



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

# **SISTEM DE MONITORIZARE ȘI REZERVARE A BIROURILOR ÎNTR-O COMPANIE**

LUCRARE DE LICENȚĂ

Absolvent: **Maria-Ioana BOBEȘ**

Coordonator **as. ing. Cosmina IVAN**  
științific:

**2017**



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

DECAN,  
Prof. dr. ing. Liviu MICLEA

DIRECTOR DEPARTAMENT,  
Prof. dr. ing. Rodica POTOLEA

Absolvent: **Maria-Ioana BOBEȘ**

**SISTEM DE MONITORIZARE ȘI REZERVARE A BIROURILOR ÎNTR-O COMPANIE**

1. **Enunțul temei:** Proiectul își propune realizarea unui sistem menit să faciliteze utilizarea birourilor neasignate într-o companie. Sistemul își propune să ofere angajaților unei companii care permite asignare dinamică a angajaților la birouri, în funcție de prezența acestora în firmă, posibilitatea de a rezerva un birou în prealabil, de a ocupa un birou liber la un moment dat sau de a vizualiza amplasarea unui coleg în cadrul companiei.
2. **Conținutul lucrării:** Cuprins, Introducere, Obiectivele Proiectului, Studiu Bibliografic, Analiză și Fundamentare Teoretică, Proiectare și Implementare, Testare și Validare, Manual de Instalare și Utilizare, Concluzii, Bibliografie, Anexe.
3. **Locul documentării:** Universitatea Tehnică din Cluj-Napoca, Departamentul Calculatoare
4. **Consultanți:**
5. **Data emiterii temei:** 1 noiembrie 2016
6. **Data predării:** 8 septembrie 2017

Absolvent: \_\_\_\_\_

Coordonator științific: \_\_\_\_\_

---

## Cuprins

<b>Capitolul 1. Introducere – Contextul proiectului .....</b>	<b>1</b>
1.1. Contextul proiectului .....	1
1.2. Motivația.....	2
1.3. Conținutul lucrării.....	3
<b>Capitolul 2. Obiectivele Proiectului .....</b>	<b>5</b>
2.1. Obiectivul principal .....	5
2.2. Obiective secundare.....	6
2.2.1. Studiul metodelor de aranjare și asignare a birourilor într-o unitate .....	6
2.2.2. Analiza problemei resurselor de spațiu de lucru într-o companie .....	6
2.2.3. Soluția problemei prezentate .....	7
2.2.4. Dezvoltarea aplicației care să implementeze soluția .....	7
2.2.5. Evaluarea prototipului realizat.....	7
<b>Capitolul 3. Studiu Bibliografic.....</b>	<b>9</b>
3.1. Gestionarea spațiului de lucru într-o companie .....	9
3.2. Munca centrată pe activitate .....	10
3.3. Influența psihologică asupra angajaților .....	10
3.4. Avantaje ale sistemului de birouri neasignate .....	11
3.5. Dezavantaje ale sistemului cu birouri neasignate .....	12
3.6. Concluzii.....	12
3.7. Sisteme similare.....	13
3.7.1. Desk Booking Software.....	13
3.7.2. Sistemul de rezervări electronice EMS .....	13
3.7.3. DeskFlex .....	14
3.7.4. Smartway2.....	14
3.7.5. Room & Resource booking .....	14
3.7.6. Comparația sistemului dezvoltat în această lucrare cu sistemele similare ..	15
<b>Capitolul 4. Analiză și Fundamentare Teoretică.....</b>	<b>17</b>
4.1. Cerințele aplicației .....	17
4.1.1. Cerințe funcționale .....	17
4.1.2. Cerințe non-funcționale .....	19
4.2. Cazuri de utilizare.....	22
4.2.1. Actorii sistemului .....	23

---

4.2.2. Cazuri de utilizare.....	23
4.2.3. Descrierea detaliată a cazurilor de utilizare.....	24
4.3. Context tehnologic.....	30
<b>Capitolul 5. Proiectare de Detaliu și Implementare .....</b>	<b>33</b>
5.1. Arhitectura generala a aplicației .....	33
5.2. Client.....	34
5.3. Server.....	38
5.4. Modelul bazei de date.....	41
5.5. Precondiții de compilare a aplicației.....	43
5.5.1. Npm .....	43
5.5.2. Apache Cordova .....	43
5.5.3. Android SDK.....	45
5.5.4. Microsoft Visual Studio.....	45
5.6. Compilarea aplicației .....	45
<b>Capitolul 6. Testare și Validare .....</b>	<b>47</b>
6.1. Clase de test la nivel de server.....	47
6.2. Testare manuală .....	48
<b>Capitolul 7. Manual de Instalare și Utilizare .....</b>	<b>51</b>
7.1. Instalarea aplicației și configurarea serverului .....	51
7.2. Conectarea la server.....	51
7.3. Manual de utilizare .....	52
<b>Capitolul 8. Concluzii .....</b>	<b>55</b>
8.1. Realizările obiectivelor propuse .....	55
8.2. Dezvoltări și îmbunătățiri ulterioare .....	56
<b>Bibliografie .....</b>	<b>57</b>
<b>Anexa 1 – Lista figurilor și a tabelor din lucrare .....</b>	<b>59</b>
<b>Anexa 2 – Glosar de termeni și abrevieri .....</b>	<b>61</b>

## Capitolul 1. Introducere – Contextul proiectului

În ziua de astăzi, tehnologia evoluează cu o viteză remarcabilă iar sistemele inteligente se pot regăsi în tot mai multe domenii. În această lucrare vorbim de gestionarea spațiului într-o companie care se poate realiza în manieră automată prin intermediul unui sistem datorită evoluției tehnologice, care impune realizarea de sisteme informatice din ce în ce mai performante și competitive.

Reducerea timpului de procesare, a costurilor, a resurselor umane, dar și a spațiului utilizat este ceea ce se urmărește a fi realizat printr-o automatizare și totodată posibilitatea desfășurării activităților în diferite medii respectiv în diferite locații sau chiar în spațiul virtual.

### 1.1. Contextul proiectului

În societatea actuală, într-o companie, se pune accentul pe performanțe cât mai mari aduse de angajați cât mai numeroși, păstrându-se costurile cât mai scăzute. Una din problemele de cost constă în spațiul necesar pentru desfășurarea activităților.

Dacă analizăm o companie la noi în țară din punct de vedere al logisticii putem observa că fiecare angajat deține propriul birou, propriul spațiu de lucru, chiar și în cazul în care acesta deține un laptop cu ajutorul căruia își poate desfășura activitatea independent de celelalte resurse pe care le are la dispoziție în spațiul personal de lucru precum, monitoare adiționale, tastatură, mouse, telefon, sau elemente de birotică.

Problema se pune însă în momentul în care numărul de angajați crește iar resursele necesare pentru spațiul lor de lucru cresc proporțional cu aceștia.

Soluția clasică este de a extinde desfășurarea activităților unei companii în mai multe clădiri, astfel ca spațiul să fie întotdeauna suficient pentru ca fiecare angajat să aibă propriul birou.

Există însă un aspect important care de cele mai multe ori nu este luat în considerare și anume faptul că într-o companie nu vor fi prezenți toți angajații simultan, unii fiind în delegații, alții fiind în concediu de odihnă sau de boală, alții lucrează de acasă sau din alte spații externe companiei, iar alții au program flexibil. Privind acest aspect, putem considera asignarea unui birou propriu fiecărui angajat, o risipă de resurse de spațiu.

Această lucrare vine cu o soluție modernă în rezolvarea problemei risipei de resurse de spațiu printr-un sistem de monitorizare a spațiului care permite alocarea dinamică de birouri angajaților. În acest fel fiecare angajat va avea posibilitatea să lucreze la un birou când este prezent la serviciu, însă locul său va fi ocupat de alt coleg când acesta nu se află în firmă. Astfel birourile nu vor mai sta neocupate, iar spațiul de lucru necesar va fi mai redus.

## 1.2. Motivația

În momentul de față există numeroase companii care își măresc numărul de angajați de la an la an și totodată și resursele de spațiu. Astfel vom avea din ce în ce mai multe clădiri de birouri, spații utilizate doar opt ore pe zi, căci acesta este programul de lucru al unui angajat cu normă întreagă. În celelalte șaisprezece ore dintr-o zi, biroul unui angajat stă liber și neutilizat.

Privind un aspect important și anume faptul că în general, în multe domenii de activitate precum domeniul tehnologiei informației angajații nu lucrează simultan, ci au program de lucru flexibil se consideră necesară propunerea sistemului de birouri neasignate, care reduce resursele de spațiu.

Pentru început s-au studiat cele două sisteme, cel clasic, al birourilor asignate și cel modern, al birourilor neasignate. Luând în considerare opiniile angajaților după experiența lor cu cele două sisteme, s-a constatat faptul că este necesar ca cele două sisteme să poată fi utilizate simultan, deoarece nu toți angajații se pot adapta cu ușurința la sistemul de birouri neasignate.

Așadar, analizând situația dată, s-a propus un sistem care să ajute angajaților la gestionarea birourilor neasignate. Fiecare angajat face o rezervare pentru un birou, fie în momentul în care îl vede liber și se așază la el, biroul rămânând ocupat până la sfârșitul zilei sau până în momentul în care angajatul marchează eliberarea acestuia, fie în prealabil.

Prin intermediul aplicației, denumite MyWorkplace, un angajat poate să vizualizeze situația birourilor în orice moment, iar pentru a se asigura că biroul este liber și în momentul în care acesta dorește să lucreze acolo, poate să facă rezervarea în prealabil. Realizarea unei rezervări în prealabil presupune căutarea biroului dorit, fie după numărul de identificare în cazul în care acesta este cunoscut, fie prin căutarea clădirii și a încăperii în care acesta se află. După alegerea biroului, se pot vizualiza detalii despre acest în pagina de detaliere.

Pagina de detaliere a birourilor, vizualizată în aplicație, oferă și informații despre locația birourilor în încăpere, astfel se poate realiza o căutare și după acest aspect sau după alte caracteristici ale biroului, precum deținerea unui monitor, a unui telefon, a perifericelor sau posibilitatea unei conexiuni la Internet.

Pentru angajații care nu sunt siguri de locația în care vor lucra în perioada care urmează sau nu își cunosc programul de lucru decât pentru perioade scurte de timp în viitor, există posibilitatea ocupării unui birou ad-hoc. Angajatul caută birourile libere din aplicație, fie făcând o căutare după status, fie se află într-o încăpere și observă un birou liber. În cazul în care acesta găsește un birou liber într-o încăpere, va scana codul QR al biroului, iar în cazul în care biroul nu este rezervat de alt angajat, acesta poate să își facă rezervare ad-hoc.

Această rezervare ad-hoc este valabilă din momentul realizării până la sfârșitul zilei curente sau până în momentul eliberării biroului care se marchează tot prin intermediul unei operații posibile din aplicația MyWorkplace.

Aplicația este utilă pentru a îmbina cele două sisteme de gestionare a birourilor, cel al birourilor asignate și cel al birourilor neasignate, deoarece pentru angajații care preferă modelul clasic de a avea un birou asignat, pot să-și facă o rezervare pentru o perioadă îndelungată de timp, astfel vor putea ocupa același birou în perioada în care sunt prezenți la serviciu.

### 1.3. Conținutul lucrării

În următoarele paragrafe se va prezenta structura lucrării, capitolele și conținutul pe scurt pentru fiecare din ele.

**Capitolul 1** - Introducere - Acest capitol conține o scurtă prezentare a contextului problemei care se dorește a fi rezolvată prin sistemul care s-a realizat, motivația alegerii acestei teme și o scurta descriere a conținutului fiecărui capitol.

**Capitolul 2** - Obiectivele Proiectului - Acest capitol conține prezentarea atât a obiectivului principal al proiectului cât și a obiectivelor secundare care au fost propuse pentru sistemul prezentat.

**Capitolul 3** - Studiul Bibliografic - În acest capitol sunt descrise principalele aspecte legate de sistemele de asignare a birourilor într-o companie, avantaje și dezavantaje ale acestora, dar și impactul pe care acestea le au asupra anjagațiilor. Totodată, acest capitol prezintă o comparație a sistemului care s-a realizat și a altor sisteme similare disponibile pe piața.

**Capitolul 4** - Analiză și Fundamentare Teoretică - Acest capitol conține o descriere a tehnologiilor utilizate în dezvoltarea sistemului precum și motivele pentru care acestea au fost alese, dar și a conceptelor folosite în dezvoltare. Tot în acest capitol sunt prezentate cerințele funcționale, cerințele non-funcționale și cazurile de utilizare ale sistemului, cele mai semnificative fiind prezentate într-un mod mai detaliat.

**Capitolul 5** - Proiectare de Detaliu și Implementare - În acest capitol se prezintă modul în care a fost proiectat sistemul. Se va prezenta schema arhitecturală a sistemului, arhitectura bazei de date și structura tabelor, diagrama de deployment, diagramele de clase, cele mai importante diagrame de secvență precum și diagrama de pachete. Totodată, în acest capitol vor fi detaliate anumite componente ale aplicației considerate relevante în dezvoltarea sistemului.

**Capitolul 6** - Testare și Validare - Acest capitol conține o prezentare a principalelor procese de testare și rezultatele obținute în urma testării. Testarea și validarea sistemului a fost realizată cu scopul de a evalua o parte din cerințele funcționale și cerințele non-funcționale ale sistemului, astfel s-au realizat clase de test la nivel de server, iar pentru aplicația mobilă s-a realizat testare manuală prin diferite scenarii de test, care să ducă la succes cât și scenarii care să ducă la afișarea unui mesaj de eroare.

**Capitolul 7** - Manual de Instalare și Utilizare - Acest capitol prezintă resursele hardware și software necesare pentru instalarea și rularea aplicației. De asemenea acest capitol prezintă modalitatea de configurare a serverului și pașii de urmat în pornirea acsetuia, dar și un manual de utilizare al aplicației.

**Capitolul 8** - În acest capitol sunt prezentate obiectivele realizate prin implementarea sistemului, dar și eventualele dezvoltări ulterioare ale sistemului.



## Capitolul 2. Obiectivele Proiectului

În acest capitol vor fi prezentate obiectivele propuse spre a fi realizate de acest sistem. Aplicația dezvoltată operează la nivel local, într-o firmă și are scopul de a eficientiza gestiunea de birouri și managementul acesteia, întreaga aplicație fiind orientată pe nevoile și cerințele utilizatorului. Aceste obiective sunt necesare și au o importanță ridicată în dezvoltarea proiectului din următoarele motive: este necesar ca la începutul unui proiect să se știe obiectivele care se dorește a fi atinse, pentru a putea constata la final, dar și pe parcurs, dacă scopul inițial al proiectului a fost atins, astfel se reduce riscul realizării unui proiect care să nu satisfacă nevoile clientului.

Obiectivele sunt de două tipuri: obiective principale și obiective secundare. Acest sistem are un singur obiectiv principal care este de asemenea și scopul proiectului, iar obiectivele secundare sunt subscopuri ale obiectivului principal. Obiectivele secundare sunt necesare în vederea realizării de scopuri intermediare. Astfel se poate urmări dacă realizarea proiectului se îndreaptă spre îndeplinirea obiectivului principal sau dacă obiectivele propuse inițial nu au fost respectate întocmai.

Pentru ca obiectivele propuse inițial să fie cât mai calitative și să nu se ajungă la nerealizarea scopului principal, acestea trebuie să fie evaluate după tehnica SMART, conform studiului realizat de Dawson în lucrarea sa [5]. Cea mai potrivită definiție a acronimului SMART este, conform [5] următoarea: specific, măsurabil, adecvat, realist și realizabil într-un timp definit.

Un obiectiv este specific dacă furnizează informație suficientă pentru definirea sarcinii aferente acestuia. Un obiectiv este măsurabil, dacă există o metodă de măsurare a progresului acestuia. Din această caracteristică trebuie să se deducă data de finalizare a acestuia. Un obiectiv este adecvat dacă prin realizarea acestuia se urmărește scopul principal al proiectului. Un obiectiv este realist dacă acesta se poate realiza în timpul pe care proiectul îl are alocat. Un obiectiv deține proprietatea de gestiune a timpului, atâta timp cât în momentul formulării obiectivului se pot identifica durata necesară finalizării lui.

În continuare se va prezenta obiectivul principal al sistemului de gestiune a birourilor, urmat de obiectivele secundare, care vor fi analizate conform tehnicii SMART.

### 2.1. Obiectivul principal

Obiectivul principal al acestui proiect este definirea și implementarea unui sistem care să faciliteze utilizarea birourilor neasignate într-o companie, oferind utilizatorilor posibilitatea de a rezerva birouri, dar și de a căuta un anume coleg în vederea informării legate de amplasarea acestuia în cadrul firmei, la un moment dat. Acest obiectiv se poate atinge prin realizarea obiectivelor secundare prezentare în continuare.

## 2.2. Obiective secundare

Pentru început se vor prezenta câteva obiective funcționale propuse în realizarea sistemului care se vor analiza conform setului de principii SMART, apoi se vor prezenta obiectivele nonfuncționale ale sistemului.

### 2.2.1. Studiul metodelor de aranjare și asignare a birourilor într-o unitate

Ca prim obiectiv secundar se menționează studierea în amănunt a metodelor de aranjare și asignare a birourilor într-o unitate, birouri asignate angajaților și birouri neasignate angajaților. Realizând analiza SMART a acestui obiectiv, observăm că este specific, deoarece ne furnizează informații suficiente necesare pentru a ști cum se realizează.

În continuare se pune accentul pe măsurabilitatea obiectivului. Fiind un obiectiv de cercetare, nu se va ști de la început când se acumulează suficientă cercetare în acest domeniu sau când se parcurge suficientă literatură pentru ca acest obiectiv să fie complet realizat. De asemenea obiectivul este adecvat în urmărirea scopului principal al proiectului, deoarece se urmărește realizarea unui sistem care facilitează metoda de alocare a birourilor neasignate într-o companie.

Obiectivul în discuție este realist în strânsă legătură cu proprietatea de măsurabilitate. În momentul în care s-a decis asupra unui set de referințe bibliografice de bază necesare și suficiente, putem deduce și dacă acest obiectiv va fi realizabil sau nu în timpul pus la dispoziție. De aici deducem și proprietatea de gestiune a timpului propus pentru proiect.

### 2.2.2. Analiza problemei resurselor de spațiu de lucru într-o companie

Un alt obiectiv secundar îl constituie studierea în amănunt a problemei resurselor de spațiu de lucru într-o companie de IT având în vedere flexibilitatea programului angajaților. Realizând analiza SMART constatăm că obiectivul este specific, se poate deduce cu exactitate ceea ce se dorește a fi realizat.

Fiind tot un obiectiv de cercetare, putem afirma faptul că măsurabilitatea nu se poate determina odată cu formularea obiectivului ci doar ulterior. Obiectivul este adecvat deoarece ideea realizării acestui sistem a pornit de la costurile ridicate pe care le aduce spațiul de lucru neutilizat unei companii.

În ceea ce privește realismul obiectivului, acesta este în strânsă legătură cu măsurabilitatea. Când s-a decis cantitatea de informație studiată se poate stabili și timpul necesar îndeplinirii sarcinii, ca apoi să se observe dacă obiectivul este realist sau nu, respectiv dacă se încadrează în timpul alocat de proiect acestui obiectiv.

### *2.2.3. Soluția problemei prezentate*

Realizând cu succes obiectivele anterioare, se ajunge la elaborarea unei soluții în vederea utilizării birourilor neasignate într-o companie. Analiza SMART a acestui obiectiv este mult mai concretă, astfel obiectivul îndeplinește toate cele cinci caracteristici ale tehnicii SMART.

Obiectivul este specific, deoarece definește ceea ce se urmărește prin intermediul acestui obiectiv, este măsurabil, deoarece progresul obiectivului poate fi urmărit, întrucât se cunoaște finalizarea concretă a acestuia, este adecvat, deoarece scopul principal este de a dezvolta un sistem care vine ca soluție a problemei specificate. Este realist, deoarece este realizabil în timpul propus de proiect și este și măsurabil, iar din aceste două proprietăți deducem și o gestiune optimă a timpului. Se poate stabili cât va dura realizarea obiectivului și când va fi finalizat încă de la formularea acestuia.

### *2.2.4. Dezvoltarea aplicației care să implementeze soluția*

În continuare se prezintă obiectivul care specifică implementarea soluției propriuzise. Obiectivul este specific, deoarece având soluția de la obiectivul anterior se cunoaște în amănunt ceea ce se dorește a fi implementat. Obiectivul este măsurabil, întrucât se poate urmări progresul dezvoltării, fie prin interații definite în prealabil, fie prin simpla urmărire a procesului de dezvoltare.

De asemenea, obiectivul este adecvat, deoarece în urma acestui obiectiv se obține o aplicație care urmează a fi utilizată în îndeplinirea scopului principal. Obiectivul acesta este realist, deoarece în urma realizării planului de dezvoltarea a aplicației, s-a constatat ca aplicația v-a fi finalizată în timpul alocat de proiect acestei etape, motiv pentru care putem sublinia și o gestionare a timpului adecvată resurselor proiectului.

### *2.2.5. Evaluarea prototipului realizat*

Ca ultim obiectiv al acestui sistem se menționează evaluarea prototipului realizat la punctul anterior. Acest obiectiv are scopul de a verifica dacă aplicația care a fost realizată este conform cerințelor utilizatorului și îndeplinește toate cerințele propuse de sistem.

Acest obiectiv este specific, deoarece definește concret sarcinile propuse, este măsurabil, deoarece se poate urmări un progres al evaluării, prin diferite cazuri de utilizare, este adecvat, deoarece propune evaluarea și testarea aplicației propuse de obiectivul principal, este realist, deoarece se poate realiza în timpul propus de proiect pentru această etapă, iar încadrându-se în timpul propus, deducem și o gestionare optimă a timpului alocat.

Pe lângă obiectivele funcționale prezentate anterior, sistemul propune și obiective secundare nonfuncționale, care pot fi evaluate conform tehnologiei SMART, în scopul realizării sistemului.

Primul obiectiv secundar pe care sistemul dorește să-l atingă este reprezentat de eficiența în utilizare a acestuia. *Eficiența în utilizare* a sistemului este măsurată prin gradul de expresivitate al interfeței utilizator și prin consistența acesteia. Un utilizator este eficient în utilizarea unui sistem în momentul în care interfața îi este familiară, iar timpul de interacțiune scade din ce în ce mai mult. O interfață eficientă este simplă - ușor de utilizat, care oferă acces rapid la funcții sau comenzi comune, - bine organizată și intuitivă - pentru a fi ușor de utilizat, explicațiile oferite utilizatorilor la primul contact cu aplicația trebuie fiind minime.

Un alt obiectiv este reprezentat de faptul că se dorește realizarea unui sistem care să fie *extensibil*, să permită dezvoltări ulterioare și care să aibă posibilitatea de extindere a funcționalităților. În general, nici un sistem nu este complet; tot timpul acesta poate fi îmbunătățit sau extins, ținându-se pasul și cu schimbările atât la nivel funcțional cât și la nivel tehnologic, fără efort prea mare.

O altă proprietate importantă a sistemului o reprezintă *securitatea*. Se dorește evitarea situațiilor în care un utilizator neautentificat să aibă acces la resursele aplicației, să le poată vizualiza sau modifica. Acestea trebuie protejate prin intermediul unui nume de utilizator și o parolă, astfel fiecare accesare a sistemului necesită autentificare. Obiectivul propus este ca administratorul să aibă posibilitatea să administreze conturile utilizatorilor, să furnizeze acestora nume de utilizator și parolă, după care utilizatorul să aibă posibilitatea de schimbare a parolei, în cazul în care aceasta a fost uitată.

Un alt obiectiv este *accesul diferit la resursele sistemului* în funcție de utilizatorul logat în sistem. Tipurile de utilizatori sunt administratorul de sistem, care este în realitate managerul firmei sau persoana desemnată pentru a se ocupa de logistica firmei și utilizatorul propriu-zis care este de fapt angajatul firmei. Fiecare dintre aceștia trebuie să beneficieze, pe baza autentificării în sistem, de anumite funcționalități specifice pe care sistemul trebuie să le ofere, funcționalități care sunt considerate obiective de implementare.

În concluzie, acest capitol a prezentat obiectivul principal și subobiectivele acestuia, obiectivele secundare, necesare în realizarea sistemului. S-a realizat o analiză SMART a fiecărui obiectiv secundar, evidențiindu-se astfel proprietățile principale ale obiectivelor și importanța lor în dezvoltarea sistemului.

## Capitolul 3. Studiu Bibliografic

În acest capitol se introduc conceptele de birouri neasignate și de rezervare de birouri într-o companie. Deasemenea se prezintă diverse aspecte care compară sistemul nou de asignare a birourilor cu sistemul clasic. Cele mai importante concepte și abordări din domeniu identificate în setul de resurse bibliografice selectate fiind: gestionarea spațiului de lucru, munca centrată pe activitate, influența psihologică asupra angajaților, avantaje și dezavantaje ale sistemului de birouri neasignate. Pentru o cât mai bună specificare a cerințelor funcționale dar și pentru o eventuală poziționare a produsului software în piață se prezintă o analiză și evaluare a sistemelor similare existente pe piață și a funcționalităților pe care acestea le oferă utilizatorilor.

### 3.1. Gestionarea spațiului de lucru într-o companie

Conceptul de *birouri neasignate* definește un model relativ nou de gestionare a spațiului într-o companie, provocând angajații la socializare mult mai amplă în cadrul acesteia. În acest sens au fost făcute mai multe studii care analizează modul în care companiile își organizează spațiul de lucru și felul în care angajații se pot adapta sistemelor diferite de organizare.

Sistemul clasic de organizare a unei companii este acela ca fiecare angajat să aibă propriul birou permanent, un birou, asignat în momentul angajării, care să îl reprezinte. Astfel fiecare angajat interacționează zilnic doar cu colegii din apropiere, fără a schimba vreun parere sau idee cu alți colegi. S-a constatat însă că cele mai originale idei și cele mai bune decizii au fost luate pe coridorul sau în cafeneaua firmei [16], loc unde se întâlnesc colegi din toată compania, nu doar de pe proiectul la care lucrează sau din încăperea unde au amplasat biroul personal.

Pentru a face posibilă interacțiunea cu persoane diferite de cele care sunt pe proiect, s-a introdus noțiunea de birouri neasignate, denumită internațional *hot-desking*. Acest sistem de organizare presupune ca un birou să fie utilizat de mai mulți angajați, lucru posibil în companiile unde angajații lucrează în ture, au program flexibil, sunt în concedii, delegații sau lucrează din alte locații, externe companiei. Astfel, costul în privința spațiului utilizat poate scădea până la 30% [19].

Pentru o bună organizare a spațiului și pentru ca angajații să se poată orienta în căutarea unui birou se menționează în [19] că birourile trebuie să arate la fel, să dețină aceleași facilități iar pentru ca lucrul pe echipe să fie posibil iar angajații să stea așezați grupat, se utilizează pereți colorați, mousepaduri similare sau chiar placute de identificare digitale.

### 3.2. Munca centrată pe activitate

În lucrarea „The Oneness of Eastern Heart & Western Mind in Future Workspaces”, [15] ,Thanh Thuy Truong și-a pus problema dacă o persoană este mai productivă când se concentrează asupra lucrului sau atunci când colaborează sau comunică cu alte persoane, luând în considerare diferși factori externi. Cele două mari teme studiate sunt spațiul interactiv și munca centrată pe activitate.

Sistemul de birouri neasignate pune accent pe munca centrată pe activitate și nu pe locația unde se desfășoară activitatea. Astfel este important ca angajatul să dețină dispozitivele necesare desfășurării activității și să aibă acces la tehnologia corespunzătoare.

Ideea de bază a muncii centrate pe activitate este ca angajatul să poată fi productiv din orice locație, având libertate de alegere în privința modului de lucru, al colegilor, a locației, al timpului de lucru, dar și a instrumentelor necesare activității. Astfel capacitatea unei companii poate fi aproximativ 80% din numărul total de angajați.

Deși pare o modalitate optimă de lucru în sistemul de birouri neasignate, munca centrată pe activitate are pe lângă avantaje și un dezavantaj destul de important și anume angajații își pierd intimitatea pe care o au la birou în sistemul de birouri asignate. Ideea de a se muta zilnic la alt birou presupune și mutarea lucrurilor personale de la un birou la altul, ceea ce aduce după sine frustrare.

Așadar, munca centrată pe activitate redă angajaților libertate de alegere în modul de lucru, dar poate fi și o influență psihică negativă. Acest ultim aspect se va discuta în subcapitolul următor.

### 3.3. Influența psihologică asupra angajaților

Odată cu implementarea unui nou sistem de organizare într-o companie apar probleme de acomodare pentru angajați, care pot duce la tulburări emoționale și psihice.

O parte emoțională care devine confuză este identificarea propriei persoane cu echipa sau cu organizația. Sistemul de organizare clasic, cu birouri asignate fiecărui angajat, duce la o identificare personală cu echipa, pentru că se lucrează în preajma colegilor de echipă, iar comunicarea se realizează direct, față-în-față. Coprezența fizică este necesară în formarea atașamentelor organizaționale, întrucât un angajat se poate identifica cu o grupă de lucru sau cu o echipă.

Pe de alta parte, sistemul de organizare bazat pe birouri neasignate aduce după sine o comunicare directă cu persoane din afara echipei, iar comunicarea în echipă se face online, prin intermediul diferitelor mijloace de comunicare la distanță. Acest aspect duce la o identificare personală cu organizația, de unde deducem faptul că având un birou propriu și împărțind spații apropiate cu aceiași colegi de birou zilnic, se crează legături apropiate cu acei colegi, iar comunicarea decurge natural, cu lejeritate, pe când neavând un birou propriu și schimbând colegii de birou zilnic, comunicare cu aceștia nu mai este antrenată în afara necesităților activității depuse, iar angajatul poate fi mult mai productiv, conform [10].

Dintr-un alt punct de vedere, Luke Stevenson prezintă în publicația sa [14] diferite aspecte care susțin faptul că sistemul de birouri neasignate duce la stres. Conform unui studiu făcut de acesta s-a dovedit că lipsa unui birou stabil duce la o dezorientare emoțională care se remarcă chiar și după ce angajatul părăsește locul de muncă, iar astfel se crează tensiuni în viața de familie iar influențele negative se pot revărsa în special asupra copiilor. De multe ori, angajatul ajunge să ducă o luptă cu sine însuși, pentru a evita conflicte familiale din motive și tensiuni acumulate la locul de muncă, fapt care îi poate provoca traume emoționale și stres.

### 3.4. Avantaje ale sistemului de birouri neasignate

Chiar dacă influențele psihologice nu sunt tocmai pozitive, sistemul de birouri neasignate are a mulțime de avantaje și motive pentru care se încurajează implementarea acestuia. Kim Peterson datează în articolul său [11] avantajele pentru care companiile au trecut la sistemul de birouri asignate. S-a constatat inutilitatea spațiilor de lucru deoarece angajații erau bolnavi, erau în vacanță sau aveau program flexibil în ceea ce privea orele de muncă [4].

Compania Long Island City a transformat un întreg etaj în săli de conferință, săli de ședințe și birouri neasignate angajaților. Conform unei recenzii de afaceri făcută de Harvard în articolul său, s-a constatat că în companie erau doar 150 de spații de lucru disponibile pentru cei 200 de angajați iar fiecare angajat deține câte un dulăpior în care să-și depoziteze lucrurile personale. De asemenea se pune accentul pe eliminarea resurselor fizice precum, foile sau diverse alte notițe și să se treacă la digitalizarea resurselor folosite.

Kevin Roose relatează în articolul său [12] faptul că biroul tipic american, pe care se află un computer personal, câteva hârtii, câteva cărți și o multitudine de lucruri personale care cel mai probabil nu îi sunt necesare angajatului în vederea unei activități productive. Orice se lucrează de la un birou personal, se poate lucra și din alt loc, fie el o sala de ședințe, un birou neasignat sau poate chiar o cafenea.

Sistemul de birouri neasignate are și beneficii din punct de vedere medical. Având în vedere că birourile asignate încurajează munca sedentară, iar „șezutul este noul tip de fumat” [12], prin implementarea sistemului de birouri neasignate, angajații sunt oarecum obligați să se deplaseze de la un birou la altul, făcând astfel mișcare involuntar.

Un alt avantaj prezentat de Kim Peterson în articolul său [11] este importanța coliziunii dintre angajați, care duce la o colaborare intensă și la o performanță crescută, deoarece s-a constatat faptul că statul în fața monitorului nu aduce după sine idei creative, ci este necesară coliziunea (engleza: „collide”) [16] cu alte persoane diversificate și comunicarea directă dintre acestea

Coliziunea cu aceleași persoane zilnic nu duce la idei noi. Deși într-o companie, accentul se pune pe activitate calitativă cât mai multă, s-a dovedit că cel mai bun mod de a realiza activități calitative este de a interacționa direct față-în-față. Iar această comunicare realizată nu numai în interiorul firmei, ci și cu oameni din afara organizației, clienții de exemplu, este importantă în creșterea performanței care reduce gândirea rigidă a angajaților și contribuie la o diversitate mai mare în gândire.

### 3.5. Dezavantaje ale sistemului cu birouri neasignate

Există însă și păreri negative în ceea ce privește sistemul de organizare bazat pe birouri neasignate. Unii angajați preferă „pur și simplu să aibă propriul birou”. Birourile neasignate sunt comparate cu perioada liceului, unde, la finalul zilei, „îți aduni cărțile și caietele și mergi acasă”[2].

De asemenea Peter Ames aduce argumente în publicația sa [1] cum că sistemul de birouri neasignate este preferat în totalitate de conducerea unei companii și nu neaparat de angajații acesteia, enunțând trei teorii în acest sens.

Prima teorie subliniază faptul că sistemul de birouri neasignate este o metoda de eficientizare a spațiului utilizat într-o companie, plecând de la premiza că nu toți angajații sunt simultan la birou. În practică însă, se pune problema ”Ce se întâmplă dacă se decid prea mulți angajați să vină la birou simultan?” [1], întrebare la care trebuie identificate răspunsuri potrivite.

A doua teorie încurajează abordarea colaborativă a lucrului, astfel încât schimbul colegilor de birou să nu influențeze capacitatea de muncă și performanțele pe care un angajat pe poate avea. În practică însă oamenii, în general, nu sunt deschiși la schimbări și preferă să dețină propriul spațiu de lucru, schimbarea biroului și deplasarea dintr-un loc într-altul cu toate lucrurile personale poate deveni un inconvenient pentru majoritatea acestora.

Cea de-a treia teorie [1] susține flexibilitatea orelor de lucru, astfel încât se poate lucra în ture, același birou putând fi utilizat de mai mult de un angajat într-o zi. În practică însă este destul de complicat de gestionat acest aspect, deoarece coordonatorilor le este dificil să țină evidența birourilor.

Tot ca dezavantaj al sistemului de birouri neasignate este menționată în publicația [4] neclaritatea ierarhiei de muncă. Având în vedere că fiecare angajat are dreptul să se așeze unde dorește, se pierde din principiul ierarhic de arganizare a unei companii, astfel încât unii pot fi intimidați dacă la biroul de langă se află managerul și pot avea productivitate redusă din acest motiv.

### 3.6. Concluzii

Asadar, după cum s-a detaliat în acest capitol, sistemul de birouri neasignate ajută la gestionarea mai eficientă a spațiului de lucru într-o companie, dar și la producerea de idei mult mai creative și la o productivitate ridicată.

Există de asemenea și aspecte negative și influențe psihologice aduse de acest sistem, însă făcând un bilant s-a constatat că aspectele pozitive ale implementării sistemului de birouri neasignate sunt mult peste cele negative iar sistemul aduce beneficii întregii companii, atat angajatorilor, conducerii, cât și angajatului [10].



### 3.7. Sisteme similare

Odata cu implementarea sistemului de birouri neasignate apare un disconfort al angajaților care poate fi rezolvat prin sistemul creat prin intermediul acestei lucrări. Fiecare angajat se întreabă zilnic ce birou este liber sau dacă biroul la care a stat cu o zi înainte e încă disponibil sau chiar și amplasarea colegilor îi este necunoscută în acest caz.

Sistemul creat prin intermediul acestei lucrări vine în rezolvarea acestei probleme. Pentru o cat mai corectă și completă identificare a funcționalităților unui astfel de sistem, au fost căutate și selectate un set de sisteme similare a căror analiză va fi prezentată în continuare.

#### 3.7.1. Desk Booking Software

Desk Booking Software [3] furnizat de firma Condeco este un sistem de monitorizare a birourilor într-o firma. Oferă posibilitatea de a organiza birourile în funcție de preferința angajaților, birouri neasignate, birouri fixe sau birouri flexibile, de a rezerva un birou, de a vizualiza toate rezervările, dar și de a găsi ușor colegii și biroul unde aceștia sunt amplasați.

Ideea de bază este ca locul de muncă să fie mult mai dinamic și să aibă un cost redus. Rezervările sunt integrate în Microsoft Outlook, astfel rezervările se fac direct din calendarul specific Microsoft Outlook. De asemenea sistemul permite rezervările de grup, în cazul unei echipe care trebuie să lucreze împreună.

#### 3.7.2. Sistemul de rezervări electronice EMS

EMS (Electronic Meeting Systems) [7] este un sistem care ajută la gestionarea spațiului, a resurselor, angajaților în vederea organizării de ședințe sau evenimente, economisind timp scurt și resurse financiare puține. EMS este de asemenea util pentru rezervarea de birouri sau săli de conferință pentru diverse ședințe sau întâlniri și permite căutarea acestora după diferite criterii precum tipul, capacitatea sau echipamentul disponibil în aceste săli.

Sistemul este o aplicație Desktop Client care permite interacțiunea simplă a utilizatorului cu sistemul. Aplicația este destinată mai multor utilizatori, cu experiență sau fără, utilizatori care necesită îndrumare sau care nu necesită îndrumare. Înainte ca aplicația să fie implementată, procesul manual de rezervare dura 13 minute. Cu ajutorul aplicației, procesul de rezervare a fost redus la 6 minute, iar managementul și comunicarea între ședințe sunt realizate mai eficient, astfel încât s-a reușit eliminarea ședințelor duplicate realizate involuntar.

### 3.7.3. *DeskFlex*

DeskFlex [6] este un sistem de rezervări pentru spații de lucru, săli de conferință, birouri, locuri de parcare sau echipament tehnic. Sistemul este destinat oricărei persoane, interne sau externe unei firme, astfel, persoane străine pot rezerva un birou, o sală de conferință sau un loc de parcare.

De asemenea sistemul permite o viziune de ansamblu asupra întregii clădiri, astfel încât un angajat poate vedea ce birouri sunt libere respectiv ocupate. În acest fel, angajatul poate face o rezervare a unui birou în apropierea unui coleg de echipă.

Sistemul permite rezervări prelabile sau ocuparea birourilor libere într-un anumit moment actual fără rezervare prealabilă. Toate aceste facilități sistemul le oferă dintr-o aplicație integrată în Microsoft Outlook.

### 3.7.4. *Smartway2*

Smartway2 [13] este un sistem de rezervări a sălilor de sedințe și a birourilor într-o companie. Aplicația permite vizualizarea facilităților unei săli de sedințe, disponibilitatea sălilor și permite utilizatorului să aleagă sala dorită. Acesta poate să rezerve sala în prealabil sau poate să o ocupe imediat, odată ce aceasta este liberă.

Modalitățile de rezervare a unei săli, respectiv a unui birou, sunt autentificarea pe baza unei parole, scanare de coduri QR, bluetooth, sau chiar și recunoaștere facială, toate acestea pentru a evita orice conflicte cauzate de rezervări duble sau birouri ocupate, chiar dacă nu este nimeni așezat la acestea.

De asemenea în cazul în care se ajunge mai repede la o ședință, Smartway2 are facilitatea de a îndruma participanții acelei sedințe la cea mai apropiată cafenea față de sala de ședințe pe care aceștia o au rezervată.

### 3.7.5. *Room & Resource booking*

Room & Resource booking [8] este un sistem dezvoltat de firma Essential pentru a rula în Microsoft Exchange și în Microsoft Outlook. Acesta oferă posibilitatea de rezervare și de viziune de ansamblu asupra tuturor sălilor de sedințe, a birourilor neasignate, dar și a altor facilități precum, capacitatea maximă a unei încăperi sau săli de sedințe, echipamentul disponibil, planul clădirii și cum se ajunge acolo.

De asemenea sistemul oferă și posibilitatea de solicitare catering, vizualizare și rezervare a locurilor de parcare și a permiselor de acces ale vizitatorilor. Toate aceste rezervări se pot realiza în prealabil sau ad-hoc.

3.7.6. *Comparația sistemului dezvoltat în această lucrare cu sistemele similare*

Având în vedere sistemele similare prezentate mai sus, se constată atât asemănări cât și deosebiri între aplicația MyWorkplace și cele prezentate anterior. Aceste diferențe se pot observa în tabelul următor.

Conform tabelului 3.1 constatăm ceea ce aduce în plus aplicația MyWorkplace, comparativ cu cele existente deja pe piață.

Pe lângă faptul că aceasta a fost dezvoltată ca necesitate a unei aplicații interne în cadrul unei companii pentru gestionarea de birouri, aceasta aduce funcționalități noi care implică funcțiile native ale telefonului, precum calendarul și camera foto, cea din urmă fiind utilizată pentru scanarea de coduri QR.

	Desk Booking Software	EMS	DeskFlex	Smartway2	Room & Resource booking	MyWorkplace
MS Outlook	X	-	X	-	X	-
Aplicație Mobilă	-	-	-	-	-	X
Rezervări prealabile	X	X	X	X	X	X
Rezervări ad-hoc	-	-	X	X	X	X
Vizualizare rezervare in calendar	-	-	-	-	-	X
Indicații google maps	-	-	-	-	-	X
Facilități birouri	-	X	-	-	X	X
Căutare birou	X	X	X	X	X	X
Căutare coleg	X	-	-	-	-	X
Scanare coduri QR	-	-	-	X	-	X

Tabel 3.1 Comparație între sisteme

Realizând această analiză comparativă, putem concluziona pe lângă faptul că aplicația MyWorkplace oferă funcționalități noi față de celelalte produse similare existente, că în acest capitol s-au prezentat diverse aspecte care compară sistemul nou de asignare a birourilor cu sistemul clasic. Acestea fiind: gestionarea spațiului de lucru, munca centrată pe activitate, influența psihologică asupra angajaților, avantaje și dezavantaje ale sistemului de birouri neasignate.

## Capitolul 4. Analiză și Fundamentare Teoretică

Acest capitol va prezenta cerințele funcționale descrise sub forma cazurilor de utilizare, cerințele non-funcționale ale sistemului și tehnologiile care au fost utilizate pentru implementarea sistemului.

### 4.1. Cerințele aplicației

Cerințele unui sistem pot fi clasificate în cerințe funcționale și cerințe non-funcționale. Cerințele funcționale prezintă o descriere completă a funcționalităților pe care sistemul trebuie să le îndeplinească, care să se poată realiza utilizând sistemul. Cerințele non-funcționale dictează proprietăți și constrângeri asupra sistemului și sunt specificate atribute de calitate pe care sistemul trebuie să le dețină.

#### 4.1.1. Cerințe funcționale

În ingineria software, o cerință funcțională definește o funcționalitate a sistemului, sau a unei componente a sistemului. O funcționalitate poate fi descrisă ca un set de variabile de intrare, un comportament specific funcționalității și un set de variabile de ieșire. Cerințele funcționale pot fi calcule, detalii tehnice, manipulări sau procesări de date, sau alte funcționalități specifice care definesc ce ar trebui să realizeze un sistem.

Pentru o gestiune mai ușoară a cerințelor funcționale ale sistemului, acestea vor fi prezentate în subcapitolele următoare în funcție de tipurile de utilizatori care pot beneficia de acestea, respectiv vor fi grupate în funcție de tipurile de acțiuni pe care le reprezintă.

#### Cerințe funcționale corespunzătoare secțiunii de autentificare:

Identificator	Cerință funcțională	Utilizatori
CF 1.1	Autentificare	Toți utilizatorii
CF 1.2	Resetare parola	Toți utilizatorii
CF 1.3	Logout	Toți utilizatorii

Tabel 4.1 Cerințe funcționale corespunzătoare secțiunii de autentificare

**Cerințe funcționale corespunzătoare administratorului:**

<b>Identificator</b>	<b>Cerință funcțională</b>	<b>Utilizatori</b>
CF 2.1	Administrare conturi utilizatori	Administrator
CF 2.2	Blocarea unui cont utilizator	Administrator
CF 3.1	Administrare sedii	Administrator
CF 3.2	Administrare încăperi	Administrator
CF 3.3	Administrare birouri(mese)	Administrator
CF 4	Administrare rezervări	Administrator
CF 5.1	Căutare după rezervări	Administrator
CF 5.2	Căutare după utilizatori	Administrator

Tabel 4.2 Cerințe funcționale corespunzătoare administratorului

**Cerințe funcționale corespunzătoare căutării unui birou:**

<b>Identificator</b>	<b>Cerință funcțională</b>	<b>Utilizatori</b>
CF 6.1	Căutare clădire și vizualizare disponibilitate birouri	Angajat
CF 6.2	Căutare încăpere și vizualizare disponibilitate birouri	Angajat
CF 7	Filtrare în lista de birouri după caracteristici	Angajat
CF 8	Vizualizare localizare clădire	Angajat

Tabel 4.3 Cerințe funcționale corespunzătoare căutării unui birou

**Cerințe funcționale corespunzătoare asignării unui birou fără rezervare prealabilă:**

<b>Identificator</b>	<b>Cerință funcțională</b>	<b>Utilizatori</b>
CF 9.1	Marcare birou prin scanare de cod QR	Angajat
CF 9.2	Rezervare birou ad-hoc	Angajat

Tabel 4.4 Cerințe funcționale corespunzătoare asignării unui birou

**Cerințe funcționale corespunzătoare rezervării unui birou:**

<b>Identificator</b>	<b>Cerință funcțională</b>	<b>Utilizatori</b>
CF 10.1	Rezervare birou	Angajat
CF 10.2	Anulare rezervare birou	Angajat
CF 10.3	Vizualizare listă de rezervări	Angajat
CF 10.4	Trimitere email când s-a rezervat sau anulat o rezervare	Angajat

Tabel 4.5 Cerințe funcționale corespunzătoare rezervării unui birou

Precum sunt definite în ingineria software, cerințele funcționale specifică rezultatele particulare ale unui sistem. Acestea sunt în contrast cu cerințele non-funcționale care specifică caracteristicile de calitate ale sistemului care referă modul de implementare și funcționare a sistemului, cum ar fi costul sau fiabilitatea.

#### *4.1.2. Cerințe non-funcționale*

Spre deosebire de cerințele funcționale ale sistemului, cerințele non-funcționale reprezintă proprietăți și constrângeri impuse asupra sistemului. Analizând orice sistem informatic observăm că cerințele non-funcționale sunt catalogate ca factori de calitate ai sistemului. În următoarele paragrafe v-a fi descris și analizat care sunt principalele calități și constrângeri impuse sistemului.

#### **Utilizabilitatea**

Această proprietate a unui sistem reprezintă ușurința cu care un sistem poate fi învățat și utilizat, astfel sistemul propus își dorește ca timpul de acomodare și învățare să fie cât mai redus. O astfel de cerință non-funcțională poate face diferență între un sistem de real succes, sau un sistem care eșuează.

Utilizabilitatea sistemului este dată și de designul interfeței grafice, dar și intuitivitatea acesteia, plasarea elementelor în locații specifice, locații cu care utilizatorul este obișnuit de la sistemele pe care le utilizează frecvent. O aplicație cu o interfață grafică cât mai prietenoasă și intuitivă, chiar incompletă din punctul de vedere al funcționalităților poate fi preferată de utilizatori în detrimentul unei aplicații complete care acoperă multe din cerințe funcționale de interes utilizatorului, dar care nu beneficiază de o interfață grafică adecvată.

Utilizabilitatea este susținută și de eficiența pe care o au utilizatorii în îndeplinirea diferitelor sarcini, eficiență care se garantează prin accesul rapid la resursele disponibile.

### **Disponibilitatea**

Această proprietate a sistemului reprezintă măsura în care sistemul trebuie să funcționeze pentru utilizatori. Ținând cont de faptul că sistemul dorește să devină o alternativă a procesului de management al birourilor dintr-o companie, în care activitatea se poate desfășura pe tot parcursul unei zile, poate chiar și noaptea, în funcție de necesitate, disponibilitatea este una din cele mai importante cerințe pe care sistemul trebuie să le acopere.

Disponibilitatea maximă care se impune este ca sistemul să fie disponibil utilizatorilor 24 ore din 24, iar timpul de revenire într-o stare stabilă a sistemului în cazul unui eșec să nu depășească 30 de minute, maxim o oră. Disponibilitatea unui sistem este susținută și de cât este de sigur un sistem, cât este de robust și tolerabil la erorile care pot să intervină.

Validarea datelor reprezintă cheia către un sistem robust și tolerant, datele introduce de utilizatori pot să fie adesea incorecte sau neformatate corect, lucrul care sistemul trebuie să-l detecteze.

### **Performanța**

Această proprietate a unui sistem se rezumă în cele mai multe cazuri la timpul maxim de răspuns care este pus la dispoziție sistemului pentru a oferi feedbackul necesar în urma unei acțiuni declanșate de utilizator. Depinzând de context, performanța poate implica timpul de răspuns pentru un request care trebuie să fie mic, rata de procesare a informației, utilizarea redusă a resurselor calculatorului sau timpul scurt de transmitere a datelor.

În cazul sistemului prezentat, principala activitate pe care o execută utilizatorul este de a căuta un birou și de a efectua o rezervare a acestuia. Din punct de vedere al performanței, un proces nu trebuie să depășească 10 secunde pentru rezervarea unui birou. De asemenea încărcarea unei pagini nu trebuie să dureze mai mult de 2 secunde.

### **Scalabilitatea**

În ingineria software, scalabilitatea este un principiu de design al sistemului, astfel încât implementarea este făcută luând în considerare posibilitatea de creștere ulterioară a sistemului. Este o măsură sistemică a capacității de a extinde un sistem și nivelul de efort necesar pentru a pune în aplicare sau a implementa extinderea.

Scalabilitatea aplicației MyWorkplace este măsurabilă prin numărul de utilizatori care pot utiliza simultan aplicația. Având în vedere resursele cloud pe care le folosește aplicația, putem menționa cele două forme de scalabilitate, pe verticală și pe orizontală. Scalabilitatea pe verticală se obține prin extinderea resurselor hardware precum, procesorul, memoria RAM, spațiul de stocare sau lățimea de bandă de rețea. Scalabilitatea pe orizontală se obține prin adăugarea unor noi noduri precum, calculatoare destinate procesării aceleiași cereri, la cele existente [9] și este realizată în manieră simplă în context cloud.



### **Extensibilitatea**

Această proprietate, numită și **modificabilitate** este definită ca abilitatea unui sistem de a avea noi funcționalități sau de a le extinde pe cele existente, unde structura internă a sistemului și fluxul de date sunt afectate minim sau deloc.

### **Securitatea**

Această proprietate a unui sistem este caracterizată pe lângă mecanismele folosite pentru protejarea datelor și împiedicarea accesului neautorizat în aplicație, de mecanisme folosite pentru restricționarea accesului asupra resurselor importante ale sistemului. Parolele salvate în baza de date sunt criptate iar pentru fiecare cerere care ajunge la server se verifică dacă există un utilizator autentificat.

### **Fiabilitatea**

Capabilitatea sistemului de a funcționa corect în anumite condiții date pentru o anumită perioadă de timp este definită prin fiabilitate. Aceasta se poate măsura ca probabilitate de succes, (Fiabilitatea = 1 - Probabilitate de eșec) sau ca frecvență de eșec.

### **Transparenta**

Aceasta este proprietatea sistemului de a permite utilizatorului să vizualizeze în orice moment acțiunile care au fost făcute anterior. Transparenta se definește ca și cantitatea percepută a informațiilor partajate în mod intenționat de la un expeditor [20].

### **Portabilitate**

Această proprietate definește capabilitatea sistemului de a fi utilizat în diverse medii. Sistemul MyWorkplace este produs pentru diferite sisteme de operare, iar portabilitatea este cheia pentru implementarea unor costuri reduse.

### **Compatibilitatea multiplatformă**

Proprietatea de **compatibilitatea multiplatformă** o regăsim la acest sistem cu ajutorul framework-ului SAPUi5 care ne permite să scriem codul o singură dată și să compilăm codul pentru toate sistemele de operare mobile cunoscute, Android, Windows Phone și iOS.

ID	Descriere
CNF 1	Performanta
CNF 2	Scalabilitate
CNF 3	Utilizabilitate
CNF 4	Fiabilitate
CNF 5	Disponibilitate
CNF 6	Securitate
CNF 7	Mentenabilitate
CNF 8	Modificabilitate/Extensibilitate
CNF 9	Transparenta
CNF 10	Portabilitate
CNF 11	Multi-platform compatibility / compatibilitatea multi-platformă

Tabel 4.6 Cerințe nonfuncționale

## 4.2. Cazuri de utilizare

În ingineria sistemelor software, un caz de utilizare este o listă de acțiuni sau etape de evenimente, care definesc în mod tipic interacțiunile dintre un rol, cunoscut în UML (Unified Modeling Language) ca actor și un sistem, pentru a atinge un obiectiv. Actorul poate fi un om sau un sistem extern.

Cazurile de utilizare au menirea să ofere o perspectivă globală asupra comportamentului și funcționalităților puse la dispoziție de către un sistem pentru utilizatori. Cerințele funcționale ale sistemului sunt cazuri de utilizare posibile, însă nu toate cerințele funcționale trebuie tratate ca și cazuri de utilizare. Cerințe precum create, read, update sau delete nu necesită prezentarea sub forma unui caz de utilizare.

Un scenariu al cazului de utilizare poate fi definit ca o secvență de acțiuni și interacțiuni ce descriu actorii care utilizează sistemul pentru a îndeplini o anumită sarcină. Cazurile de utilizare corespunzătoare aplicației vor fi descrise atât sub forma unei diagrame de cazuri de utilizare cât și în reprezentare textuală.

### 4.2.1. Actorii sistemului

Tipurile de utilizatori care se găsesc în sistem sunt:

- Administrator
- Utilizator autentificat

### 4.2.2. Cazuri de utilizare

În diagramele de cazuri de utilizare reprezentate în acest sub-capitol sunt ilustrate acțiunile pe care le poate realiza administratorul și utilizatorul autentificat, angajatul. Se poate observa o diferență majoră din punct de vedere al acțiunilor care se pot executa. În figura 4.1 sunt prezentate cazurile de utilizare specifice utilizatorului cu drepturi de administrator.

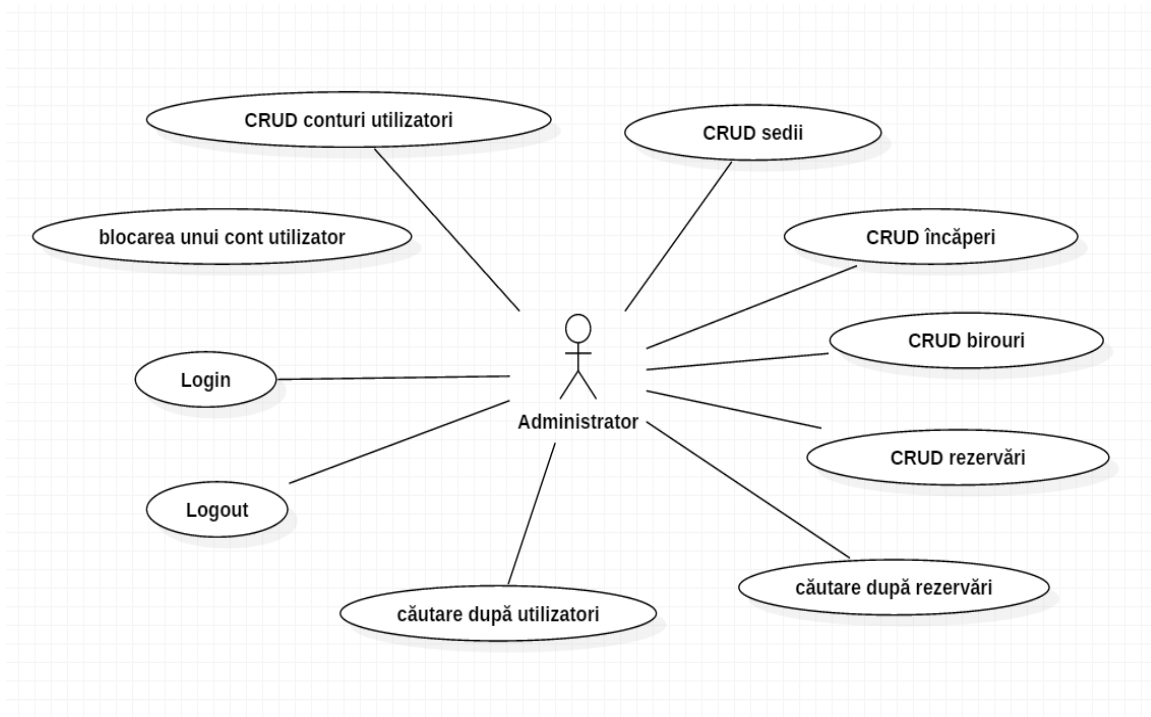


Figura 4.1 Cazuri de utilizare ale administratorului

În figura 4.2 se prezintă cazurile de utilizare specifice utilizatorului autentificat, care este angajatul.

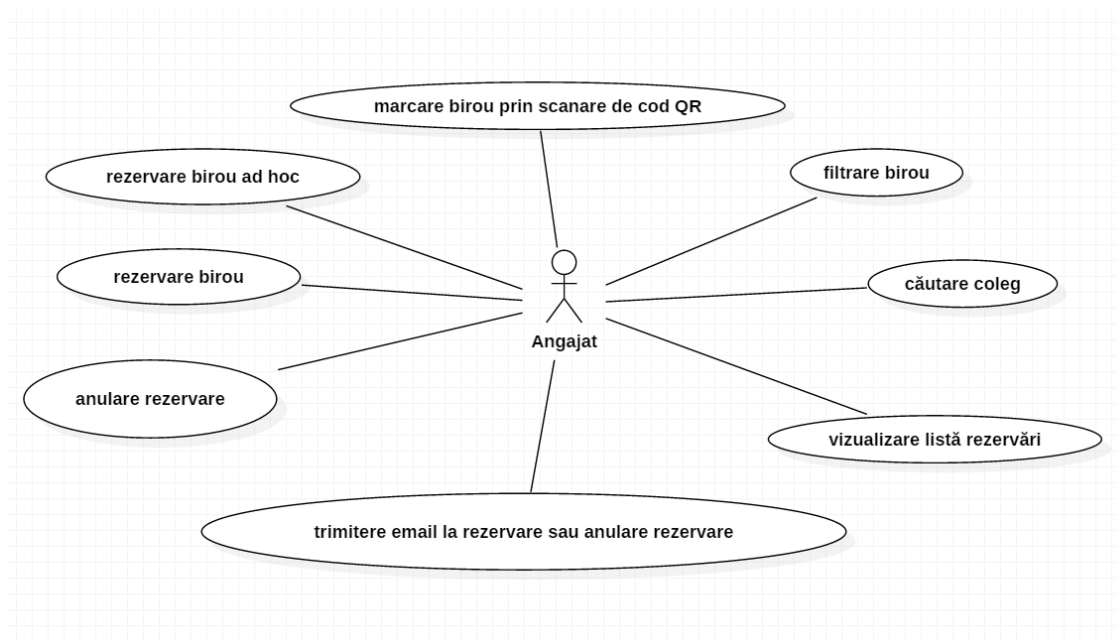


Figura 4.2 Cazuri de utilizare specifice angajatului

### 4.2.3. Descrierea detaliată a cazurilor de utilizare

În continuare vor fi analizate cele mai importante cazuri de utilizare, pentru fiecare din acestea specificându-se actorul principal, participanții cu interes, precondiții, postcondiții, scenariul de success și scenarii alternative.

#### CU1

**Numele cazului de utilizare:** Rezervare birou

**Actor principal:** Actorul principal al scenariului de utilizare este angajatul unei firme în care s-a introdus sistemul de birouri neasignate iar ca mediu de gestionare a acestora se utilizează aplicația MyWorkplace dezvoltată prin aceasta lucrare.

**Participanți cu interes (Stakeholders):** Principalul participant cu interes al acestui scenariu este angajatul, deoarece este în interesul său să facă acea rezervare de birou, pentru a ști de unde va lucra în perioada ce urmează, sau poate doar să aleagă un birou preferat din companie. Tot ca participanți cu interes putem considera și managerul firmei, care poate fi interesat sau doar curios de rezervările făcute de angajații săi, sau poate dorește să facă doar un bilanț legat de disponibilitatea birourilor din firmă.

**Precondiții:** Precondițiile necesare pentru realizarea acestui scenariu sunt conectarea la o sursă de internet, descărcarea aplicației, instalarea acesteia și autentificarea în sistem, care se realizează printr-un nume utilizator și o parolă furnizate de manager în momentul angajării.

**Postcondiții:** După realizarea acestui scenariu, rezervarea trebuie să fie văzută în calendarul dispozitivului mobil, iar emailul trimis de aplicație în momentul rezervării trebuie să poată fi citit din aplicația de email a utilizatorului. De asemenea aplicația poate fi utilizată pentru realizarea unei alte rezervări, sau în alte scopuri, de exemplu, căutarea unui coleg, în vederea informării amplasării acestuia în cadrul firmei la un moment dat.

**Scenariul de succes:**

1. Utilizatorul alege din pagina principală opțiunea de „căutare birou” (search desk).
2. Se încarcă lista cu clădirile disponibile și se observă disponibilitatea birourilor din fiecare clădire.
3. Utilizatorul selectează clădirea dorită, vizualizează detaliile despre clădire, apoi apasă butonul „vizualizare încăperi” (view rooms).
4. Se încarcă încăperile clădirii alese de utilizator și se afișează disponibilitatea birourilor la nivel de încăpere.
5. Utilizatorul selectează încăperea dorită, vizualizează detaliile despre încăpere, apoi apasă butonul „vizualizare birouri” (view desks).
6. Se încarcă lista cu birourile încăperii alese anterior și se afișează disponibilitatea acestora.
7. Utilizatorul alege un birou.
8. Utilizatorul apasă butonul „rezervă” (reserve).
9. Se încarcă o nouă pagină unde utilizatorului îi este cerut intervalul rezervării care este format din data și ora de început și de final a rezervării.
10. Utilizatorul introduce datele cerute la pasul anterior și apasă butonul „ok”.

Scenariul este realizat în momentul în care mesajul de informare a rezervării este afișat, utilizatorul primește un email de informare în legătură cu rezervarea realizată.

**Scenarii alternative:**

- \*a. În orice pas al scenariului de succes sistemul se blochează:
  1. Angajatul se reautentifică în sistem și reia rezervarea.
- 2a. Nu sunt birouri disponibile în clădirea selectată.
  1. Utilizatorul alege altă clădire.
- 4a. Nu sunt birouri disponibile în încăperea respectivă sau birourile disponibile nu sunt pe placul său.
  1. Utilizatorul alege altă încăpere.
- 10a. Datele și orele dorite nu sunt disponibile.
  1. Utilizatorul alege alt birou.
- 10b. Datele și orele nu sunt valide.
  1. Se afișează mesaj de eroare.
- 10c. Sistemul nu poate salva rezervarea.
  1. Se afișează mesaj de eroare.

Diagrama urmatoare concretizează fluxul de evenimente declanșat în cazul de utilizare pentru rezervarea unui birou.

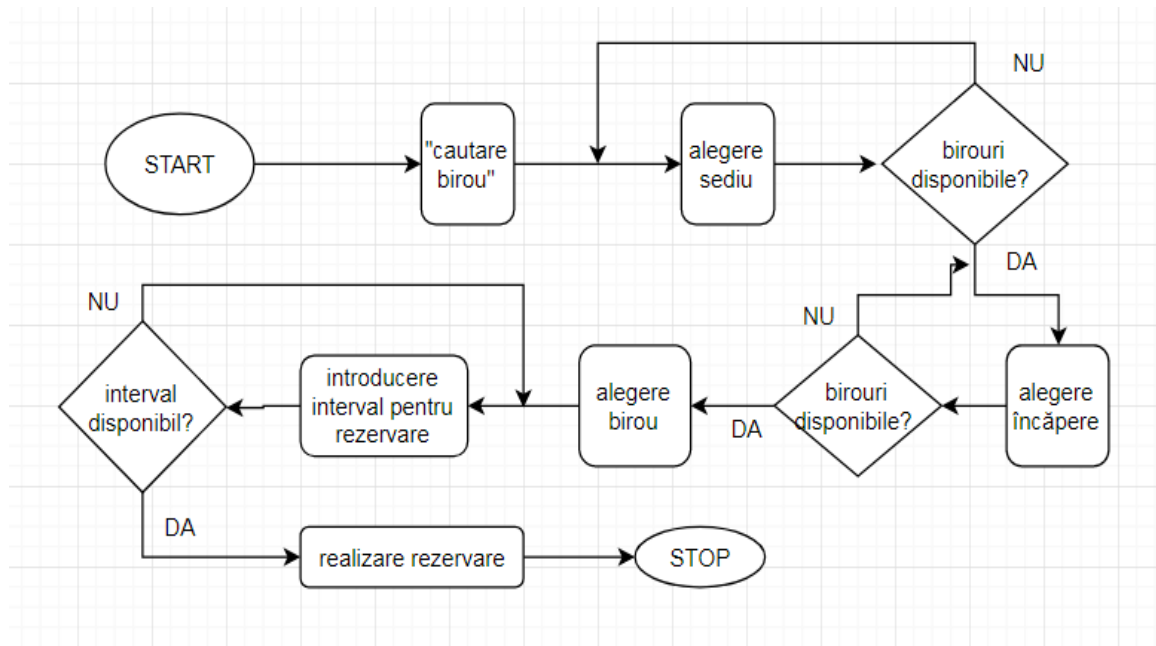


Figura 4.3 Diagrama de flux pentru cazul de utilizare CU1

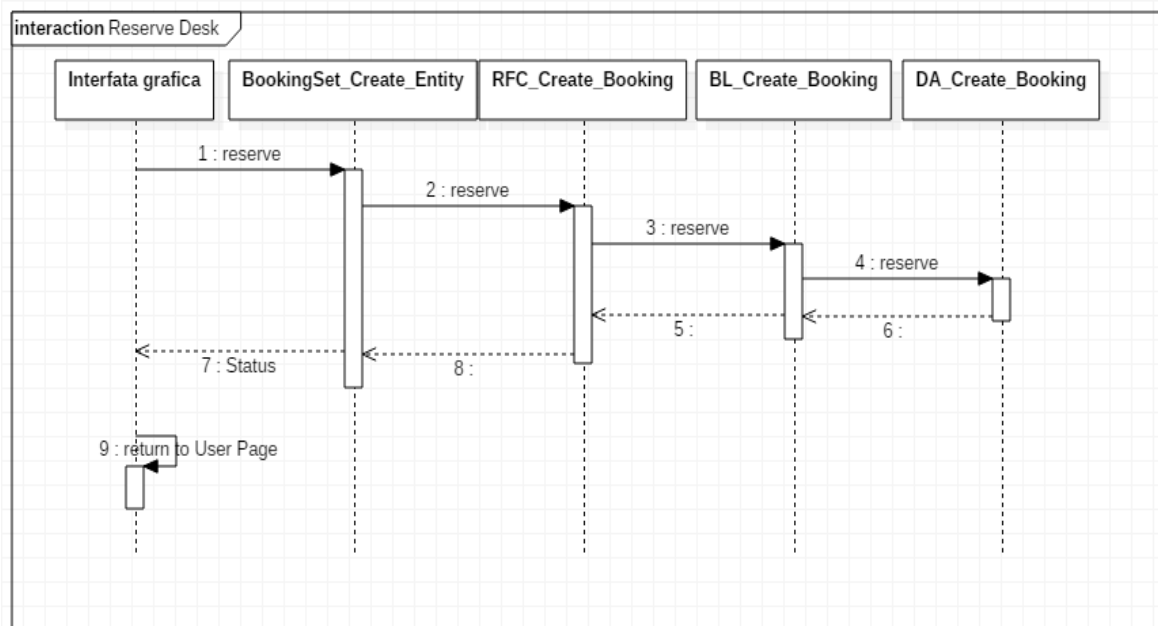


Figura 4.4 Diagrama de secventa a cazului de utilizare CU1

## CU2

**Numele cazului de utilizare:** Anulare rezervare birou

**Actor principal:** Actorul principal al scenariului de utilizare este angajatul unei firme în care s-a introdus sistemul de birouri neassignate iar ca mediu de gestionare a acestora se utilizează aplicația MyWorkplace, dezvoltată prin această lucrare.

**Participanți cu interes (Stakeholders):** Principalul participant cu interes al acestui scenariu este angajatul, deoarece este în interesul său să anuleze o rezervare de birou, pentru a nu ține blocat un birou, dacă nu intenționează să îl ocupe în perioada respectivă sau poate doar pentru a putea alege un alt birou preferat din companie. Tot ca participanți cu interes putem considera și managerul firmei, care poate fi interesat sau doar curios de rezervările făcute de angajații săi, sau poate dorește să facă doar un bilanț legat de disponibilitatea birourilor din firmă.

**Precondiții:** Precondițiile necesare pentru realizarea acestui scenariu sunt conectarea la o sursă de internet, descărcarea aplicației, instalarea acesteia și autentificarea în sistem, care se realizează printr-un nume utilizator și o parola furnizate de manager în momentul angajării.

**Postcondiții:** După realizarea acestui scenariu, rezervarea trebuie să nu mai fie văzută în calendarul dispozitivului mobil, iar emailul trimis de aplicație în momentul anulării trebuie să poată fi citit din aplicația de email a utilizatorului. De asemenea aplicația poate fi utilizată pentru realizarea unei alte rezervări, sau în alte scopuri, de exemplu, căutarea unui coleg, în vederea informării amplasării acestuia în cadrul firmei la un moment dat.

### Scenariul de succes:

1. Utilizatorul alege din pagina principală opțiunea „rezervările mele” (myBookings).
2. Se încarcă lista cu toate rezervările utilizatorului logat în sistem.
3. Utilizatorul selectează rezervarea dorită.
4. Se încarcă o nouă pagină unde utilizatorul poate vedea detaliile rezervării, apoi apasă pe butonul „anulare” (cancel).

Scenariul este realizat în momentul în care mesajul de informare a anulării este afișat, utilizatorul primește un email de informare în legătură cu anularea realizată.

### Scenarii alternative:

\*a. În orice pas al scenariului de succes sistemul se blochează:

1. Angajatul se reautentifică în sistem și reia procesul de anulare.
- 4a. Sistemul nu poate anula rezervarea.
1. Se afișează mesaj de eroare.

### CU3

**Numele cazului de utilizare:** Căutare birou

**Actor principal:** Actorul principal al scenariului de utilizare este angajatul unei firme în care s-a introdus sistemul de birouri neasignate iar ca mediu de gestionare a acestora se utilizează aplicația dezvoltată prin aceasta lucrare.

**Participanți cu interes (Stakeholders):** Principalul participant cu interes al acestui scenariu este angajatul, deoarece este în interesul său să caute un birou, pentru a alege un birou preferat din companie sau doar pentru a vizualiza disponibilitatea acestuia.

**Precondiții:** Precondițiile necesare pentru realizarea acestui scenariu sunt conectarea la o sursă de internet, descărcarea aplicației, instalarea acesteia și autentificarea în sistem, care se realizează printr-un nume utilizator și o parolă furnizate de manager în momentul angajării.

**Postcondiții:** După realizarea acestui scenariu, angajatul este informat despre facilitățile biroului căutat și de disponibilitatea acestuia. De asemenea aplicația poate fi utilizată pentru realizarea unei rezervări, sau în alte scopuri, de exemplu, căutarea unui coleg, în vederea informării amplasării acestuia în cadrul firmei la un moment dat.

#### Scenariul de succes:

1. Utilizatorul alege din pagina principala opțiunea „vizualizare birouri” (view desks).
2. Se încarcă lista completă de birouri.
3. Utilizatorul scrie în bara de căutare numărul biroului căutat și apasă „căutare” (search).
4. În lista inițială apare doar biroul căutat.
5. Utilizatorul selectează biroul.
6. Se încarcă o nouă pagină unde utilizatorul vizualizează detaliile despre birou.

Scenariul este realizat în momentul în care utilizatorul vizualizează detaliile despre biroul ales.

#### Scenarii alternative:

\*a. În orice pas al scenariului de succes sistemul se blochează:

1. Angajatul se reautentifică în sistem și reia rezervarea.

4a. Nu s-a găsit biroul căutat, lista este goală.

1. Utilizatorul introduce alt număr de birou, sau alte caracteristici ale biroului precum, denumirea clădirii în care se află, denumirea încăperii, sau chiar și denumirile facilităților biroului căutat.



## CU4

**Numele cazului de utilizare:** Blocare cont utilizator

**Actor principal:** Actorul principal al scenariului de utilizare este administratorul unei firme în care s-a introdus sistemul de birouri neasignate iar ca mediu de gestionare a acestora se utilizează aplicația myWorkplace dezvoltată prin aceasta lucrare.

**Participanți cu interes (Stakeholders):** Principalul participant cu interes al acestui scenariu este managerul, deoarece este în interesul său, dar și al firmei, ca un angajat care nu mai lucrează în firmă să nu mai poată realiza rezervări în cadrul acesteia.

**Precondiții:** Precondițiile necesare pentru realizarea acestui scenariu sunt conectarea la o sursă de internet, descărcarea aplicației, instalarea acesteia și autentificarea în sistem, care se realizează printr-un nume utilizator și o parolă specifice administratorului.

**Postcondiții:** După realizarea acestui scenariu, utilizatorul care se dorește a fi blocat, nu mai poate să se autentifice în sistem. De asemenea aplicația poate fi utilizată pentru introducerea unui nou angajat în sistem, sau în alte scopuri, de exemplu, căutarea unui coleg, în vederea informării amplasării acestuia în cadrul firmei la un moment dat, căutarea unui birou sau vizualizare tuturor rezervărilor în vederea realizării unor bilanțuri..

### Scenariul de succes:

1. Administratorul alege din pagina principala opțiunea „utilizatori” (users).
2. Se încarcă lista cu toți utilizatorii.
3. Administratorul introduce în bara de selectie numele, prenumele, numele de utilizator sau emailul utilizatorului dorit și apasă butonul „căutare” (search)
4. În listă inițială apare doar utilizatorul căutat.
5. Administratorul selectează utilizatorul căutat.
6. Se deschide o noua pagină unde administratorul poate vizualiza detalii despre acel utilizator.
7. Administratorul apasa butonul „inactivare” (inactivate).
8. Utilizatorul devine inactiv.

Scenariul este realizat în momentul în care în dreptul utilizatorului blocat apare statusul „inactiv” (inactive), iar acesta nu mai are dreptul de a se loga în sistem.

### Scenarii alternative:

- \*a. În orice pas al scenariului de succes sistemul se blochează:
  1. Angajatul se reautentifică în sistem și reia rezervarea.
- 4a. În lista afișată după cautare apare mai mult de un utilizator.
  1. Administratorul alege din lista utilizatorul dorit.
- 7a. Butonul de „inactivare” nu este vizibil.
  1. Utilizatorul este deja blocat, se observă statusul acestuia.

### 4.3. Context tehnologic

Ca și context tehnologic pentru acest sistem am ales mediul *SAP*. *SAP ERP*, [22], este un software de planificare a resurselor întreprinderii dezvoltat de compania germană SAP SE. *SAP ERP* încorporează funcțiile esențiale ale unei organizații, permite o integrare mai ușoară la nivel mondial (barierele de curs valutar, de limbă și de cultură pot fi suprapuse automat), actualizările trebuie efectuate o singură dată pentru a fi implementate la nivelul întregii companii, oferă informații în timp real, reducând posibilitatea erorilor de redundanță, furnizorii au cunoștințe anterioare și expertiză cu privire la modul de a construi și implementa cel mai bine un sistem, interfața utilizator este complet personalizabilă, permițând utilizatorilor finali să dicteze structura operațională a produsului.

Ca limbaj de programare pentru partea de back-end a acestui sistem s-a ales *ABAP*. Acesta este un limbaj de programare de nivel înalt creat de compania software germană SAP SE. Acesta este poziționat în prezent ca limbă pentru programarea *SAP Application Server*, care face parte din platforma *NetWeaver* pentru dezvoltarea de aplicații de afaceri.

*SAP NetWeaver Application Server* este o componentă a soluției care funcționează ca server de aplicații web pentru soluțiile *SAP*. Acest server este fundamentul pe care rulează cea mai mare parte a gamei de produse *SAP*. Toate serverele de aplicații *SAP* reprezintă layerul de aplicație al arhitecturii sistemului *SAP* bazat pe *ABAP*. Aceste servere execută aplicații *ABAP* și comunică atât cu componentele de prezentare cât și cu baza de date.

Pe parte de front-end s-a utilizat tehnologia *SAPUI5*, care este o tehnologie UI client bazată pe *Javascript*, *CSS* și *HTML5*.

*Javascript* este un limbaj de programare client-side, care permite clientului să interacționeze foarte mult cu server-ul, în urma fiecărei interacțiuni generându-se conținut dinamic care trebuie afișat în pagina Web. Este un limbaj de programare orientat obiect, bazat pe conceptul prototipurilor. *Javascript* este folosit pentru introducerea funcționalităților în paginile Web, codul fiind rulat de către browser-ul Web.

*HyperText Markup Language 5* prescurtat ca *HTML5* este cea de-a cincea versiune a standardului *HyperText Markup Language*, prescurtat *HTML*, care este o formă de marcare orientată către prezentarea documentelor text pe o singură pagină, utilizând un software de randare specializat, numit agent utilizator *HTML*, cel mai bun exemplu în acest sens fiind browser-ul Web.

*Cascading Style Sheets* prescurtat ca *CSS* este un standard pentru formatarea elementelor unui document *HTML*. Stilurile definite pentru elementele documentului se pot defini prin fișiere externe, referințele spre acestea făcându-se în zona head a documentului *HTML*, sau prin introducerea etichetelor `<style></style>` în zona head a documentului.

Aplicațiile dezvoltate cu *SAPUI5* se execută într-un browser pe orice dispozitiv, fie acesta mobil, tabletă sau desktop PC. În funcție de mediul în care este utilizat *SAPUI5*, bibliotecile sau aplicațiile pot fi stocate pe un server *SAP NetWeaver Application Server* sau pe o platforma *SAP Cloud*, iar datele de afaceri pot fi accesate prin *SAP Gateway*, utilizând modelul *OData*.

Platforma Cloud furnizata de SAP este *SAP HANA Cloud Platform*, prescurtat HCP. Aceasta este o platformă-ca-serviciu (platform-as-a-service PaaS) pentru transformări digitale cu servicii și capabilități curprinzătoare de dezvoltare a aplicațiilor. Cu ajutorul acestei platforme se pot dezvolta aplicații mult mai ușor, astfel se accelerează timpul de la idee la produs, de asemenea crește productivitatea, se reduce costul și riscul, iar aplicațiile pot fi rulate atât pe telefoane inteligente, tablete sau desktop PC.

Unul din serviciile disponibile pe platforma de SAP HANA Cloud este *OData Provisioning*, [21], care combină capabilitățile SAP Gateway pentru serviciile SAP OData într-un context Cloud. OData Provisioning expune datele de afaceri și logica de afaceri ca servicii OData pe SAP Cloud, permițând astfel clienților să ruleze aplicații orientate pe utilizator pe SAP Cloud.

*SAP Gateway* este parte a SAP NetWeaver Application Server și este utilizat pentru a stabili o conexiune între suita de afaceri (business suite) SAP și clienții țintă, platforme și framework-uri. Oferă instrumente de dezvoltare și generare pentru a crea servicii OData pentru diferite instrumente de dezvoltare a clienților. SAP Gateway oferă o modalitate mai ușoară de utilizare a logicii de afaceri (business logic) și a conținutului sistemului SAP Back-end pe aplicațiile web. De asemenea, reduce complexitatea pentru a accesa datele SAP și oferă interfețe simple pentru a reduce timpul de dezvoltare.

*Open Data Protocol*, prescurtat OData este un protocol care permite crearea și consumul API-urilor interogabile și interoperabile, care utilizează modelul arhitectural REST, într-un mod simplu și standard.

Modelul arhitectural *Representation State Transfer*, prescurtat REST, reprezintă cea mai populară metodă de construire a unui serviciu Web. REST este un model arhitectural folosit pentru a citi, crea, actualiza și șterge date de pe server prin intermediul apelurilor HTTP. Un apel REST reprezintă o simplă cerere HTTP către server.

Pentru sistemul MyWorkplace, am utilizat un cont utilizator de proba (trial) pentru autentificarea pe platforma de HCP, motiv pentru care conexiunea dintre SAP Netweaver Application Server și SAP HANA Cloud Platform s-a realizat prin intermediul *SAP HANA Cloud Connector*. Prin intermediul acestuia s-a realizat expunerea serviciilor de pe SAP Netweaver Server pentru a fi utilizate pe platforma HCP.

Tehnologiile menționate anterior se pot observa în diagrama conceptuală prezentă în figura 4.5. Așadar se poate observa comunicarea dintre logica de afaceri și poarta SAP prin intermediul interfeței Remote Function Call, prescurtat RFC, care este o interfață standard de comunicare între sistemele SAP. Apoi, de la poarta SAP prin protocolul HTTP serviciile se expun cu ajutorul aplicației SAP HANA Cloud Connector spre platforma de Cloud. Pe această platformă se găsesc mai multe servicii disponibile care oferă dezvoltatorului o utilizare mult mai ușoară a acesteia. De menționat sunt SAP Web IDE, care este mediul de dezvoltare online, oferit de platforma SAP Hana Cloud Platform, dar și framework-ul SAPUI5 și posibilitatea unei conexiuni Git.

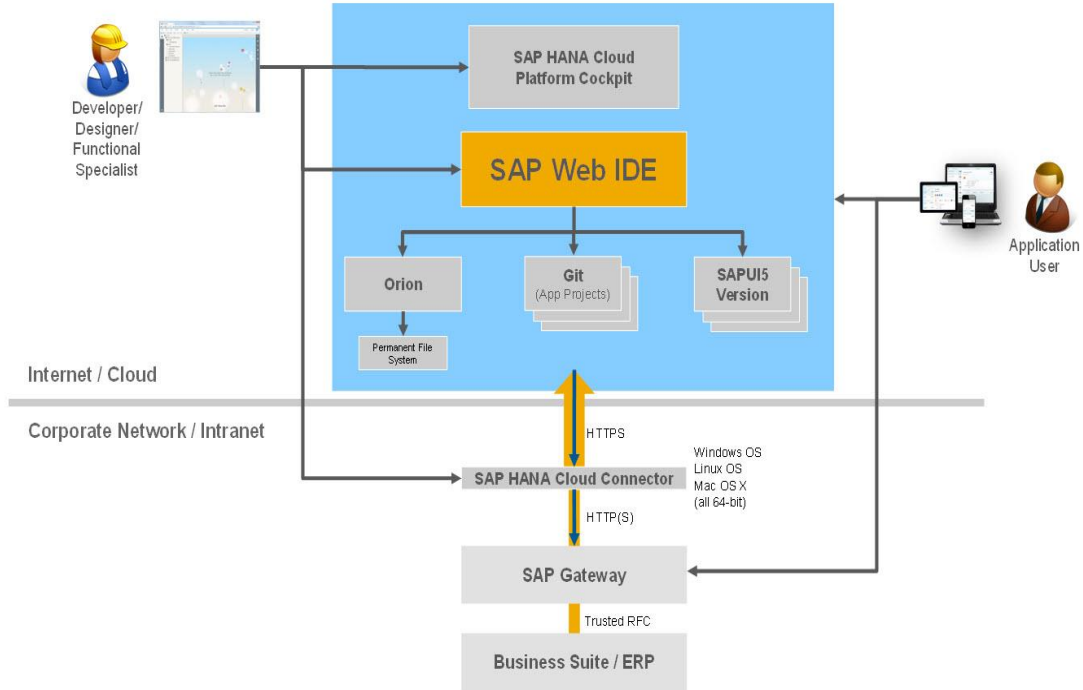


Figura 4.5 Arhitectura conceptuală a sistemului<sup>1</sup>

*Motivația de a alege tehnologii SAP este suținută de rapiditatea și simplitatea de dezvoltare a aplicațiilor cross platform prin intermediul framework-ului SAPUI5, care datorită bazării pe limbajul Javascript, are capacitatea de a gestiona un număr mare de conexiuni simultane, oferind un timp de răspuns scăzut. Motivația pentru alegerea SAP NetWeaver Application Server și a limbajului de programare ABAP este susținută de performanța sistemului oferită de utilizarea aceleași game de tehnologii atât pentru front-end cât și pentru back-end.*

<sup>1</sup> [https://help.hana.ondemand.com/SAP\\_RDE/frameset.htm?2a11c8d3a9e74ca98861c3db5794042b.html](https://help.hana.ondemand.com/SAP_RDE/frameset.htm?2a11c8d3a9e74ca98861c3db5794042b.html)

## Capitolul 5. Proiectare de Detaliu și Implementare

Acest capitol cuprinde modul în care sistemul a fost proiectat. Se va prezenta schema arhitecturală a sistemului, modelul bazei de date, diagrama de deployment și vor fi detaliate anumite componente ale aplicației considerate relevante.

### 5.1. Arhitectura generală a aplicației

În această secțiune se va prezenta arhitectura generală a aplicației, cum a fost proiectat sistemul, iar în următoarele subcapitole se vor descrie detaliat componentele sistemului.

MyWorkplace este o aplicație orientată pe arhitectura client-server. Pe partea de client avem o aplicație implementată cu tehnologia SAPUI5, iar partea de server este implementată în limbajul de programare ABAP. Legătura dintre client și server se realizează prin protocolul OData, prin servicii REST.

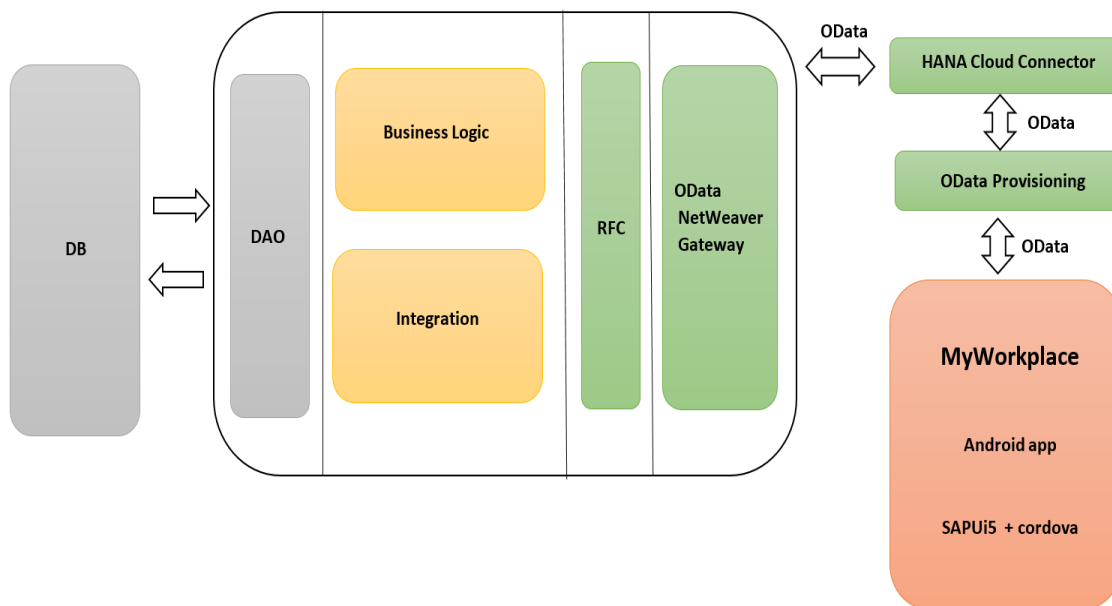


Figura 5.1 Arhitectura generală a sistemului

În figura 5.1 putem observa o schemă generală care evidențiază prin cod de culori nivelele întregii aplicații.

Un prim nivel al aplicației este nivelul de date, care este reprezentat prin culoarea gri și care realizează legătura cu baza de date.

Urmatorul nivel este cel logic, care este reprezentat prin culoarea galben. Acest nivel reprezintă întreaga funcționalitate a aplicației, dar și integrarea cu sisteme externe, în cazul acesta trimiterea de emailuri.

Comunicarea dintre client și server este reprezentată ca nivel separat cu verde. Remote Function Call, prescurtat RFC, este interfața standard de comunicare între sistemele SAP. RFC apelează o funcție care trebuie executată într-un sistem la distanță. Aplicațiile de la distanță pot fi asociate cu software-ul SAP și cu programarea ABAP și oferă o modalitate pentru utilizarea unui program extern pentru a utiliza datele returnate de pe server.

Tot pe nivelul de comunicare observăm și cele trei componente necesare pentru comunicarea prin protocolul OData. SAP NetWeaver Gateway este poarta de acces la și de la server. HANA Cloud Connector este componenta care face legătura dintre sistemele SAP, în acest caz SAP NetWeaver Application Server și HANA Cloud Platform, iar OData Provisioning este un serviciu al Platformei Cloud care permite expunerea serviciilor REST spre exterior.

Ultimul nivel este reprezentat cu portocaliu și înglobează întreaga parte de client.

În capitolele următoare s-a realizat o descriere detaliată a celor două componente principale, client și server, cât și a modelului bazei de date.

### 5.2. Client

Aplicația Client a sistemului este realizată cu tehnologia SAPUI5, care implementează modelul arhitectural software pentru implementarea interfețelor utilizator Model View Controller, prescurtat MVC. Conceptul MVC este utilizat pentru a separa reprezentarea informațiilor de interacțiunea cu utilizatorul. Această separare facilitează dezvoltarea și schimbarea părților în mod independent.

Fiecare din cele trei componente are propriul său rol în realizarea execuției unei operații. Vederea (View-ul) este responsabilă pentru definirea și redarea interfeței utilizator. Modelul gestionează datele aplicației, iar controlerul reacționează pentru a se vizualiza evenimentele și interacțiunea cu utilizatorul modificând vizualizarea și modelul. Aceste roluri se pot observa în figura 5.2.

Modelul arhitectural MVC a fost ales pentru dezvoltarea aplicației MyWorkplace, având anumite avantaje precum, o mai bună lizibilitate, mentenabilitate și extensibilitate. MVC permite modificarea vizualizării fără a atinge logica de afaceri (business logic) și permite definirea mai multor vizualizări ale aceluiași date.

În general, vizualizările și controlerul formează o relație 1:1, dar este posibil să existe și controleri fără vizualizări, acești controleri fiind controleri de aplicație. Este, de asemenea, posibilă crearea de vizualizări fără controleri.

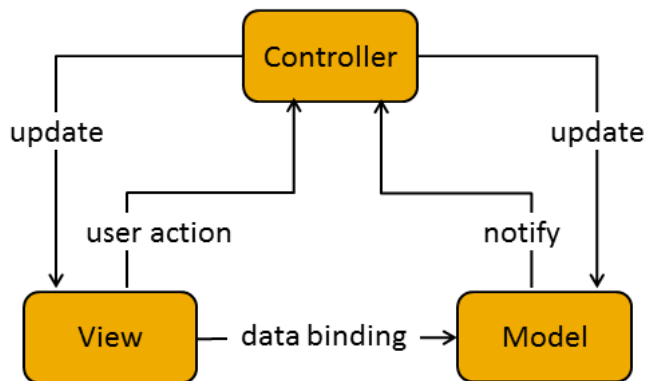


Figura 5.2 Arhitectura MVC<sup>2</sup>

Aplicația MyWorkplace are câte o vizualizare și câte un controler pentru fiecare pagină separată a aplicației, acest aspect se poate vizualiza în figura 5.2 care reprezintă diagrama celor mai relevante clase controler a aplicației. Fiecare controler are asignat câte o vizualizare, iar modelul este creat cu ajutorul clase „sap.ui.model.json.JSONModel”, care crează modele mapate pe datele primite de la server, în format JSON.

JavaScript Object Notation, prescurtat JSON este un format de fișier standard care utilizează text lizibil și inteligibil de oameni, pentru a transmite obiecte constând din perechi atribut-valoare sau tipuri de date în formă matriceală.

În diagrama de clase se pot distinge cele două ramuri ale aplicației, cea de administrator și cea de utilizator autentificat. Pe ramura de administrator, aplicația permite acestuia să acceseze secțiunea de gestiune a conturilor utilizator, cea de gestiune a birourilor și cea de vizualizare a rezervărilor întregii companii. În secțiunea de gestiune a conturilor utilizator, administratorului îi este permis să introducă un nou client în sistem, să modifice datele personale ale unui anumit client sau să dezactiveze un cont utilizator.

În cazul în care administratorul dezactivează un cont utilizator din greșeală, sau persoana căreia i-a fost dezactivat contul în prealabil, din diverse motive precum, concedii fără plată, acesta are posibilitatea de a reactiva un cont. În secțiunea de vizualizare rezervări, administratorul are posibilitatea de a căuta rezervările după un anumit birou, după birourile dintr-o anumită încăpere sau clădire sau după un anumit nume de utilizator, astfel acesta poate obține informații precum disponibilitatea unui birou sau amplasarea unui coleg în cadrul companiei.

Secțiunea de gestiune a birourilor este comună celor două tipuri de utilizatori, însă fiecăruia îi sunt permise diferite operații. Administratorului îi este permis să creeze, modifice sau să ștergă din sistem clădiri, încăperi și birouri. Ștergerea fiind permisă doar în cazul în care nodul sters, de exemplu încăperea, are la rândul său toate subnodurile șterse, de exemplu birourile.

<sup>2</sup> <https://openui5beta.hana.ondemand.com/1.30.0/docs/guide/91f233476f4d1014b6dd926db0e91070.html>

Pe de altă parte, utilizatorului autentificat, secțiunea de gestiune a birourilor, îi permite să caute un birou după clădirea sau încăperea în care acesta se află. Apoi, după alegerea biroului, acesta are posibilitatea de rezervare ad-hoc (check in), rezervare pentru utilizare ulterioară (reserve) sau anularea unei rezervări ad-hoc (check out).

Utilizatorului autentificat îi mai este permis să vizualizeze o listă cu toate birourile de unde acesta poate realiza o căutare rapidă a unui birou deja cunoscut sau a biroului preferat, de unde poate realiza aceleași operații de rezervare ad-hoc, rezervare pentru utilizare ulterioară sau anularea unei rezervări ad-hoc ca și în secțiunea de căutare amănunțită a birourilor.

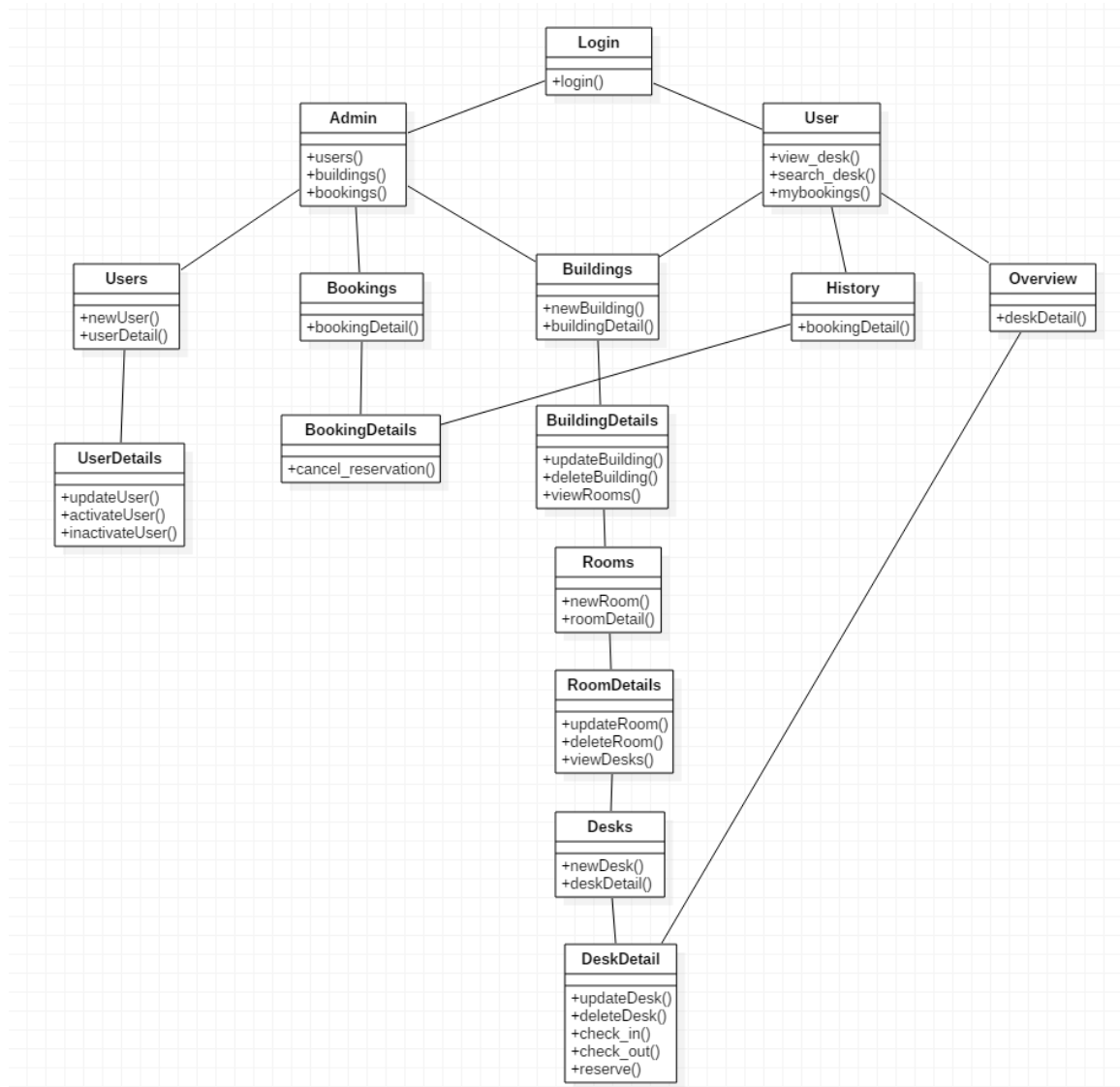


Figura 5.3 Diagrama de clase a aplicației Client



O altă secțiune destinată utilizatorului este cea de vizualizare a rezervărilor proprii, de unde acesta poate, pe lângă vizualizarea rezervărilor sale, atât din trecut, cât și din prezent și viitor, să realizeze anularea unei rezervări (cancel).

Tot pe diagrama de clase se poate observa și legătura dintre paginile aplicației. Fiecare pagină reprezintă o vizualizare separată iar navigarea dintre pagini se realizează conform legăturilor reprezentate pe diagramă. Comunicare între vizualizări se realizează însă prin intermediul controlerelor aferente fiecărei vizualizări, controlerul fiind cel care încarcă modelul pentru a fi reprezentat în vizualizarea corespunzătoare.

În continuare se va prezenta o diagrama care exemplifică fluxul de date în realizarea unei operații la nivel de aplicație client. Luând în considerare operația de vizualizare rezervări, se poate descrie în șase pași, conform figurii 5.4 cum se realizează aceasta.

Primul pas reprezintă cererea pe care dispozitivul mobil o realizează prin intermediul protocolului HTTP, în cazul de față, apăsarea butonului de vizualizare rezervări. Următorul pas reprezintă extragerea de date și maparea acestora pe modelul de rezervări. Apoi se transmite modelul creat la controler, sub denumirea de rezultat. Controlerul transmite rezultatul vizualizării, în vederea afișării acestora, iar ultimul pas este reprezentat de afisarea propriu-zisă a vizualizării pe ecranul dispozitivului.

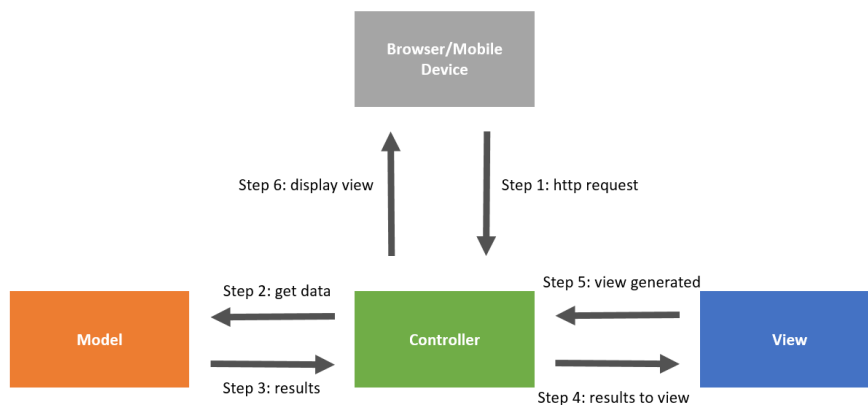


Figura 5.4 Flow SAPUI5 MVC<sup>3</sup>

În continuare se va prezenta biblioteca Barcode Scanner, utilizată în vederea integrării aplicației MyWorkplace cu funcțiile native ale dispozitivului mobil, cu ajutorul căreia se realizează scanarea de coduri QR. Această librărie pornește camera foto a dispozitivului și citește instant un cod QR, apoi returnează valoarea citită.

<sup>3</sup> <http://saphanatutorial.com/what-is-sapui5/>

### 5.3. Server

Aplicația Server a acestui sistem se bazează pe modelul arhitectural pe trei nivele (Three-tier). Arhitectura three-tier este un model de arhitectură software client-server în care logica de stocare a datelor și accesul la date, logica de proces funcțional și interfața cu utilizatorul sunt dezvoltate independent, iar de cele mai multe ori sunt menținute pe platforme diferite. Modelul three-tier al aplicației conține următoarele nivele: nivelul de prezentare (presentation tier), nivelul logic (business tier) și nivelul de stocare a datelor (data tier). Nivelul de prezentare, este reprezentat de clasele RFC și de serviciul OData, care comunică cu aplicația client. Nivelul logic înglobează cea mai mare parte din logica aplicației, transformând cererile venite de la client în interogări sau modificări ale bazei de date și convertind datele extrase din baza de date în modele de date interpretate de către client. Acest nivel la rândul său interacționează cu nivelul de stocare al datelor, în vederea comunicării cu baza de date.

Împărțirea pe nivele a aplicației MyWorkplace se poate observa în diagrama de pachete, reprezentată în figura 5.5. Pentru fiecare nivel a fost realizat câte un pachet. Nivelul de aplicație este reprezentat de pachetul Interface, nivelul logic este reprezentat de pachetul Business Logic iar nivelul de date este reprezentat de pachetul Data Access.

Pentru o mai bună gestiune a implementării, pachetele de Interface și cel de Business Logic au fost împărțite în subpachete. Subpachetul RFC conține clase care implementează interfața de comunicare între sistemele SAP. Pachetul OData conține serviciul dezvoltat, care este utilizat de Aplicația Client. Pachetul Processing conține partea de calcul și de logică a aplicației iar pachetul Integration conține partea de integrare cu emailul.

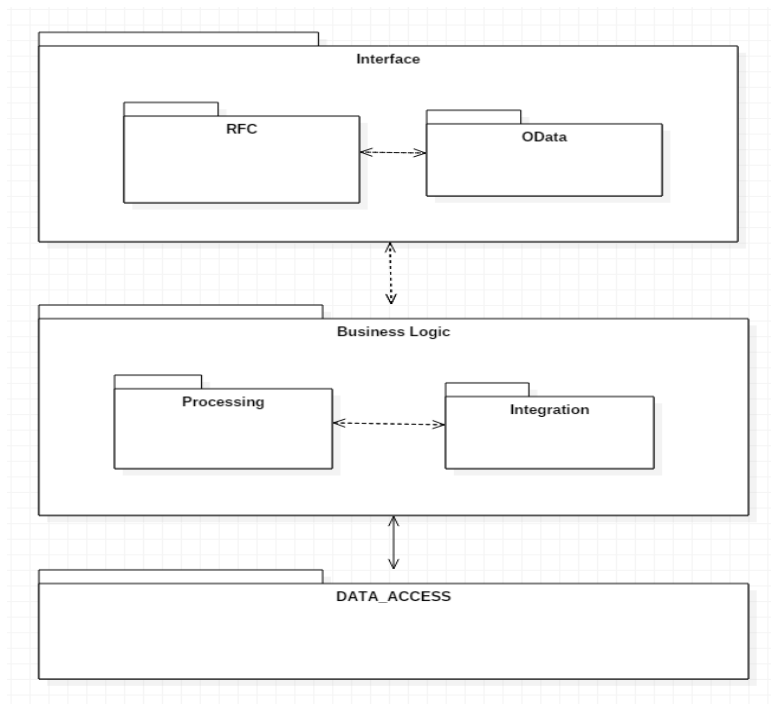


Figura 5.5 Diagrama de pachete la nivel de server

În continuare se va detalia structura aplicației server la nivel de clase. În pachetul Data\_Access s-a proiectat interfața CRUDQ care definește metodele elementare de acces la baza de date și anume, creare (create), modificare (update), citire singulară (get), citire multiplă (get\_table) și stergere (delete). Metodele acestei interfețe sunt implementate în clasele de acces către baza de date. Pentru fiecare tabel propriu al aplicației, s-a creat câte o clasă de acces, iar acestea au fost denumite asemănător tabelelor cu care fac legătura.

În pachetul Business\_Logic au fost proiectate clase corespunzătoare celor din pachetul anterior, doar că acestora le-a fost adăugată și logică de afaceri. În aceste clase, datele extrase din baza de date sunt prelucrate pentru a fi transmise spre aplicația client, iar datele transmise de aplicația client sunt prelucrate pentru a fi mapate în baza de date.

De asemenea în acest nivel au fost introduse metodele de validare a parolei și de criptare și decriptare a acesteia, dar și metoda prin care se realizează trimiterea de emailuri. Aceste metode s-au realizat prin intermediul unor servicii predefinite de sistemul SAP.

În pachetul Interface a fost proiectată, pentru fiecare operație realizabilă, câte o clasă care implementează interfața de comunicare între sistemele SAP. De asemenea, în acest pachet a fost implementat și serviciul prin care se face comunicarea dintre partea de client a aplicației MyWorkplace și partea de server a acesteia.

Diagrama de clase a aplicației Server este reprezentată în figura 5.6. De menționat este faptul că au fost reprezentate doar clasele metodele relevante ale fiecărei clase, pentru a nu încărca diagrama atât de tare și pentru a păstra lizibilitatea cât mai bună. Tot pentru lizibilitate mai bună au fost reprezentate clase RFC generice pentru operațiile de creare, modificare, stergere, citire singulară și citire multiplă.

În cadrul sistemului sunt însă pentru fiecare operație a fiecărei entități câte o clasă separată. Din acest motiv s-a abordat notația X, care reprezintă, pe rând denumirea fiecărei entități. De asemenea s-au omis din diagramă atât legăturile de asociere directă între clasele aceluiași nivel, dar și dependențele mai puțin relevante în procesul de rezervări, care îngreunau lizibilitatea diagramei.

În continuare, pe baza diagramei de clase, reprezentată în figura 5.6 se prezintă operațiile permise la nivel de server și modul de realizarea a acestora.

Prima operație este autentificarea, unde la nivelul de logică se verifică parola extrasă din baza de date, pentru utilizatorul care dorește autentificarea, cu cea introdusă de acesta prin intermediul interfeței grafice.

Umatoarele patru operații sunt cele de rezervare ad-hoc, rezervare pentru utilizare ulterioară, anulare rezervare ad-hoc, anulare rezervare. Pentru aceste operații, sistemul prelucrează la nivelul logic datele primite din interfața grafică și le mapează pentru a realiza operațiile necesare cu baza de date.

Ultimele operații posibile sunt operațiile care sunt permise doar administratorului și anume cele de administrare a bazei de date. Acestea sunt operațiile de creare, modificare, stergere, citire singulară și citire multiplă.

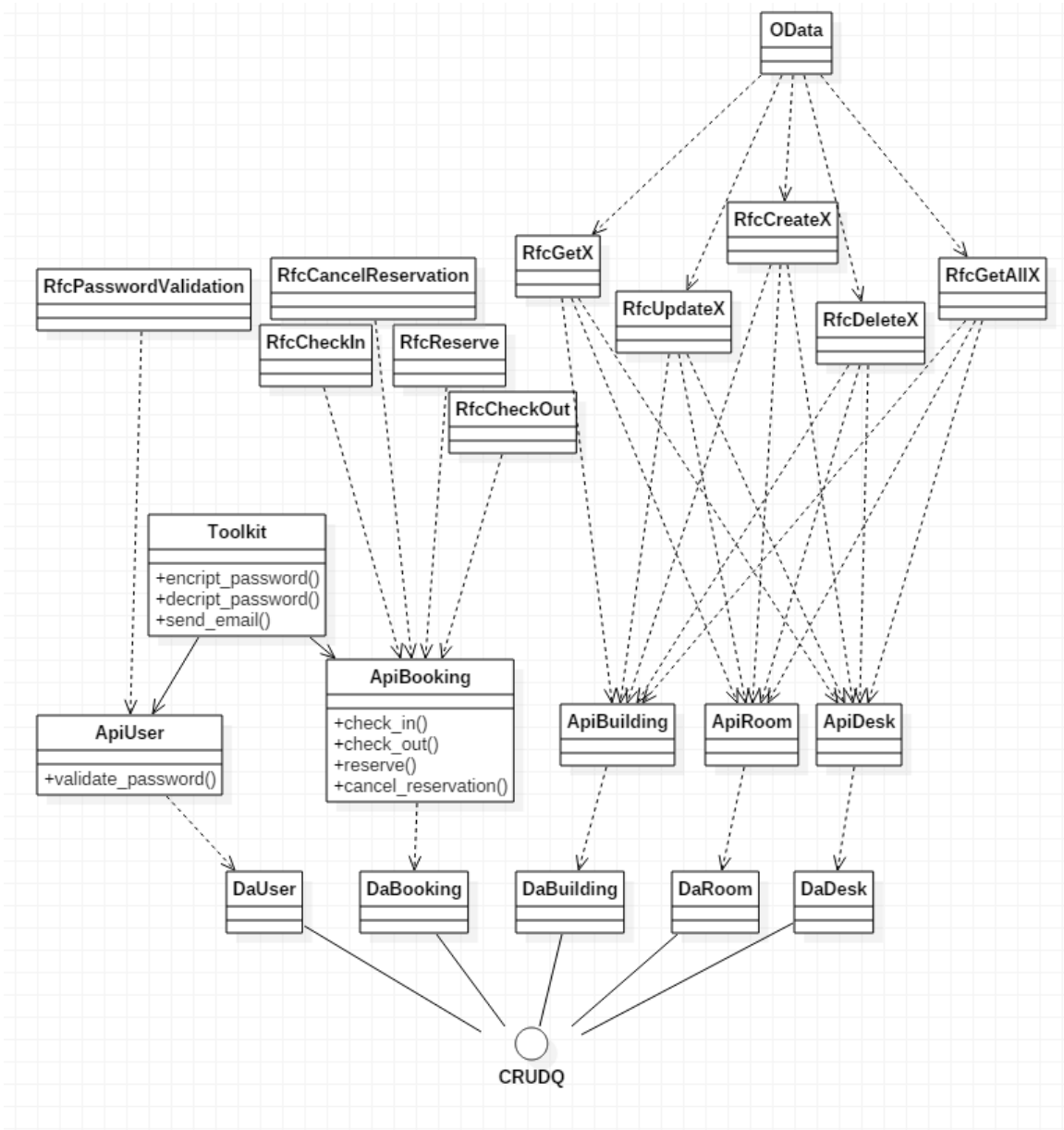
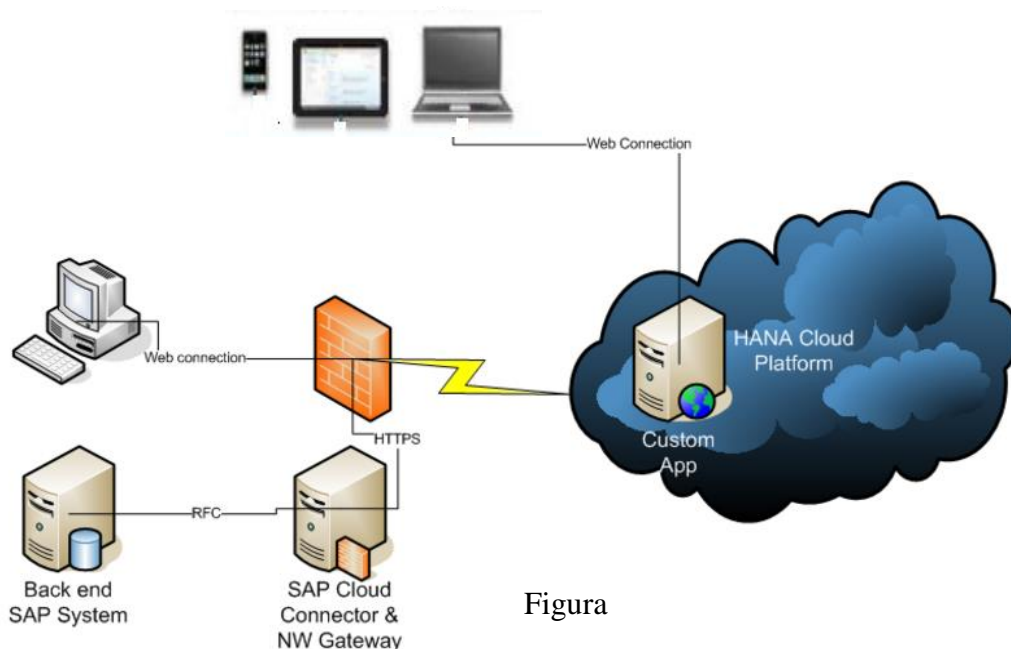


Figura 5.6 Diagrama de clase a aplicației Server

În figura 5.7 este reprezentată diagrama de deployment, unde sunt prezentate componentele hardware sau nodurile care rulează componentele software ale sistemului prezentat, dar și modul în care aceste componente sunt interconectate. Scopul acestei diagrame este de a prezenta structura hardware necesară rulării sistemului.



Figura

#### 5.7 Diagrama de deployment a sistemului<sup>4</sup>

Aplicația MyWorkplace poate fi accesată de pe orice dispozitiv, având proprietatea de compatibilitate multi-platăformă, însă, datorită integrării cu funcții native ale dispozitivelor mobile, se recomandă rularea acesteia pe telefonul mobil sau pe tabletă.

Clientul comunică cu serverul aplicației prin intermediul protocolului OData.

Componentele fizice ale sistemului sunt serverele SAP, unde se află și baza de date, mașina pe care rulează SAP Cloud Connector și Netweaver Gateway, platforma de Cloud și dispozitivele mobile pe care este rulată aplicația.

#### 5.4. Modelul bazei de date

Sistemul propus utilizează baza de date construită în mediul SAP. Baza de date conține 7 tabele. În figura 5.8 sunt prezentate tabelele din baza de date și legăturile dintre acestea.

Fiecare tabel din baza de date conține date dintr-un singur model din lumea reală. Datele nu se țin în mai multe locuri în baza de date, nu există redundanță la nivelul de date stocate, lucru care este facilitat la modificările care pot interveni, nefiind nevoie să se facă modificările în mai multe tabele.

Tabelul denumit T005 este un tabel intern al sistemului SAP utilizat care conține denumirea fiecărei țări și cheia acestuia. Am utilizat acest tabel pentru extragerea cheii fiecărei țări pentru a evita redundanța prin construirea unui tabel propriu, separat, cu

<sup>4</sup> <https://blogs.sap.com/2015/01/14/hana-cloud-platform-why-it-is-important-from-a-customer-point-of-view/>

aceiași date. Tabelul City conține orașele predefinite de dezvoltator în care se pot crea ulterioare sedii de birouri. Tabelul Building menține datele relevante pentru o clădire de birouri sau un sediu. Tabelul Room conține date legate de încăperile unei clădiri, iar tabelul Desk conține date legate de birourile care pot fi ulterior ocupabile. Pe diagrama bazei de date se pot observa atributele specifice fiecărei entități. Tabelul User conține utilizatorii sistemului, iar tabelul Booking salvează rezervările realizate asupra birourilor.

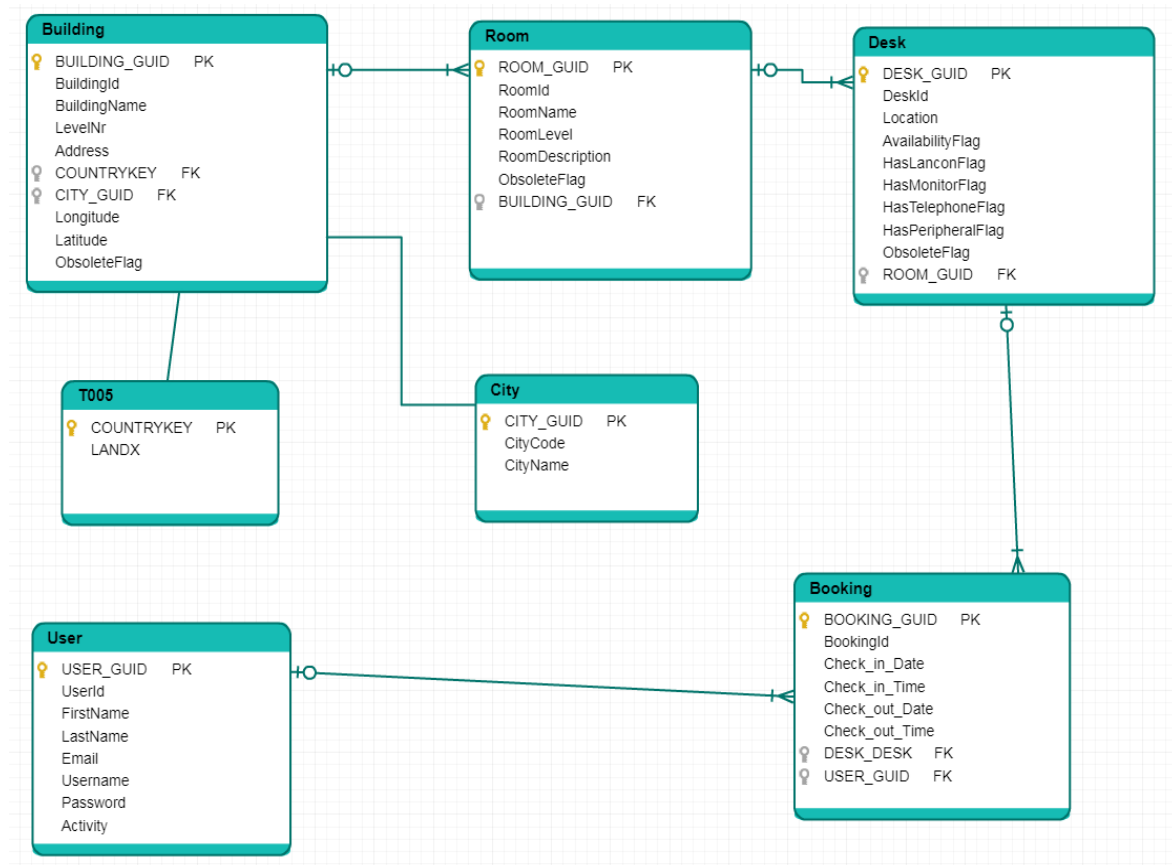


Figura 5.8 Modelul bazei de date

În proiectarea bazei de date s-a dorit reprezentarea cât mai corectă a informațiilor și diminuarea posibilității de a ajunge la informații eronate. În vederea atingerii acestui scop, s-a utilizat tehnica de normalizare a bazei de date. În continuare sunt descrise formele normale pe care baza de date le respectă.

Forma normală unu, prescurtat FN1, presupune că la intersecția unei linii cu oricare coloană, să se găsească un câmp care conține exact o valoare. O tabelă nu poate avea grupuri repetitive iar fiecare atribut poate să primească doar o valoare atomică. Baza de date construită pentru aplicația MyWorkplace satisface aceste constrângeri, deoarece fiecare atribut al unui tabel primește doar o valoare atomică.

Forma normală doi, prescurtat FN2, este îndeplinită dacă forma normală unu este îndeplinită și fiecare atribut care nu aparține cheii primare, este total dependent funcțional de cheia primară. Forma normală doi se verifică la relațiile care au cheie compusă din

mai multe atribute pe post de cheie primară. Baza de date a aplicației MyWorkplace conține doar chei formate dintr-un singur atribut, motiv pentru care se află în forma normală doi.

Forma normală trei, prescurtat FN3, este îndeplinită dacă sunt îndeplinite forma normală unu și forma normală doi și dacă nu există nici un atribut care să nu aparțină cheii principale și care să fie tranzitiv dependent de cheia principală. În general, astfel de dependențe există doar dacă tabelul stochează date din mai multe domenii, iar tabelele bazei de date ale aplicației MyWorkplace stochează date doar dintr-un singur domeniu.

Forma normală Boyce Code, prescurtată BCNF presupune eliminarea dependențelor funcționale parțiale și tranzitive, astfel încât fiecare determinant este o cheie primară și toate atributele neprime sunt determinate funcțional de cheia primară aleasă. Fiecare tabel care s-a creat conține o cheie primară alcătuită dintr-un singur atribut care este determinantul tuturor celorlalte atribute conținute în tabel, astfel, condiția de apartenență la forma normală Boyce Code este îndeplinită.

## 5.5. Precondiții de compilare a aplicației

### 5.5.1. Npm

Npm [18] este un manager de pachete pentru limbajul de programare JavaScript. Acesta este managerul de pachete implicit pentru mediul de runtime JavaScript Node.js. Se compune dintr-un client de linie de comandă, numit si npm și o bază de date online a pachetelor publice, numită registrul npm. Npm poate gestiona pachetele care sunt dependențe locale ale unui anumit proiect precum și instrumentele JavaScript instalate la nivel global. Atunci când este instalat ca manager de dependență pentru un proiect local, npm poate instala, într-o singură comandă, toate dependențele unui proiect prin intermediul pachetului package.json.

### 5.5.2. Apache Cordova

Apache Cordova [17] este un cadru de dezvoltare mobilă open source care permite utilizarea tehnologiilor web standard - HTML, CSS3 și JavaScript pentru dezvoltarea pe mai multe platforme. Aplicațiile se execută în limbajele direcționate către fiecare platformă și se bazează pe legăturile API compatibile cu standardele pentru a accesa capacitățile fiecărui dispozitiv, cum ar fi senzori, datele, starea rețelei, etc.

Apache Cordova este utilă pentru a dezvolta o aplicație mobilă pe mai multe platforme, fără a fi necesară reimplementarea cu limbajul și setul de instrumente al fiecărei platforme. De asemenea, acest mediu de dezvoltare este util în conectarea componentelor aplicațiilor native mobile cu o fereastră WebView care poate accesa API-uri de nivel de dispozitiv sau în dezvoltarea unei interfețe de plugin între componentele native și WebView.

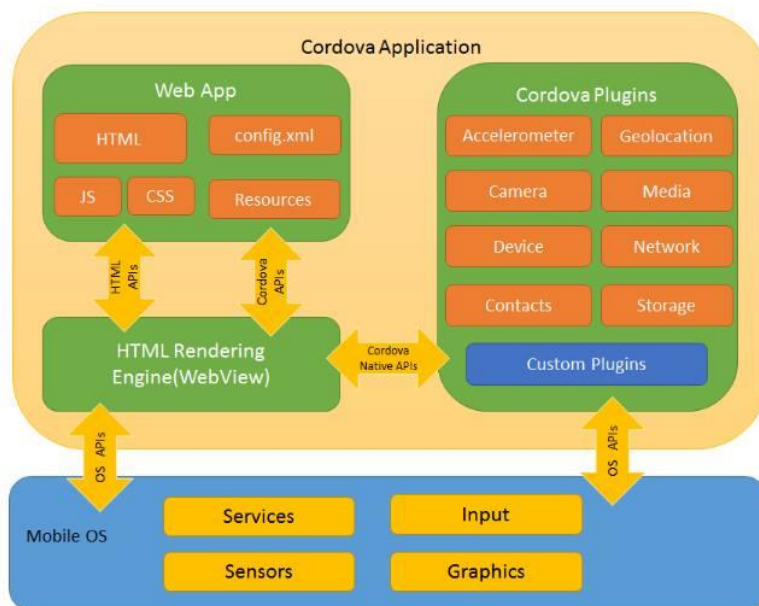


Figura 5.9 Arhitectura aplicației Cordova<sup>5</sup>

În figura 5.9 se prezintă o arhitectură generală a unei aplicații Cordova. Componenta *WebView* cu funcția Cordova poate furniza aplicației întreaga interfață cu utilizatorul. Pe unele platforme, aceasta poate fi, de asemenea, o componentă într-o aplicație mai mare hibridă, care amestecă *WebView* cu componente native ale aplicației.

*WebApp* este partea în care se află codul aplicației. Aplicația însăși este pusă în aplicare ca pagină web. În mod implicit este necesar un fișier local numit *index.html* care are referințe la CSS, JavaScript, imagini, fișiere media sau alte resurse necesare pentru a fi rulat. Aplicația se execută într-un *WebView* împreună cu funcțiile native, care poate fi apoi rulată pe dispozitive. Această componentă are un fișier foarte important – fișierul *config.xml* care furnizează informații despre aplicație și specifică parametrii care afectează modul în care funcționează.

Pluginurile sunt o parte integrată a ecosistemului Cordova. Ele oferă o interfață pentru Cordova și componentele native pentru a comunica între ele și pentru legăturile cu dispozitivele standard API. Acesta va permite invocarea codului nativ din JavaScript. Proiectul Apache Cordova menține un set de pluginuri numite Core Plugins. Aceste pluginuri de bază oferă aplicației posibilitatea de acces la capacitățile dispozitivului, cum ar fi bateria, camera, contactele, etc.

Cordova oferă două fluxuri de lucru de bază pentru a crea aplicații mobile, fluxul cross-platform și fluxul centrat pe o platformă specifică. Având în vedere că aplicația *MyWorkplace* are la bază fluxul cross-platform, se va prezenta doar acesta. Fluxul de lucru cross-platform este utilizat pentru ca aplicația să ruleze pe cât mai multe sisteme de operare mobile diferite. Acest flux de lucru se concentrează asupra cordova CLI. CLI este un instrument de nivel înalt care permite dezvoltarea de proiecte pentru mai multe platforme simultan, abstractizând o mare parte din funcționalitatea scripturilor de nivel inferior. CLI copiază un set comun de elemente web în subdirectoare pentru fiecare

<sup>5</sup> <http://cordova.apache.org/docs/en/7.x/guide/overview/index.html>



platformă mobilă, efectuează modificări de configurare necesare pentru fiecare și execută scripturi de construire pentru a genera aplicații în format binar. CLI oferă, de asemenea, o interfață comună pentru a aplica pluginuri aplicației.

*S-a ales realizarea aplicației bazată pe Apache Cordova, deoarece este un framework care permite dezvoltarea aplicației pe mai multe sisteme de operare mobile, fără necesitatea rescrierii codului în limbajele specifice fiecărei platforme. De asemenea, acest framework îmi permite integrarea funcțiilor native ale telefonului cu aplicația.*

### 5.5.3. Android SDK

Pentru compilarea aplicației pentru sistemul de operare Android, Cordova necesită Android SDK. Acest kit de dezvoltare software pentru Android include un set de instrumente de dezvoltare, precum biblioteci, emulator de telefon, documentație, cod exemplu, tutoriale și chiar și un program de depanare.

Kitul de dezvoltare software pentru Android este necesar în compilarea aplicației cordova, deoarece pentru a realiza aplicația pentru sistemul de operare Android, Cordova compilează codul generic JavaScript împreună cu librăriile necesare furnizarea de Android SDK.

### 5.5.4. Microsoft Visual Studio

Pentru compilarea aplicației în vederea rulării acesteia pe platforma Windows phone, este necesar ca platformele de Windows de pe dispozitivul mobil să fie la fel, mai exact să aibă aceeași versiune. În plus, este necesară instalarea mediului de dezvoltare Microsoft Visual Studio. Acesta este utilizat în general pentru dezvoltarea aplicațiilor destinate utilizării pe platformele Windows. În momentul compilării, cordova compilează codul generic JavaScript împreună cu librăriile furnizare de platforma windows prin intermediul mediului de dezvoltare Microsoft Visual Studio.

## 5.6. Compilarea aplicației

Compilarea aplicației pentru android, respectiv pentru windows phone se realizează în mod asemănător. Aplicația se compilează din Windows Command Line. Se deschide linia de comandă și se merge cu comanda „cd” până în folderul unde se regăsesc sursele aplicației MyWorkplace, aplicația dezvoltată în SAPUi5 și anume parte de client a întregului sistem. Apoi în funcție de platforma pentru care se dorește compilarea, se realizează următorii pași:

Pentru platforma Android se execută comanda de compilare specifică, reprezentată mai jos. Dacă s-a realizat cu succes compilarea, se afișează în consolă un mesaj de informare și locația unde a fost salvată aplicația.

```
$ cordova build android
```

Pentru platforma Windows se execută comanda de compilare specific, reprezentată mai jos, în care este specificată și versiunea de windows pe care va rula aplicația, prin intermediul paramentului *-appx*. În cazul de față, s-a realizat compilare pentru platforma Windows Phone 8.1. Dacă s-a realizat cu succes compilarea, se afișează în consolă un mesaj de informare și locația unde a fost salvată aplicația.

```
cordova build windows -- --appx=8.1-phone
```

După realizarea compilării, fișierul aplicație rezultat se transmite pe dispozitivul mobil iar pașii de instalare pot fi executați cu succes.

## Capitolul 6. Testare și Validare

În acest capitol se dorește prezentarea principalelor procese de testare realizate asupra întregului sistem. Realizarea etapelor de testare nu garantează în general funcționalitatea completă și corectă a sistemului în toate condițiile, însă poate ajuta la identificarea anumitor probleme și să ducă la găsirea de soluții și rezolvări. Testarea poate fi definită ca un proces de validare și verificare a faptului că sistemul corespunde cerințelor și constrângerilor impuse.

Procesul de dezvoltare a sistemului a fost iterativ, testarea fiecărei componente realizându-se la finalul implementării. Pentru parte de backend s-au implementat clase de test, care au fost rulate atât cu date corecte cât și cu date eronate pentru a se verifica modul de răspuns al aplicației și starea în care trece aceasta în caz de eroare. Partea de frontend și anume aplicația mobilă s-a testat manual pornind de la scenarii simple de success până la scenarii mult mai complexe.

Îndiferent de metodele de testare folosite, rezultatul testării oferă dezvoltatorului un nivel de siguranță asupra componentelor dezvoltate, dar în același timp rezultatele obținute îi oferă o idee privind rata de defectare a sistemului.

Testarea sistemului s-a realizat în două etape, în fiecare etapă fiind testate anumite componente ale sistemului. Principalele metode de testare folosite sunt: clasele de test la nivel de server și testarea manuală.

### 6.1. Clase de test la nivel de server

```

METHOD do_testcase.

DATA: lv_input1    TYPE sysuuid_x,
      lv_input2    TYPE sysuuid_x,
      lv_act_result TYPE zmwp_s_booking,
      lv_exp_result TYPE zmwp_s_booking.

lv_input1 = mt_input[ name = 'DESK_GUID' ]-value.
lv_input2 = mt_input[ name = 'USER_GUID' ]-value.

lv_exp_result-desk_guid = lv_input1.
lv_exp_result-user_guid = lv_input2.

CALL FUNCTION 'ZMWP_CREATE_BOOKING'
EXPORTING
  iv_user_guid = lv_exp_result-user_guid
  iv_desk_guid = lv_exp_result-desk_guid.

SELECT SINGLE desk_guid user_guid FROM zmwp_booking
  INTO CORRESPONDING FIELDS OF lv_act_result
  WHERE desk_guid = lv_exp_result-desk_guid
  AND user_guid = lv_exp_result-user_guid.

cl_aunit_assert=>assert_equals(
  exp = lv_exp_result
  act = lv_act_result
  msg = |Expected result <> Actual result |
  ).

ENDMETHOD.

```

Figura 6.1 Exemplu clasă de test la nivel de server

La nivel de server, au fost realizate clase de test, ca în exemplul din figura 6.1. În acest exemplu s-a verificat realizarea unei rezervări de tip ad-hoc. S-au introdus datele de intrare, apoi s-a realizat crearea unei rezervări, ca apoi să se verifice dacă câmpul creat anterior se află în baza de date.

Aceste teste au trecut cu succes, astfel s-a trecut la următoarea interacție de testare și anume la testarea aplicației mobile, care s-a realizat manual.

### 6.2. Testare manuală

Aplicația MyWorkplace a fost testată în mod manual, testarea realizându-se pentru cele mai importante componente ale sistemului. Principalele scenarii de test au fost întocmite pe baza funcționalităților sistemului.

În continuare se vor prezenta cele mai semnificative cazuri de testare folosite pentru testarea aplicației.

**CT1:** Rezervarea unui birou

**Precondiții:** Angajatul este autentificat în sistem

#### Scenariul 1:

**Pasul 1:** Utilizatorul alege din pagina principală opțiunea de „căutare birou” (search desk).

**Pasul 2:** Se încarcă lista cu clădirile disponibile și se observă disponibilitatea birourilor din fiecare clădire.

**Pasul 3:** Utilizatorul selectează clădirea dorită, vizualizează detaliile despre clădire, apoi apasă butonul „vizualizare încăperi” (view rooms).

**Pasul 4:** Se încarcă încăperile clădirii alese de utilizator și se afișează disponibilitatea birourilor la nivel de încăpere.

**Pasul 5:** Utilizatorul selectează încăperea dorită, vizualizează detaliile despre încăpere, apoi apasă butonul „vizualizare birouri” (view desks).

**Pasul 6:** Se încarcă lista cu birourile încăperii alese anterior și se afișează disponibilitatea acestora.

**Pasul 7:** Utilizatorul alege un birou.

**Pasul 8:** Utilizatorul apasă butonul „rezervă” (reserve).

**Pasul 9:** Se încarcă o nouă pagină unde utilizatorului îi este cerut intervalul rezervării care este format din data și ora de început și de final a rezervării.

**Pasul 10:** Utilizatorul introduce datele cerute la pasul anterior, astfel încât data de final să fie înaintea datei de început și apasă butonul „ok”.

**Rezultat așteptat:** Se afișează mesaj de eroare cum că datele nu sunt corecte. Data de început trebuie să fie înaintea datei de final.

**Scenariul 2:**

**Pasul 1:** Utilizatorul alege din pagina principală opțiunea de „căutare birou” (search desk).

**Pasul 2:** Se încarcă lista cu clădirile disponibile și se observă disponibilitatea birourilor din fiecare clădire.

**Pasul 3:** Utilizatorul selectează clădirea dorită, vizualizează detaliile despre clădire, apoi apasă butonul „vizualizare încăperi” (view rooms).

**Pasul 4:** Se încarcă încăperile clădirii alese de utilizator și se afișează disponibilitatea birourilor la nivel de încăpere.

**Pasul 5:** Utilizatorul selectează încăperea dorită, vizualizează detaliile despre încăpere, apoi apasă butonul „vizualizare birouri” (view desks).

**Pasul 6:** Se încarcă lista cu birourile încăperii alese anterior și se afișează disponibilitatea acestora.

**Pasul 7:** Utilizatorul alege un birou care este ocupat.

**Pasul 8:** Utilizatorul apasa butonul „rezervă” (reserve).

**Pasul 9:** Se încarcă o noua pagină unde utilizatorului îi este cerut intervalul rezervării care este format din data și ora de început și de final a rezervării.

**Pasul 10:** Utilizatorul introduce datele cerute la pasul anterior astfel încât data de început să fie momentul actual și apasă butonul „ok”.

**Rezultat așteptat:** Mesaj de eroare cum că biroul este ocupat.

**Scenariul 3:**

**Pasul 1:** Utilizatorul alege din pagina principală opțiunea de „căutare birou” (search desk).

**Pasul 2:** Se încarcă lista cu clădirile disponibile și se observă disponibilitatea birourilor din fiecare clădire.

**Pasul 3:** Utilizatorul selectează clădirea dorită, vizualizează detaliile despre clădire, apoi apasă butonul „vizualizare încăperi” (view rooms).

**Pasul 4:** Se încarcă încăperile clădirii alese de utilizator și se afișează disponibilitatea birourilor la nivel de încăpere.

**Pasul 5:** Utilizatorul selectează încăperea dorită, vizualizează detaliile despre încăpere, apoi apasă butonul „vizualizare birouri” (view desks).

**Pasul 6:** Se încarcă lista cu birourile încăperii alese anterior și se afișează disponibilitatea acestora.

**Pasul 7:** Utilizatorul alege un birou.

**Pasul 8:** Utilizatorul apasa butonul „rezervă” (reserve).

**Pasul 9:** Se încarcă o noua pagină unde utilizatorului îi este cerut intervalul rezervării care este format din data și ora de început și de final a rezervării.

**Pasul 10:** Utilizatorul introduce datele cerute la pasul anterior și apasă butonul „ok”.

**Rezultat așteptat:** Mesaj de confirmare cum că rezervarea a fost realizată. Se realizează autentificarea în contul personal de email și se observă primirea emailului de confirmare a rezervării.

**CT2:** Căutare birou

**Precondiții:** Angajatul este autentificat în sistem

**Scenariul 1:**

**Pasul 1:** Utilizatorul alege din pagina principala opțiunea „vizualizare birouri” (view desks).

**Pasul 2:** Se încarcă lista completă de birouri.

**Pasul 3:** Utilizatorul scrie în bara de căutare numărul biroului căutat și apasă „căutare” (search).

**Pasul 4:** Se presupune că biroul nu există.

**Rezultat așteptat:** Lista de birouri apare goală.

**Scenariul 2:**

**Pasul 1:** Utilizatorul alege din pagina principala opțiunea „vizualizare birouri” (view desks).

**Pasul 2:** Se încarcă lista completă de birouri.

**Pasul 3:** Utilizatorul scrie în bara de căutare numele clădirii în care dorește să caute un birou.

**Pasul 4:** În lista inițială apar doar birourile din clădirea aleasă.

**Pasul 5:** Utilizatorul alege un birou din acea listă și îl selectează.

**Rezultat așteptat:** Se încarcă o nouă pagină unde utilizatorul vizualizează detaliile despre birou.

**Scenariul 3:**

**Pasul 1:** Utilizatorul alege din pagina principala opțiunea „vizualizare birouri” (view desks).

**Pasul 2:** Se încarcă lista completă de birouri.

**Pasul 3:** Utilizatorul scrie în bara de căutare numărul biroului căutat și apasă „căutare” (search).

**Pasul 4:** În lista inițială apare doar biroul căutat.

**Pasul 5:** Utilizatorul selectează biroul.

**Rezultat așteptat:** Se încarcă o nouă pagină unde utilizatorul vizualizează detaliile despre birou.

## Capitolul 7. Manual de Instalare și Utilizare

În acest capitol sunt prezentate resursele necesare pentru instalarea software-ului, instalarea cu acces a componentelor sistemului pe o mașină locală, pașii pentru conectarea la server precum și manualul de utilizare al aplicației.

### 7.1. Instalarea aplicației și configurarea serverului

Pentru rularea cu succes a aplicației, este necesar un *sistem SAP*, în care să fie importat pachetul cu aplicația de server. Acesta va fi furnizat împreună cu lucrarea. Pentru conectarea sistemului SAP la Platforma de Cloud este necesar *SAP Cloud Connector*, care trebuie configurat conform sistemului SAP. SAP Cloud Connector se poate descărca de pe pagina oficială, de la adresa <https://tools.hana.ondemand.com/#cloud>, iar configurațiile pas cu pas, care diferă în funcție de sistemul de operare ales și de sistemul SAP care urmează a fi legat prin SAP Cloud Connector, sunt tot pe pagina oficială, la adresa <https://help.sap.com/viewer/cca91383641e40ffbe03bdc78f00f681/Cloud/enUS/ec68ee242c3d4c7797fc53bb65abcd71.html>.

Pentru expunerea serviciului OData, este necesară crearea unui cont pe *platforma HANA Cloud*, iar la secțiunea Services se alege *OData Provisioning* și se setează destinația conform setărilor din HANA Cloud Connector. Detalii de configurare a destinației se pot consulta pe site-ul oficial, la adresa <https://help.sap.com/viewer/a7c6d8a0bd1f415887f6246d8cf8e68e/Cloud/enUS/bcd66a91489f4da8b7b1cf36758d5499.html>

După realizarea acestor conexiuni, serverul este configurat, iar aplicația poate rula cu succes. Pentru a rula aplicația, însă, aceasta trebuie instalată. Ea va fi furnizată împreună cu această lucrare iar pașii de instalare sunt foarte ușor de realizat. Aplicația trebuie salvată pe dispozitivul mobil, trebuie instalată, iar apoi se poate utiliza cu succes.

### 7.2. Conectarea la server

Dacă server-ul este deja configurat, atunci pornirea acestuia se realizează în câțiva pași. Primul pas este conectarea la sistemul SAP, care se poate realiza local, sau remote prin intermediul unui token, care oferă o cheie de acces către acel sistem. Apoi se pornește SAP Cloud Connector, după care, conexiunile sunt deschise.

Pentru verificarea conexiunilor, acestea se pot testa, fie din pagina de configurare a Cloud Connector-ului, apăsând butonul de „verificare conexiune” (check connection), fie din serviciul de OData Provisioning, de la secțiunea de configurare a destinațiilor, unde se regăsește același buton de „verificare conexiune”.

Dacă a fost verificată conexiunea, aplicația mobilă poate fi pornită, iar utilizarea acesteia se va realiza cu succes.

### 7.3. Manual de utilizare

După ce conectarea la server a fost realizată cu succes și aplicația a fost instalată, aceasta se poate rula.

Manualul de utilizare cuprinde o parte din scenariile prezentate prin cazurile de utilizare descrise în capitolul 4.

Pentru ca autentificarea în sistem să se poată realiza cu succes, utilizatorii trebuie să introducă numărul de numele de utilizator și o parolă validă, aceste date fiind furnizate de administrator în momentul angajării. După autentificarea unui angajat, acesta vizualizează pagina principală cu opțiunile disponibile utilizatorilor. Dacă autentificarea a fost realizată de administrator, atunci se vor vizualiza în pagina principală opțiunile acestuia. Secțiunea de autentificare și paginile principale aferente celor două tipuri de utilizatori se pot vizualiza în figura 7.1.

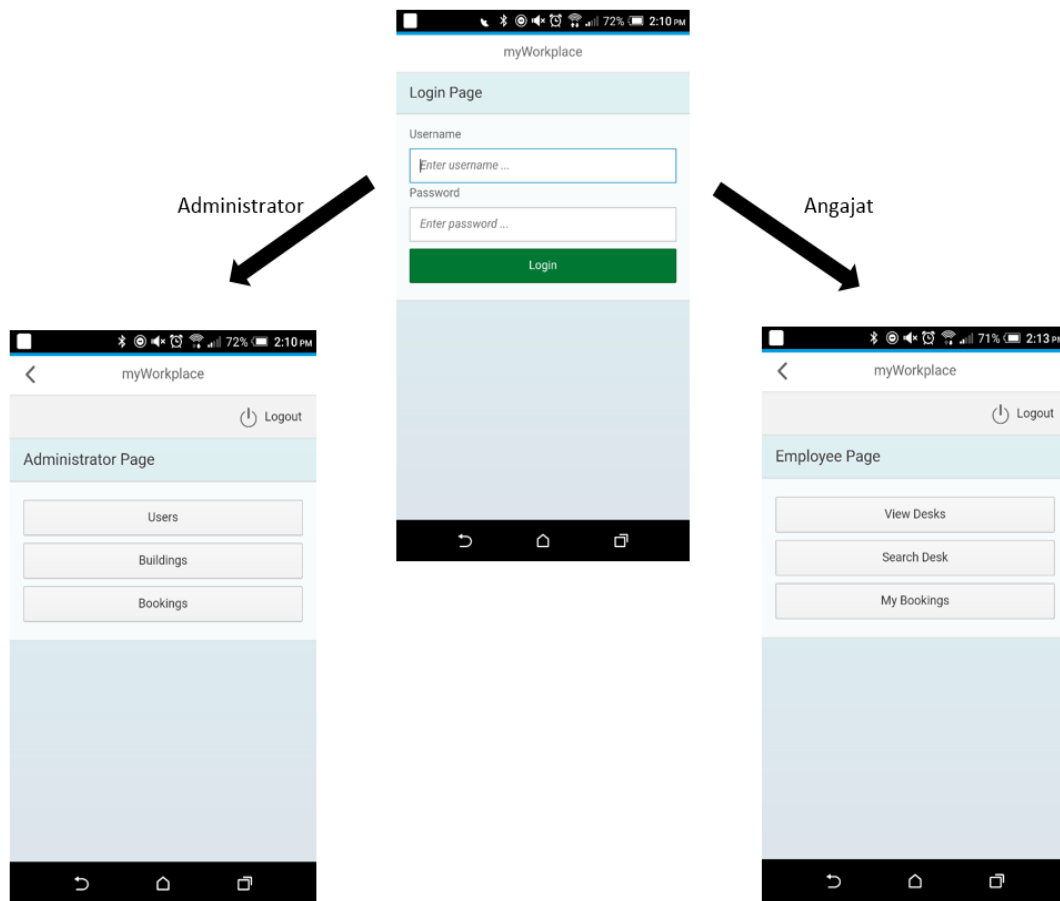


Figura 7.1 Autentificare



Utilizatorul poate să aleagă ce operații dorește să efectueze. În continuare, se va prezenta modalitatea de activare și dezactivare a unui utilizator, operații realizate de administrator. Pentru efectuarea acestor operații, administratorul alege opțiunea „Users”, apoi alege utilizatorul dorit, iar în pagina de detalii va putea vizualiza doar butonul aferent operației posibile în funcție de statusul utilizatorului. Dacă acesta este activ, va fi vizibil doar butonul „Inactive”, iar dacă acesta este inactiv, va fi vizibil doar butonul „Active”. În figura 7.2 se poate vizualiza pagina de detalii în ambele cazuri.

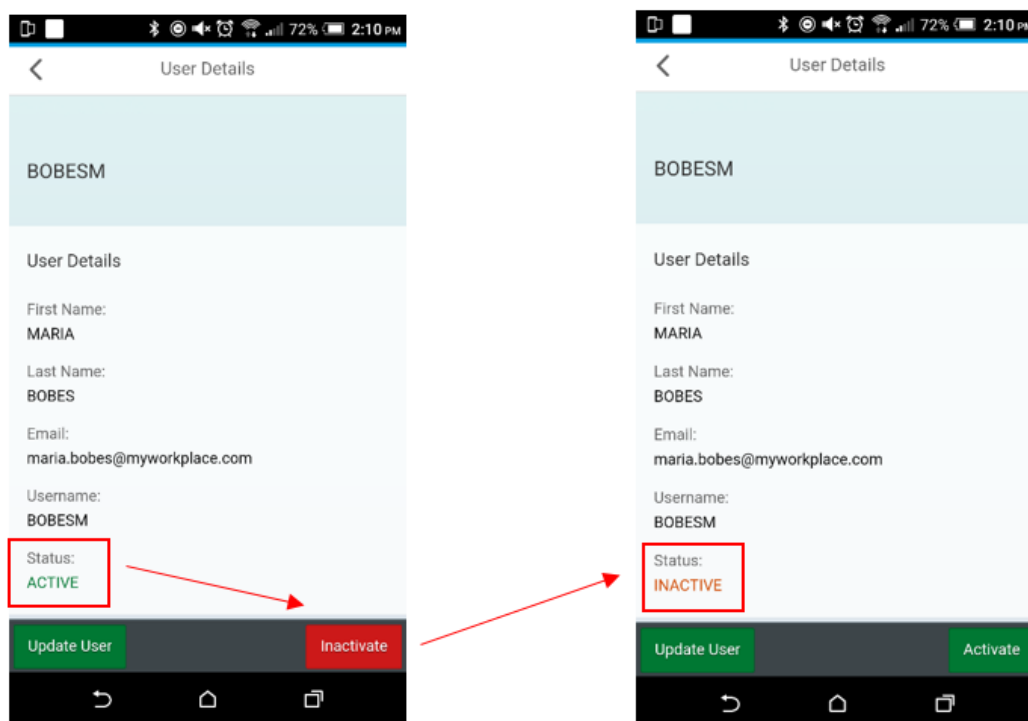


Figura 7.2 Dezactivare utilizator

În ceea ce privește rezervarea unui birou, aceasta poate fi realizată ad-hoc, prin intermediul butonului „Check-in”, operație care poate fi urmată de o anulare, posibilă prin butonul „Check-out” sau poate fi realizată cu scopul de a fi utilizată ulterior, ceea ce este posibil prin intermediul butonului ”Reserve”. Aceste operații sunt disponibile în pagina de detalii a birourilor, dacă în sistem s-a autentificat un angajat.

Pentru a vizualiza pagina de detalii, angajatul are două posibilități. Prima posibilitate este de a alege din pagina principală opțiunea „View Desks”, iar apoi să aleagă din lista tuturor birourilor pe cel dorit. A doua posibilitate este să aleagă din pagina principală opțiunea ”Search Desk”, să aleagă o clădire din lista afișată, apoi să aleagă o încăpere din lista de încăperi, iar în final să aleagă biroul dorit din lista birourilor disponibile în acea încăpere. Figura 7.3 prezintă pagina de detalii a unui birou și butoanele disponibile pentru efectuarea operațiilor. Butoanele pentru rezervările ad-hoc sunt disponibile în funcție de disponibilitatea biroului ales .

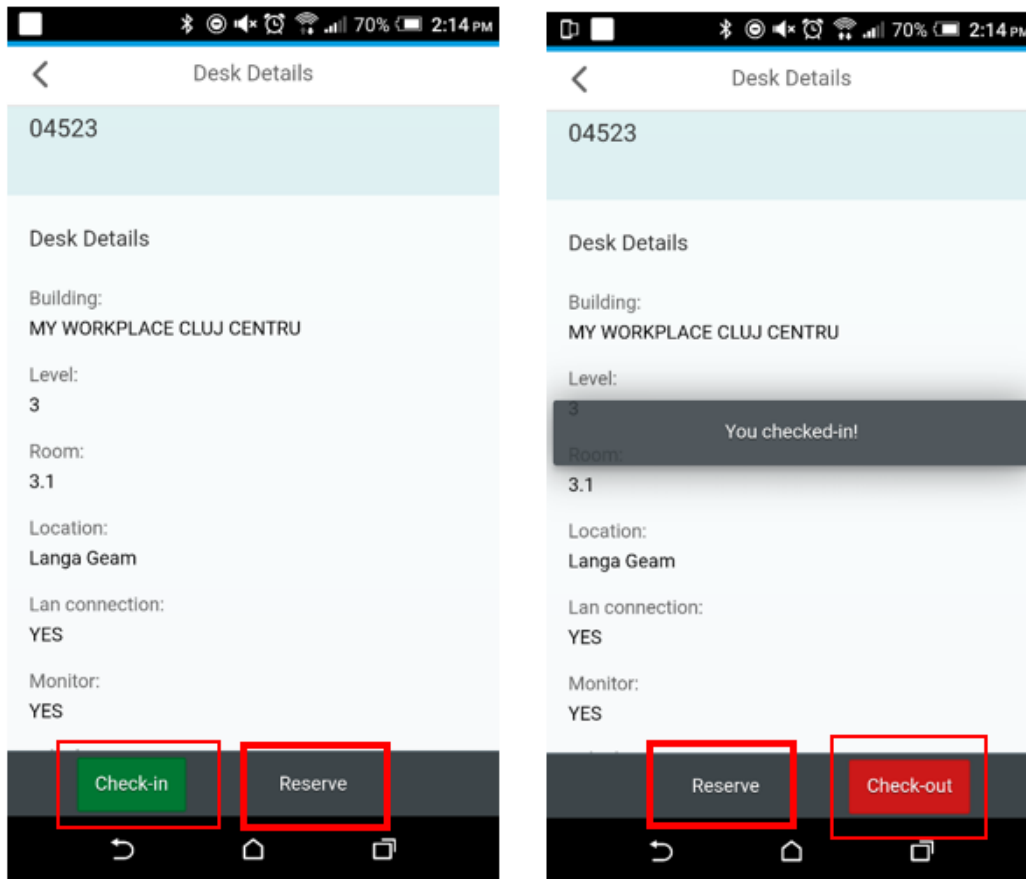


Figura 7.3 Rezervare birou

## Capitolul 8. Concluzii

În acest capitol vor fi prezentate obiectivele atinse de acest proiect, realizările acestuia, dar și descrierile dezvoltărilor ulterioare și îmbunătățirilor viitoare.

### 8.1. Realizările obiectivelor propuse

Sistemul realizat reușește să-și atingă scopul, de a putea fi un sistem competitiv cu sistemele similare. Majoritatea obiectivelor propuse au fost realizate, utilizatorii fiind clasificați în funcție de rolul pe care îl au în aplicație prin crearea de pagini pentru fiecare tip de utilizator. Dezvoltarea unui astfel de sistem a reprezentat o oportunitate de dezvoltare tehnică pentru dezvoltatorul sistemului, deoarece s-au dobândit cunoștințe solide în domeniul aplicațiilor create cu framework-ul SAPUi5. Dezvoltatorul sistemului a înțeles care sunt pașii necesari într-un proces de dezvoltare iterativă și ce presupune fiecare pas, plus etapele de analiză și de design ale sistemului care au reprezentat cea mai mare provocare pentru dezvoltator în momentul în care a trebuit să ia decizii cu privire la modelul de date și la arhitectura bazei de date folosite.

Sistemul reușește să livreze o aplicație cu o interfață grafică intuitivă și ușor de învățat, astfel că sistemul nu necesită un tutorial complex pentru a înțelege funcționalitatea oferită de sistem. Aceste aspecte sunt susținute de design-ul creat și de modul de implementare al componentelor.

Sistemul dezvoltat permite utilizatorilor autentificați să:

- Caută o clădire și să vizualizeze disponibilitatea birourilor din acea clădire.
- Caută o încăpere și să vizualizeze disponibilitatea birourilor din acea încăpere.
- Filtreze în lista de birouri după caracteristici precum, locație sau existența perifericelor.
- Marcheze un birou prin scanare de cod QR.
- Rezerve un birou ad-hoc.
- Anuleze o rezervare realizată ad-hoc.
- Rezerve un birou pentru o utilizare ulterioară.
- Anuleze o rezervare.
- Vizualizeze lista propriilor rezervări.
- Primească emailuri la fiecare rezervare sau anulare de rezervare.
- Vizualizeze o listă cu toate birourile.
- Caută birouri după o specificație aleasă precum, numărul biroului, numele încăperii sau a clădirii în care se află acesta, localizarea biroului în încăpere sau numele utilizatorului care se află la birou, în cazul care biroul este ocupat.

Sistemul dezvoltat permite administratorului să:

- Blocheze un cont utilizator
- Creeze un utilizator și să-i modifice datele personale.
- Creeze, modifice sau să steargă sedii, încăperi și birouri.
- Vizualizeze toate rezervările.
- Caută o anumită rezervare după numărul acesteia.
- Caută un anumit utilizator după numele de utilizator.

### 8.2. Dezvoltări și îmbunătățiri ulterioare

Orice sistem informatic poate să beneficieze de dezvoltări ulterioare, prin adăugarea de noi funcționalități sistemului sau prin îmbunătățirea funcționalităților existente. Sistemul dezvoltat nu reprezintă o excepție în acest sens, acesta permițând adăugarea de noi funcționalități într-o manieră foarte simplă.

În cele ce urmează se va prezenta o listă cu posibilitățile de dezvoltare ulterioară:

- Funcționalitatea de salvare în calendarul nativ al dispozitivului.
- Funcționalitatea de vizualizare a clădirii pe harta oferită de aplicația google maps.
- Un sistem de rapoarte pentru utilizatorul cu drepturi de administrator. Astfel acesta poate să facă un bilant al rezervărilor pentru o anumită perioadă de timp, pentru un anumit birou sau pentru un anumit angajat.
- Funcționalitatea de rezervare de grupuri de birouri, acest lucru fiind util în cazul în care este necesară munca pe echipe sau în cazul realizării unor traininguri.
- Funcționalitatea de rezervare a sălilor de sedinte, precum și a resurselor necesare. Din resursele posibile amintim video proiectorul, tabla, monitoare suplimentare și care asigură conexiunea la internet. Astfel resursele pot fi mutate dintr-o sală de ședințe în alta, în funcție de necesitate lor.
- Compilarea aplicației și pentru alte platforme, precum iOS. În momentul de față mulți utilizatori beneficiază de un dispozitiv mobil cu sistem de operare iOS, astfel oferirea aplicației și în această variantă este necesară.

**Bibliografie**

- [1] Peter Ames, “Hot-desking: hot or not?”, Management Today, 19 februarie 2015, last updated 23 februarie 2015, Disponibil: <http://www.managementtoday.co.uk/hot-desking-hot-not/article/1334680>  
Accesat: 14 martie 2017
- [2] Ben Collins, “ ‘Hot desking’ Is A Big Trend -- Here’s Why A Lot Of People Hate It”, Business Insider Australia, 11 Aprilie 2013, Disponibil: <https://www.businessinsider.com.au/hot-desking-is-a-big-trend-heres-why-a-lot-of-people-hate-it-2013-4> Accesat: 15 Martie 2017
- [3] Condeco, “Desk Booking Software”, Disponibil: <http://www.condecosoftware.com/products/desk-booking> , Accesat 17 Martie 2017
- [4] Doug Darroch, “Hot Desking: The New Work Style”, The Farm SoHo, 11 iulie 2016, Disponibil: <http://www.thefarmsoho.com/magazine-blog/hot-desking>  
Accesat: 14 martie 2017
- [5] Christian W. Dawson, “Projects in Computing and Information Systems. A student’s guide.”, Pearson Education Limited, Edinburgh 2005
- [6] Deskflex, “Room Scheduling”, Disponibil: <https://www.deskflex.com/Products/Room-Scheduling> Accesat: 17 Martie 2017
- [7] EMS, “Meeting and room scheduling software benchmark survey”, Disponibil: <https://www.emssoftware.com/pdf/executive-summaries/benchmark-survey-executive-summary.pdf> Accesat: 17 Martie 2017
- [8] Essential, “Room & Resource booking build on Mictosoft Exchange & Outlook”, Disponibil: <http://www.resource-essentials.com/brochures/room-desk-resource-booking-solution.pdf> Accesat: 17 Martie 2017
- [9] Valerică Greavu-Stefan, „Cloud Computing: Caracteristici și Modele”, Ianuarie 2015, Disponibil: <https://www.researchgate.net/publication/289952478>  
Accesat: 3 Septembrie 2017
- [10] Lynne J. Millward, S. Alexander Haslam, Tom Postmes, “Putting Employees in Their Place: The Impact of Hot Desking on Organizational and Team Identification” , in Organization Science, Vol. 18, No. 4, Iulie-August 2007, pp 547-559, Disponibil: [https://www.researchgate.net/publication/255579541\\_Putting\\_Employees\\_in\\_Their\\_Place\\_The\\_Impact\\_of\\_Hot\\_Desking\\_on\\_Organizational\\_and\\_Team\\_Identificatio\\_n](https://www.researchgate.net/publication/255579541_Putting_Employees_in_Their_Place_The_Impact_of_Hot_Desking_on_Organizational_and_Team_Identificatio_n) Accesat: 4 Martie 2017
- [11] Kim Peterson, “Why companies are doing away with assigned desks”, Moneywatch, 14 Noiembrie 2014, 1:30PM, Disponibil: <http://www.cbsnews.com/news/why-companies-are-doing-away-with-assigned-desks> Accesat: 15 Martie 2017
- [12] Kevin Roose, “In defense of ‘hot desking’, the controversial new workplace seating arrangement”, Fusion, 24 Septembrie 2015, 3:37PM, Disponibil: <http://fusion.net/story/203509/in-defense-of-hot-desking> Accesat: 14 Martie 2017
- [13] The Smartway2, “Successful Metting Room Resource Scheduling”, Disponibil: <http://www.smartway2.com/assets/smartway2-pr-09-dec-2014.pdf>  
Accesat: 17 Martie 2017

- [14] Luke Stevenson, “Hotdesking can add to social worker stress, study finds”, Workforce, 5 August 2016, Disponibil: <http://www.communitycare.co.uk/2016/08/05/hotdesking-can-add-social-worker-stress-study-finds> Accesat la: 2 Martie 2017
- [15] Thanh Thuy Truong, “The Oneness of Eastern Heart & Western Mind in Future Workspace”, Master of Architecture Explanatory Document, Unitec Institute of Technology, 2015, Disponibil: <http://unitec.researchbank.ac.nz/handle/10652/3177> Accesat: 5 Martie 2017
- [16] Ben Waber, Jennifer Magnolfi, Greg Lindsay, “Workspaces That Move People”, Harvard Business Review, Octombrie 2017, Disponibil: <https://hbr.org/2014/10/workspaces-that-move-people> Accesat: 15 Martie 2017
- [17] Documentație oficială Apache Cordova, <http://cordova.apache.org/docs/en/7.x/guide/overview/index.html> Accesat: 15 Martie 2017
- [18] Documentație Oficială npm, <https://docs.npmjs.com/all> Accesat: 15 Martie 2017
- [19] Wikipedia: “Hot desking”, last modified on 31. December 2016, Disponibil: [https://en.wikipedia.org/wiki/Hot\\_desking](https://en.wikipedia.org/wiki/Hot_desking) Accesat: 2 Martie 2017
- [20] Wikipedia, Transparența software, Disponibil: [https://en.wikipedia.org/wiki/Transparency\\_\(human%E2%80%93computer\\_interaction\)](https://en.wikipedia.org/wiki/Transparency_(human%E2%80%93computer_interaction)), Accesat: 03 Septembrie 2017
- [21] SAP Cloud Platform OData Provisioning - <https://help.sap.com/viewer/a7c6d8a0bd1f415887f6246d8cf8e68e/Cloud/en-US/9751653cf5b2456887aa3825cc089d13.html> Accesat: 10 Aprilie 2017
- [22] ERP System - <https://www.sap.com/products/enterprise-management-erp.html> Accesat: 11 Aprilie 2017

## Anexa 1 – Lista figurilor și a tabelelor din lucrare

### 1. Lista figurilor

Figura 4.1 Cazuri de utilizare ale administratorului.....	23
Figura 4.2 Cazuri de utilizare specifice angajatului.....	24
Figura 4.3 Diagrama de flux pentru cazul de utilizare CU1.....	26
Figura 4.4 Diagrama de secventa a cazului de utilizare CU1.....	26
Figura 4.5 Arhitectura conceptuală a sistemului.....	32
Figura 5.1 Arhitectura generală a sistemului.....	33
Figura 5.2 Arhitectura MVC.....	34
Figura 5.3 Diagrama de clase a aplicației Client.....	36
Figura 5.4 Flow SAPUi5 MVC.....	37
Figura 5.5 Diagrama de pachete la nivel de server.....	38
Figura 5.6 Diagrama de clase a aplicației Server.....	40
Figura 5.7 Diagrama de deployment a sistemului.....	41
Figura 5.8 Modelul bazei de date.....	42
Figura 5.9 Arhitectura aplicației cordova.....	44
Figura 6.1 Exemplu clasă de test la nivel de server.....	47
Figura 7.1 Autentificare.....	52
Figura 7.2 Dezactivare utilizator.....	53
Figura 7.3 Rezervare birou.....	54

### 2. Lista tabelelor

Tabel 3.1 Comparatie între sisteme.....	15
Tabel 4.1 Cerințe funcționale corespunzătoare secțiunii de autentificare.....	17
Tabel 4.2 Cerințe funcționale corespunzătoare administratorului.....	18
Tabel 4.3 Cerințe funcționale corespunzătoare căutării unui birou.....	18
Tabel 4.4 Cerințe funcționale corespunzătoare asignării unui birou.....	18
Tabel 4.5 Cerințe funcționale corespunzătoare rezervării unui birou.....	18
Tabel 4.6 Cerințe nonfuncționale.....	22





## Anexa 2 – Glosar de termeni și abrevieri

### 1. Abrevieri

Termen	Definiție
ABAP	Advanced Business Application Programming
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheet
HANA	High performance ANalytics Appliance
HTML	Hypertext Mark-up Language
JSON	JavaScript Object Notation
REST	Representational State Transfer
RFC	Remote Function Call
SAP	Systems, Applications & Products in Data Processing
UML	Unified Modeling Language

### 2. Glosar de termeni

Termen	Definiție
Apache Cordova	Un framework de dezvoltare a aplicațiilor mobile care permite programatorilor software să construiască aplicații pentru dispozitive mobile folosind CSS3, HTML5 și JavaScript în loc să se bazeze pe API-uri specifice fiecărei platforme.
CSS	Un limbaj pentru descrierea prezentării unui document scris în HTML
HTML 5	limbajul standard de programare pentru descrierea conținutului și a aspectului unei pagini web
OData	Un protocol care permite crearea și consumarea de servicii REST
Protocol HTTP	Protocol request-response care stă la baza aplicațiilor distribuite
Request	Ațiune inițiată de client pentru a obține anumite resurse aflate pe server respectând specificațiile protocolului HTTP.
Response	Ațiune inițiată de server pentru a-i oferi clientului datele cerute respectând specificațiile protocolului HTTP
REST	Un stil arhitectural precum și o abordare a comunicațiilor, care se utilizează în dezvoltarea serviciilor web
RFC	Interfață standard de comunicare între sistemele SAP
SAP HANA Cloud Platform	O platformă care permite crearea și testarea aplicațiilor bazate pe HANA cloud
SAPUI5	Interfața utilizator a companiei SAP pentru HTML 5
Server	Aplicație care rulează pe un server Web și furnizează servicii altor aplicații numite aplicații client