



(12) CERERE DE BREVET DE INVENȚIE

(21) Nr. cerere: a 2012 00808

(22) Data de depozit: 13.11.2012

(41) Data publicării cererii:  
29.03.2013 BOPI nr. 3/2013

(71) Solicitant:  
• GIURESCU LAURENȚIU EMANUEL,  
PIAȚA CUZA VODĂ NR. 3, SLĂNIC, PH, RO

(72) Inventatori:  
• GIURESCU LAURENȚIU EMANUEL,  
PIAȚA CUZA VODĂ NR. 3, SLĂNIC, PH, RO

(54) PROGRAM DE CALCULATOR DE TIP PLATFORMĂ ON-LINE  
PENTRU GESTIUNEA AFACERILOR - KYBS

(57) Rezumat:

Invenția se referă la un program de calculator de tip platformă Cloud Computing, dedicată mediului de afaceri. Aceasta poate fi utilizată, prin intermediul internetului, de pe orice dispozitiv hardware, precum calculator desktop, laptop, tabletă etc. Pe lângă aspectele comune tuturor aplicațiilor software disponibile on-line, precum arhitectura pe niveluri sau diverse funcționalități specifice (autentificare, autorizare, operarea datelor etc.), această aplicație software prezintă o serie de caracteristici inovatoare. Principalul element de noutate constă în integrarea mai multor sisteme informatice, de dimensiuni mari, specifice mediului de afaceri, precum ERP, CRM și DMS, într-o platformă unică, asigurând astfel eficientizarea actului managerial al beneficiarului, și evitarea dublării inutile a informațiilor. Astfel, în cadrul aceleiași aplicații informatice, pe baza unei autentificări unice, utilizatorii pot accesa un set foarte larg de funcționalități și date care, în momentul de față, se regăsesc

disipate și necorelate, în cadrul unor sisteme informatice distincte. O altă inovație este gradul de scalabilitate, personalizare și flexibilitate al aplicației. Aceasta furnizează o serie de instrumente și unelte ce permit utilizatorilor să definească din interfața aplicației, în timp real, obiectele cu care aplicația funcționează (de exemplu, documente, entități, validări, reguli de business etc.). O altă caracteristică extrem de importantă, care asigură un grad ridicat de accesibilitate și performanță, se referă la faptul că aplicația software este special realizată și pregătită, astfel încât să interfațeze cu diverse dispozitive IT și/sau electronice (de exemplu, POS-uri, token-uri, telefoane inteligente, scannere etc.), în conformitate cu standardul M2M (machine to machine).

Revendicări: 1  
Figuri: 3



“PROGRAM DE CALCULATOR DE TIP PLATFORMĂ ON-LINE PENTRU GESTIUNEA AFACERILOR - KYBS (Keep Your Business Smart)”

Programul de calculator de tip platformă on-line pentru gestiunea afacerilor – KYBS (Keep Your Business Smart), conform invenției, este realizat și utilizat în domeniul tehnologiei informației și comunicației, respectiv dezvoltarea la comandă a programelor de calculator (software). Activitatea este clasificată în CAEN în grupa 620 – Activități de servicii în tehnologia informației, clasa 6201 Activități de realizare a soft-ului la comandă (software orientat client). În dezvoltarea programelor de calculator cu unele funcții similare pe piața tehnologiei informației se folosesc tehnologii și concepte precum:

- i. Limbaje de programare: Microsoft .Net C#, Java, HTML, JavaScript, Apex, VisualForce.
- ii. Baze de date: Microsoft SQL Server, Oracle Database, MySQL, etc.
- iii. Concepte/Principii:
  - a) Multitenancy – se referă la un principiu în arhitectura software prin care o singură instanță a unui produs software rulează pe un server și este utilizată concurențial de către mai multe organizații client (chiriași). Un produs software bazat pe această arhitectură este conceput astfel încât datele și configurațiile să fie partiționate astfel încât fiecare organizație-client lucrează cu o instanță virtuală personalizată a aplicației. Acest principiu este considerat a fi un atribut esențial al Cloud Computing.
  - b) Software as a Service – reprezintă o nouă paradigmă în modul de livrare al aplicațiilor software, un model arhitectural construit pe caracteristicile principiului Multi Tenant, caracteristici precum: eficiența, scalabilitate masivă și configurare prin metadate.

*Stadiul critic al tehnicii programelor de calculator* din domeniul adresat de această invenție, observat în piața românească de profil, este caracterizat prin (1) dezvoltarea proprietară a sistemelor informatice integrate (un furnizor dezvoltă un anume tip de sisteme informatice fără a avea un alt produs competitor dezvoltat și plasat în altă clasă calitativă) și (2) lipsa de integrare a mai multor sisteme informatice de la terți producători la nivelul unui singur client (în absența unor interfețe de conectare sau funcții de integrare). Cercetarea documentară efectuată de noi prin consultarea bazelor de date publicate de OSIM a relevat absența înregistrărilor pentru un astfel de program de calculator de tip platformă on-line pentru gestiunea afacerilor. Anexăm la această descriere și lista rezultatelor căutărilor din cercetarea documentară proprie.

*Problema tehnică* pe care o rezolvă invenția este (1) de natura conectării și funcționalității componentelor hardware (în principal calculatoare) și software ale sistemelor informatice utilizate într-o arhitectură multi-modulară precum și (2) de natura managementului informatic asociat în utilizarea simultană a mai multor sisteme informatice de către aceeași companie.

Invenția constă, de o manieră novatoare pe piața românească, într-o platformă on-line ce permite eliminarea implementării sistemelor informatice integrate de tip ERP (Enterprise Resources Planning), CRM (Customer Relationship Management) și DMS (Document Management System) într-un sistem informațional unic la nivelul unei societăți comerciale sau entitate publică, prin externalizarea către platforma on-line inventată.

Activitatea inventivă este în principal reprezentată de (1) programarea funcționalităților platformei ce vor permite (\*) eliminarea barierelor de accesibilitate, bariere de care companiile mici și mijlocii se lovesc în momentul în care doresc să utilizeze tehnologii performante, motivul principal fiind prețul foarte ridicat al acestor tehnologii (sisteme informatice de tip business) și echipamentele informatice asociate utilizării, (\*\*), și (2) aspecte ale managementului informatic asociate în utilizarea platformei ce implică managementul schimbării (sistemelor informatice) și al calității (informațiilor procesate).

Invenția KYBS are aplicabilitate industrială prin

- i. unele funcționalități ale platformei care vor permite unor module ale sistemelor informatice integrate de tip ERP, CRM și DMS să colaboreze tehnic cu echipamentele informatice, video și de birotică asociate utilizării acestor sisteme în diverse medii de operare industrială și de afaceri.
- ii. dezvoltarea, în cadrul programului de calculator, a unor funcții tehnice, prin rularea instrucțiunilor de program ale platformei pe calculatoarele clienților, vor avea efecte tehnice ulterioare ce vor îmbunătăți performanțele echipamentelor din sistemele informatice sau vor permite controlul unor procese tehnologice ale clienților platformei (prin intermediul rutinelor sau modulelor).

În sinteză, invenția (1) implică o rețea de calculatoare și alte aparate cu tehnologia informației și comunicațiilor programabile și (2) prezintă mai multe caracteristici inovatoare ce sunt realizate totalmente și în parte de modulele programului pentru calculator. Platforma online de business are o arhitectură bazată pe metadata, structurată pe 3 nivele (3-tier). Acest tip de arhitectură, bazat pe metadata, permite dezvoltarea de aplicații multitenant cu un grad foarte ridicat de customizare, facilitate ce este încorporată în invenție prin modificarea modalității de utilizare și

partajare a resurselor asociate utilizării sistemelor informatice integrate. Nivelurile acestei arhitecturi sunt: nivelul server, nivelul de stocare (database) și nivelul interfață client. Fiecare nivel este compus din mai multe componente, astfel:

1. Nivelul server este compus din componentele: Compiled Runtime Engine (Kernel), Web Services Facade, Data Acces Layer, Business Logic Layer, Business Entities ș.a..
2. Nivelul de stocare este compus din componentele: Tabele de date, Tabele de Metadate, Tabele Pivot ș.a.
3. Nivelul interfață client este compus din componentele: Model, View, Controller, Repository ș.a.

Platforma va fi găzduită într-un data center propriu și clienții vor accesa platforma prin internet. Astfel, clienții noștri nu vor mai nevoiți să dețină un data center propriu (camera servere) și să gestioneze costurile aferente acestuia, precum: costuri licențe, costuri echipamente/hardware, costuri întreținere (electricitate, chirie spațiu, etc.), costuri personal specializat, etc.

*Avantajele* pe care invenția Platforma on-line pentru Gestiunea Afacerilor - KYBS le prezintă sunt următoarele:

- 1) Platforma integrează, în mod inventiv, în același sistem funcționalități specifice de business, funcționalități regăsite în sisteme informatice complexe, de tip business: ERP, CRM și DMS (sisteme informatice care până acum, luate individual, erau inaccesibile companiilor mici și mijlocii). Astfel, prin integrarea acestor funcționalități se obține o eficiență maximă în gestiunea afacerilor cu ajutorul unor sisteme informatice integrate. Exemple de funcționalități specifice sistemelor informatice de tip ERP, CRM și DMS: gestiunea financiar contabilă, gestiunea mijloacelor fixe, gestiunea contractelor de vânzare, gestiunea forței de vânzare (lead-uri, campanii, oportunități, etc.), Full Text Document Search (căutare bazată pe metadate sau pe funcționalitatea motoarelor de baze de date Full Text Search), Forme electronice, Check in/Check out pentru editarea documentelor, etc.
- 2) Accesibilitate și mobilitate: Platforma este accesibilă de oriunde există internet, indiferent de calculatorul sau dispozitivul de pe care este accesată. Astfel, utilizatorul se poate conecta la platforma de pe orice dispozitiv (PC, laptop, smartphone, tableta, etc.) care are conexiune la internet; în felul acesta, se obține o eficientizare operațională în zona de mentenanță/suport IT, nemaifiind necesară instalarea de aplicații client pentru a putea utiliza platforma și de asemenea utilizatorul nu mai este constrâns să se conecteze la platforma dintr-un punct fix (de pe un desktop).

3) Platforma are un grad ridicat de customizare/personalizare, putând modela diverse tipuri de afaceri. Astfel, aceasta platformă poate modela afaceri precum: leasing, asigurări, executări silite, asset management, colectare creanțe, service auto, etc.

De asemenea, clientul/utilizatorul platformei poate customiza aplicația conform necesităților afacerii. Astfel, printr-un cont de administrator, utilizatorul poate adăuga câmpuri, elimina câmpuri, modifica tipuri de date, adăuga reguli, constrângeri, etc. Avantajele majore ale acestui mod de administrare sunt:

- Reducerea costurilor: deoarece clientul nu mai este nevoit să solicite furnizorului aplicației servicii dezvoltare software (modelul clasic de dezvoltare aplicații software funcționează astfel: utilizatorul solicită furnizorului software customizări, iar furnizorul le implementează contra cost), acesta putând să implementeze customizările în aplicație.
- Independența utilizatorului: utilizatorul nu mai este nevoit să aștepte ca furnizorul aplicației să aloce timp și resurse pentru implementarea customizărilor cerute.
- Lipsa conflicte customizări: modelul actual de implementare a customizărilor prin care furnizorul gestiona implementarea acestora putea conduce la incompatibilități între customizările solicitate de mai mulți clienți, astfel rezultând respingerea unor customizări. Uneori, furnizorul refuză customizările și din cauza creșterii complexității administrării versiunilor aplicației.

4) Un alt avantaj important al acestei platforme îl reprezintă scalabilitatea. Platforma este construită pe baza conceptelor Cloud Computing și Multitenant, aceste concepte asigurând prin definiție scalabilitatea. Astfel resursele hardware și software sunt utilizate la comun de către toți clienții iar sistemul prin motoare (engine-uri) și servicii specializate va furniza clienților conectați la platforma cea mai bună performanță tehnică posibilă. De asemenea sistemele ce funcționează pe baza Cloud Computing sunt foarte ușor de upgradat din punct de vedere al resurselor hardware, în cazul în care gradul de încărcare al resurselor fizice necesită acest lucru. Prin implementare în platformă a conceptului Multitenant, se va realiza o partiționare a datelor în funcție de client, astfel întregul volum de date gestionat de către platforma va fi împărțit fizic în subvolum de date aferente fiecărui client. Acest lucru nu numai că asigură scalabilitatea sistemului, dar în același timp asigură o performanță foarte bună a sistemului.

48

- 5) Securitate sporită – deoarece platforma este gazduită într-un singur data center este mult mai eficient și ușor de gestionat prin operațiunile ce țin de securitatea datelor (backup, actualizări software, alocarea celor mai bine pregătite resurse umane, etc.). De asemenea conexiunea între client și platforma se realizează printr-o conexiune securizată/criptată (de ex. AES 128-bit), neexistând riscul interceptării datelor sau al transmiterii datelor „in alb”, așa cum se mai întâmpla în cazul unor sisteme informatice.
- 6) Tehnologiile și conceptele de ultim moment ce vor fi utilizate (Cloud Computing, multitenant, etc.) precum și gradul ridicat de inovare vor avea un impact pozitiv semnificativ asupra mediului, în principal din punct de vedere al eficienței energetice (consumul de energie electrică este optimizat, economia de energie electrică fiind direct proporțională cu numărul de clienți, astfel cu cât numărul de clienți este mai mare cu atât se obține o economie mai mare de energie electrică).
- 7) Costurile inițiale/de achiziție și costurile de utilizare/mentenanță ale acestei platforme sunt nule pentru clienții noștri. Soluțiile tradiționale presupuneau costuri ridicate atât pentru achiziția sistemului (servere, licențe, etc.) cât și pentru mentenanță/suport. Aceste costuri au constituit de-a lungul timpului o barieră pentru companiile mici și mijlocii în accesarea tehnologiilor moderne și complexe, lucru care reprezintă un mare dezavantaj în mediul afacerilor. Datorită costurilor de achiziție și costurilor de întreținere/mentenanță nule, platforma facilitează accesul la tehnologii noi/performante tuturor companiilor interesate indiferent de dimensiunea acestora (până acum prin modul tradițional de a dezvolta și vinde software companiile mici și mijlocii nu aveau acces la soluții software complexe din cauza costurilor de achiziție și mentenanță foarte ridicate). Prin facilitarea accesului companiilor la tehnologii noi/performante această platformă contribuie la îmbunătățirea mediului de afaceri prin optimizarea tehnologică și operațională a companiilor.

Descrierea invenției din domeniul programelor de calculator este însoțită de *reprezentări grafice* care prezintă etapele sau pașii operaționali aferenți procesării datelor, după cum urmează:

1. asimilarea conceptului Multitenancy în platforma on-line
2. arhitectură platforma KYBS, respectiv arhitectura funcțională și arhitectura hardware.

Invenția va fi realizată dintr-o serie de programe, subansamble, subprograme și alte componente informatice (proceduri stocate, add-on-uri, plug-in-uri, etc.). După cum se observă în graficul *Arhitectura platformei* subansamblele platformei sunt: subansamblul DataBase (DataBase Tier), subansamblul Server (Server Tier) și subansamblul Client (Client Tier).

i. Subansamblul DataBase va fi alcătuit din obiecte specifice bazelor de date, precum: tabele de date, proceduri stocate, indexi, view-uri, etc. La nivelul acestui subansamblu abordarea este una inovativă prin faptul că datele sunt stocate într-un grup de tabele de tip „Warehouse”. Acest tip de organizare se referă la organizarea datelor sub forma unor depozite de date, indiferent de proveniența acestora. Toate proprietățile specifice tabelor unei baze de date sunt gestionate printr-o structură de baze de date denumite Metadata Business Logic Tables. În cadrul acestei structuri vor fi gestionate relațiile dintre entități, validările existente pe entități, dependențele dintre entități, etc. Un al treilea grup de tabele, denumite Metadata System Tables, va gestiona toate informațiile referitoare la întregul sistem, precum: entitățile gestionate de sistem, chirișii care se conectează la sistem - „tenants”, etc.

ii. Subansamblul Server este compus dintr-o serie de programe a căror funcționalitate permite gruparea acestora în următoarele subansamble de programe:

- Metadata Services
- Security Services
- System Services
- Business Services
- Process Services.

De asemenea la nivelul subansamblului Server va exista o interfață API de interconectare a sistemului cu sisteme din exterior. Astfel, prin dezvoltarea de diverse plug-in-uri sistemul va putea fi integrat cu diverse sisteme informatice.

iii. Subansamblul client va conține componente precum Model, View și Controller, conform design pattern-ului MVC. Acest subansamblu reprezintă interfața utilizator a invenției care va fi accesibilă dintr-un browser de internet.

Tehnologiile utilizate în dezvoltarea acestui subansamblu sunt tehnologii moderne, de ultim moment: ASP.Net MVC 4, HTML5, JavaScript/Jquery.

*Desenul nr.1* reprezintă o privire de ansamblu asupra platformei KYBS în care figurile numerotate de la (1) la (5) reprezintă următoarele:

- (1) Nivel Client/Interfața utilizator: de la acest nivel se efectuează accesul la aplicație/platforma prin utilizarea unui browser web și a unei conexiuni la internet
- (2) Internet
- (3) Firewall: aplicații ce blochează accesul neautorizat

(4) Nivel Server: colecție de ansamble/subansamble de programe

(5) Nivel DataBase: colecție de structuri de date și obiecte specifice bazelor de date ce facilitează stocarea, operarea și manipularea datelor și informațiilor aplicației/platformei

*Desenul nr.2* reprezintă Arhitectura funcțională a nivelului Client în care figurile numerotate de la (1) la (10) reprezintă următoarele:

(1) Browser internet (Internet Explorer, Firefox Mozilla, Chrome, etc.)

(2) Solicitare HTTP, din browser către Controller

(3) Controller: subansamble de programe ce gestionează acțiunile pe care utilizatorii le efectuează/solicita

(4) Solicitare obiecte (poate fi solicitare HTTP, Websockets, etc.) de la Controller către Model

(5) Model: subansamblu de programe ce gestionează obiectele și regulile de business ale aplicației

(6) Raspuns solicitare obiecte (de ex. WebResponse) de la Model către Controller

(7) Transmitere obiecte: Controllerul trimite obiectele către View pentru ca acestea să fie populate cu date

(8) View: subansamblu de programe ce gestionează popularea obiectelor cu date

(9) Raspuns transmitere obiecte, de la Model către Controller

(10) Raspuns HTTP, Controller trimite către interfața/browser web obiectul solicitat, acesta conținând datele corespunzătoare.

*Desenul nr.3* reprezintă Arhitectura platformei în care figurile numerotate de la (1) la (20) reprezintă următoarele elemente de arhitectură cu 3 nivele (3 Tier).

(1) Nivelul UI – reprezintă interfața utilizator. Interfața utilizator este accesibilă dintr-un browser de internet, iar această interfață va fi realizată folosind limbaje de programare precum: HTML5, JavaScript, ASP.NET și platforma Microsoft .Net Framework 4.5.

(2) Nivelul Server – reprezintă structura de programe ce oferă setul de funcționalități ale platformei. Astfel, nivelul server, furnizează funcționalitățile necesare gestionării autentificării pe platformă, autorizării operațiilor efectuate, salvării datelor în baza de date, încărcării datelor din baza de date, etc. De asemenea, nivelul server, conține instrumentele și funcționalitățile necesare astfel încât platforma să fie scalabilă, care să permită accesul concurențial și care să permită un înalt grad de personalizare/customizare. Acest nivel va fi realizat utilizând limbajul de programare C#, ASP.NET și platforma Microsoft .Net Framework 4.5.



- (3) Nivelul DataBase – reprezinta baza de date in care sunt stocate datele. Pentru realizarea acestui nivel se va utiliza Microsoft SQL Server 2012.
- (4) Interfata utilizator – set de programe ce faciliteaza accesul utilizatorului la aplicatie/platforma. Arhitectura ptr interfata utilizator va fii o combinatie intre modelele de proiectare (design patterns) MVC (Model-View-Controller), Facade și Repository.
- (5) Servicii Metadata (subansamblu de programe): mecanism prin intermediul caruia sunt gestionate asa zisele metadatae. Metadataele reprezinta un set de informatii despre anumite aspecte ale aplicatiei (de ex. metadataele pot reprezenta informatii despre un model de date: tipuri de date utilizate, valori implicite, denumire in baza de date, denumire in interfata utilizator, validari, formatari, etc.)
- (6) Servicii Securitate (subansamblu de programre) – prin acest set de servicii se implementeaza nivelul de securitate in aplicatie platforma (autentificare, autorizare, audit, conexiuni criptate și securizate, etc.).
- (7) Servicii procese – categorie de programe ce include punctul (8)
- (8) Web Services Facade (subansamblu de programe) – colectie/set de servicii web prin care nivelul server este interfatat cu nivelul client.
- (9) Servicii Business – categorie de programe ce include punctele (10), (11) si (12)
- (10) Data Acces Layer – DAL (subansamblu de programe) – set de programe prin care se faciliteaza accesul la nivelul DataBase.
- (11) Business Logic Layer – BLL (subansamblu de programe) – set de programe prin care se implementeaza logica/modelul de business in aplicatie. Acest set de servicii/functionalitati contin validari, fluxuri de procese/business, etc.
- (12) Business Entities – BE (subansamblu de programe) – reprezinta o colectie de obiecte ce modeleaza businessul (de ex. contracte, facturi, incasari, etc.).
- (13) Servicii Sistem – categorie de programe ce include punctele (14), (15), (16) si (17)
- (14) Nucleu/Kernel (subansamblu de programe) – reprezinta modulul central al aplicatiei/platformei. Acest modul central gestioneaza servicii de baza precum alocarea memoriei, gestioneaza procesele, etc.
- (15) Tenant Load Balancer (subansamblu de programe) – reprezinta modulul prin care platforma gestioneaza solicitarile diversilor clienti (tenants). Astfel, platforma va stii sa directioneze solicitarile clientilor catre resursele corespondente ale fiecarui client (baza de date, partitii de tabele din baza de date, mecanisme de business personalizate, etc.)

(16) Coordonator Tranzactii (subansamblu de programe) – mecanism prin care platforma/aplicatia gestioneaza tranzactiile efectuate de catre utilizatorii. Acest mecanism optimizeaza modul in care tranzactiile sunt operate avand impact direct asupra performantei sistemului (prioritatea tranzactiilor, blocajele, etc.)

(17) Autentificare & Autorizare (subansamblu de programe) – mecanism prin care se verifica și se permite accesul utilizatorilor in sistem (autentificare) și de asemenea se autorizeaza operatiunile pe care utilizatorii doresc sa le efectueze (gestiunea rolurilor și a permisiunilor utilizatorilor).

(18) Tabele Metadate Sistem – tabele din baza de date unde sunt stocate informațiile despre platformă, customizările clienților, informații despre entitățile manipulate de către platformă, etc.

(19) Tabele Metadate Logica Business – tabele din baza de date unde sunt stocate informațiile despre relațiile dintre entități, validări asupra entităților, reguli de business, etc.

(20) Tabele Entități Business – tabele de tip “Warehouse” în care sunt stocate toate datele pe care platforma le operează.

Realizarea invenției informatice va permite integrarea sistemului modular de tip platformă cu diverse dispozitive tehnice și terminale de telecomunicații:

- a) Scaner coduri de bare – aceasta funcționalitate poate fi implementată pe smartphoneuri prin instalarea unei aplicații proprii. Astfel un utilizator va putea încărca în sistem informații despre un produs doar citind cu un smartphone codul de bare al produsului prin intermediul aplicației dedicate smartphoneurilor pe care o vom dezvolta. Scanarea de coduri de bare poate fi asociată cu o funcționalitate atât a sistemului ERP (citirea automată de facturi) cât și a sistemului DMS (identificarea documentelor).
- b) Voice Recording - La nivelul unui smartphone prin intermediul unei aplicații dezvoltate de noi putem înregistra convorbirile pe care operatorii unei companii le au cu diverși clienți. Astfel, aceste convorbiri efectuate pe smartphoneuri pot fi salvate automat în sistem și înregistrate la nivelul clientului respectiv. Această funcționalitate este una specifică sistemelor informatice de tip CRM.
- c) Geo coding - La nivelul unui smartphone, prin aplicația noastră, un agent de vânzare poate fi urmărit prin intermediul GPS-ului pentru identificarea locațiilor clienților vizitați de agent într-o anumită perioadă. Datele colectate prin GPS de aplicația instalată pe smartphone sunt înregistrate în sistem în timp real. Astfel un manager de vânzări poate

verifica dacă vânzătorii își respectă agenda, poate fi notificat în legătură cu întârzierile sau neconformitățile de itinerar. Această funcționalitate se încadrează la sistemele CRM/SFA (sales force automation).

- d) Geo coding - La fel ca la punctul c), o companie își poate monitoriza flota de mașini cu aplicația inventată. GPS-urile instalate pe autovehicule pot fi programate să comunice cu sistemul nostru, astfel datele despre parcurs fiind înregistrate în sistem (distanțe, consum, etc.). Această funcționalitate se încadrează la sisteme ERP/Fleet management.
- e) Geo coding – companiile care prestează diferite tipuri de servicii pot utiliza funcționalitatea pentru a vizualiza pe terminal locația agenților și a monitoriza din dispecerat în timp real și a direcționa cel mai apropiat agent către client. Această funcționalitate se poate aplica companiilor de curierat, de taximetrie, externalizare servicii IT, etc. Aceasta funcționalitate se încadrează la sisteme de tip ERP.
- f) POS – dispozitivele de tip POS pot fi programate astfel încât să comunice cu aplicația inventată și să înregistreze plățile efectuate de către clienți. Această funcționalitate se încadrează la sisteme ERP.
- g) Scanner – astfel de dispozitive pot fi programate pentru a scana, arhiva și indexa automat documente în format electronic. Această funcționalitate se încadrează la sisteme DMS.

*Modul de realizare a invenției este prezentat detaliat și prin fragmente din programul pentru unitățile de procesare a datelor (computere). Aceste segmente de cod sursă redau o descriere sumară a funcțiilor din cele 3 subansamble.*

#### 1. Reprezentare clasa în ASP.NET MVC.

Clasa este folosită ca un container pentru a reține datele care sunt preluate din baza de date.

```
public class Plan
{
    public int ID { get; set; }
    public DateTime Data { get; set; }
    public float Costs { get; set; }
    public float InterestRate { get; set; }
    public float Equity { get; set; }
    public float Leverage { get; set; }
    public float NewBusiness { get; set; }
    public float NonPerforming { get; set; }
}
```

```

public float OnGoing { get; set; }
public float Bucket { get; set; }
}

```

## 2. Reprezentare Repository in ASP.NET MVC

Repository-ul reprezintă legătura între clase și baza de date. Această legătură se face prin intermediul procedurilor stocate, care permit definirea de parametri și care sunt ușor de apelat în C#. Aceste repository-uri sunt de fapt metodele unei clase. Sunt separate de clase, tocmai pentru a realiza o cât mai mare independență a acestora. Aceste repository-uri conțin toate metodele folosite pentru o clasă, chiar dacă returnează tipuri obișnuite de tip int, string, bool, sau liste de clase. Aceste repository-uri reprezintă DAL-ul aplicației (Data Access Layer).

```

public List<Plan> Plan_GetAll()
{
    var list = new List<Plan>();
    using (var conn = new SqlConnection(SqlHelper.ConnectionString))
    {
        var cmd = conn.CreateCommand();
        cmd.CommandText = "Reporting_GetAllPlane";
        cmd.CommandType = System.Data.CommandType.StoredProcedure;
        conn.Open();
        using (var reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                var elem = new Plan();
                elem.ID = Convert.ToInt32(reader["ID"]);
                elem.Data = Convert.ToDateTime(reader["DAte"]);
                elem.Bucket = float.Parse(reader["Bucket"].ToString());
                elem.InterestRate = float.Parse(reader["InterestRate"].ToString());
                elem.Equity = float.Parse(reader["Equity"].ToString());
                elem.Leverage = float.Parse(reader["Leverage"].ToString());
                elem.Costs = float.Parse(reader["Costs"].ToString());
                elem.NewBusiness = float.Parse(reader["NewBusiness"].ToString());
                elem.NonPerforming = float.Parse(reader["NonPerforming"].ToString());
                elem.OnGoing = float.Parse(reader["OnGoing"].ToString());
                list.Add(elem);
            }
        }
    }
}

```

```
return list;    }
```

### 3. Reprezentare Service in ASP.NET MVC

Service-ul reprezintă puntea de legătură între interfața aplicației și baza de date și în termeni de programare reprezintă interfețe și asigură o securitate sporită. Aceste service-uri seamănă destul de mult ca mod de apelare cu web service-urile standard din .NET

```
public class PlanService
{
    PlanRepository repo;
    public PlanService()
    {
        repo = new PlanRepository();
    }
    public List<Plan> Plan_GetAll()
    {
        return repo.Plan_GetAll();
    }
}
```

### 4. Reprezentare Controller in ASP.NET MVC

Controller-ele reprezintă logica aplicației și modul cum vor fi afișate elementele în interfața cu utilizatorul. Controller-ul este de fapt locul unde sunt apelate toate metodele existente în repository, prin apelarea metodelor din service. Totuși aici poate fi creată o nouă logică, care să nu țină cont în totalitate de metodele existente. Controller-ele fac parte din proiectul Web al aplicației, iar database și service sunt folosite ca referințe în acest proiect. Un controller returnează o acțiune, care are în spate un view care va fi generat pentru a putea apoi fi vizualizat fără probleme de către utilizator. Controller-ul este format dintr-o multime de acțiuni.

```
public class HomeController : Controller
{
    [Authorize]
    public ActionResult Index()
    {
        .....
    }
    [Authorize]
    public ActionResult Print(DateTime date, DateTime dateS)
    {
        .....
    }
}
```

### 5. Reprezentare Action in ASP.NET MVC

Acțiunile sunt de fapt paginile care sunt generate în interfață. Aceste acțiuni pot să aibă definite o serie de atribute, dar cel mai important atribut este cel de [Authorize], care nu permite unui utilizator care nu este conectat în aplicație printr-un user și o parolă să vizualizeze pagina generată de acțiune. Acțiunile desigur nu generează doar pagini web, ci și pagini care au alte formate. Sper exemplul din acțiuni se pot genera pagini care returnează un fișier xml.

```
[Authorize]
```

```
public ActionResult CreatePlan()
```

```
{ return View(); }
```

```
[HttpPost]
```

```
public ActionResult CreatePlan(Database.Plan model)
```

```
{new Service.PlanService().Plan_Add(model.Data.AddMonths(1).AddDays(-1),
model.InterestRate, model.Leverage, model.Equity, model.NewBusiness, model.Costs,
model.NonPerforming, model.OnGoing, model.Bucket);
return Redirect("/Home/About"); }
```

## 6. Reprezentare View in ASP.NET MVC

View-urile reprezintă generarea paginilor care sunt afișate în interfața cu utilizatorul. Aceste view-uri sunt returnate de către acțiuni și conțin elemente de tip HTML, foi CSS, elemente JQuery sau JavaScript, care sunt astfel folosite pentru a genera pagini web.

```
@{
```

```
ViewBag.Title = "Print";
```

```
Layout = null;
```

```
var dataF = ViewBag.Valo;
```

```
var dataStart = ViewBag.DataS;
```

```
var data = Convert.ToDateTime(dataF);
```

```
var dataS = Convert.ToDateTime(dataStart); }
```

```
<link href="@Url.Content("~/Content/Site.css")" rel="stylesheet" type="text/css" />
```

```
<script src="@Url.Content("~/Scripts/jquery-1.5.1.js")" type="text/javascript"></script>
```

```
<script src="@Url.Content("~/Scripts/jquery-ui-1.8.11.min.js")"
type="text/javascript"></script>
```

```
<script src="@Url.Content("~/Scripts/modernizr-1.7.min.js")"
type="text/javascript"></script>
```

```
<link href="@Url.Content("~/Content/themes/base/jquery.ui.all.css")" rel="stylesheet"
type="text/css" />
<script type="text/javascript">
    function toggle2(id, link) {
        var e = document.getElementById(id);
        if (e.style.display == "") {
            e.style.display = 'none';
            link.innerHTML = 'Descriere +';
        } else {
            e.style.display = "";
            link.innerHTML = 'Descriere -';        }    }
</script>
<script type="text/javascript">
    $(document).ready(function                ()                {
    $("#rapoarte1").load('/Home/Raport1?date=@ViewBag.Data&dateS=@ViewBag.DataS');
    $("#rapoarte2").load('/Home/Raport2?date=@ViewBag.Data&dateS=@ViewBag.DataS');
    });
</script>
<script src="@Url.Content("~/Scripts/jsapi.js")" type="text/javascript"></script>
<script type="text/javascript">
    google.load("visualization", "1", { packages: ["corechart"] });
</script>
<div id="rapoarte1" style="width:950px; ">
</div>
<div style="clear:both;"></div>
<div id="rapoarte2" style="width:950px;">
</div>
```

*Aplicabilitatea industrială* a invenției este justificată prin următoarele:

1. Prezintă utilitate tehnică în operarea sistemelor informatice integrate și a echipamentelor informatice asociate utilizării;
2. Realizarea acestei invenții informatice va permite integrarea sistemului modular de tip platformă cu diverse dispozitive tehnice și terminale de telecomunicații;

3. Platforma on-line este în mod obiectiv realizabilă pe baza următoarelor caracteristici definitorii ale acesteia: accesibilitate, flexibilitate, scalabilitate și customizare. Aceste caracteristici sunt avantaje nete în fața sistemelor informatice clasice (identificate în stadiul tehnicii) și vor conduce către un grad foarte ridicat de multiplicare a platformei.



## REVENDICARE

Programul de calculator de tip platformă on-line KYBS (Keep Your Business Smart), caracterizat prin aceea că va permite implementarea sistemelor informatice integrate de tip ERP (Enterprise Resources Planning), CRM (Customer Relationship Management) și DMS (Document Management System) într-un sistem informațional unic la nivelul unei societăți comerciale sau entitate publică, prin externalizarea către platforma on-line conform invenției, ce este alcătuită din 3 nivele ce folosesc conceptul de Cloud Computing: nivel interfață utilizator, nivelul server, și nivelul baza de date, va funcționa într-un centru de date și va fi utilizat de către utilizatori prin intermediul rețelelor de calculatoare și internetului, fiind utilizat pentru gestiunea afacerilor.

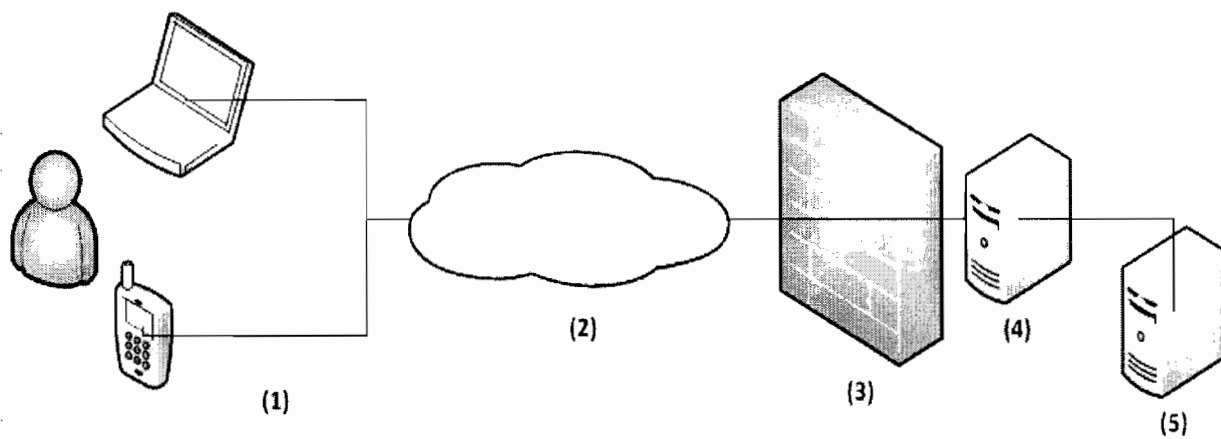


Fig. 1

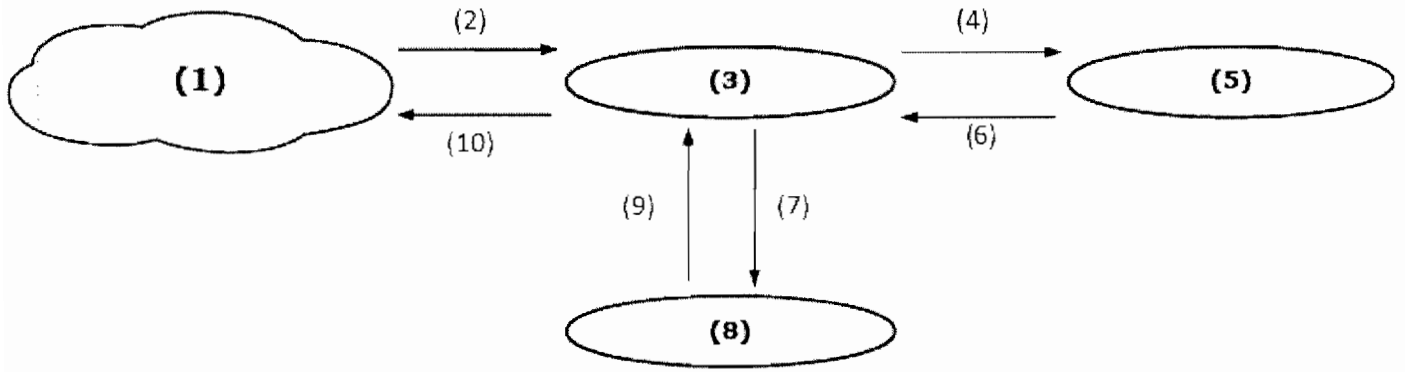


Fig. 2

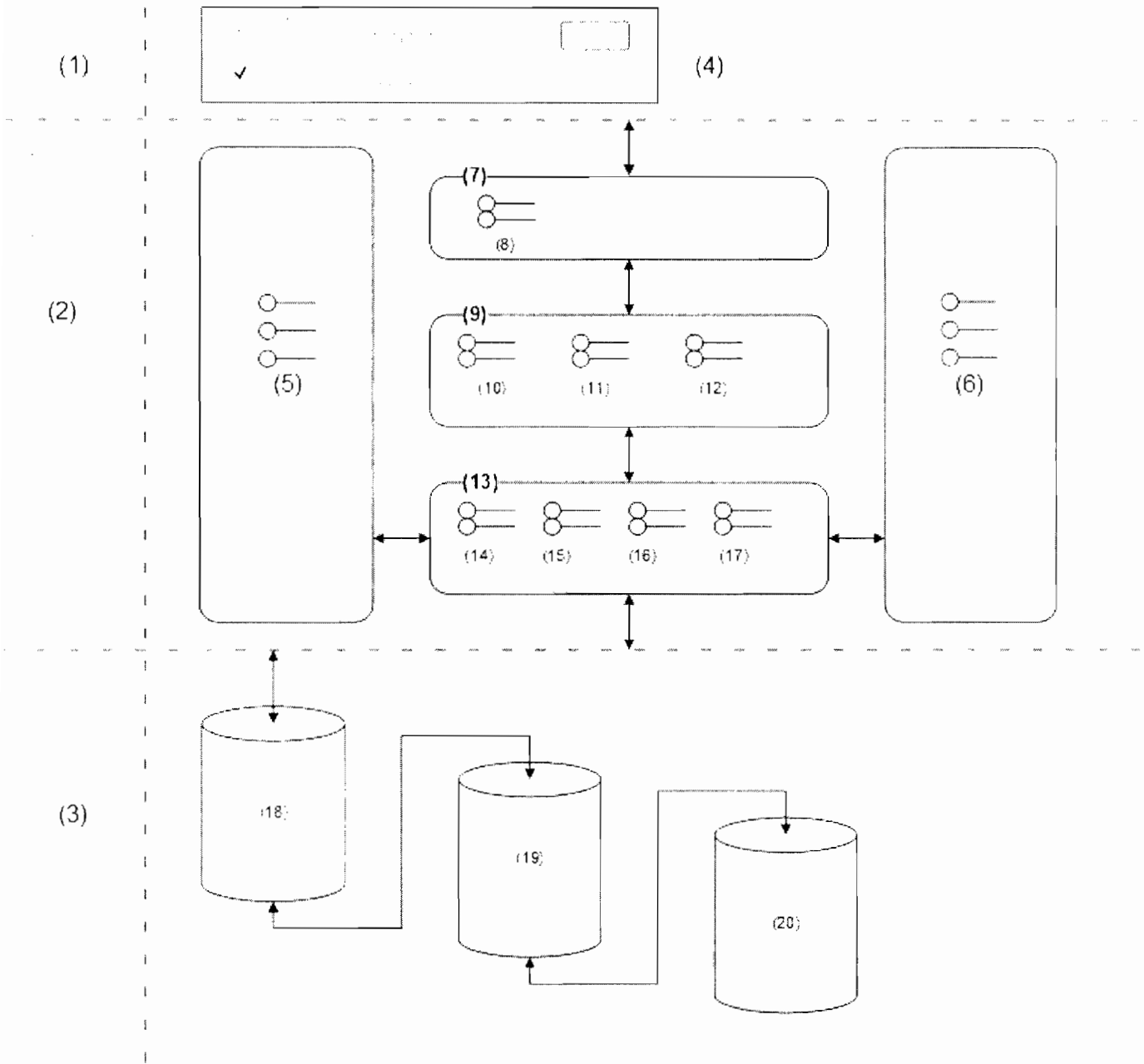


Fig. 3