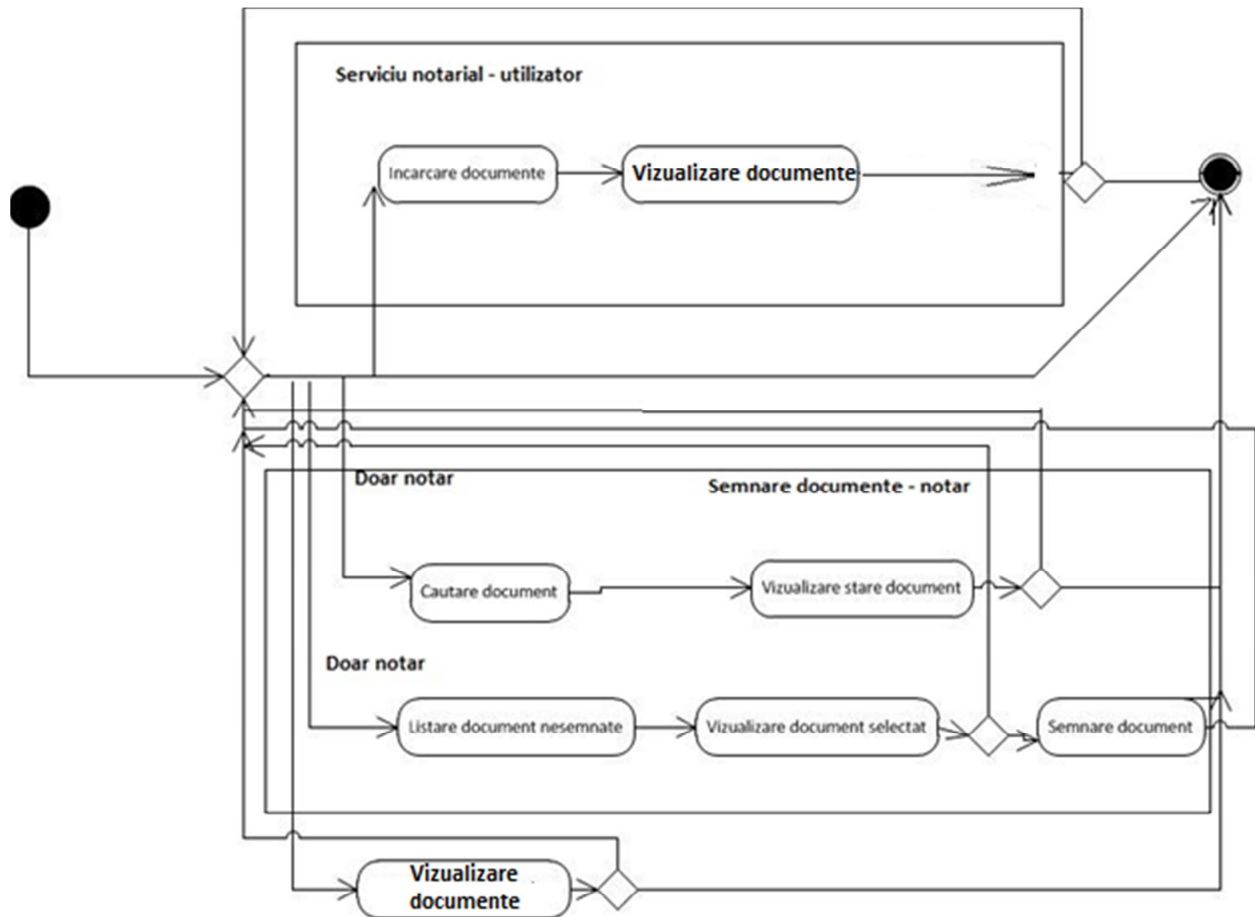


5.2.2.7. Diagrame de activitate

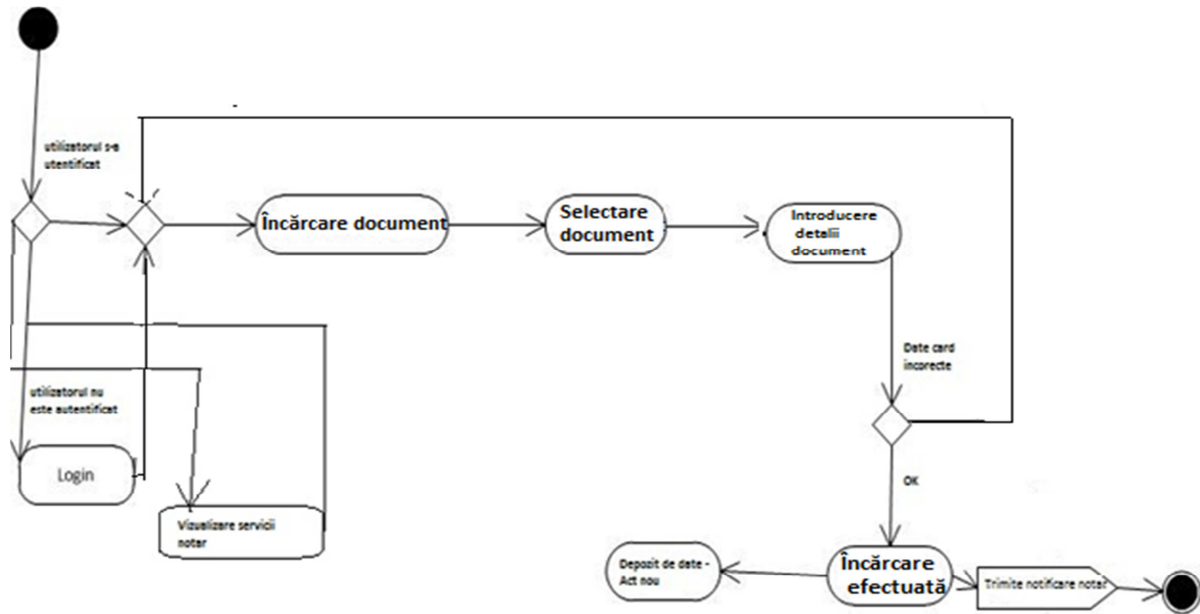
Diagramele de activitate sunt reprezentări grafice a fluxului de lucru pas cu pas a activităților și acțiunilor. Este o variantă a diagramei de stare ce modelează dinamica unui proces sau a unei operații. Acest tip de diagramă pune în evidență controlul execuției de la o activitate la alta.



Figură 5-41 Diagramă generală pentru notar și utilizator

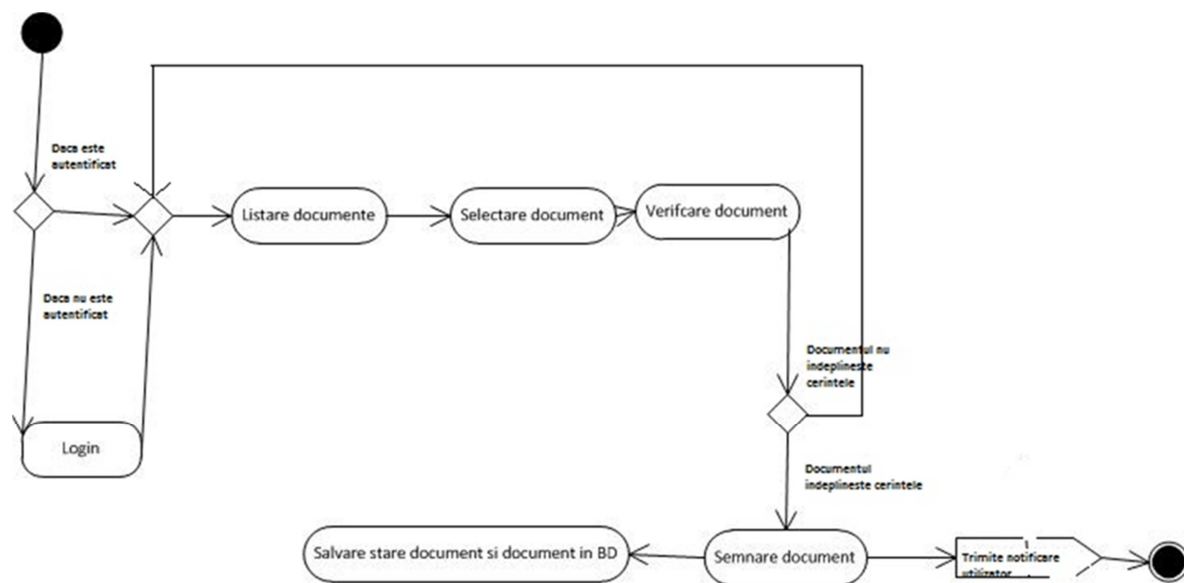
În figura 5-42, este descris modul în care utilizatorul obișnuit interacționează cu sistemul. Utilizatorul are opțiunea de autentificare dacă dorește să solicite serviciile notariale sau opțiunea de a vizualiza lista de servicii și prețuri oferite de notar. După ce se autentifică, utilizatorul poate să aleagă tipul de documente pentru care solicită servicii, să încarce documentul, să introducă detalii despre documentul încărcat pentru a putea fi preluat mai departe de către notar. În cazul în care datele sunt eronate, se revine la faza de reintroducere date. Dacă salvarea datelor și a documentului se realizează cu

succes, aceasta va fi înregistrată în baza de date și va fi trimisă o notificare pe email notarului.



Figură 5-42 Diagramă pentru utilizator

În figura 5-43, este reprezentată diagrama de activitate a notarului. Notarul începe ca și un utilizator obișnuit prin procesul de autentificare. După ce s-a autentificat, el poate cere listarea documentelor salvate în baza de date (în funcție itemii de cautare) , selectarea unui document pe care dorește să-l vizualizeze, verificarea documentului. Verificarea documentului este realizată de notar în funcție de legile din fiecare țară, dacă în urma verificării s-a observat îndeplinirea tuturor cerințelor acesta este semnat electronic, salvat în baza de date in noua stare și este trimisă o notificare către utilizatorul care a introdus în sistem acest document.



Figură 5-43 Diagramă pentru notar

5.3. Testarea aplicației și rezultatele ei

5.3.1. Noțiuni teoretice

Testarea software determină dacă un sistem software este gata de livrare și estimează nivelul de performanță al acestuia. Testarea furnizează o baza pentru interacțiunea cu persoanele implicate în proiect. Creșterea complexității sistemelor software a dus la o creștere a bugetului alocat acestei faze din procesul de dezvoltare al unui proiect (între 30 și 50%). O mare parte a efortului necesar realizării sistemelor software este alocată dezvoltării modelelor de testare și a aplicațiilor de testare automată. [31]

Testarea este procesul execuției programului cu scopul de a pune în evidență erorile. Detectarea erorilor este scopul principal al testării. În același timp, prezintă interes atât dezvoltarea unor seturi de date de test adecvate care să conducă la activarea erorilor precum și modalitățile de alocare a timpului necesar testării, în special în sistemele de mare complexitate. [31]

Metodele de testare dinamică presupun executarea programului folosind așa numitele date de test. Datele de test se construiesc conform cerințelor funcționale specificate, iar rezultatele furnizate de program se compară cu cele prezentate în specificații. [31]

Metodele de testare statică cuprind verificarea programului, analiza anomaliilor, inspecția codului. Verificarea programului necesită specificarea condițiilor la intrare și a postcondițiilor la ieșire. Analiza anomaliilor caută eventuale comportări anormale ale programului (spre exemplu, porțiuni de cod care nu sunt executate niciodată). Scopul testării statice este de a analiza sistemul software și de a deduce operațiile sale curente ca o consecință logică a deciziilor de proiectare. [31]

Această modalitate de testare nu necesită execuția programului.

Următoarele metode de testare vor fi puse în aplicare după terminarea și în timpul dezvoltării programului software:

	Testare funcțională	Testare structurală
Dinamică	Testare aleatoare Testare pe domenii Graf cauză-efect	Testare computațională Testare pe domenii Testare căi Generare date Analiza schimbărilor

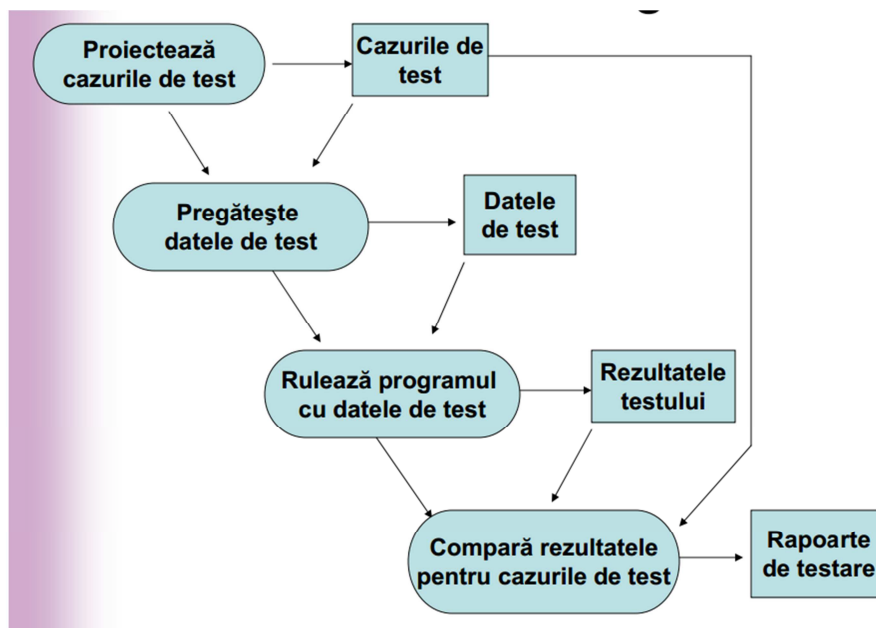
Statica	Verificare specificații	Inspectare cod Verificare program Execuție simbolică Analiză anomalii
---------	-------------------------	--

Tabel 5-1 Metode de testare [31]

5.3.2. Schema de testare urmarita

Etapele schemei de testare

- Se selectează ce trebuie măsurat (cuantificat) de testul respectiv. Înainte de realizarea testului, trebuie identificate scopurile acestuia. Scopurile pot fi diferite (spre exemplu, testarea fiabilității, testarea completitudinii cerințelor). [32]
- Se decide cum se face testarea a ceea ce trebuie testat. După ce s-a stabilit ce este de testat, trebuie decis cum se realizează testele relevante. [32]
- Se dezvoltă cazurile de test. Pentru tipurile de testare deja acceptate, trebuie creată o colecție de cazuri de test (situații de test) pentru antrenarea sistemului supus testării. [32]
- Se determină rezultatele așteptate ale testului respectiv. [32]
- Se execută cazurile de test. [32]
- Se compară rezultatele obținute cu cele așteptate [32]



Tabel 5-2 Schema generală pentru testare [33]

5.3.3. Cazuri de test

Cazurile de test sunt un set de intrări de test, condiții de execuții, și rezultate așteptate dezvoltate pentru obținerea unui anumit scop. Scopul unui caz de test este de a identifica și comunica condiții ce vor implementa testul.

Test Case ID	Scenariu	Username	Certificat	Nu este deja înregistrat	Informatii specifice	Inregistrarea se poate realiza	Rezultate așteptate
TC 1	Inregistrare realizata cu succes	V	V	V	V	V	Mesaj de confirmare
TC 2	Utilizatorul nu a fost gasit	I	N/A	N/A	N/A	N/A	Mesaj de eroare: Inapoi la pagina de logare
TC 3	Informatii incorecte	V	V	N/A	I	N/A	Mesaj de eroare: Reintroducerea datelor
TC 4	Utilizatorul iese din aplicatie	V	V	N/A	N/A	N/A	Inapoi la pagina de logare
TC 5	Sistemul nu este disponibil	V	V	N/A	N/A	N/A	Mesaj de eroare
TC 6	Inregistrare oprita	V	V	N/A	N/A	I	Mesaj de eroare
TC 7	Contul deja exista - inregistrare duplicata	V	V	I	N/A	N/A	Mesaj de eroare

Figură 5-44 Exemplu matrice pentru cazuri de testare – tabel general

Test Case ID	Scenariu	Username	Certificat	Nu este deja înregistrat	Informatii specifice	Inregistrarea se poate realiza	Rezultate așteptate
TC 1	Inregistrare realizata cu succes	test@yah.com	Certificat CertSign	Da	Da	Da	Mesaj de confirmare
TC 2	Utilizatorul nu a fost gasit	unknown@yah.com	N/A	N/A	N/A	N/A	Mesaj de eroare: Inapoi la pagina de logare
TC 3	Informatii incorecte	test@yah.com	Certificat CertSign	N/A	Invalide	N/A	Mesaj de eroare: Reintroducerea datelor
TC 4	Utilizatorul iese din aplicatie	test@yah.com	Certificat CertSign	N/A	N/A	N/A	Inapoi la pagina de logare
TC 5	Sistemul nu este disponibil	test@yah.com	Certificat CertSign	N/A	N/A	N/A	Mesaj de eroare
TC 6	Inregistrare oprita	test@yah.com	Certificat CertSign	N/A	N/A	Full	Mesaj de eroare
TC 7	Contul deja exista - inregistrare duplicata	test@yah.com	Certificat CertSign	Nu	N/A	N/A	Mesaj de eroare

Figură 5-45 Exemplu matrice pentru cazuri de test – cu date

5.3.4. Exemple de testare

Cazuri de teste executate doar de interfața pusă la dispoziție utilizatorului – teste sistem (teste functionale) (exemple):

Caz 1

Pas 1. Descrierea pasului: Logarea la o aplicatie test cu un cont valid.

Rezultate așteptate: Afișarea paginii de început.

Pas 2: Descrierea pasului : Click pe link-ul de “Logout” de pe pagina Home.

Rezultate așteptate: Pagina de logare este afișată.

Caz 2

Pas 1. Descrierea pasului : Încărcarea pe serverul a fișierelor dorite.

Rezultate așteptate: Afișarea fișierelor încărcate dacă acestea au formatul impus.

Pas 2. Descrierea pasului: Detalierea fișierelor încărcate pe server

Rezultate așteptate: Afișarea unui mesaj că detaliile despre fișier au fost salvate cu succes.

Pas 3. Descrierea pasului: Verificare document încărcat în pagina Documente.

Rezultate așteptate: Documentul și detaliile să fie afișate corespunzător în rubric “Documente nesemnate”.

.

6. Concluzii

Lucrarea de față a prezentat noțiunile teoretice despre baze de date, rolul acestora în aplicațiile Web, diferite tehnologii Web de acces la baze de date și comparații între acestea, conceptul de cloud computing, platforma Windows Azure, conceptul de Database-as-a-Service(DbaaS).

S-au detaliat noțiunile teoretice despre baze de date, cum s-a ajuns la utilizarea unei astfel de tehnologii, bazele de date relaționale și avantajele utilizării lor.

S-au prezentat caracteristicile cele mai importante ale tehnologiilor web actual de acces la baze de date. S-au realizat comparații între aceste tehnologii, subliniind avantajele și/sau dezavantajele fiecăreia descriind tipurile de aplicații pentru care se pretează fiecare în parte.

S-a prezentat conceptul de cloud computing, principalele caracteristici. Avantajele utilizării unei astfel de tehnologii și s-a pus accentul în prezentare pe conceptul de „Database-as-a-Service”. S-au prezentat principalele avantaje ale utilizării unor astfel de tehnologii (ASP, PHP, Python, Ruby)., diferențele față de tehnologiile actuale și care este rolul lor în aplicația implementată. Ca și caz particular al cloud computing-ului, s-a prezentat platforma cloud Microsoft Windows Azure. S-au detaliat diferite caracteristici, servicii și avantaje ale utilizării unei astfel de platforme și rolul lor în aplicația implementată.

S-a prezentat aplicația implementată și motivarea tehnologiilor folosite. S-a prezentat conceptul de Model-View-Controller și ASP.NET MVC 4. S-a detaliat fiecare framework utilizat și de ce a fost folosit.

S-au utilizat diagramele UML pentru reprezentarea arhitecturii aplicației, a modulelor componente și a modului în care utilizatorii acestei aplicații interacționează cu modulele componente. Prezentarea fiecărei diagrame a fost însoțită de o scurtă prezentare teoretică despre diagramă și rolul ei.

S-a realizat o matrice de testare, pentru testarea aplicației și concluziile în urma obținerii rezultatelor.

Se dorește ca pe viitor aplicația să poate avea opțiunea de semnare a documentelor din browser (fără să oblige clientul să folosească anumite tehnologii), iar autentificarea să se realizeze cu un token hardware.

7. *Bibliografie*

1. „Database as Backend to Web Tools”
2. (Feng)Thanaa Ghanem, Walid Aref „Web Databases: Deepen the Web”
3. (https://en.wikipedia.org/wiki/Relational_database)
4. (<http://searchsqlserver.techtarget.com/definition/relational-database>)
5. (http://www.ehow.com/list_5985480_benefits-relational-database_.html)
6. (<http://www.buzzle.com/articles/advantages-of-relational-databases.html>)
7. ([http://msdn.microsoft.com/en-us/library/ms524929\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms524929(v=vs.90).aspx))
8. (<http://www.techyshell.com/internet/asp-its-advantages-and-disadvantages/>)
9. (<https://wiki.php.net/>)
- 10.(http://en.wikibooks.org/wiki/PHP_Programming)
- 11.(<https://en.wikipedia.org/wiki/PHP>)
- 12.(http://en.wikibooks.org/wiki/Ruby_Programming)
- 13.(<http://wiki.python.org/moin/>)
- 14.(<http://wiki.python.org/moin/DatabaseProgramming/>)
- 15.(<http://www.aspvsp.php.com/>)
- 16.(<http://wiki.python.org/moin/PythonVsPhp>)
- 17.(http://www.wikivs.com/wiki/Python_vs_Ruby)
- 18.(http://en.wikipedia.org/wiki/Cloud_database)
- 19.(Carlo Curino) „Relational Cloud: A Database-as-a-Service for the Cloud”
- 20.(Krishnan, 2010) „Programming Windows Azure”
- 21.(Brunetti, 2011) „Windows Azure Step by Step”
- 22.(http://en.wikipedia.org/wiki/Platform_as_a_service)
- 23.(David Chapell ,2008) „Introducing Windows Azure”
- 24.(http://www.legi-internet.ro/notar_electronic.htm)
- 25.(http://en.wikipedia.org/wiki/Public_key_certificate)

- 26. (<http://www.techrepublic.com/blog/10things/10-reasons-to-use-azure-for-your-cloud-apps/1282>)
- 27. (Freeman, 2013) “Pro ASP.NET MVC 4”
- 28. (http://www.w3schools.com/aspnet/mvc_intro.asp)
- 29. Radu Mihail Obadă, „Introducere în dezvoltarea aplicațiilor”
- 30. Florin Ostafi „Ingineria sistemelor de programe”
- 31. (<http://blog.liga-group.ro/liga-software/testare-software/etapele-principale-ale-unei-scheme-de-testare>)
- 32. Adriana Gheorghies, Ovidiu Gheorghies, “Curs 11”