**Tema 1 PSSC**

**Analiza unui sistem software**

Sistem de furnizare de servicii

**Grofu Viorela**

**Grupa 1.3**

1. Prezentarea cerintelor

1.1.Functionale

1.2.Non-functionale

1.3. Cerinte care influenteaza arhitectura

1. Componente

2.1.Descompunerea in component

2.2. Definirea responsabilitatilor componentelor si a relatiilor dintre ele. Argumentare

1. Prezentarea sistemului software din doua perspective
2. Indicatori de calitate

4.1.Identificarea celor mai importanti 3 indicatori de calitate

1. Identificarea tehnologiilor middleware folosite pentru a comunica intre componente. Argumentarea alegerilor
2. Identificarea pincipalelor modele si stiluri arhitecturale folosite. Argumentarea alegerilor

7.Prezentarea scenariilor de validare a arhitecturii

**Cerinte functionale**

Sistemul va permite accesul tuturor persoanelor la informatii referitoare la o varietate de servicii, informatii actualizate in timp real.

Sistemul furnizeaza date cu privire la furnizorii de servicii.

Sistemul permite persoanelor sa primeasca online informatiile solicitate, pe baza mai multor intrebari care au menirea de a clarifica serviciul solicitat.

Sistemul trebuie sa permita centralizarea tuturor informatiilor cu privire la serviciul solicitat, fara a fi influentate de prioritizarea furnizorilor in functie de valoarea de piata.

Sistemul trebuie sa acorde drepturi totale administratorului de sistem(poate adauga informatii noi, poate actualiza date existente, poate vizualiza recenziile furnizoriilor de servicii, poate prelucra date cu caracter personal in vederea unui sondaj cu privire la acuratetea sistemului)

**Cerinte non-functionale**

Sistemul va fi implementat in Visual Studio 2015, cu ajutorul tehnologiei ASP.NET MVC.

Sistemul va acorda tuturor utilizatorilor servicii transparente. Fiecare persoana poate avea acces la toate informatiile solicitate intr-un timp foarte scurt, informatiile fiind reale si actualizate in timp real.

Baza de date implementata va fi de tip Entity Framework.

Cerinte care influenteaza arhitectura

Sistemul va fi implementat pe baza unei arhitecturi de tip client-server.

**2. Componente**

2.1 Descompunerea in componente

1. Administrator de server- drepturi- prelucreaza informatii cu privire la:

a. Furnizor de servicii

b. Domeniul de activitate

c. Servicii furnizate

d. Informatii cu privire la activitatea desfasurata

e. Adresa furnizorului

f. Numar de telefon

g. Recenziile clientilor

2. Utilizator

a. Lista servicii

b. Vizualizare intrebari cu privire la clarificarea serviciului solicitat

c. Generare rapoarte

3. Baza de date realizeaza legatura intre componentele mai sus amintite.

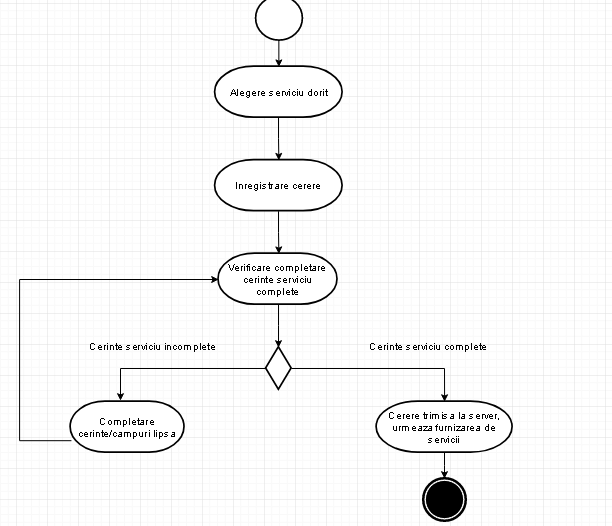
4. Baza de date a administratorului – ofera acestuia drepturile pentru preluarea anumitor informatii.

5. In functie de serviciul ales, sistemul va genera o serie de intrebari cu privire la clarificarea domeniului de activitate, adresa, momentul necesitatii serviciului, samd.

6. Generare rapoarte.

Prezentare perspectiva

1.Prima perspectiva



**Indicatori de calitate**

4.1.Identificarea celor mai importanti 3 indicatori de calitate

1. Securitatea informatiei transferate

Informatiile transmise de catre un furnizor de servicii vor fi depozitate in baza de date in conditii de siguranta.

2. Corectitudinea datelor afisate

Datele oferite sunt in concordanta cu cele transmise de furnizor, fara a suferi modificari sau a le fi omise anumite aspecte.

3. Scalabilitatea din punct de vedere al numarului de furnizor de servicii.

Numarul de furnizori de servicii poate atinge cote maxime, iar blocarea sistemului la volum mare de date nu este permisa.

**5. Identificarea tehnologiilor middleware folosite pentru a comunica intre component,argumentarea alegerilor**

Se va folosi modelul bazat pe cozi de mesaje deoarece in cazul in care cererile unor clienti nu pot fi procesate imediat de catre server, sistemul sa nu se blocheze si sa permita in continuare utilizarea interfetelor.

De asemenea, serverul poate gestiona mai mulți clienți conectați simultan.

Un astfel de server poate să gestioneze mai multe cozi simultan, respectiv mai multe cereri transmise simultan prin intermediul firelor de execuție organizate in pool-uri de fire de execuție. Un singur proces poate trimite mesaje către mai multe cozi, respectiv fiecare coadă poate fi interogată de unul sau mai mulți destinatari.

**6. Identificarea pincipalelor modele si stiluri arhitecturale folosite. Argumentarea alegerilor**

Se va utiliza modelul arhitectural client-server.

Clienţii şi serverele vor comunica printr-o reţea de calculatoare(prin Internet). Clienţii nu partajează niciuna dintre resursele proprii, ci apelează la resursele serverului prin funcţiile server, inițiând sesiuni.

Se va utiliza MVC(Model-View-Controller) deoarece separă funcţionalitatea specifică domeniului pentru care este dezvoltat sistemul software de interfaţă grafică a aplicaţiei.

**7.Prezentarea scenariilor de validare a arhitecturii**

Scenarii posibile:

1. Un constructor completeaza informatii pentru a-si oferi serviciul in cadrul acestui sistem

2. Constructorul nu completeaza toate campurile cu informatiile corecte

3. Un client decide sa utilizeze serviciul contractorului si il cauta la adresa indicata in cadrul site-ului.

4. Constructorul nu se regaseste la adresa indicata iar clientul, dezamagit, ii ofera acestuia o recenzie negativa.