să ofere o abordare completă pentru experiența academică, unde studenții sau alte persoane beneficiază de ajutorul mai multor tehnologii, și orientarea și navigarea pot fi oferite sub același concept. Cu toate acestea, avem să considerăm faptul că multe universități au fost construite când conceptul de “smart” (“inteligent”) nu era încă creat [2]. De aceea, arhitectura clădirilor sau conexiunilor dintre ele nu au fost create luând în considerare factori moderni (ca inclusivitate, scalabilitate cu tehnologia informației, etc.). De exemplu, cele mai impactate persoane care încearcă să navigheze o topologie complexă a clădirilor sunt persoanele cu dizabilități locomotorice.

Soluția generală este de a oferi persoanelor interesate informații sub formă de hărți și instrumente de ghidare, pentru a sprijini navigarea și luarea deciziilor. Cu toate acestea, deoarece fiecare campus universitar este un loc unic, cerințele pentru a construi o astfel de soluție trebuie să fie personalizate pentru acea universitate. Complexitatea pe care o implică un campus universitar constă în diversitatea mediilor (exterior versus interior), diferitele niveluri (scări, etaje etc.) și dificultatea unor topologii față de altele. Cercetările actuale aduc soluții la unele aspecte ale problemei, toate în direcția campusului „inteligent”, precum peisajul campusului [3], i-campus [4], oraș inteligent [5][6], navigație bazată pe evenimente [7], navigare iBeacon [8], o rețea socială mobilă într-un campus inteligent [9].

Aceste aplicații care oferă utilizatorilor indicații în timp real despre rutele lor de navigație fac parte din stadiul actual. Cu toate acestea, pietonii se bazează în mare parte pe metode tradiționale de orientare, deși ajungerea la destinație în spații publice mari, cum ar fi nodurile de transport, centrele comerciale și campusurile universitare, a devenit din ce în ce mai complicată și duce uneori la întârzieri cu efecte neplăcute. Diferitele soluții la această problemă sunt în general concentrate pe cerințe specifice, tratând separat probleme precum siguranța pietonilor, optimizarea drumului, sau rutarea personalului de urgență.

O soluție integrativă este propusă în această teză sub forma unei platforme de servicii, prin utilizarea serviciilor accesibile tuturor utilizatorilor dintr-un campus universitar. Serviciile prezentate în această teză sunt construite în întregime de autorul acestei teze (pe baza datelor tipice pentru un campus inteligent), printre acestea se numără și servicii de integrare a informațiilor cu servicii externe (acționează ca și consumatori de servicii terțe). Toate serviciile se bazează pe un

model efficient al campusului (care reprezintă topologia campusului), model care este prezentat și ca o contribuție personală în această teză, având ca scop principal flexibilitatea și versatilitatea (luând în considerare schimbările viitoare ale serviciilor și adăugarea de noi servicii).

De obicei, rolul oamenilor este de a consuma servicii din diverse sisteme ciber-fizice caracteristice clădirilor inteligente (sau unui oraș inteligent, în general). Totuși, abordările recente arată că rolul lor poate fi extins substanțial pentru a deveni contributori la sisteme, prin furnizarea de informații care pot fi citite de computer, luarea de decizii de validare, interacțiunea cu sistemul, sau chiar asumând rolul de actuatori [10]. Această prezență activă a agenților umani duce la dezvoltarea sistemelor socio-ciber-fizice (SCPS). Un astfel de exemplu este dat în [10] pentru planificarea rutelor de evacuare în situații de urgență când informațiile relevante ar trebui să fie obținute nu numai de la senzori, ci și de la persoanele din zona afectată. Informațiile de la oameni și contribuțiile lor la luarea și urmărirea deciziilor pot fi integrate în sistemul general controlat într-o mare varietate de moduri, cum ar fi intrări, perturbări, actuatori, sau feedback. Oamenii pot interveni astfel în multiple moduri în cuplarea dintre percepție și acțiune [11].

Teza este plasată în contextul furnizării de direcții de navigare și accesibilitate persoanelor în interiorul unui campus universitar. Acest lucru este important pentru a oferi vizitatorilor experiențe personalizate, luând în considerare o serie de nevoi și preferințe posibile ca parametri ale serviciilor de rutare. Se adresează în principal noilor veniți, care necesită informații de navigare asociate cu criteriile de personalizare. Exemple relevante sunt disponibile pentru modelele de campus inteligente care trebuie să fie centrate pe persoană [12], în care beneficiul pentru oameni este pus mai presus de alte motive bazate pe tehnologie. Serviciile software permit oamenilor să dobândească o percepție mai precisă a spațiului public și să rămână în contact cu alte persoane. Acest tip de percepție are asemănări cu cazul roboților care trebuie să navigheze în același mediu cu oamenii [13]; ideea este că percepția și acțiunile unei persoane sunt influențate de alți oameni din același mediu și de tehnologia încorporată în soluția de navigare. Un alt domeniu care a afectat studiul nostru este componenta de decizie inteligentă; o aplicație inteligentă de campus centrată pe persoană ar trebui să ofere utilizatorului siguranța și înțelegerea mediului înconjurător, în special în cazul navigației și accesibilității – o idee care este abordată mai larg în [14].

Am investigat soluții de ajutorare a oamenilor bazate pe nevoi personale specifice, De exemplu, imposibilitatea de a folosi scările sau pentru a trece prin uși înguste, preferința pentru un drum mai puțin aglomerat sau mai puțin poluat, necesitatea de a evita ploaia etc. Aceste preocupări de accesibilitate și siguranța epidemiologică sunt oportune și trebuie tratate într-un mod integrat. Cu toate acestea, dorim ca aceste servicii să fie conduse de comunitate, o abordare care a fost adoptată pe scară largă pentru aplicațiile de rutare a mașinilor. O hartă topologică a spațiului public conține un număr limitat de puncte de acces, ca intrări. Se poate considera aceasta ca o mulțime finită matematică, deoarece este dificil de adăugat resurse „reale” (la infinit) într-un spațiu public care este deja stabilit; De exemplu, o cameră nouă nu poate fi adăugată cu ușurință în interiorul unei clădiri, deoarece ar necesita să fie construită fizic. Cu toate acestea, amenajarea unui spațiu public poate reține o multitudine de configurații, adică o clădire poate fi redenumită, o cameră mare poate fi împărțită în încăperi mai mici (sau laboratoare), etc. Este, de asemenea, posibil să existe diverse preferințe sau limitări care mai departe complica traseul, de exemplu, cineva cu probleme

locomotorii nu poate urma aceeași cale ca ceilalți sau, în cazul pandemiei de COVID-19, oamenii trebuie să interacționeze mai puțin și să își desfășoare în continuare sarcinile zilnice [15]. Dorim să abordăm aceste criterii diverse într-un mod integrat, dar totuși să permitem flexibilitate în ceea ce privește alegerile oamenilor. Acest lucru este posibil pe baza unei modelări graf de uz general care este adaptată dinamic în funcție de diverse considerații precum: luarea în considerare a feedback-ului de la alți pietoni care merg în același spațiu public; încercarea de a evita zonele aglomerate; evitarea oricăror zone poluate pe baza datelor provenite de la senzori; evitarea zonelor afectate de vreme solicitând date în timp real de la API-urile meteo; evitarea riscurilor de tip locomotoric pentru persoanele cu astfel de probleme [16][17]. Acestea corespund resurselor de tip *policies* care pot fi aplicate serviciilor de rutare; atunci când mulți pietoni solicită direcții de rutare în mai multe moduri (cu *0-policy*, *1-policy*, *2-policy* etc.), serviciul recalculează ponderile din interiorul grafului pentru fiecare invocare, astfel încât fiecare persoană are o un drum în spațiul public special adaptat pentru nevoile sale speciale.

2 Analiza Stadiului Actual

În această secțiune, am inclus o analiză a stadiului tehnic în toate domeniile conexe acestei teze. Acest capitol își propune să pună în perspectivă ceea ce au realizat alții și cum putem îmbunătăți ce observăm în cercetările lor, sau cum putem folosi această cercetare în părți foarte specifice din această teză.

2.1 Planificarea drumului

Planificarea drumului se face între locații care nu au aceeași adresă. Există lucrări științifice semnificative cu privire la algoritmi de navigare folosind GPS (Global Positioning System); majoritatea se bazează pe analiza și prelucrarea datelor grafice pentru hărți, dar și a altor date în timp real referitoare la aglomerații, ambuteiaje, accidente și alte evenimente care pot influența traseul selectat. Sunt algoritmi multi-obiectiv, care încearcă să găsească un traseu care nu doar să minimizeze distanța, ci și timpul, sau un traseu selectat pentru a parcurge anumite puncte de pe hartă (obiective turistice sau alte locații). Algoritmi din acest domeniu sunt orientați spre patru tipuri de călătorie: vehicule cu roți (mașini, camioane) cu algoritmi de localizare a traseului bazați pe graf [18], cale ferată (trenuri), care sunt mai simplu de rezolvat și bazate pe o abordare a teoriei grafurilor [19], călătoria pe mare (nave) bazată pe un graf de vizibilitate la scară multiplă [20] și călătoria pe jos (pietoni) bazată pe graful de orientare pe principii cognitive [21].

2.2 Mobilitate și Accesibilitate în Spațiul Public

Navigare incluzivă pentru persoanele interesate. Pentru mobilitate și orientare în interiorul unui spațiu public (în special într-un campus), există mai multe modalități de a merge dintr-un loc în altul: folosirea bicicletelor, trotinetelor electrice, vehiculelor electrice personale [22], sau călătorind ca pieton. O altă problemă specifică este generarea de rute pentru persoanele cu dizabilități, de exemplu, persoanele cu deficiențe de vedere [23].

Graf ierarhic bazat pe vizibilitate în interior. Bazat pe un graf ierarhic bazat pe vizibilitate în interior (HiVG) pentru ghidarea navigației în clădiri cu mai multe etaje [24], WRLD este un exemplu de soluție software specializată pentru navigare și orientare (în interior și în aer liber) în medii reale [25].

Poziționare în interior. Sistemul de poziționare în interior poate fi implementat utilizând un algoritm *K-nearest-neighbor* față de planul clădirii [26], [27]. O altă soluție implică receptori Bluetooth și Beacons [28].

Tehnologia IoT. Tehnologia Long Range (LoRa) [29] poate fi utilizată pentru a implementa o soluție IoT bazată pe variabile de mediu. Tehnologia IoT este utilizată în mai multe aplicații conexe, așa cum este rezumat în [30].

Infrastructura de conectivitate. Ideea unui campus universitar inteligent este abordată de Fortes et al. ca „Smart Tree” [31] [32].

Interacțiuni cu dispozitivele mobile. Interfața pentru utilizator este pe telefoane mobile de cele mai multe ori, dar în unele situații, pentru a reduce costurile, un set de indicatori încorporați poate fi utilizat pentru navigația în aer liber sau în interior [33], (pentru nou-veniți) [34].

2.3 Evitarea Aglomerației

Un subiect mai recent este legat de controlul bolilor, în special de pandemia COVID-19. Este deosebit de relevant pentru spațiile publice, unde este foarte probabil să se formeze mulțimi. Această problemă introduce un set suplimentar de constrângeri în ceea ce privește planificarea și urmărirea traseului, în timp ce nu modifică în mod fundamental algoritmi și metodele utilizate în planificarea rutelor în spații publice precum campusuri. Desigur, atunci când se încearcă controlul răspândirii unei boli trebuie să se impună un set de restricții în trafic; în acest caz, se face referire la mobilitatea pietonilor și nu la vehicule.

Campusurile universitare sunt locuri în care transmiterea COVID-19 poate avea un impact mare asupra părților interesate datorită mobilității persoanelor afectate. Există o dependență între dinamica pietonilor și epidemiologie [35], transmiterea de aerosoli [36], modelarea matematică a răspândirii bolii [37] și riscurile expunerii studenților la COVID-19 într-o clădire universitară [38] sau în alte zone urbane [39].

Din punctul de vedere al mobilității și al planificării rutelor, controlul bolilor influențează modul în care astfel de algoritmi definesc rutele, de exemplu, prin indicarea diferitelor rute pentru pietoni în interiorul clădirilor [40], [41].

Proiectarea aplicațiilor software și hardware poate oferi strategii privind modul în care senzorii de aer pot fi utilizați pentru a indica cu exactitate riscurile COVID-19 [42]. Bidila et al. precizați clar că este ușor de încălcat procedurile (fără intenție) și că în absența testării (pentru oameni și suprafață) riscul de contaminare este dificil de calculat. Aceștia propun o soluție low-cost și implementabilă (IoT cu un sistem dual de monitorizare a calității aerului) pentru a avertiza oamenii când densitatea lor locală devine un factor de risc.

2.4 Tendințe pentru Interfața cu Utilizatorul

În această subsecțiune, prezentăm cercetările efectuate legate de tendințele pentru interfața cu utilizatorul, pe baza cercetărilor noastre pe următoarele subiecte, remarcăm:

- Interfața grafică utilizator nu este întotdeauna superioară interfeței utilizator de tip text [43]
- Interfață pentru navigare – re poziționarea continuă a utilizatorilor către următorul punct intermediar [44]
- Spațiu public incluziv social – adaptarea spațiului la exclusivitate opusă [45]
- Concepte și relații spațiale – proiectarea spațiului centrată pe utilizator [46]
- GPS vs indicații de la oameni – adesea complementare [47]
- Ordinea informațiilor hărții direcționale – pentru a simplifica asimilarea informațiilor [48]
- Sistem de baze de date relaționale pentru imagini [49]
- API-uri – reutilizare și creșterea productivității [50]

2.5 Lecții învățate

În cercetarea noastră, am urmărit modul în care alții au atins obiective similare cu această teză. De aceea, principalele subiecte pe care le-am cercetat sunt planificarea rutelor, mobilitatea în spațiile publice, evitarea aglomerației și experiența utilizatorului. Planificarea rutelor a fost cercetată pentru a obține un punct de vedere la nivel înalt față de traseul la nivel global, nu doar traseul pietonal. Mobilitatea în spațiile publice a fost cercetată cu accent pe navigația pietonală și modul în care alții folosesc diverse tehnologii pentru a realiza navigarea în astfel de spații. Evitarea aglomerației a fost cercetată ca o extensie și o oportunitate de a face utilizatorii să se simtă în siguranță în perioadele de pandemie; Este o extensie deoarece una dintre contribuțiile originale ale acestei teze este un mecanism de evitare a aglomerației într-un spațiu public. Experiența utilizatorului a fost cercetată pentru a înțelege cum li se pot oferi utilizatorilor toate detaliile și să consume serviciile pe care le-am dezvoltat eficient pentru această teză.

Rezultatul acestei analize este că un sistem care utilizează tehnologii foarte diferite (în mare parte tehnologii inteligente) nu este doar dificil de implementat în lumea reală (oferind șanse mici pentru o implementare reală), ci și dificil de implementat și de folosit în spații publice vaste. precum campusurile universitare pentru perioade extinse. Mai mult, utilizatorul și administratorul spațiului public necesită acțiuni complexe de folosit, respectiv de a produce date.

Parametrii pe care i-am dedus din cercetarea noastră pentru a-i folosi pentru domeniul nostru global de teză, este că tehnologia ar trebui să fie consumată de utilizatori mai eficient, sub formele de servicii, în special servicii web, pe interfețe vizuale minime și ușor de înțeles. Aceste servicii ar trebui să aibă opțiunea de a personaliza experiența pentru fiecare utilizator, iar aceste servicii ar trebui să fie mobile, astfel accesibile prin intermediul datelor mobile de pe dispozitivele personale.

Decizia pentru contribuțiile originale ale acestei teze a fost crearea unei interfețe cu utilizatorul care să ofere utilizatorului imagini vizuale ale aspectului spațiului public care să conțină indicații

generale pentru parcurgere. Aceste imagini sunt create pe baza imaginilor încărcate de un administrator în platformă și, pe deasupra, platforma va crea direcții vizuale pentru utilizator, personalizate după nevoile acestuia. Imaginile vor fi stocate pe un server și vor putea fi preluate de utilizator în orice moment. Cu toate acestea, utilizatorului i se vor oferi indicații simple în interfața cu utilizatorul (text intuitiv pentru a compensa partea vizuală a navigării).

3 Obiectivele Cercetării

În această secțiune, discutăm problematica cercetării, rezultatele dorite și specificațiile de dezvoltare pentru obiectivele noastre, care la un loc realizează obiectivul global al tezei.

3.1 Servicii pentru Accesibilitate Optimă în Spațiul Public

Universitățile și alte spații publice sunt locuri cu multe complexități arhitecturale. Prin urmare, în aceste spații publice pot fi percepute diverse complicații de către participanți. Principala dificultate pentru utilizatorul final este complexitatea navigării. Navigarea într-un spațiu public înseamnă parcurgerea acestuia pentru a ajunge la o destinație. Stresul timpului limitat este adăugat la complexitatea navigației. Stresul de timp limitat este crescut dacă sunt prezenți alți factori (externi sau interni). Un factor extern ar putea fi faptul că spațiul public este dificil de navigat, deoarece are un aspect arhitectural complex. Un factor intern ar putea fi faptul că utilizatorul suferă de probleme locomotorii. Dacă aceste complicații nu sunt tratate, utilizatorul final se va simți stresat și va avea o experiență generală neplăcută.

Un utilizator cu probleme locomotorii care trebuie să traverseze spațiul public, are alte nevoi decât un utilizator care nu suferă de acestea. Deci, acest utilizator trebuie să fie furnizat cu alte direcții de navigare față de un utilizator care nu suferă de acestea. Dacă nu facem această distincție, utilizatorul care suferă de probleme locomotorii va avea o experiență neplăcută sau, mai rău, siguranța acestuia ar fi compromisă. Fiecare utilizator trebuie tratat cu egalitate și incluziune.

O altă problemă atunci când se oferă servicii de navigație utilizatorilor este pe ce canal sunt livrate serviciile. Datorită naturii serviciilor care trebuie furnizate utilizatorului final, aceste soluții trebuie accesate ușor și în orice moment.

Problema completă care trebuie rezolvată este: asigurarea participanților în spațiul public cu servicii care să țină cont de posibilitatea ca participanții să aibă probleme locomotorii.

Soluțiile sunt: servicii de navigație care pot fi accesate ușor și în orice moment de către participanți și oferă soluții personalizate pentru nevoile participanților.

Specificațiile sunt: să se ofere aceste servicii pe telefoane mobile conectate cu date mobile.

De aceea, utilizatorul va acționa ca un client, iar serviciile vor fi furnizate de un server care este o entitate diferită de client. Am construit soluția noastră în jurul experienței cu telefoanele mobile și datele mobile, deoarece acestea sunt populare, utilizatorii știu deja cum să le opereze, iar serviciile pe care le oferim nu trebuie să includă instruire extinsă pentru utilizator.

3.2 Servicii pentru Reducerea Riscurilor de Pandemie

Spațiile publice sunt de obicei vizitate de mulțimi mari. Dar mulțimile mari sunt susceptibile la pandemii, ca cea prin care am trecut, pandemia SARS-COV-2.

Mulțimile dintr-un spațiu public se pot crea în mod spontan, în special într-o clădire mică, iar această mulțime ar oferi o probabilitate mai mare de transmitere a virusurilor. De asemenea, ar avea un impact asupra navigației libere dintr-un punct al spațiului public în altul.

Problema completă este: extinderea serviciilor de navigație pentru a lua în considerare impactul aglomerației în spațiul public și oferirea de rute alternative pentru ca utilizatorii să ajungă la destinație.

Soluțiile trebuie să ferească utilizatorii de potențialele căi care conțin aglomerație, reducând astfel riscurile de pandemie. Căile puse la dispoziție de serviciul de navigație în aceste cazuri încearcă să ghideze oamenii prin porțiuni din spațiul public pe care nu au fost vizitate în ultimul timp (dacă nu au fost vizitate în ultimul timp, există o șansă mai mică ca mulțimi să se acumuleze în acele zone în mod spontan).

Specificațiile trebuie să extindă funcționalitatea serviciului de navigație. Un modul care poate fi optat la alegerea utilizatorului, astfel condus de utilizator, un modul care crește automat factorii de aglomerație în interiorul unui spațiu public. Actualizarea automată a factorilor de aglomerație este utilizată pentru a arăta porțiunile din spațiul public care pot avea potențialul de a crea aglomerații mari, unde riscurile unei pandemii cresc substanțial.

3.3 Rutarea Pietonilor Generată de Date și Comunitate

Participanții într-un spațiu public pot fi priviți ca o comunitate. Această comunitate este diversă și fiecare participant are nevoi care sunt diferite de cele ale altor participanți. Este posibil ca spațiul public să nu fie actualizat pentru fiecare dintre nevoile utilizatorului, dar utilizatorii pot conduce reprezentarea spațiului public. De exemplu, utilizatorii ar trebui să aibă voie să voteze dacă o parte din spațiul public respectă anumite criterii stabilite de comunitate.

Datele pentru reprezentarea spațiului public pot fi încărcate și din alte surse. Realitatea spațiului public nu se schimbă, dar împrejurările se schimbă. Un exemplu ar fi condițiile meteorologice pentru spațiul public. În acest caz, spațiul public nu se schimbă, aspectul arhitectural este în continuare același, dar unele caracteristici arhitecturale pot fi afectate de vreme. Un alt exemplu ar fi calitatea aerului din interiorul spațiului public.

Problema completă este de a construi soluții pentru adaptarea datelor externe și a datelor de la comunitate pentru a extinde experiența utilizatorului în spațiul public.

Soluțiile sunt servicii care permit integrări în platformă, pentru extinderea reprezentării spațiului public.

Specificațiile sunt de a permite administratorilor să integreze date din surse externe și să ajute utilizatorii finali. Utilizatorul final invocă serviciile de navigare de bază opțiunilor care indică date externe sau interne.

3.4 Configurare și Validare pentru un Campus Universitar

Ca exemplu de spațiu public, alegem un campus universitar pentru implementarea conceptului personal. Această alegere vine ca rezultat al unei analize în funcție de doi factori: configurație și validare. Având în vedere înclinațiile noastre academice, putem modela serviciile pentru un campus universitar într-o manieră eficientă. De asemenea, un campus universitar poate include diverși participanți, deci nevoi multiple. Structura campusului universitar este în mare parte statică, ceea ce înseamnă că proprietățile arhitecturale nu se modifică dinamic în funcție de anumite evenimente. Un campus universitar poate avea mai multe clădiri și, cel mai probabil, diferiți administratori.

Problema de configurare completă este crearea unei configurații extensibile a campusului universitar care să poată sprijini nevoile utilizatorilor.

Soluția de configurare este de a permite administratorilor să decidă granularitatea detaliilor arhitecturale pe care le va avea modelul pentru campusul universitar atunci când este utilizat de utilizatorii finali.

Specificațiile de configurare sunt pentru a permite administratorului să încarce detalii de configurare cu privire la campusul universitar real și detalii de configurare care trebuie să respecte liniile directoare susținute de platformă.

Problema completă de validare: configurația spațiului public este valabilă dacă utilizatorul poate folosi informațiile furnizate de platformă, pentru a lua decizii în interiorul campusului universitar.

Soluția de validare: să construiască pe deasupra configurației inițiate de administrator în pasul de configurare

Specificația de validare: pentru a crea servicii care sunt decuplate de codul de configurare și obiectul spațiului public.

4 Metoda de Cercetare

În această secțiune, discutăm scenarii generale pe care le-am analizat pentru a determina forma finală a serviciilor pe care le oferim utilizatorului final. Discutăm despre abordarea orientată către servicii pe care am ales-o pentru a realiza proiectarea acestor servicii, precum și implementarea lor. Discutăm despre traseul pietonal cu mai multe resurse de tip *policies*, unde explicăm ce înseamnă *policy* pentru platforma noastră și cum ajută utilizatorul să aibă o experiență personalizată folosind soluția de navigare. Discutăm despre validarea contribuțiilor acestei teze pentru un campus universitar și de ce am ales să ne concentrăm pe această temă pentru această teză și pe implementarea conceptului prezentată mai târziu în această teză.

4.1 Analiza scenariilor

Oamenii navighează cu ușurință în spațiile publice cu care sunt foarte familiarizați. Cu toate acestea, atunci când se îndepărtează această familiaritate, navigarea și orientarea devin complexe. Pentru a înțelege scenariile cu care se confruntă o persoană într-un spațiu public care nu îi este bine cunoscut, trebuie să analizăm în primul rând cum arată un spațiu public și apoi care este scopul acelui spațiu public.

Scenarii de precondiții de navigare. Pentru a naviga într-un spațiu public, oamenii trebuie să știe unde merg (să își cunoască destinația/să recunoască că au ajuns la destinație) și să știe ce direcție ar trebui să aibă (să fie îndreptați în direcția corectă/să recunoască punctele intermediare ulterioare de-a lungul punctelor de referință de interes).

Scenarii de complexitate a navigației. Complexitatea spațiului public dictează complexitatea navigației pentru utilizator. A nu ști cum să navighezi în diferite tipuri de complexități ar putea implica un cost suplimentar (timp pierdut, sau pierderea orientării); Acesta este motivul pentru care recunoașterea destinației și direcția contează în designul platformei noastre.

Scenarii de complexitate personalizate. Nevoile diferiților utilizatori implică alte tipuri de dificultăți pe care o platformă de navigare ar trebui să le ia în considerare. Aici pot fi menționate mai multe scenarii, cum ar fi complexitatea locomotorie (care ar trebui să fie cel mai important aspect pentru o persoană cu probleme locomotorii), complexitatea de evitare a mulțimii (în cazul în care utilizatorii doresc să evite aglomerațiile care sunt create fără un model cunoscut anterior în interiorul unui public), complexitatea informațiilor de la părți terțe (în cazul în care utilizatorii doresc să obțină o recomandare corectă pentru navigarea lor, luând în considerare datele care provin de la furnizori externi, de exemplu, informații despre vreme).

4.2 Reprezentarea Spațiului Public

În centrul platformei descrise în această teză se află modelarea spațiului public. Acest model trebuie să exprime realitatea și să ofere aceste informații pentru construirea serviciilor necesare pentru utilizatori. Aceste informații trebuie să fie ușor stocate, întreținute și utilizate pentru algoritmi de navigare. Acestea sunt motivele pentru care am ales modelul nostru ca spațiul public să fie un graf ponderat de uz general.

- Un graf ponderat de uz general este o structură care este creată în jurul ideii de vecini.
- Un graf ponderat de uz general are algoritmi standardizați pentru interogarea unei căi în interiorul acestuia. Acest lucru implică faptul că algoritmi sunt bine cunoscuți și testați, De aceea, utilizatorilor finali li se vor furniza în mod fiabil informațiile de navigare. Un graf ponderat de uz general are mai multe moduri de configurație: noduri, margini și ponderi.
- Un graf ponderat de uz general poate fi creat apoi din mai multe puncte de date, în implementarea conceptului prezentat în capitolele următoare descriem cum din date statice simple reprezentate într-o foaie Excel într-un anumit format, am construit modelul grafului de bază, model utilizat de toate serviciile platformei.

Granularitatea reprezentării grafului poate fi stabilită de nevoile spațiului public sau extinse în viitor. În implementarea conceptului am ales granularitatea modelului grafului de bază după cum urmează:

- nodurile grafului reprezintă fiecare topologie de arhitectură ca încăperi, holuri, scări etc.
- marginile dintre noduri sunt conexiuni abstracte directe (pot fi citite ca un nod este vecin cu altul)
- ponderile grafului reprezintă nevoile utilizatorilor (sunt cuantificări actualizabile care reprezintă ceea ce are nevoie utilizatorul din navigare)

4.3 Design Orientat pe Servicii

Designul nostru se bazează pe expunerea serviciilor de orientare, pentru a ajuta oamenii să aleagă traseul cel mai potrivit pentru a merge printr-un spațiu public și a ajunge la destinația dorită. Partea centrală a sistemului este capacitatea de a modela spațiul din lumea reală într-o structură asemănătoare unui graf, pentru a permite adaptabilitatea algoritmilor de rutare, așa cum este detaliat de Costa și colab. în [51]. Traseul adoptat de fiecare persoană influențează fluxul pietonal general, iar serviciile de rutare țin cont și de feedback-ul dat de comunitatea de persoane care beneficiază de astfel de servicii (Figura 1).

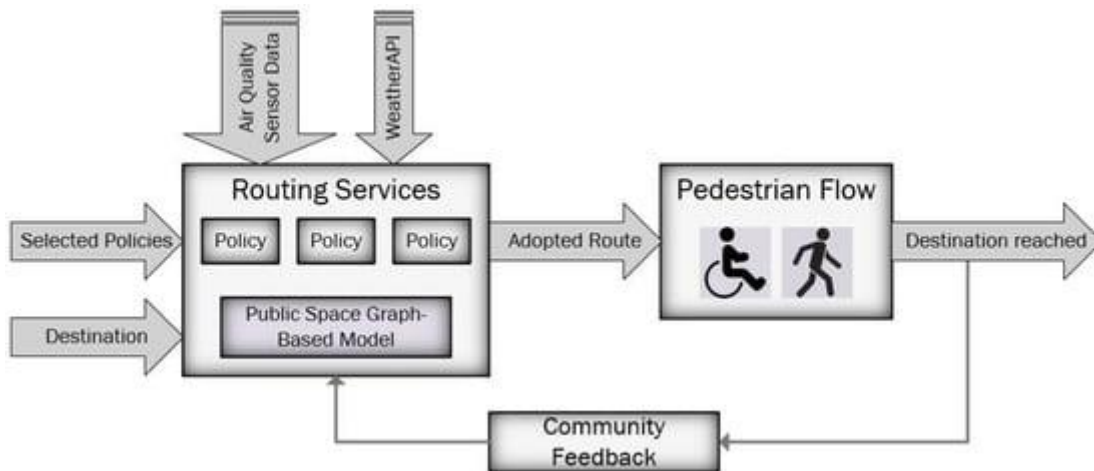


Figura 1. Sistemul de rutare pietonală.

Serviciul de rutare ia ca parametri de intrare nodurile generate de administratori și parametri externi, cum ar fi vremea, riscul epidemiologic, problemele personale precum dificultățile locomotorii, sau situațiile de urgențe. Din acest motiv, nodurile trebuie create în conformitate cu un model care să susțină astfel de resurse *policies*; prin urmare, aceste *policies* trebuie să fie cunoscute înainte ca resursele reale din lumea reală să fie modelate ca noduri. Pentru ca sistemul să fie robust, aceste *policies* reprezintă o resursă în interiorul sistemului, expusă de alte servicii; ar

trebuie să fie o sursă de conformitate pentru nodurile care sunt create prin invocarea lor. Serviciile sunt scrise într-o arhitectură client-server care expun punctele finale HTTP.

4.4 Rutarea Multi-policies pentru Pietoni

Designul nostru se bazează pe ideea că un spațiu public care oferă o platformă de navigare și orientare trebuie să ia în considerare nevoile utilizatorilor finali. Iar nevoile utilizatorilor sunt tehnic infinite. Oricât de mult putem presupune nevoile utilizatorului, adevărul va fi întotdeauna deținut de utilizatorul final.

Modele multiple de grafuri. O modalitate de a satisface toate nevoile este de a crea mai multe modele de grafuri ale spațiului public, mai multe modele pentru ponderile din interiorul grafurilor și/sau mai mulți algoritmi care ar putea transforma toate datele într-o secvență ordonată care poate fi furnizată utilizatorului pentru navigare. Această metodă ar include o mulțime de structuri și costuri de stocare, mai multă întreținere pe modele și mai multe informații specializate pe care un administrator ar trebui să le mențină în continuare, deci mai multă pregătire tehnică a personalului. Această abordare (Figura 2) prezintă o problemă cu privire la modul de rezolvare a integrării tuturor modelelor de grafuri diferite (care reprezintă nevoile utilizatorilor), pentru cazul unui utilizator care are nevoi multiple (de exemplu un utilizator dorește să evite aglomerația dar și riscurile locomotorice).

Model de graf centralizat cu ponderi generate dinamic. O altă modalitate de a satisface toate nevoile este de a crea un model de graf de bază pentru a reprezenta mai întâi un spațiu public și pentru a permite ponderilor să reprezinte nevoile utilizatorilor. O pondere mare pe o margine a grafului ar însemna un cost mai mare pentru utilizator pentru a utiliza în lumea reală acea porțiune a spațiului public. De aceea, ponderile sunt generate dinamic și conduse de utilizator. Adică utilizatorul selectează când dorește să călătorească în spațiul public luând în considerare mai multe nevoi. Selectarea în avans a nevoilor multiple (și nu a modelelor diferite de bază pentru nevoie diferite) este o abordare mai scalabilă și mai modulară decât cea prezentată anterior. Această abordare seamănă cu un set de filtre multiple pe o căutare (asemănătoare unei baze de date).

Ultima abordare, bazată pe un graf cu ponderi generate dinamic, a fost aplicată în această teză (Figura 3). Modul în care permitem nevoilor multiple ale utilizatorului să existe în platformă este prin crearea de *policies*. *Policies* sunt resurse care sunt stocate în platformă și descriu modul în care ponderile grafului sunt afectate dacă utilizatorul final invocă aceleași *policies*. Le-am numit *policies* pentru că au forma unui contract, un contract care este creat de o parte (administrator), un contract care are puncte de interes de unde obține datele care reprezintă nevoile utilizatorului, un contract care descrie modul în care prelucrează datele pentru a fi înțelese de obiectul de bază, care este modelul graf și algoritmul de navigare, și un contract care poate fi invocat de utilizatorul final să fie furnizat cu beneficiile contractului respectiv.

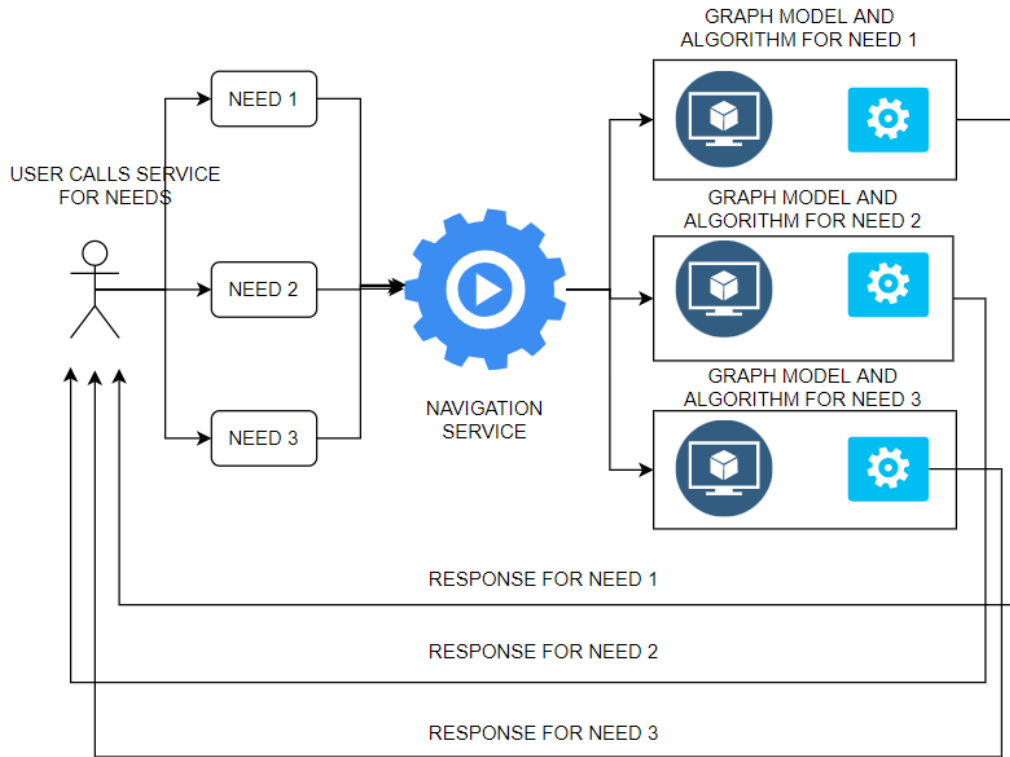


Figura 2. Utilizatorul e nevoit să invoce servicii pentru fiecare nevoie.

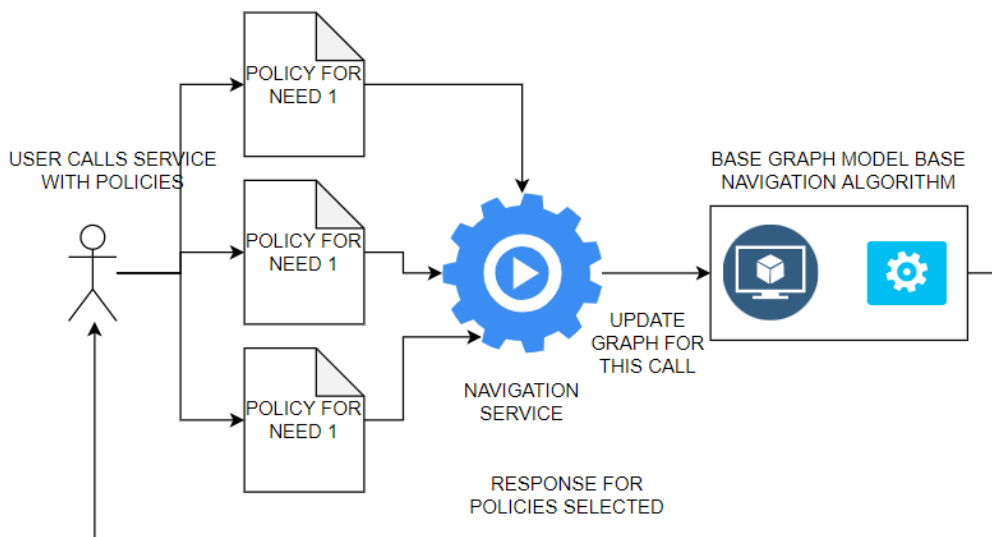


Figura 3. Model centralizat de graf cu politici particulare.

4.5 Validare pentru un Campus Universitar

Există multe tipuri de spații publice care ar fi putut fi alese pentru implementarea noastră de concept pentru a valida designul sistemului nostru, dar considerăm alegerea unui campus universitar, pentru a verifica și a valida sistemul nostru, mai eficientă din mai multe considerente.

Campusurile universitare nu sunt la fel de închise ca spațiile publice (secția de poliție/spital), deoarece încă încurajează navigarea liberă, nu servesc scopului de divertisment de masă (mall) și participanții dintr-un campus universitar nu au luxul timpului. Campusurile universitare sunt zone vaste în care participanții cunosc de obicei doar o parte din ele. Campusurile universitare sunt locuri unde un participant ajunge cu o sarcină definită și limitată în timp. Aceasta implică o anumită urgență în rezolvarea sarcinii.

Campusurile universitare combină diferite categorii de topologii arhitecturale care prezintă interes pentru participanți. Aceste topologii arhitecturale ar putea ușura sau îngreuna navigarea participanților. Prin urmare, este necesar să se valideze implementarea conceptului pentru un campus universitar pentru preferințele utilizatorilor.

Participanții într-un campus universitar au sarcini diferite zi de zi și nevoi diferite care sunt fie de natură personală, fie de natură obiectivă (create chiar de campusul universitar). Aceste nevoi sunt rezolvate cu ajutorul resurselor *policies*, astfel încât fiecare participant să se simtă inclus în campusul universitar și să aibă o experiență plăcută în traversarea acestui spațiu public.

5 Contribuțiile Științifice referitoare la Orientare și Accesibilitate

În această secțiune, discutăm contribuțiile științifice construite pe baza detaliilor prezentate în secțiunile anterioare:

- am ales un model care acceptă găsirea rutelor pe baza grafurilor ponderate multicriteriale.
- modelăm spațiul public sub formă de graf, iar acest lucru este benefic pentru găsirea rutei.
- discutăm despre integrarea mai multor criterii în acest graf, cum modelăm criteriile pentru a satisface nevoile utilizatorilor și cum se potrivește cu un scenariu de navigare în interiorul unui spațiu public.
- Resursele *policies* de rutare care sunt derivate din nevoile din lumea reală ale utilizatorilor și modul în care acestea pot extinde scopul platformei.
- fluxul general de lucru din cadrul acestei platforme, unde descriem modul în care administratorii și utilizatorii finali interacționează cu platforma.
- arhitectura generală care susține designul conturat și modul în care serviciile furnizează și consumă date în interiorul platformei, precum și serviciul de vizualizare pentru utilizatorii finali.

5.1 Generarea de Drum Bazată pe Graf Ponderat Multicriterial

Alegem să reprezentăm un spațiu public din lumea reală sub formă de graf, deoarece teoria grafurilor oferă deja algoritmi care urmăresc comportamentul în lumea reală, unde navigarea de la locația curentă (nodul sursă) către o destinație (nodul destinație) este un set de pași care poate fi modelat prin secvența de noduri ale grafului.

Un algoritm de rutare cu un singur criteriu funcționează pe baza unui graf ponderat care ia în considerare doar consumul necesar pentru a traversa resursele (nodurile). Să analizăm un exemplu de graf ponderat, în care A, B și C sunt nodurile și X, Y și Z costurile de navigare de la A la B, de la B la C și, respectiv, de la A la C. Dacă pe ruta $A \rightarrow B \rightarrow C$, costul este $X + Y$, și este mai economic decât alegerea $A \rightarrow C$ care costă Z, atunci ruta aleasă ar trebui să fie $A \rightarrow B \rightarrow C$.

Această abordare poate fi folosită ca algoritm multicriterial. Schimbările către algoritmul multicriterial trebuie să se reflecte atât asupra algoritmului, cât și asupra reprezentării; nodurile ar trebui să aibă proprietăți speciale pe care algoritmul ar trebui să le ia în considerare. De aceea, ponderile dependențelor (muchiiilor) devin funcții ale proprietăților nodurilor conectate, așa cum este reprezentat în Figura 4. Unele proprietăți ale nodurilor ar putea să nu fie interesante pentru anumite persoane. Nodurile trebuie să aibă mai multe proprietăți care reprezintă interesele anumitor utilizatori; astfel, precum un abonament la buletin informativ, toți oamenii au interese, dar adesea sunt diferite de la unul la altul și nu ar trebui să afecteze interesele altora. Un utilizator poate „apela” algoritmul multicriterial în funcție de interesele sale particulare (proprietățile nodurilor); De aceea, algoritmul multicriterial calculează rezultatul într-o manieră personalizată.

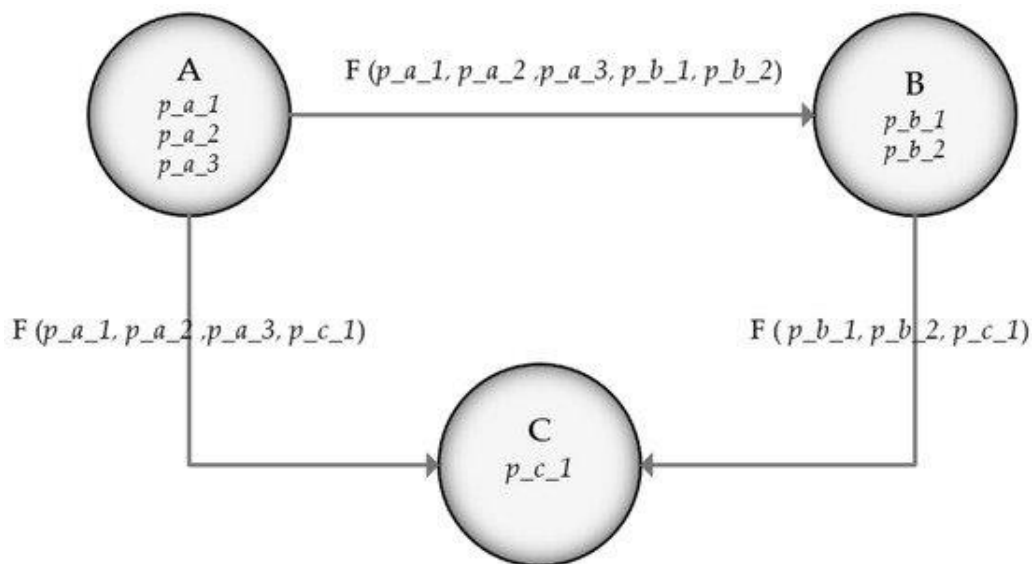


Figura 4. Graf general ponderat pentru rutarea multicriterială.

Într-un graf ponderat în general cu trei noduri, fiecare conexiune între noduri are un cost cunoscut sub numele de pondere în teoria grafurilor. În abordarea multicriterială, se poate presupune că există un cost al funcției pentru calcularea ponderii. Figura 4 prezintă un graf ponderat cu trei noduri, într-o configurație de conexiune similară ca în exemplul de mai sus. Diferența este că și costul traversării nodurilor nu mai este o informație independentă. Fiecare nod are proprietăți desemnate. Aceste proprietăți de pe fiecare nod reprezintă cât de dificil este să acceseze acest nod. Pentru a trece de la un nod la altul, trebuie luată în considerare o funcție de cost care integrează proprietățile celor două noduri. Această abordare ne permite să reprezentăm cât de dificil este să acceseze o resursă și să extragem costul dintre două astfel de resurse. Considerăm că fiecare dintre aceste proprietăți are o valoare numerică, iar funcția F este funcția SUM (Sumă), adică adună toate proprietățile și schimbă practic ponderile grafului. Dacă aceste proprietăți se modifică dinamic, ponderile grafului se schimbă și ele dinamic. Dacă abstractizăm aceste proprietăți sub anumite categorii (sau criterii), utilizatorii pot apela algoritmul de rutare multicriterial care oferă acești parametri. Aceasta înseamnă că utilizatorul care dorește să navigheze în clădire/campus/spațiul public poate minimiza impactul pentru criteriile date (deoarece algoritmul de bază minimizează costul de navigare).

5.2 Modelul de Bază a Grafului pentru Spațiul Public

Modelul graf al clădirii, reprezentat în UML (Unified Modeling Language), este prezentat în Figura 5. Spațiul public include mai multe clădiri și mai multe conexiuni între clădirile care se află la fața locului. Atributele unei clădiri includ numele spațiului (adică numele clădirii), numărul de noduri (corespunzător caracteristicilor arhitecturale) și numărul de conexiuni între nodurile din aceeași clădire. O conexiune are două proprietăți (sursă și destinație) pentru a reprezenta vecinii unei clădiri sau nod. Fiecare nod are proprietăți de nod (a se vedea „*” pentru a afișa multiplicitatea > 0 în Figura 5), inclusiv tipul de nod (hol, scări, cameră etc.) și *policies*. O resursă *policies* este definită prin nume, sursa de date (de unde provin datele, pentru a construi particularitățile legate de *policy*) și „dataManipulation” (o procedură de interacțiune cu datele din sursa de date).

După cum s-a explicat anterior, proprietățile unui nod ar trebui grupate sub criterii comune, pentru a oferi utilizatorului posibilitatea de a invoca aceste criterii atunci când apelează algoritmul de rutare (altfel algoritmul nu poate centraliza informațiile și nu poate crea ponderi pe graf); De aceea, ar trebui să fie folosită o abstractizare specifică pentru a descrie această cerință. În acest scop, definim o serie de resurse *policies* și care să fie disponibile pietonului care dorește să obțină indicații de traseu pentru a ajunge la destinația dorită. *Policies* sunt definite în abordarea noastră ca reguli cu privire la modul în care algoritmul multicriterial ar trebui să calculeze costurile pe un graf, unde să recupereze informațiile de stocare despre noduri și cum ar trebui să fie utilizate informațiile de la noduri în timp ce algoritmul multicriterial rulează.

Astfel de *policies* sunt aplicate întregului graf și apoi activează anumite proprietăți pentru fiecare nod; atributul „nodeProperties” din Figura 5 poate fi populat cu date referitoare la resursele *policies* alese, pentru a fi apoi luate în considerare la costul fiecărei conexiuni, prin utilizarea funcției SUM (Sumă) pe toate datele cuantificate din datele de la resursele *policies*. În acest fel, algoritmul minimizează costul total și oferă ruta cu cel mai mic cost. Cu toate acestea, alte proprietăți ale

nodurilor pot fi populate cu date de acces care vor fi furnizate utilizatorului ca rezultat al generatorului de rută.

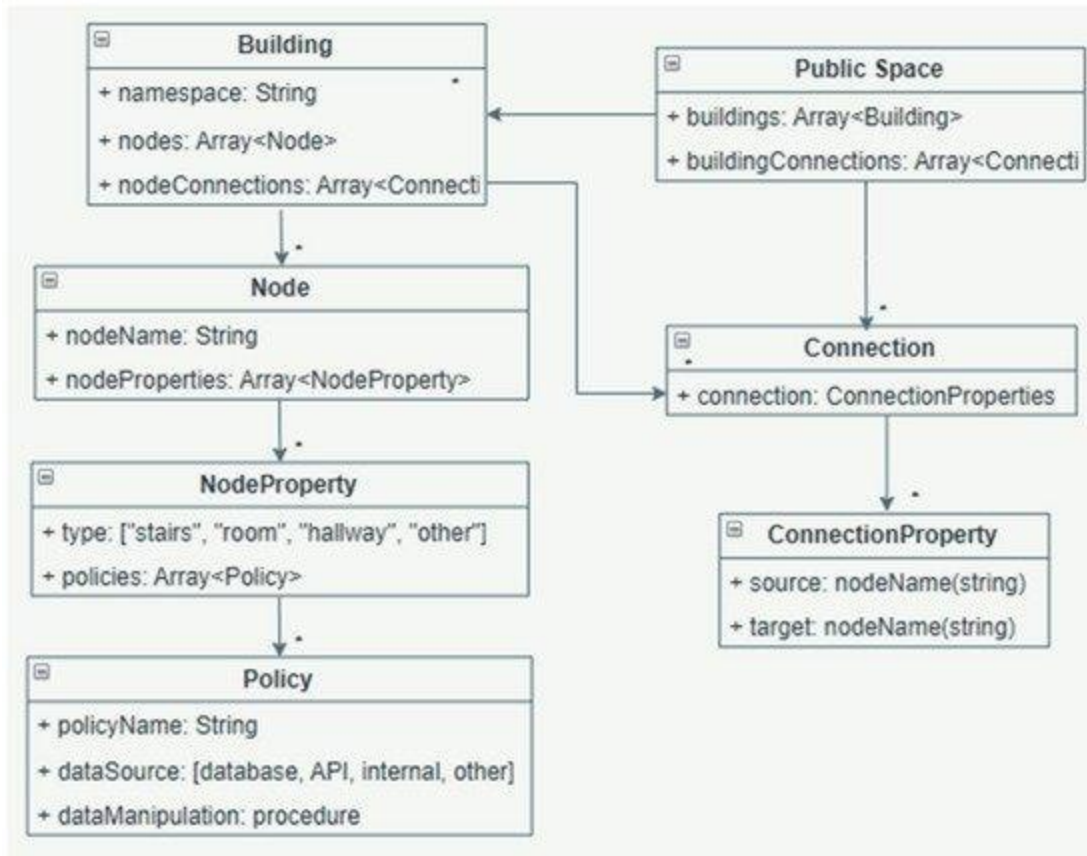


Figura 5. Modelul Graf cu resurse *policies* al clădirilor.

5.2.1 Scenariile de Navigare pentru Spațiile Publice

Pentru a analiza mai multe scenarii, ne vom referi la Figura 6. Această figură prezintă planul unei clădiri ipotetice cu 2 intrări conectate la exteriorul clădirii. Intrările sunt denumite „Entrance 1” și „Entrance 2”. „Hallway 1” este conectat la ambele intrări. „Room A” și „Room B” sunt conectate la „Hallway 1”, iar cele două camere nu sunt conectate. Pentru acest caz, Este nevoie de un model pentru realitatea spațiului public. Un model care poate fi ușor de extins dacă o caracteristică arhitecturală ar fi creată sau distrusă în realitate. Și un model care ar putea rula algoritmi de navigare pentru utilizator. Un astfel de model este un graf. Un graf este o reprezentare eficientă a realității dacă se fac ipoteze corecte. Un model graf realizat pentru clădirea ipotetică este prezentat mai jos în Figura 6, lângă clădirea ipotetică.

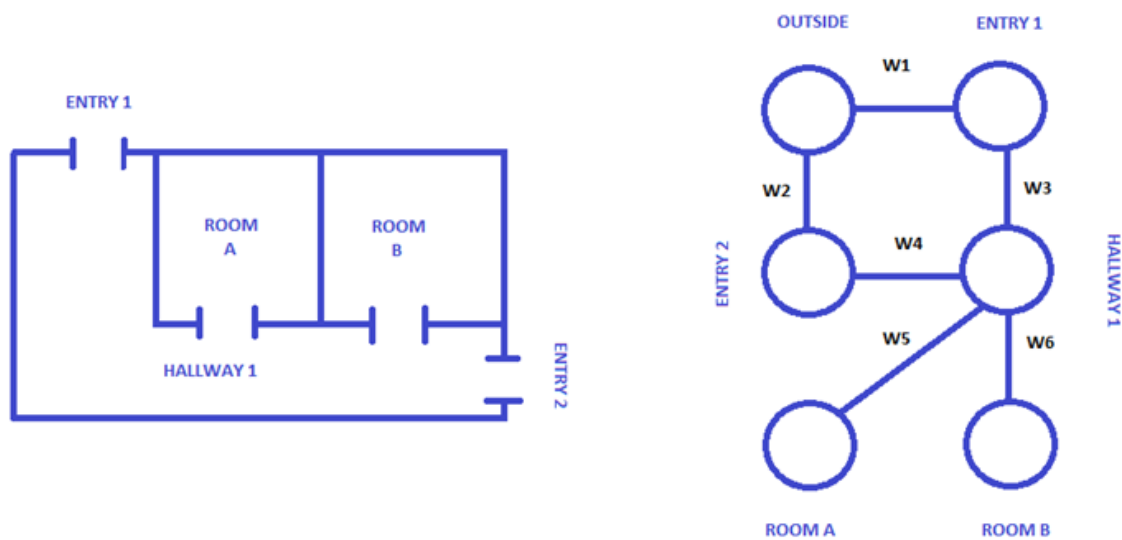


Figura 6. Cladire Ipotetică și Modelul Graf respective.

Notațiile W1, W2, ... W6 din figura grafului din Figura 6 reprezintă ponderile grafului care sunt atribuite muchiilor. Modul în care am lucrat cu ponderile în acest model este să presupunem că ponderile sunt o descriere generală a costului. Costul este cuantificabil, ceea ce îl face măsurabil și ușor de integrat cu alte costuri. Această abstractizare face posibilă extinderea funcționalității serviciilor care rulează deasupra acestei modelări, ceea ce este important pentru teza de față în ansamblu. Pentru un scenariu de navigare simplu în care un utilizator trebuie să traverseze spațiul public, ponderile pot fi considerate un cost constant. Și un algoritm de navigare bazat pe o reprezentare graf (ca Dijkstra) este posibil să scoată cea mai scurtă cale posibilă între punctul de pornire al navigației și destinația stabilită de utilizator.

5.3 Resurse Policies pentru Rutare

De obicei, un algoritm de generare a rutei pentru un graf urmează principiul reducerii costului călătoriei; aceasta poate fi realizată prin calea cea mai scurtă, sau prin calea cea mai rentabilă, alegând ruta cu cel mai mic cost pentru a ajunge la o destinație. Algoritmul nostru adoptă abordarea rentabilă din următoarele motive:

- un algoritm de cale rentabil funcționează pe baza unui graf ponderat.
- se poate crea un graf ponderat pentru a cuantifica dificultatea accesării unui nod.
- integrăm informații diverse (pe baza nevoilor utilizatorilor) în valoarea ponderii pentru fiecare mușe (*polices*, utilizate opțional).

Am decis să privim un nod ca pe un concept de sine stătător, cum ar fi o pagină web, și să lăsăm nodul să dicteze cât de dificil este de accesat, pe baza resurselor *polices* care îi sunt atașate. Algoritmul generator de rute analizează fiecare nod din graf și decide, pe baza resurselor *polices* date la apelul algoritmului, dacă trecerea prin acel nod este de dorit sau nu. Aceste *polices* sunt

cuantificate și furnizate ca o adăugare de pondere în fiecare muchie, permițându-ne să folosim algoritmi bine stabiliți de generare a căilor într-un graf.

Policy 1. Considerarea voturile comunității. O resursă *policy* de vot poate oferi utilizatorului general posibilitatea de a vota dacă o resursă respectă anumite criterii. Voturile în timp se adună și sunt cuantificate printr-o valoare medie și luate în considerare atunci când se solicită acest *policy*.

Policy 2. Evitarea zonelor aglomerate. O altă resursă *policy*, condusă tot de comunitatea de pietoni care utilizează serviciul de rutare, ia în considerare factorul de aglomerație. Când utilizatorii cer sistemului să obțină o anumită cale, înseamnă că, atunci când accesează locațiile respective din lumea reală, aceștia ocupă nodurile corespondente din graf. Mulți utilizatori ar putea avea nevoie să parcurgă unele secțiuni din lumea reală, care se suprapun cu secțiunile traversate de alți utilizatori, rezultând din nevoia lor de a-și găsi drumul; prin urmare, mulțimile se pot forma spontan. Implementarea acestei resurse *policy* se bazează pe rezultatele nevoilor de rutare ale utilizatorilor anteriori, prin creșterea ponderii nodurilor corespondente. De aceea, atunci când următorul utilizator sună serviciul de rutare cu *policy* de a evita aglomerația, ponderile pe legăturile resurselor vizitate anterior sunt crescute, iar algoritmul oferă o rută alternativă mai eficientă din punct de vedere al costurilor (unde costul se referă felul în care spațiul este aglomerat). După un anumit interval de timp, ponderile pe noduri sunt automat reduse dacă nu mai sunt furnizate de serviciul de rutare, ceea ce înseamnă că locațiile corespondente devin mai puțin aglomerate.

Policy 3. Evitarea zonele poluate. Această resursă *policy* este bazată pe date și poate fi aplicată dacă spațiul public este prevăzut cu dispozitive de detectare pentru a detecta diferite proprietăți chimice pentru a caracteriza calitatea aerului.

Policy 4. Evitarea condițiilor meteorologice nefavorabile. Această resursă *policy* asigură, de asemenea, o rutare bazată pe date, bazată pe date provenite de la furnizori externi, cum ar fi furnizorii de prognoză meteo. Serviciul de rutare ia în considerare minimizarea efectului factorilor externi creați de condițiile meteorologice.

Policy 5. Evitarea zonelor cu risc locomotoric. Această resursă *policy* deține informații despre accesibilitatea locațiilor din spațiul public care pot crea dificultăți utilizatorilor cu nevoi de mobilitate redusă. Atunci când această *policy* este utilizat în algoritmul de rutare, ruta calculată elimină nodurile sau conexiunile lipsite de accesibilitate, sau cel puțin utilizatorului îi sunt prezentate informații privind potențialele riscuri din lumea reală.

5.3.1 Algoritmul Multi-Policy de Rutare a Pietonilor

Algoritm de rutare 0-Policy. În mod implicit, graful de bază al unei clădiri are ponderile tuturor conexiunilor egale, deoarece nodurile reprezintă locații fizice care sunt foarte apropiate unele de altele. Acest lucru se aplică în cazul unei solicitări de rutare cu *0-policy*, în care algoritmul rentabil prioritizează calea care cumulează o valoare minimă, adică cea mai scurtă cale. Cu toate acestea, utilizatorii cresc indirect ponderile pe căile care au fost date ca soluții de rutare dacă utilizatorul urmează direcțiile de rutare. Acest lucru este valabil și pentru rutarea cu *0-policy*, deoarece, în cazul în care un utilizator dorește să evite locurile aglomerate, este necesar să știe ce locații (nodurile din graf) au fost cele mai vizitate, chiar dacă utilizatorii care au solicitat serviciul de rutare nu au selectat *Policy 2*. Dacă un nod este prezent într-o ieșire a căii de rutare calculată în

trecutul recent, proprietatea factorului de aglomerare este mărită cu o sumă fixă; dacă un nod nu a fost prezent într-o cale de rutare de ieșire, factorul de aglomerare este scăzut cu o sumă fixă.

Algoritm de rutare multi-policy. În cazul algoritmului de rutare *multi-policy*, fiecare nod are proprietăți specifice care urmează resursele *policies*, iar aceste proprietăți sunt folosite pentru a actualiza ponderile dintre noduri. Algoritm de rutare eficient din punct de vedere al costurilor emite calea care cumulează o valoare minimă, care nu poate fi întotdeauna calea cea mai scurtă, așa cum a fost în cazul cererii de *0-policy*, ci cea care este cea mai convenabilă pentru utilizator, având în vedere nevoile acestuia. Fiecare *policy* analizează proprietatea/prorietățile specifice nodului care o indică. Resursa *policy* analizează proprietățile pereche de pe fiecare nod; Algoritmul 1 arată că sistemul ascultă întotdeauna solicitările utilizatorului și furnizează rezultatul apelurilor serviciului de rutare. Anumite detalii sunt necesare pentru a calcula proprietățile nodurilor cu privire la cât de aglomerate sunt:

- *crowdProperty* – proprietate specială care aparține nodului referitor la *policy* de evitare a aglomerației.
- *increaseCrowd* – cantitatea cu care proprietatea *crowdProperty* pe nod trebuie să mărească costul dacă acest nod a fost oferit în rezultatul apelurilor trecute ale algoritmului de generare de drum.
- *crowdTimeStamp* – când proprietatea *crowdProperty* a fost actualizată ultima oară.
- *currentDate* – data curentă a sistemului când apelul utilizatorului a fost efectuat

Algorithm 1 Multi-policy pedestrian routing algorithm

Inputs: *increaseCrowd*

while routing service provided **do**

get from user call: *start, finish, policies*

for *policy* in *policies*:

switch *policy*:

 Policy1: **call** community feedback algorithm

 Policy2: **call** crowd avoiding algorithm

 Policy3: **call** pollution avoiding algorithm

 Policy4: **call** weather conditions algorithm

 Policy5: **call** reduced mobility algorithm

end switch

receive *customized_graph*

end for

```
call multicriterial_route_finding_algorithm(customized_graph)  
send directions to user  
get from user call: accepted_directions  
if accepted_directions  
for element in path do  
    if element.crowdTimeStamp is_aprox currentDate  
        element.crowdProperty = element.crowdProperty + increaseCrowd  
        element.crowdTimeStamp = currentDate  
    end if  
end for  
end if  
end while
```

Când utilizatorul cheamă serviciul de rutare, sistemul preia detaliile de punct de start (de unde începe navigarea) și punctul destinație (destinația de navigare). Și creează drumul prin graful actualizat recent (*customized_graph*) luând ca parametru de intrare startul și destinația și servește utilizatorului rezultatul obținut cu algoritmul multicriterial de rutare. După aceasta, pentru fiecare nod din ruta oferită utilizatorului (dacă proprietatea nodului *crowdTimeStamp* este aproximativ egală cu timpul stocat *currentDate*), atunci *crowdProperty* pentru acest nod este crescut cu *increaseCrowd* (configurat pentru a determina *crowdProperty* cu cât trebuie majorat după ce a fost vizitat), și *crowdTimeStamp* al acestui nod este suprascris cu valoarea *currentDate*.

5.3.2 Algoritmul de Feedback de la Comunitate

După ce a folosită calea oferită de serviciul de navigare, utilizatorul poate oferi un feedback dacă calea a fost utilă și ușor de utilizat pentru nevoile sale. Acest feedback ia forma unui vot, în care utilizatorul poate vota dintr-un interval, De exemplu, de la 1 la 10, unde 1 înseamnă că cel mai probabil utilizatorul nu și-ar dori niciodată această cale a doua oară când apelează la serviciul de navigație și 10 înseamnă că cel mai probabil utilizatorul ar dori această cale a doua oară când apelează la serviciul de navigație. Aceste voturi sunt stocate în platformă cu numărul total de voturi (incremental după ce fiecare cale furnizată este dată utilizatorului, iar utilizatorul o votează). Atunci când următorul apel pentru serviciul de navigație este efectuat cu *policy* de feedback a comunității, serviciul va lua în considerare feedback-ul anterior de la utilizatori și va crește ponderea nodurilor care au fost incluse în ieșirea serviciului de navigație pe care utilizatorul le-a votat cu scoruri de nedorit. De aceea, serviciul de navigație va calcula o rută care să intersecteze cât mai puțin nodurile pe care nu au fost votate favorabile de utilizatorii din trecut.

Algorithm 2 Community Feedback algorithm

```
Inputs: graph, communityFeedbackList, limit1, limit2, limit3, limit5, limit6  
for node in graph do  
  get from communityFeedbackList nodeVoteScore  
  for connection in nodeConnections do:  
    if nodeVoteScore > limit1 and nodeVoteScore < limit2  
      addingUpNorm = 1  
    end if  
    if nodeVoteScore > limit2 and nodeVoteScore < limit3  
      addingUpNorm = 2  
    end if  
    if nodeVoteScore > limit3 and nodeVoteScore < limit4  
      addingUpNorm = 3  
    end if  
    if nodeVoteScore > limit4 and nodeVoteScore < limit5  
      addingUpNorm = 4  
    end if  
    if nodeVoteScore > limit5 and nodeVoteScore < limit6  
      addingUpNorm = 5  
    end if  
    weightConnection = weightConnection + addingUpNorm  
  end for  
end for  
return graph
```

5.3.3 Algoritm de Evitarea Aglomerației

Metoda prin care modelăm factorul de aglomerație în interiorul unui spațiu public este de a crește ponderea grafului în funcție de un factor de aglomerație. Aceasta înseamnă că fiecare utilizator care traversează graful, crește ponderile marginilor grafului (reprezentând legătura dintre caracteristicile arhitecturale reale) care au fost vizitate. Cu ponderile crescute ale marginilor stocate în graf, algoritmul de navigare care rulează deasupra grafului va oferi o altă rută decât o

rută pentru un utilizator anterior (dacă spațiul public real este construit pentru a o oferi această alternativă).

În aplicația noastră, factorul de aglomerație graf se adună de fiecare dată, un utilizator cheamă și acceptă o rută furnizată de serviciul de navigație. Deoarece după ceva timp caracteristicile arhitecturale reale ar putea fi considerate neocupate, factorul de aglomerație are timp să descrească (TTL, Time-To-Leave). Dacă acel timp trece și niciun alt apel de serviciu de navigație nu returnează o caracteristică arhitecturală care a crescut anterior ponderile conexiunilor, atunci, ponderile crescute sunt reduse cu un alt factor.

Algoritmul de evitare a aglomerației corespunde la *Policy 2*. După cum a rezultat din descrierea anterioară a algoritmului de rutare *multi-policy*, chiar dacă nu este selectată niciun *policy*, graful de bază este întotdeauna actualizat cu ponderile crescute pe secțiunile care au fost recent cele mai vizitate de alți pietoni. Algoritmul 3 are următoarele configurații:

- *crowdProperty*—valoarea de aglomerație pe un nod (cu cât valoarea este mai mare cu atât este nodul mai vizitat).
- *crowdTimeStamp*— când a fost *crowdProperty* actualizat ultima oară.
- *crowdTimeFrame* — un interval de timp după care se consider scăderea *crowdProperty*.
- *decreaseCrowdFactor* — cât de mult *crowdProperty* pe un nod trebuie să scadă după ce a trecut *crowdTimeframe*.

Algorithm 3 Crowd avoiding algorithm

Inputs: *decreaseCrowdFactor, crowdTimeFrame, graph, limit1, limit2, limit3, limit4, limit5, limit6*

for *node* in *graph* **do**

get from *Policy2*: *crowdProperty, crowdTimestamp*

timeDifference = *currentDate* – *crowdTimestamp*

timeDifferenceFrame = *timeDifference* / *crowdTimeFrame*

decreaseCrowdProperty = *decreaseCrowdFactor* **X** *timeDifferenceFrame*

crowdProperty = *crowdProperty* – *decreaseCrowdProperty*

if *crowdProperty* < 0

crowdProperty = 0

end if

for *connection* in *nodeConnections* **do**:

if *crowdProperty* > *limit1* **and** *crowdProperty* < *limit2*

addingUpNorm = 1

end if

if *crowdProperty* > *limit2* **and** *crowdProperty* < *limit3*

```

addingUpNorm = 2
end if
if crowdProperty > limit3 and crowdProperty < limit4
addingUpNorm = 3
end if
if crowdProperty > limit4 and crowdProperty < limit5
addingUpNorm = 4
end if
if crowdProperty > limit6
addingUpNorm = 5
end if
weightConnection = weightConnection + addingUpNorm
end for
end for
return graph

```

Algoritmul calculează *crowdProperty* ca și valoarea curentă *crowdProperty* minus *decreaseCrowdFactor*, multiplicat cu rezultatul *currentDate* minus *crowdTimestamp*, și împărțit la *crowdTimeframe*. În acest fel, *crowdProperty* este mereu actualizat în funcție de cât timp a trecut de când a fost nodul vizitat ultima oară. Dacă după această operație *crowdProperty* este mai mic decât 0, înseamnă că a trecut mult timp de când nodul a fost vizitat, deci este un nod viabil de traversare pentru utilizatorii care au ales să evite aglomerații. După ce *crowdProperty* pentru acest nod a fost calculat, se actualizează ponderea pentru fiecare conexiune al acestui nod. Cele cinci limite pot fi setate ca o ajustare de ponderi prin adunarea unui factor de normare, care definește cât de multă aglomerație este considerată decentă.

5.3.4 Algoritmul de Evitarea Poluare

Policy de evitare a poluării crește proprietățile conexe ale nodurilor pe baza datelor luate de la senzori; De aceea, valoarea data de un sensor este oglindită în valoarea conexă a nodului. Analog, cum a fost explicat în paragraful anterior, ponderile nodului cu *policy* de sensor sunt actualizate conform *Policy 3* invocat de utilizator, pentru a favoriza drumurile ce conțin nodurile cu valorile de poluare cât mai mici. Algoritmul 4 are configurațiile *sensorDataSource* (sursa de date care oferă citirea de la senzorii de calitate aerului); *sensorValue* (valoarea actuală convertită la *addingUpNorm* care este folosită la actualizarea ponderilor grafului).

Algorithm 4 Pollution avoiding algorithm

Inputs: *graph*, *limit1*, *limit2*, *limit3*, *limit4*, *limit5*, *limit6*

for *node* in *graph* **do**

get from *SensorPolicy*: *sensorDataSource*, *sensorValue*

sensorValue = call *sensorDataSource*

for *connection* in *nodeConnections* **do:**

if *sensorValue* > *limit1* **and** *sensorValue* < *limit2*

addingUpNorm = 1

end if

if *sensorValue* > *limit2* **and** *sensorValue* < *limit3*

addingUpNorm = 2

end if

if *sensorValue* > *limit3* **and** *sensorValue* < *limit4*

addingUpNorm = 3

end if

if *sensorValue* > *limit4* **and** *sensorValue* < *limit5*

addingUpNorm = 4

end if

if *sensorValue* > *limit5* **and** *sensorValue* < *limit6*

addingUpNorm = 5

end if

weightConnection = *weightConnection* + *addingUpNorm*

end for

end for

return *graph*

În general, un algoritim *multi-policy* necesită normare. Soluția noastră propune ca fiecare valoare să fie mapată la valori discrete dintr-un set {1, 2, 3, 4, 5}. În Algoritmii 2, 3, 4, 5 și 6, aceste valori de normare sunt adăugate la ponderile grafului, ele reprezentând nivelele de desirabilitate (1—foarte desirabil, 2—desirabil, 3—neutru, 4—nedesirabil, 5—foarte nedesirabil).

5.3.5 Algoritmul de Evitare a Condițiilor Meteo Nefavorabile

Policy meteo poate fi invocată pentru a ajuta utilizatorii să evite condițiile meteorologice nefavorabile deoarece exteriorul este mai afectat de condițiile meteorologice, ceea ce înseamnă că resursele care sunt situate în afara clădirilor vor fi cel mai mult afectate de vreme. Serviciul de navigație care este invocat de această *policy* va trimite un apel despre condițiile meteo către o sursă externă (cel mai probabil un API meteo). Dacă acest API meteo va găsi condiții favorabile care sprijină navigarea în afara clădirilor, atunci resursele exterioare care suportă o astfel de navigare vor fi returnate în calea utilizatorului.

Descrierea serviciului de navigare care utilizează acest algoritm este în Algoritmul 5 de mai jos:

Algorithm 5 Weather condition shelter algorithm

Inputs: *graph*, *WeatherAPI*, *limit1*, *limit2*, *limit3*, *limit4*, *limit5*, *limit6*

```
for node in graph do  
  get from WeatherAPI weather  
  for connection in nodeConnections do:  
    if connection == outside  
      if weather > limit1 and weather < limit2  
        addingUpNorm = 1  
      end if  
      if weather > limit2 and weather < limit3  
        addingUpNorm = 2  
      end if  
      if weather > limit3 and weather < limit4  
        addingUpNorm = 3  
      end if  
      if weather > limit4 and weather < limit5  
        addingUpNorm = 4  
      end if  
      if weather > limit5 and weather < limit6  
        addingUpNorm = 5  
      end if  
    end if  
  end if
```

weightConnection = *weightConnection* + *addingUpNorm*

end for

end for

return *graph*

5.3.6 Algoritm de Evitare a Riscurilor Locomotorice

Spațiile publice ar trebui să fie spații incluzive pentru ca orice participant să navigheze în siguranță. Acesta este un alt scenariu pe care l-am luat în considerare în platforma noastră. Aici viziunea noastră abstractă asupra pondere-cost în interiorul grafului ajută la susținerea acestor scenarii. Costul ar putea fi considerat de către un utilizator final ca fiind dificultatea de a traversa o porțiune a spațiului public. Dificultatea poate veni din multe criterii în cazul persoanelor afectate de probleme locomotorii. De exemplu, scări fără rampe, uși grele, diferite niveluri de înălțime de podea, etc. Aceste caracteristici arhitecturale dificile sunt incluse în modelul graf, într-un mod care poate crește ponderile marginilor legate de acestea.

Pentru un scenariu unde un utilizator încearcă să acceseze din “Outside” “Room A”, modelul graf suportă integrarea cu un *difficulty map*. Dacă presupunem ca ușa de la “Entry B” este grea și ușa de la “Entry A” este favorabilă problemelor locomotorice, atunci drumul care conține “Entry A” va fi mai puțin costisitor și va fi oferit utilizatorului.

Pentru a defini un nivel de dificultate pentru fiecare topologie arhitecturală care reprezintă un nod în modelul graf, administratorul platformei trebuie să încarce ceea ce ne-am referit mai sus ca *difficulty map*. Această *difficulty map* este creată prin cuantificarea pericolului real creat de o topologie arhitecturală. Pentru a crea această *difficulty map*, administratorul platformei poate să folosească multiple manuale care sunt oferite pentru a decide gradul de risc a clădirii pentru un utilizator cu probleme locomotorice. În implementarea noastră de concept am folosit ghidurile de la [45].

Algorithm 6 Reduced mobility algorithm

Inputs: *graph*, *locomotoryModel*, *limit1*, *limit2*, *limit3*, *limit5*, *limit6*

for *node* in *graph* **do**

get from *locomotoryModel* *locomotoryDifficulty*

for *connection* in *nodeConnections* **do:**

if *locomotoryDifficulty* > *limit1* **and** *locomotoryDifficulty* < *limit2*

addingUpNorm = 1

end if

if *locomotoryDifficulty* > *limit2* **and** *locomotoryDifficulty* < *limit3*

addingUpNorm = 2

```
end if  
if locomotoryDifficulty > limit3 and locomotoryDifficulty < limit4  
addingUpNorm = 3  
end if  
if locomotoryDifficulty > limit4 and locomotoryDifficulty < limit5  
addingUpNorm = 4  
end if  
if locomotoryDifficulty > limit5 and locomotoryDifficulty < limit6  
addingUpNorm = 5  
end if  
weightConnection = weightConnection + addingUpNorm  
end for  
end for  
return graph
```

5.4 Operațiile Generale pentru Platformă

În ipoteza noastră, datele pentru un serviciu public trebuie să fie conduse de comunitatea care beneficiază de el. Prin urmare, platforma noastră dispune de servicii care permit administratorilor oricărui spațiu public să încarce versiunea spațiului public de care au nevoie pentru comunitățile lor. De aceea, nu creăm o reprezentare statică a spațiului public, dar administratorul trebuie să o creeze. Odată ce reprezentarea a fost creată, ea este încărcată pe platforma care o convertește în modelul de graf unde pot rula alte operațiuni legate de graf, cum ar fi identificarea căii. Responsabilitatea platformei este de a crea serviciul de conversie. Cu alte cuvinte, modelul hărții, deoarece mapează valorile înțelese și create de administrator la obiecte de calcul. Singura modalitate de a respecta un flux corect de informații între administrator și platformă este un set de reguli sau standarde care pot fi ușor de înțeles de către administratori și ușor de implementat de către platformă.

Operațiunile prezintă 3 faze:

- *bootstrap* (care creează resurse computaționale)
- *consumption* (care consumă API și servicii)
- *maintenance* (care implică toate activitățile care actualizează platforma și o mențin relevantă) [52].

5.4.1 Design pentru Faza de Bootstrap

În implementarea noastră de concept, am stabilit acele reguli în jurul vizualizării planurilor de construcție a spațiilor publice și a vecinătății topologiilor arhitecturale indicate în acele planuri de construcție. Concret, am implementat servicii de încărcare a imaginilor de fișiere raster care sunt o reprezentare la scară a etajelor clădirilor și un fișier Excel care reprezintă ce topologii arhitecturale sunt în vecinătatea altor topologii arhitecturale și la ce coordonate, pe ce plan al clădirii. Aceleași imagini încărcate vor fi folosite de serviciile de navigație pentru a desena deasupra lor indicații pentru apelul utilizatorului. Am ales Excel deoarece este mai ușor de utilizat pentru majoritatea oamenilor, inclusiv pentru administratorii de spații publice. Orice date furnizate de administrator sunt folosite de platforma pentru a crea toate obiectele suport necesare pentru serviciul principal care este serviciul de navigare.

5.4.2 Fazele platformei. Exemplu de Campus Universitar

Faza de *bootstrap* este aceea în care administratorul încarcă prin serviciul platformei specializate toate datele necesare în platformă pentru a crea modelul graf de bază și hărțile vizuale. Pentru această fază, platforma necesită date ușor de înțeles de oameni. În acest moment platforma poate înțelege fișiere Excel care stochează intersecțiile resurselor reale ale spațiului public și pe ce plan se găsesc acestea. Acest plan trebuie să corespundă cu imaginile planurilor care sunt încărcate inițial. Fișierul Excel are o structură simplă care poate fi ușor creată și înțeleasă de oameni (în afară de modul în care arată modelul graf de bază, sau harta proprietăților nodurilor), iar hărțile vizuale sunt fișiere de imagine simple care sunt denumite în același mod cum sunt menționate în fișierul Excel pentru corespondență. Aceasta este cea mai critică fază, fără această fază, utilizatorii nu vor putea apela serviciul de traseu pentru a obține un traseu și o hartă vizuală intuitivă prin spațiul public.

Faza de *consumption* este locul în care utilizatorii folosesc platforma pentru a găsi modalități de deplasare în spațiul public, unde serviciul cel mai important este serviciul de traseu care le va oferi traseul de parcurs, direcții care pot fi citite ca text și hărți vizuale. Acest serviciu de cale poate fi apelat cu *policies* pentru a lua în considerare ceea ce are nevoie cel mai mult utilizatorul atunci când traversează spațiul public.

Faza de *maintaining* este faza în care resursele generate din faza de bootstrap sunt îmbunătățite și menținute la zi. În această fază sunt create *policies* (din resurse externe, ca API-uri meteo, sau resurse interne, ca aglomerarea în spațiul public) pentru a oferi utilizatorului o opțiune de personalizare. Figura 7 prezintă principalele operațiuni pe care administratorii și utilizatorii le pot face pe această platformă, fără operațiunea inițială a administratorilor de pe platformă, utilizatorul nu poate începe să-și consume serviciile.

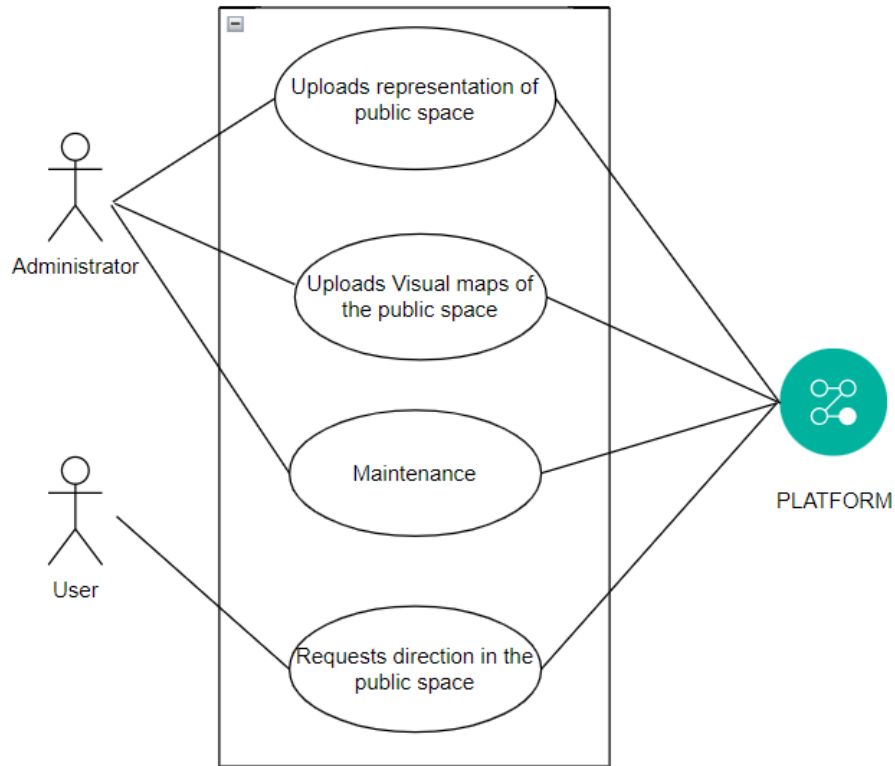


Figura 7. Operațiunile Administratorului și Utilizatorului cu platforma.

5.5 Servicii Integrate pentru Orientare și Accesibilitate

Arhitectura software pe care am ales-o pentru proiectarea noastră se bazează pe o Arhitectură Orientată pe Servicii (SOA) cu ajutorul serviciilor RESTful. Am ales să construim platforma bazată pe SOA, deoarece urmează designul nostru explicat mai sus, care poate fi rezumat ca centrat pe servicii. Elementul de bază al unei aplicații SOA este serviciul, care este un modul logic care poate îndeplini aceeași funcție de business într-o manieră repetabilă și previzibilă. Am folosit servicii web RESTful care pot comunica între ele prin intermediul unui client HTTP. Am ales să folosim serviciile web RESTful, deoarece am considerat că dispozitivele mobile sunt populare în prezent și că aceste dispozitive pot comunica cu ușurință prin intermediul unui client HTTP, astfel punctul de intrare pentru utilizatorii finali în platformă este mai puțin costisitor sau necunoscut.

5.5.1 Arhitectura Software

Sistemul este proiectat cu 3 secțiuni prezentate în Figura 8. O secțiune este utilizată pentru interacțiunea utilizatorului cu platforma (User Interaction with platform). O altă secțiune este utilizată pentru dezvoltarea serviciilor web RESTful platformei care sunt consumate de Interacțiunea cu utilizatorul cu platforma (Platform RESTful Webservices). O altă secțiune este folosită pentru sursele de date care sunt consumate de secțiunea RESTful Webservices (Data sources). Fiecare secțiune comunică cu cealaltă pentru a satisface nevoia utilizatorului final. Fără

una dintre aceste trei secțiuni valoarea nu ar fi furnizată utilizatorului, sau ar fi furnizată cu date eronate.

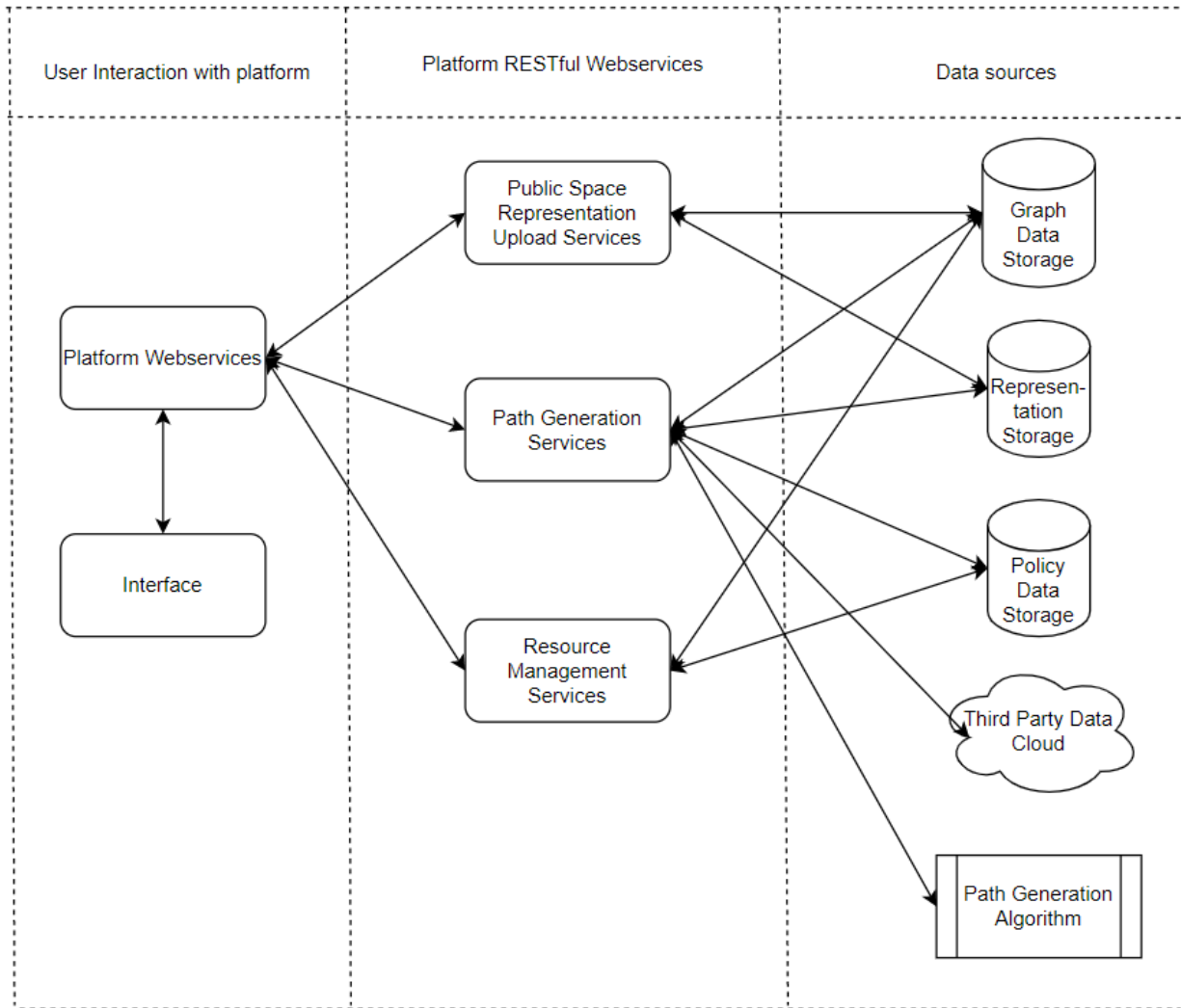


Figura 8. Sistemul ca Arhitectură Orientată Obiect.

Interface ajută utilizatorul și administratorul să ajungă la logica pe care o expune platforma. Interfața este utilizată pentru scop dublu. Primul domeniu furnizează date serviciilor web ale platformei și consumă date de la serviciile web ale platformei. La acest nivel, datele sunt furnizate și consumate prin apeluri HTTP.

Platform Webservices sunt punctele finale care comunică cu backend-ul platformei în care întreaga logică este creată ca servicii web RESTful. Platforma Webservices este creată pentru a consuma date din interfață și a le furniza backend-ului și pentru a consuma date din backend și a le furniza interfeței.

Public Space Representation Upload Services au scopul de a prelua datele de reprezentare pentru spațiul public (Excel, fișiere imagine) furnizate de administrator în faza de bootstrap a platformei.

Path Generation Services preia intrări din interfața cu utilizatorul și, împreună cu datele stocate în Graph Data Storage, Representation Storage și Policy Data Storage, Third Party Data Cloud le trimite la algoritmul de generare a căii pentru a returna calea dintre intrările date de utilizator. Care este apoi transmis utilizatorilor ca răspuns la solicitările acestora.

Resource Management sunt serviciile care preiau informații de la administrator și actualizează informațiile stocate în secțiunea Data sources (Sursă de date), în special pentru Stocarea datelor graf și Stocarea datelor de *policies*.

Graph Storage este independent din punct de vedere tehnologic, poate fi stocat pe o bază de date relațională, sau într-un obiect în memorie și disc.

Representation Storage se ocupă de stocarea fișierelor de imagine care reprezintă planuri de construcție. Deoarece aceste imagini trebuie trimise către interfața utilizatorului pentru un consum ușor, ele trebuie să fie expediate acolo cu ajutorul acelorași servicii web pe care utilizatorul le apelează atunci când trimite cereri.

Policy Data Storage este o stocare de resurse ca **Graph Storage**. Aceste resurse stochează logici specifice care sunt utilizate de algoritmul de generare a căilor pentru a actualiza graful de bază în memorie.

Third Party Data Cloud sunt resurse stocate eventual la crearea *policies*. Path Generation Service va comunica acestor părți terțe Data Cloud, pentru a colecta informațiile care sunt mapate de *policy* care face referire la aceasta.

Path Generation Algorithm este utilizat pentru a crea o rută prin versiunea de graf furnizată de Path Generation Service (ar putea fi graful de bază într-un scenariu cu *0-policy* sau un graf de bază actualizat cu ponderi într-un scenariu cu *n-policies*). Acest algoritm este un algoritm standard Dijkstra care are nevoie doar ca intrare de date graf tradiționale și de puncte pentru a crea traseul între ele.

5.5.2 Servicii RESTful

Serviciile RESTful sunt folosite pentru a transporta date între utilizator și platformă. Fiecare serviciu RESTful are un punct final și un standard de date care trebuie respectat pentru ca datele să circule în interiorul sistemului. Punctul final este de tip URL și standardul este, în general, în implementarea noastră de concept, un format JSON, dar avem cazuri când standardul trebuie să fie format de fișier.

Definiția tehnică a serviciului de încărcare a hărților vizuale pentru spațiul public este descrisă în Tabelul 1. Scopul acestui serviciu este de a prelua de la administrator o imagine care reprezintă spațiul public, ca o imagine la scară a etajului unei clădiri și de a o stoca pe platformă.

Tabel 1. Descrierea Tehnică a Serviciului de Încărcare a Hărților Vizuale pentru Spațiul Public

Metodă HTTP	POST
URL	/mapUpload
Tip Date	tip image
Ieșire	200 OK – pentru imagini acceptate 400 Bad Request – pentru imagini neacceptate
Scop	Încărcarea imaginii reprezintă spațiul public pentru platformă.

Definiția tehnică a Serviciului de încărcare a metadatelor în spațiul public este descrisă în Tabelul 2. Scopul acestui serviciu este de a prelua de la administrator un document care descrie aspectul clădirii pentru a crea modelul graf pentru aceasta, scopul final fiind rularea unui algoritm de navigare deasupra acestuia.

Tabel 2. Descrierea Tehnică a Serviciului de Încărcare a Meta-datelor pentru Spațiul Public

Metodă HTTP	POST
URL	/metadataUpload
Tip Date	tip text
Ieșire	200 OK – pentru document acceptat 400 Bad Request – pentru document neacceptat
Scop	Încărcarea de documente cu meta-date ce reprezintă spațiul public pentru platformă.

În plus față de aceste două servicii de bază pentru serviciile de încărcare pentru reprezentarea spațiului public, există un serviciu de încărcare a informațiilor suplimentare. Încărcarea informațiilor suplimentare are posibilitatea de a adăuga date într-un format de document, care poate fi folosit pentru a descrie spațiul public, dar nu este la fel de important ca crearea modelului graf de bază.

Tabel 3. Descrierea Tehnică a Serviciului de Încărcare a Informațiilor Extra

Metodă HTTP	POST
URL	/extrasUpload
Tip Date	tip text
Ieșiri	200 OK – pentru document acceptat 400 Bad Request – pentru document neacceptat
Scop	Încărcarea informațiilor extra care reprezintă date adiționale despre spațiul public (e.g. dificultate locomotorică).

Path Generation Services sunt servicii care ajută utilizatorul final să acceseze cel mai important aspect al acestei platforme, adică informațiile de navigare și orientare care pot fi folosite eficient în lumea reală. Serviciile de generare a căilor conțin 3 servicii de bază: Serviciu de rutare, Serviciu de direcții, Serviciu de hărți.

Scopul serviciului de rută este de a prelua intrarea utilizatorului într-un format JSON care conține punctul de start, punctul de sosire și orice *policies* opționale optate de utilizatorul final. Definiția tehnică a serviciului de rută se găsește în tabelul 4.

Tabel 4. Descrierea Tehnică a Serviciului de Rutare

Metodă HTTP	GET
URL	/route
Tip Date	tip JSON
Ieșire	200 OK – ruta între start și destinație
Scop	Oferă utilizatorului o rută ordonată secvențial. Conține punctul de start, punctele intermediare ordonate, punctul destinație.

Definiția tehnică a Serviciului de Direcții este descrisă în Tabelul 5. Scopul Serviciului de Direcții este de a oferi utilizatorului indicații sintactice pentru a urma traseul secvențial ordonat furnizat de Serviciul de Rutare.

Tabel 5. Descrierea Tehnică a Serviciului de Direcții

Metodă HTTP	GET
URL	/directions
Tip Data	tip JSON cu ieșirea Serviciului de Rutare
Ieșire	200 OK – ruta dintre punctul de start și destinație cu direcții sintactice
Scop	Oferă utilizatorului o rută ordonată secvențial cu direcții sintactice.

Definiția tehnică a serviciului de hărți este descrisă în tabelul 6. Scopul serviciului de hărți este de a oferi utilizatorului hărți vizuale care pot fi urmărite pentru a ajunge la destinație.

Tabel 6. Descrierea Tehnică a Serviciului de Hărți

Metodă HTTP	GET
URL	/map
Tip Date	tip JSON cu ieșirea Serviciului de Rutare
Ieșire	200 OK – hartă vizuale cu ruta desenată între start și destinație
Scop	Oferă utilizatorului o hartă vizuală cu ruta necesară.

Serviciile de gestionare a resurselor sunt servicii folosite pentru a menține resursele platformei, cum ar fi nodurile sau *policies*. De aceea, Serviciile de menținere a resurselor platformei conțin 2 servicii de bază, Serviciul Managementul Resurselor Graf și Serviciul Managementul Policy.

Definiția tehnică a serviciului Managementul Resurselor Graf descrisă în Tabelul 7. Intrarea pentru acest serviciu este un nod care necesită întreținere.

Tabel 7. Descrierea Tehnică a Serviciului Managementul Resurselor Graf

Metodă HTTP	POST
URL	/graph
Tip Date	tip JSON cu resursa de schimbat
Ieșire	200 OK – dacă resursa a fost schimbată 400 Bad Request – dacă resursa nu a fost găsită
Scop	Oferă administratorului posibilitatea de a actualiza detalii minore legate de modelul graf.

Definiția tehnică a Serviciului Managementul Policy este descrisă în Tabelul 8. Scopul serviciului este de gestionare a politicilor este de a adăuga și modifica *policies* în platformă.

Table 8. Descrierea Tehnică a Serviciului Managementul Policy

Metodă HTTP	POST
URL	/policy
Tip Date	Tip JSON cu <i>policy</i> de actualizat sau schimbat
Ieșire	200 OK – dacă <i>policy</i> a fost actualizat sau creat 400 Bad Request – dacă <i>policy</i> nu a fost găsit
Scop	Oferă administratorului posibilitatea de a crea sau actualiza <i>policies</i> în platformă.

5.5.3 Sistemul de Recomandare

Scopul final al platformei este de a oferi utilizatorilor posibilitatea de a se ajuta singuri în procesul de luare a deciziilor. Problema pe care o abordăm în această teză implică faptul că un utilizator final se confruntă cu necunoscutul. Pentru a da câteva exemple, fie utilizatorul nu știe cum să ajungă la o destinație, fie nu cunoaște ce rută este cea mai sigură. Acest aspect îl împiedică pe utilizator să știe dinainte ce opțiuni are să se orienteze. Prin urmare, atunci când platforma oferă rezultatul pentru apelul de serviciu de navigație, trebuie să îl prezinte într-o manieră clară, ușor de urmărit, și de încredere. Încrederea este un element de impact în experiența cu orice sistem, în special pentru unul care sprijină luarea deciziilor și recomandările [53].

În platforma propusă de autor, din punct de vedere tehnic, serviciul de care este cel mai interesat utilizatorul final este serviciul de navigație. Serviciul de navigare rulează deasupra modelului graf corespunzător spațiului public. Serviciul de navigare actualizează, de asemenea, ponderile grafului de bază pe baza *policies* pe care utilizatorul le invocă. Din punct de vedere al experienței

utilizatorului, orientarea ar putea fi covârșitoare dacă acesta este nou în acel spațiu public. Pentru această problemă, folosim hărțile vizuale încărcate de administrator în procesul de bootstrap, hărți vizuale care sunt conectate la nodurile pe care serviciul de navigație (prin intermediul unui sistem de recomandare) le folosește și le furnizează utilizatorului sub formă de indicații. Pe deasupra acestor hărți vizuale, sistemul de recomandare desenează săgeți de la poziția de start la următoarea poziție intermediară, apoi la următoarea poziție intermediară și așa mai departe până la poziția finală.

În cazul în care harta vizuală are detalii extinse ca numele caracteristicilor arhitecturale, utilizatorul le poate folosi pentru a afla că este pe calea cea bună, de asemenea utilizatorul i se dau sintactic instrucțiuni text cu detalii care sugerează „turn left” („mergeți la stânga”), „turn right” („mergeți la dreapta”), „go ahead on” („mergeți înainte”), etc.

Recomandări textuale. Figura 9 prezintă interfața cu utilizatorul pentru un apel de serviciu de navigație din camera ED003 spre camera ED012 într-o aplicație implementată de concept creată pentru domeniul tezei noastre pentru clădirea ED a Facultății de Automatică și Calculatoare din cadrul Universității Politehnica din București. Figura 10 oferă utilizatorului rezultatul serviciului de navigare în format text cu detalii sintactice, De exemplu „from room ED 003” („din camera ED 003”), „go ahead on” („mergeți înainte”), sau „go right on” („mergeți la dreapta”), etc. Aceste detalii ajută utilizatorul să se orienteze și să aibă o direcție generală de parcurgere a spațiului public dacă lipsesc unele indicații din lumea reală (ca numele camerelor).

Recomandări vizuale. Figura 10 prezintă un traseu în interiorul clădirii ED a Facultății de Automatică și Calculatoare din cadrul Universității Politehnica București care este furnizat utilizatorului final care a solicitat detalii în Figura 9. Acest tip de hartă vizuală cu săgeți roșii este furnizată utilizator atunci când apelează serviciul de navigare.

ed003

ed012

Crowd
 Sensor
 Locomotory

Get Path

from
 room ED 003
 go ahead on
 hallway A on first floor in ED building
 go right on
 hallway D on first floor in ED building
 go right on
 hallway C on first floor in ED building
 go right on
 room ED 012
 destination reached

Get Map

Figura 9. Interfața cu Utilizatorul Implementată cu detalii de navigare.

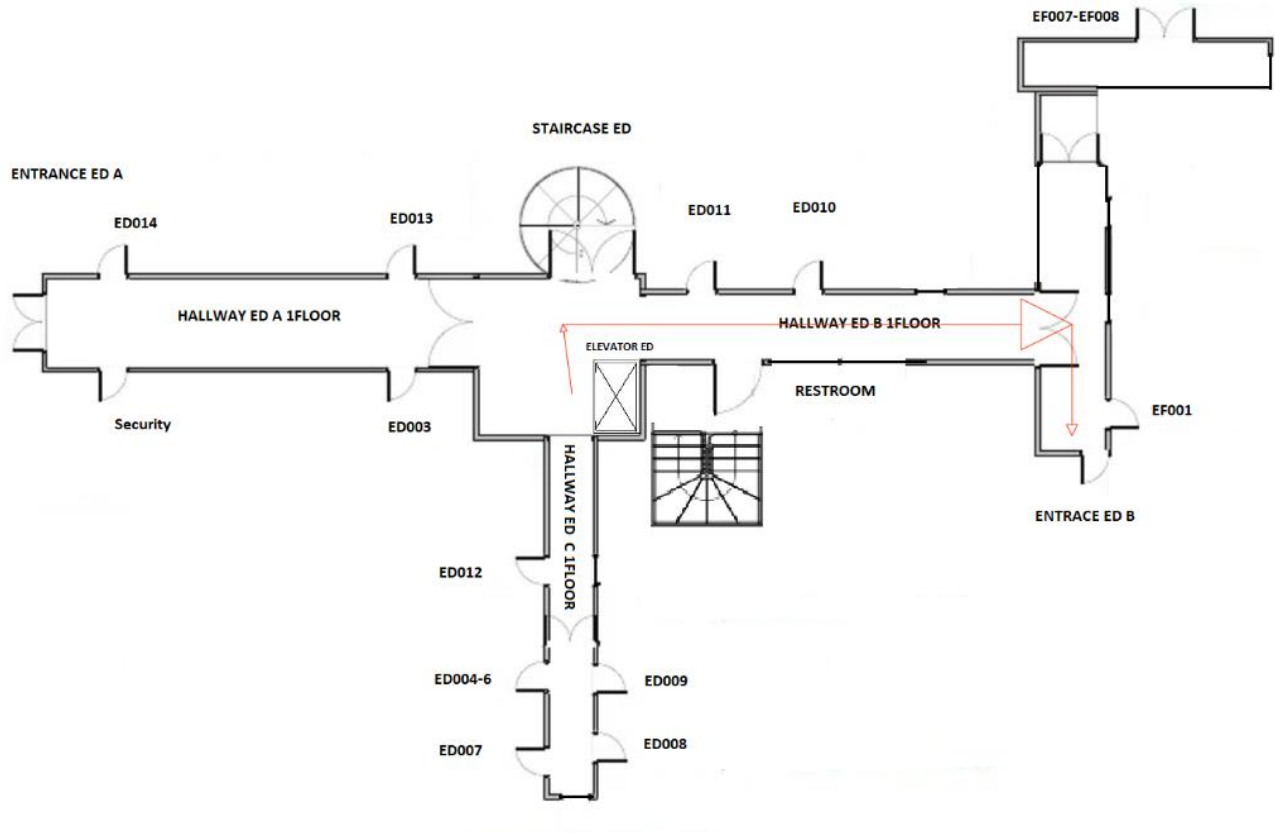


Figura 10. Harta Vizuală Implementată cu rută în spațiul public real.

6 Evaluarea și Validarea Serviciilor Integrate de Orientare și Accesibilitate

În această secțiune discutăm despre modul în care am efectuat verificarea și validarea platformei noastre. Discutăm despre metoda de evaluare pe care am folosit-o pentru această platformă de orientare și accesibilitate, discutând ce resurse de calcul am folosit și ce cazuri am ales pentru validare. Discutăm rezultatele pentru platforma de validare (realizată pentru Facultatea de Automatică și Calculatoare), algoritmi pe care i-am folosit, performanța și experiența vizuală pentru utilizatorul final. Discutăm rezultatele și cum putem evalua ce ar fi diferit în alte scenarii din viața reală.

6.1 Metoda de Evaluare

Dezvoltarea acestor servicii de rutare se bazează pe Arhitectura Orientată pe Servicii (SOA), iar serviciile web sunt bazate pe o abordare RESTful. Administratorii pot accesa o interfață pentru a mapa obiectele de resurse din lumea reală (care reprezintă puncte de interes, adică locații fizice care sunt importante pentru spațiul public) la cele computaționale din interiorul grafului. Ne referim la aceste obiecte computaționale drept resurse, care sunt obiecte dinamice care pot fi interogate și actualizate, dar care reprezintă obiecte din lumea reală în același timp. Este mai întâi important să stabilim un model de graf eficient de uz general, pentru a avea opțiunea de a adapta cu ușurință resursele în viitor. Acest graf de uz general operează cu concepte precum resurse și dependențe (noduri și, respectiv, muchii). Resursele trebuie să aibă posibilitatea de a evolua în timp și de a oferi informații părților interesate. Configurația oferă resurse care acceptă mai multe proprietăți, pe baza nevoilor tuturor părților interesate (*policies*). Am trecut de la un cost de sine stătător (care în cele din urmă ar fi o anumită valoare stocată într-o bază de date) la un cost integrat în proprietățile resursei pentru a o stoca cu alte proprietăți pe care o resursă fizică le-ar putea avea într-o bază de date și pentru a o expune ca o resursă pe serviciile web REST. Datele de conectare rămân neafectate.

6.1.1 Evaluarea Performanței

Pentru scopul testării, am modelat o clădire din facultatea noastră cu metoda descrisă în Capitolul 5. Graful rezultat conține 107 noduri și 338 conexiuni între ele. Testele de performanță au fost executate pentru 1000, 2000, 3000, 4000 și 5000 de utilizatori virtuali (VU); selecția acestor numere de utilizatori este discutată în continuare în Secțiunea 6.3.2 referitoare la Cerințele de încărcare comparative cu realitatea. Aceste VU-uri sunt generate în sistem cu numărul total dorit pentru VU, împărțit la 10 pe secundă. De exemplu, la testarea sistemului cu 1000 VUs, factorul de generare este de 100 VUs/s, pentru 2000 VUs Este 200 VUs/s, și așa mai departe. După atingerea numărului dorit de VU-uri în sistem, testul continuă timp de 60 s, după care VU-urile sunt evacuate

din sistem; Prin urmare sistemul nu mai primește cereri. Cererile sunt trimise către sistem după un timp aleator, între 0,5 s și 2 s.

Testele au fost efectuate pe o mașină cu procesor Intel(R) Core (TM) i5-1035G1 și 8 GB RAM, pentru a evalua dacă poate face față stresului de la utilizatori reali pe o topologie arhitecturală din lumea reală, oferind astfel un serviciu de încredere. Utilizatorii apelează serviciile pentru a calcula traseul dintre două puncte, cu un punct start și un punct final aleatoriu luate din lista de noduri. Această teză prezintă rezultatele pentru trei dintre cazurile de testare, executate cu configurația descrisă mai sus:

- Rutarea *0-policy*.
- Rutarea *1-policy* pentru *Policy 2. Evitarea aglomerației*.
- Rutare *2-policy* pentru *Policy 2. Evitarea aglomerației + Policy 3. Evitarea zonelor poluate*.

6.2 Rezultate

6.2.1 Exemple pentru *policy locomotor* pentru un campus universitar

Această secțiune descrie abordarea noastră pentru rezolvarea nevoilor de accesibilitate și navigare pentru spațiile publice, luând în considerare posibilele constrângeri locomotorii ale utilizatorilor. Prezintă cum se modelează un spațiu public, cum se extinde acest model, cum se integrează conceptul de *policy* și cum se furnizează informații utilizatorului final.

Cazurile de testare descrise în această teză sunt navigație interioară și intra-nivel în interiorul clădirii ED (între două etaje, unul deasupra celuilalt). Scopul este de a verifica dacă serviciile oferă utilizatorului (care nu optează pentru un *policy locomotor*) cea mai scurtă cale către destinație, iar utilizatorul (care a optat pentru un *policy locomotor*) calea cea mai sigură, inclusiv mesaje de avertizare cu privire la riscurile existente.

Să luăm în considerare două cazuri de testare, unul cu niciun *policy* selectat și unul cu *policy locomotor*. Ambele au aceleași puncte de început și de destinație trimise la algoritmul de generare de cale, pentru a obține o comparație între scenariile *zero-policy* și *one-policy*. Evaluarea se bazează pe:

- Criteriul 1. Lungimea drumului recomandat
- Criteriul 2. Dacă se iau în considerare zonele imposibile de acces pentru o persoană cu dizabilități locomotorice, e.g. scări fără rampe, sau uși prea mici pentru un scaun cu rotile.
- Criteriul 3. Nivelul de dificultate pentru persoanele cu dizabilități locomotorice, din cauza rampelor, ușilor grele sau spațiilor restrânse.

Punctul de plecare ales în testele noastre este camera ED117, situată la primul etaj al clădirii ED, conectată la holul B de la primul etaj. Punctul de destinație ales în testul nostru este camera ED012, situată la parterul clădirii ED, conectată cu Holul C de la parter. Figura 11 afișează rezultatul algoritmului de generare a traseului fără opțiunea pentru *policy locomotor*, și rezultatul cu opțiunea pentru *policy locomotor*.

Interpretând Figura 11, se observă că indicațiile oferite utilizatorului îi sugerează să urmeze calea cea mai scurtă din graf, corespunzătoare în realitate scării principale din ED, care este mai aproape de ED117. Această soluție răspunde criteriului 1, dar nu îndeplinește criteriul 2, de aceea este nepotrivită pentru un pieton cu deficiențe locomotorii. Apoi, navigația continuă până la destinație de la parterul clădirii.

Interpretând Figura 11, se observă că indicațiile oferite sunt benefice pentru o persoană cu deficiențe locomotorii; ele ghidează pietonul pe calea cea mai sigură (Criteriul 3) și nu pe cea mai scurtă (Criteriul 1). În acest caz, serviciul returnează o cale care include un lift, care este mai potrivit pentru persoana cu probleme locomotorii, iar Criteriul 2 este îndeplinit.

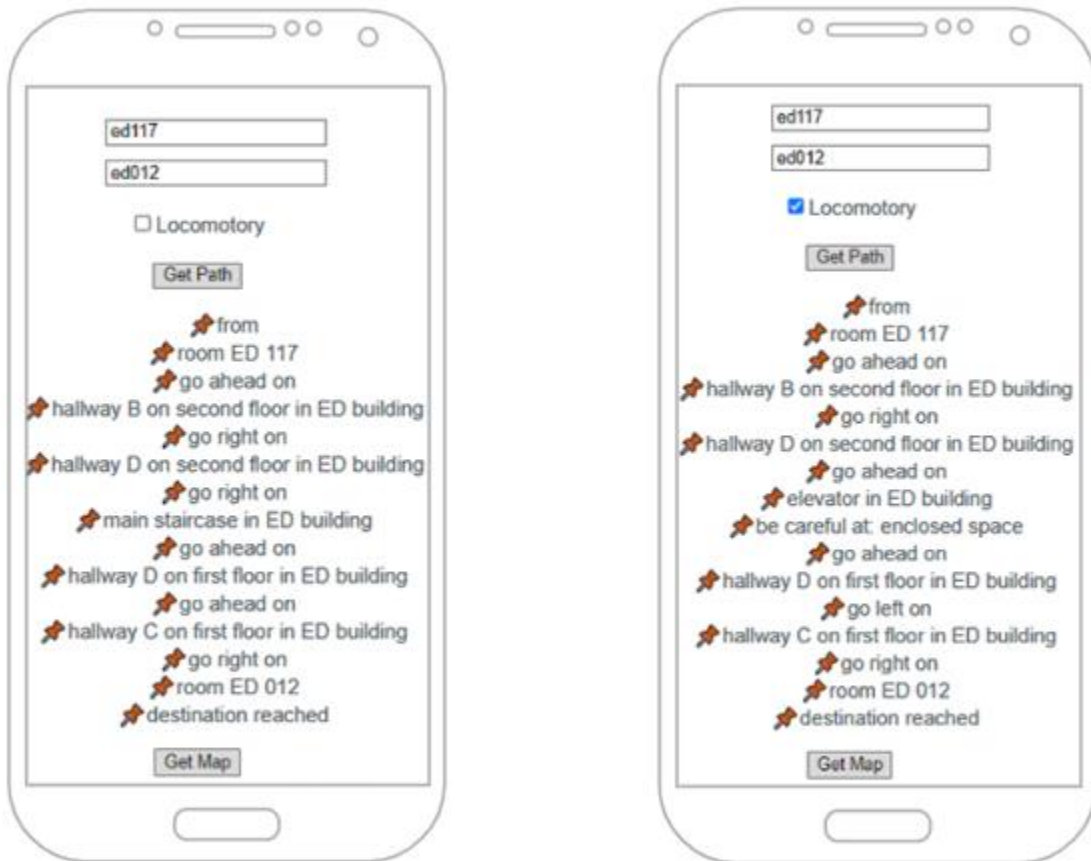


Figura 11. Drumul generat dintre ED117 și ED012. Comparație între *0-policy* și *policy* locomotor.

Cu toate acestea, sunt furnizate informații despre riscuri; riscul liftului este „enclosed space” („spațiul restrâns”), așa cum este definit de modelul *policy* locomotor.

6.2.2 Rezultatele de Performanță

Pentru a vizualiza mai bine cum funcționează serviciile în cadrul diferitelor *policies*, Figura 12 prezintă un graf care arată timpul mediu de răspuns față de VU pentru cele trei cazuri de testare

menționate. Se poate observa că cea mai bună curbă a timpului mediu de răspuns aparține testului *0-policy*; pentru aceasta, serviciile folosesc cel mai mic efort de calcul. Pentru cazurile cu *1-policy* și *2-policy*, evoluțiile timpului mediu de răspuns devin mai mari cu numărul de utilizatori (de exemplu, pentru 4000 VU-uri și 5000 VU-uri); cu toate acestea, chiar și pentru această încărcare, diferențele dintre *0-policy*, *1-policy* și *2-policy* sunt sub 1500ms, păstrând performanța acceptabilă pentru utilizatorul final.

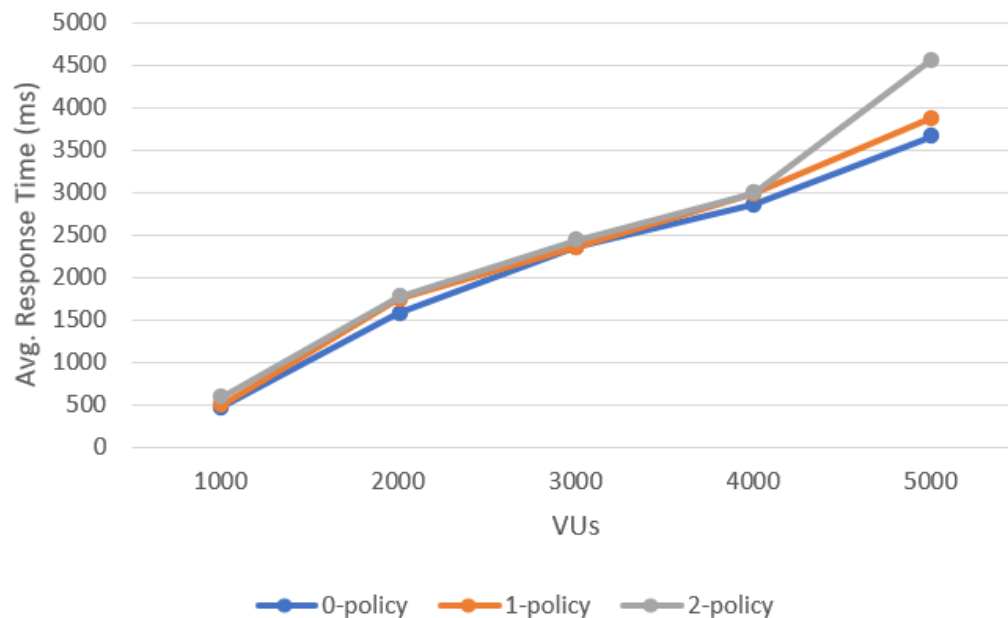


Figura 12. Timpul mediu de răspuns pentru *0-policy* vs *1-policy* vs *2-policy*.

Un alt factor care afectează direct utilizatorul final este rata de eșec a serviciilor de rutare. În Figura 13, se pot observa ratele de eșec (%) în raport cu numărul de VU-uri care apelează serviciile de rutare pentru cele trei cazuri de testare luate în considerare. Ratele de eșec sunt în general sub 4% și mai mari pentru un număr ridicat de VU-uri. Pentru un număr diferit de *policies* aplicate, valorile ratei de eșec sunt destul de apropiate unele de altele, demonstrând că sistemul este construit pentru a gestiona toate cazurile în același mod, fără un impact puternic al VU-urilor și al efortului de calcul implicat.

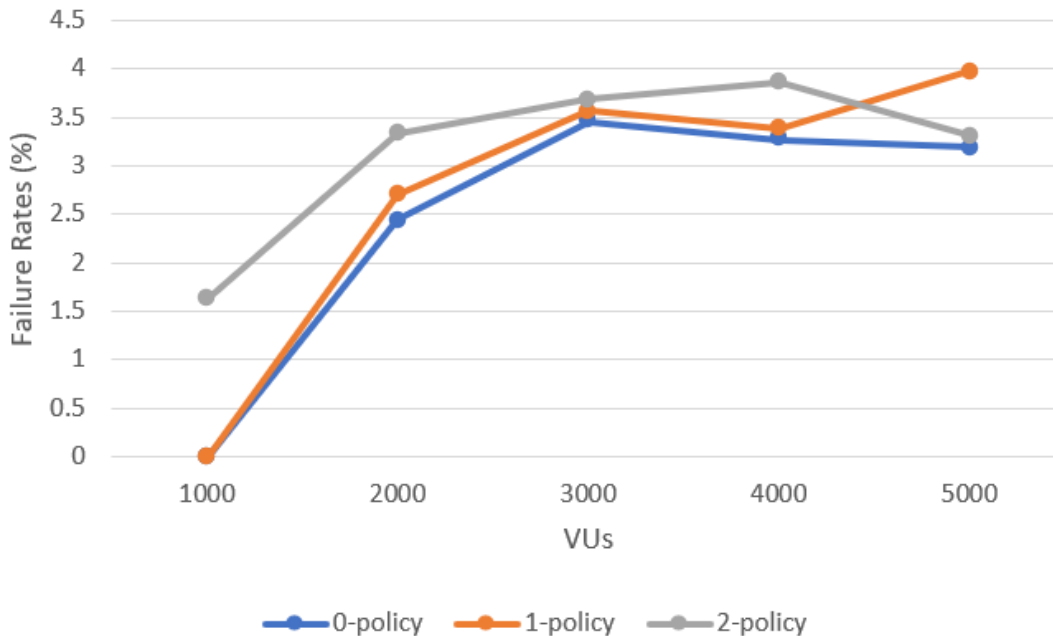


Figura 13. Ratele de eșec pentru *0-policy* vs *1-policy* vs *2-policy*.

6.3 Discuții

6.3.1 Interpretarea Rezultatelor

Pentru a analiza dacă rezultatele obținute indică o experiență bună de utilizare a serviciilor de rutare, să analizăm Figura 12 și Figura 13. Observăm că există diferențe între rezultatele testării cu *0-policy*, *1-policy* și *2-policy*, dar, în fiecare caz, diferența dintre 2000, 3000, 4000 și 5000 VU-uri este neglijabilă în ceea ce privește timpul de răspuns; rata de eșec este de 0% pentru 1000 de VU, în scenariile *0-policy* și *1-policy* și sub 4% pentru un număr mai mare de utilizatori în toate scenariile. Solicitățile pe secundă sunt substanțial similare pentru toate cele trei cazuri de testare, evaluând că serviciul de rutare poate gestiona în mod fiabil acest tip de sarcini. Indicatorul timpului de răspuns poate fi evaluat în comparație cu o navigare tipică a paginii web, unde 1 până la 2 s pentru încărcarea conținutului care reprezintă preferința utilizatorului, iar maxim 5 s pentru încărcarea paginii este de asemenea considerat acceptabil, conform [54]. Serviciile noastre consumă mai multe resurse decât o simplă găsire a unei pagini web, dar rămân în aceleași valori menționate. O arhitectură software mai modulară ar oferi posibilitatea de a scala diferite module. Scalabilitatea și sistemele distribuite pot oferi rezultate mai bune de performanță, în special în producție [55]; în plus, modul de stocare a grafului de bază are un impact asupra modului în care se preiau datele din acesta.

6.3.2 Cerințe de Încărcare Comparative cu Realitatea

Numărul de utilizatori virtuali utilizați pentru testare a fost dedus din studiul cerințelor din viața reală pentru diverse universități, pentru a testa aplicația cât mai aproape de realitate.

Cerințele din viața reală pentru un campus universitar pot rezulta din studierea ratelor de admitere în fiecare an de la unele universități binecunoscute din lume. Universitatea din Cambridge a avut 3528 de nou-veniți în 2019, 3465 în 2018 și 3480 în 2017, conform [56]. Universitatea din Oxford a admis 3280 de studenți noi în 2019, 3309 în 2018 și 3270 în 2017 [57]. University College London a avut 9145 de nou-veniți în 2019, 6110 în 2018 și 5885 în 2017. Eidgenössische Technische Hochschule Zürich a avut 3357 de nou-veniți în 2020 [58]. Imperial College of London a avut 3045 de nou-veniți în 2019, 2845 în 2018 și 2795 în 2017 [59]. Universitatea din Edinburgh a primit 7344 de nou-veniți în 2019, 6346 în 2018 și 6221 în 2017 [60].

Apoi, distribuim numărul total de studenți față de numărul total posibil de ani necesari pentru a obține o diplomă de licență. În acest fel, se poate extrapola că dintr-o universitate care are aproximativ 25.000 de studenți, cum ar fi cea folosită pentru testele noastre, pe un interval mediu de 4 ani pentru licență, vin 6000 de studenți noi în fiecare an.

Având în vedere că nu toată lumea accesează serviciile de rutare în același timp, selecția încărcării utilizate în testele noastre, adică numărul de Utilizatori Virtuali, urmărește realitatea cât mai aproape posibil. Pentru un alt tip de spațiu public, cum ar fi un nod de transport, aceste cerințe pot diferi.

7 Concluzii

În concluzie, problema traseului pietonal acoperă, în general, următoarele aspecte: modul de modelare a grafului care corespunde spațiului real, modul de parcurgere a grafului în funcție de diferite funcții de cost și modul de luare în considerare a altor preocupări specifice pentru anumite categorii de oameni. Trebuie să mapeze cunoștințele matematice abstracte la elementele arhitecturale concrete, circumstanțele momentului și nevoile diferitelor pietoni. Această teză a propus un design orientat spre servicii care ia în considerare toate aceste aspecte pentru furnizarea de direcții de rutare personalizate într-un spațiu public, prin alegerea unui set de *policies*. Două dintre acestea sunt determinare de comunitate, luând în considerare voturile altor pietoni care circulă în același spațiu public, sau evitare locuri frecventate de multe alte persoane. Celelalte două sunt determinate de date care provin din mai multe surse, cum ar fi senzorii pentru măsurarea calității aerului sau serviciile externe de monitorizare a vremii. În plus, un *policy* suplimentar are în vedere nevoi speciale pentru persoanele cu mobilitate redusă.

La adăugarea de *policies* suplimentare, testele pentru până la 5000 de utilizatori virtuali au arătat o mică diferență în timpii de răspuns (mai puțin de 1 s) și, de asemenea, în ratele de eșec (mai puțin de 0,8%) - chiar dacă execuția nu a fost efectuată într-un mediu de performanță ridicat. Prin urmare, furnizarea unor astfel de servicii software, inclusiv flexibilitatea în alegerea combinațiilor specifice de *policies*, este accesibilă atunci când se gestionează un spațiu public. Testele de performanță au demonstrat că serviciile de rutare pot suporta încărcări mari de solicitări într-o infrastructură de spațiu public modelată cu o granularitate medie.

Recomandarea noastră atunci când se implementează acest tip de platformă este să înceapă cu cel mai potrivit model graf care reprezintă spațiul public real. Granularitatea reprezentării contează, și

este mai bine să se efectueze o analiză în prealabil, să se determine care sunt punctele de interes specifice și să se stabilească conexiunile care leagă punctele de aproximativ aceeași dimensiune ca în realitate. În plus, este recomandat să se determine tipurile de persoane care folosesc spațiul public. În cele din urmă, recomandările noastre pentru asigurarea performanței corecte în spații publice foarte mari sunt în direcția unei strategii de modularizare, implementare în Cloud și scalare adecvate, pentru a oferi performanțe și experiență mai bună pentru utilizator.

7.1 Rezumatul Contribuțiilor Originale

Contribuțiile originale prezentate în această teză se încadrează în categoriile de servicii de navigație și orientare pentru spații publice inteligente. Pentru a rezuma contribuțiile noastre originale, menționăm întrebările care ne-au determinat să le dezvoltăm.

O viziune la nivel înalt a contribuțiilor personale prezentate în această teză sunt:

- Analiza stadiului actual pentru software de navigare și orientare
- Definierea unui set de scenarii de navigare și orientare în interiorul unui campus universitar
- Propunerea unei abordări a traseului pietonal bazat *multi-policy* (evitarea aglomerației, constrângeri locomotorii, voturile comunității, evitarea zonelor poluate, evitarea condițiilor meteorologice nefavorabile)
- Proiectarea algoritmului pentru rutare într-un graf bazat pe *policies*
- Crearea serviciilor REST pentru a gestiona o platformă de navigare centrată pe utilizator
- Integrarea informațiilor externe pentru a fi integrate cu navigarea recomandată utilizatorului final
- Elaborarea unui model pentru un exemplu de campus universitar
- Configurarea algoritmilor pentru accesibilitate
- Dezvoltarea unei platforme orientată spre servicii care integrează serviciile pentru a manipula modele, hărți și soluții flexibile de navigare
- Validarea abordării propuse pentru un campus universitar
- Evaluarea performanței
- Discutarea oportunității aplicării acestei abordări pe baza studiului nevoilor campusurilor existente

7.2 Lista Publicațiilor

1. **Ioan Damian**, Anca Daniela Ionita, Restful Services for Orientation and Accessibility in a University Campus; University Politehnica of Bucharest Scientific Bulletin (2021)– Series C Rank:Q4 (Electrical Engineering and Computer Science); Impact factor 0.37; **WOS:000628640200003**
2. **Ioan Damian**, Marian Lacatusu, Anca Daniela Ionita, Florin Lacatusu, Software Services to Support Faculty Management in Times of Pandemic, *15th International Technology, Education and Development Conference (INTED 2021) Proceedings* (2021), pp. 4634-4639, IATED Digital Library, doi: 10.21125/inted.2021.0943

3. **Ioan Damian**, Anca Daniela Ionita, Silvia Oana Anton, Community - and Data-Driven Services for Multi-Policy Pedestrian Routing, *Sensors* (2022) Rank: **Q2** (Engineering, Electrical & Electronic), Impact factor: 3.576, **WOS: 000817661800001**
4. Florin Lacatusu, **Ioan Damian**, Anca Daniela Ionita, Marian Lacatusu, Smart Building Manager Software in Cloud, *University Politehnica of Bucharest Scientific Bulletin* (2021)– Series C Rank:Q4 (Electrical Engineering and Computer Science); Impact factor 0.37; **WOS: 000741473700003**
5. Marian Lacatusu, Florin Lacatusu, **Ioan Damian**, Anca Daniela Ionita, Multicloud Deployment to Support Remote Learning, *15th International Technology, Education and Development Conference (INTED 2021) Proceedings* (2021), pp. 4601-4606, IATED Digital Library, doi: 10.21125/inted.2021.0936
6. **Ioan Damian**, Marian Lacatusu, Florin Lacatusu, Anca Daniela Ionita, Web Services for Guiding persoanele cu Locomotor Impairments in Public Spaces, *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*, 2022, pp.639-644, IEEE, INSPEC, doi:10.1109/ICSTCC55426.2022.9931861, **WOS: 000889980600107**
7. Marian Lacatusu, Anca Daniela Ionita, Florin Lacatusu and **Ioan Damian**, "Decision support for multicloud deployment of a modeling environment," *2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 2022, pp. 247-251, doi: 10.1109/ICCP56966.2022.10053951
8. Florin Lacatusu, Anca Daniela Ionita, Marian Lacatusu, **Ioan Damian** and Daniela Saru, "A Comparison of Cloud Edge Monitoring Solutions for a University Building," *2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 2022, pp. 253-257, doi: 10.1109/ICCP56966.2022.10053978

7.3 Perspective de Viitor

Pentru a vorbi despre perspective de viitor, trebuie să începem cu contribuțiile originale prezentate în această teză. Acestea sunt punctele pe care dorim să evoluăm în viitor și le considerăm posibile la acest moment.

Aplicația minimalistă este un punct forte pentru teza noastră, dar beneficiul real poate fi observat doar în producție. Aceasta înseamnă că o platformă cu designul nostru trebuie să fie implementată într-o soluție Cloud, astfel încât trebuie formulată o strategie de implementare în Cloud.

O caracteristică de bază a platformei rămâne modelul legat de reprezentarea sub formă de graf. Această caracteristică nu este menită să fie modificată din motivele discutate mai sus în capitolele de modelare. O dezvoltare viitoare este reprezentarea spațiului public cu componente din CAD (design asistat de computer), pentru care trebuie implementate și alte module de modelare. Ceea

ce s-ar putea schimba este adăugarea mai multor algoritmi de navigare, care pot fi de uz general, astfel încât să poată fi integrați cu *policies* la nivelul platformei.

Aspectul *policies* platformei nu trebuie schimbat, este o caracteristică de bază și singurul mod în care ar putea integra interesele multiple ale utilizatorilor în interiorul platformei. Ceea ce ar putea fi extins se referă la serviciile care modelează interacțiunea cu realitatea și stocarea și utilizarea în interiorul platformei.

O altă perspectivă de viitor este dezvoltarea notificărilor de tip „push” bazate pe sistem care să ajute utilizatorii să fie mai conștienți dacă o anumită porțiune din spațiul public nu mai este accesibilă. De exemplu, dacă un hol a fost închis în timpul zilei și poate dimineața înca era deschis.

O perspectivă de viitor privind experiența utilizatorilor este integrarea platformei cu platformele de dispozitive inteligente. Dispozitive inteligente ca smartwatch-urile sunt componente care ar putea beneficia de progrese tehnice în următorii ani și credem că integrarea acestora este un aspect important, deoarece sunt deosebit de populare în rândul generațiilor mai tinere.

Referințe

1. Ioan Damian, Anca Daniela Ionita, Restful Services for Orientation and Accessibility in a University Campus; University Politehnica of Bucharest Scientific Bulletin (2021)– Series C (Electrical Engineering and Computer Science)
2. Florin Lacatusu, Ioan Damian, Anca Daniela Ionita, Marian Lacatusu, Smart Building Manager Software in Cloud, University Politehnica of Bucharest Scientific Bulletin (2021)– Series C (Electrical Engineering and Computer Science)
3. R. Drober, Campus Landscape – Functions, Forms, Features, John Wiley & Sons, 2000, pp. 10-60
4. M. Hamblen, "Just what is a smart City?," 2015. [Online]. Available: <https://www.computerworld.com/article/2986403/just-what-is-a-smart-city.html>. [Accessed 17 December 2020]
5. T. Nam et al, "Conceptualizing smart city cu dimensions of technology, people, and institutions" in 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, 2011.
6. S. Bendre et al, "Event Based Campus Navigation System" International Journal of Computer Science and Information Technologies, vol. 7, no. 1, pp. 462-464, 2016.
7. D. Merode et al, "Smart Campus based on iBeacon Technology" in International Symposium on Ambient Intelligence end Embedded System, 2015.

8. J. Yim et al, "Design and Implementation of a Smart Campus guide Android App", in *Appl Math Inf Sci*, vol. 8, no. 1, pp.47-53, 2014.
9. L. Kwok, "A vision for the development of i-campus," *Smart Learning Environments*, vol. 2, pp. 1-12, 2015.
10. Calinescu, R.; Cámara, J.; Paterson, C. Socio-cyber-physical systems: Models, opportunities, open challenges. In *Proceedings of the 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems, SEsCPS@ICSE*, Montreal, QC, Canada, 28 May 2019; pp. 2–6.
11. Martensen, B. The Perception-Action Hierarchy and its Implementation Using Binons (Binary Neurons). In *Proceedings of the 10th Annual International Conference on Biologically Inspired Cognitive Architectures (BICA)*, Seattle, WA, USA, 15–18 August 2019; pp. 489–500.
12. Ebrahim, A. People-Centric Smart Campus. In *Proceedings of the 2021 International Conference on Transport and Smart Cities (ICoTSC 2021)*, Frankfurt, Germany, 17 December 2021; pp. 264–267.
13. Sánchez, C.M.; Zella, M.; Capitán, J.; Marrón, P.J. From Perception to Navigation in Environments cu Persons: An Indoor Evaluation of the State of the Art. *Sensors* 2022, 22, 1191.
14. Coito, T.; Firme, B.; Martins, M.S.E.; Vieira, S.M.; Figueiredo, J.; Sousa, J.M.C. Intelligent Sensors for Real-Time Decision-Making. *Automation* 2021, 2, 62–82.
15. Ioan Damian, Marian Lacatusu, Anca Daniela Ionita, Florin Lacatusu, Software Services to Support Faculty Management in Times of Pandemic, *15th International Technology, Education and Development Conference (INTED 2021) Proceedings (2021)*, pp. 4634-4639, IATED Digital Library, doi: 10.21125/inted.2021.0943
16. Ioan Damian, Anca Daniela Ionita, Silvia Oana Anton, Community- and Data-Driven Services for Multi-Policy Pedestrian Routing. *Sensors*. 2022, 22(12):4515. doi: <https://doi.org/10.3390/s22124515>
17. Ioan Damian, Marian Lacatusu, Florin Lacatusu, Anca Daniela Ionita, Web Services for Guiding persoanele cu Locomotor Impairments in Public Spaces, *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*, 2022, pp.639-644, IEEE, INSPEC, doi:10.1109/ICSTCC55426.2022.9931861
18. Ntakolia, C.; Iakovidis, D.K. A swarm intelligence graph-based pathfinding algorithm (SIGPA) for multi-objective route planning. *Comput. Oper. Res.* 2021, 133, 105358.
19. Wang, D.; Chen, X.; Huang, H. A graph theory-based approach to route location in railway interlocking. *Comput. Ind. Eng.* 2013, 66, 791–799.
20. Wu, G.; Atilla, I.; Tahsin, T.; Terziev, M.; Wang, L.C. Long-voyage route planning method based on multi-scale visibility graph for autonomous ships. *Ocean Eng.* 2021, 219, 108242.

21. Kielar, P.M.; Biedermann, D.H.; Kneidl, A.; Borrmann, A. A unified pedestrian routing model for graph-based wayfinding built on cognitive principles. *Transp. A Transp. Sci.* 2018, 14, 406–432.
22. Zhao, J.; Fang, Z. Research on Campus Bike Path Planning Scheme Evaluation Based on TOPSIS Method: Wei'shui Campus Bike Path Planning as an Example. *Procedia Eng.* 2016, 137, 858–866.
23. Najjar, A.B.; Al-Issa, A.R.; Hosny, M. Dynamic indoor path planning for the visually impaired. *J. King Saud Univ. Comput. Inf. Sci.* 2022, in press.
24. Zhou, Z.; Weibel, R.; Richter, K.-F.; Huang, H. HiVG: A hierarchical indoor visibility-based graph for navigation guidance in multi-storey buildings. *Comput. Environ. Urban Syst.* 2022, 93, 101751.
25. WRLD Environments Cases. Indoor Navigation. Benefits and Use Cases. Available online: <https://www.wrld3d.com/blog/indoor-navigation/> (accessed on 20 March 2021).
26. Wang, J.; Hu, A.; Liu, C.; Li, X. A Floor-Map-Aided WiFi/Pseudo-Odometry Integration Algorithm for an Indoor Positioning System. *Sensors* 2015, 15, 7096–7124.
27. Prandi, C.; Delnevo, G.; Salomoni, P.; Mirri, S. On Supporting University Communities in Indoor Wayfinding: An Inclusive Design Approach. *Sensors* 2021, 21, 3134.
28. Ferreira, J.C.; Resende, R.; Martinho, S. Beacons and BIM Models for Indoor Guidance and Location. *Sensors* 2018, 18, 4374.
29. Mataloto, B.; Ferreira, J.C.; Resende, R.; Moura, R.; Luís, S. BIM in People2People and Things2People Interactive Process. *Sensors* 2020, 20, 2982.
30. Lin, Y.-B.; Chou, S.-L. SpecTalk: Conforming IoT Implementations to Sensor Specifications. *Sensors* 2021, 21, 5260.
31. Fortes, S.; Hidalgo-Triana, N.; Sánchez-La-Chica, J.-M.; García-Ceballos, M.-L.; Cantizani-Esteva, J.; Pérez-Latorre, A.-V.; Baena, E.; Pineda, A.; Barrios-Corpa, J.; García-Marín, A. Smart Tree: An Architectural, Greening and ICT Multidisciplinary Approach to Smart Campus Environments. *Sensors* 2021, 21, 7202.
32. Tseng, K.-H.; Chung, M.-Y.; Chen, L.-H.; Chang, P.-Y. Green Smart Campus Monitoring and Detection Using LoRa. *Sensors* 2021, 21, 6582.
33. Dasler, P.; Malik, S.; Mauriello, M.L. “Just Follow the Lights”: A Ubiquitous Framework for Low-Cost, Mixed Fidelity Navigation in Indoor Built Environments. *Int. J. Hum.-Comput. Stud.* 2021, 155, 102692.
34. Chou, T.-L.; ChanLin, L.-J. Augmented Reality Smartphone Environment Orientation Application: A Case Study of the Fu-Jen University Mobile Campus Touring System. *Procedia-Soc. Behav. Sci.* 2012, 46, 410–416.

35. Li, C.-Y.; Yin, J. A pedestrian-based model for simulating COVID-19 transmission on college campus. *Transp. A Transp. Sci.* 2022, 1–25.
36. Zhao, M.; Zhou, C.; Chan, T.; Tu, C.; Liu, Y.; Yu, M. Assessment of COVID-19 aerosol transmission in a university campus food environment using a numerical method. *Geosci. Front.* 2022, 101353.
37. Muller, K.; Muller, P.A. Mathematical modelling of the spread of COVID-19 on a university campus. *Infect. Dis. Model.* 2021, 6, 1025–1045.
38. Bartolucci, A.; Templeton, A.; Bernardini, G. How distant? An experimental analysis of students' COVID-19 exposure and physical distancing in university buildings. *Int. J. Disaster Risk Reduct.* 2022, 70, 102752.
39. von Seidlein, L.; Alabaster, G.; Deen, J.; Knudsen, J. Crowding has consequences: Prevention and management of COVID-19 in informal urban settlements. *Build. Environ.* 2020, 188, 107472.
40. Polanco, L.D.; Siller, M. Crowd management COVID-19. *Annu. Rev. Control* 2021, 52, 465–478.
41. Geneletti, D.; Cortinovis, C.; Zardo, L. Simulating crowding of urban green areas to manage access during lockdowns. *Landsc. Urban Plan.* 2022, 219, 104319.
42. T. Bidila, R.N. Pietraru, A.D Ionita, A. Olteanu “Monitor Indoor Air Quality to Assess the Risk of COVID-19 Transmission” 23rd International Conference on Control Systems and Computer Science Technologies, CSCS 2021 ; : 356-361, 2021.
43. Chen JW, Zhang J. “Comparing Text-based and Graphic User Interfaces for novice and expert users.” *AMIA Annu Symp Proc.* 2007 Oct
44. Julian Keil, Dennis Edler, Lars Kuchinke, Frank Dickmann “Effects of visual map complexity on the attentional processing of landmarks”. *PLoS One* 2020 March
45. Landman K. “Inclusive public space: rethinking practices of mitigation, adaptation, and transformation”. *Urban Des Int.* 2020
46. Van Hoogdalem, H., Van Der Voordt, T. J. M., & Van Wegen, H.B.R. “Comparative floorplan-analysis as a means to develop design guidelines.” *Journal of Environmental Psychology*, 1985
47. Yu Li, Weijia Li, Yingying Yang, Qi Wang “Feedback and Direction Sources Influence Navigation Decision Making on Experienced Routes”. *Front. Psychol.*, September 2019
48. Hermann Bulf, Maria Dolores de Hevia, Valeria Gariboldi, Viola Macchi Cassia “Infants learn better from left to right: a directional bias in infants' sequence learning”. *Sci Rep* 2017
49. Chang, N. S., & Fu, K. S. “A relational database system for images.” *Lecture Notes in Computer Science*, 1980

50. Mark Wallis, Frans Henskens, Michael Hannaford “A Distributed Content Storage Model for Web Applications”. Conf. INTERNET 2010
51. da Fontoura Costa, L.; Oliveira, O.N., Jr.; Travieso, G.; Rodrigues, F.A.; Boas, P.R.V.; Antiqueira, L. Analyzing and Modeling Real-World Phenomena cu Complex Networks: A Survey of Applications. *Adv. Phys.* 2011, 60, 329–412.
52. Bauer, V., & Heinemann, L. “Understanding API Usage to Support Informed Decision Making in Software Maintenance.” 2012 16th European Conference on Software Maintenance and Reengineering
53. Nunes, I., Jannach, D. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Model User-Adap Inter* 27, 393–444 (2017)
54. Bartuskova, A.; Krejcar, O. Loading Speed of Modern Websites and Reliability of Online Speed Test Services. In *Computational Collective Intelligence. Lecture Notes in Computer Science*; Núñez, M., Nguyen, N., Camacho, D., Trawiński, B., Eds.; Springer: Cham, Switzerland, 2015; Volume 9330.
55. Alshinina, R.; Elleithy, K. Performance and Challenges of Service-Oriented Architecture for Wireless Sensor Networks. *Sensors* 2017, 17, 536.
56. UCAS. End of Cycle 2016 Data Resources DR4_001_02 Main Scheme Acceptances by Provider. Available online: <https://www.ucas.com/data-and-analysis/ucas-undergraduate-releases/ucas-undergraduate-end-cycle-data-resources/applicants-and-acceptances-universities-and-colleges-2016>
57. University of Oxford. Annual Admissions Statistical Report: May 2020. Available online: <https://www.ox.ac.uk/sites/files/oxford/AnnualAdmissionsStatisticalReport2021.pdf>
58. UCAS. 2020 Entry Provider-Level End of Cycle Data Resources. Available online: <https://www.ucas.com/data-and-analysis/undergraduate-statistics-and-reports/ucas-undergraduate-end-cycle-data-resources-2020/2020-entry-provider-level-end-cycle-data-resources>
59. ETH. ETH in Figures. Available online: <https://ethz.ch/en/the-eth-zurich/portrait/eth-zurich-in-figures.html>
60. University of Edinburgh. Undergraduate Admissions Statistics. Available online: <https://www.ed.ac.uk/student-recruitment/admissions-advice/admissions-statistics>