Arhitectura aplicatiei Web : Password Manager(Passer)

1.Introducere:

Password Manager este o aplicatie ce va permite utilizatorului gestionarea cu usurinta a parolelor folosite in cadrul unor aplicatii/siteuri Web.

In cadrul proiectului se va construi o aplicatie ce va implementa un sistem client-server ce va fi folosit pentru a afisa sub forma unui site web informatii despre conturile asociate unui utilizator.

2.Tehnologii utilizate:

Front-end:

HTML impreuna cu CSS vor fi folosite pentru partea de front-end a aplicatiei. Interactiunea dintre front-end si back-end se va face prin preluarea si afisarea de informatii precum si campuri de introdus informatii si butoane pentru procesarea lor. Aplicatia va avea partea de front-end decuplata de cea de back-end.

Back-end:

Javascript va fi folosit ca si principalul limbaj de programare pentru partea de back-end. Se va folosi design pattern de tip MVC.

Node.JS va fi folosit pentru crearea de micro servicii.

Folosim JSON/XML pentru importarea de informatii de pe alte sisteme informatice.

MySQL va fi folosit pentru partea de baze de date. Va stoca informatiile de autentificare a userilor precum si conturile lor intr-un mod sigur. Parolele vor fi criptate cu ajutorul criptosistemului RSA sau Vigenere.

Pachetul Node-Mandrill din Node.JS va fi folosit pentru sistemul de trimitere e-mail cu scopul resetarii parolei unui user sau al inregistrarii lui.

3.Folosire REST(REpresentational State Transfer):

REST este o cale simpla de a transmite si de a primi date intre client si server si nu are multe standarde definite. REST foloseste HTTP GET, POST si PUT pentru a realiza transferul de informatii. Datele pot fi trimise prin JSON, XML sau plain text. Pentru acest proiect entitatile vor fi in general de tipul JSON.

4. Structura aplicatiei:

Avantaje monolith:

- echipa noastra este mica, iar organizarea nu este o problema atat de mare
- teoretic este mai usor sa pornesti de la o aplicatie de la acest gen si mai apoi sa o modularizezi, decat sa o faci din prima cu microservicii(mai ales cand nu ai experienta cu ele)
- pentru ca nu se merita sa faci microservicii fara frameworkuri, avand in vedere ca este destul de dificil de implementat si nu sunt prea multe avantaje la o aplicatie mica

Avantaje microservicii:

- cod modularizat, usor de inteles si cu sisteme independente
- securitate mai mare

- daca apare o eroare, aceasta nu risca sa se propage in tot programul

O aplicatie ce se bazeaza pe modelul MVC are 3 nivele:
-view: cu care este controlat accesul la aplicatie, el va trimite cererile userului mai departe(ce poate fi reprezentat in aplicatia noastra de continutul static, ce va avea diverse pagini(scris in HTML, CSS si JS), fiind interfata grafica.)
-controller: cu care se efectueaza operatiile logice si de utilizare de informatii; el accepta intrarea de la view si o proceseaza, urmand sa o trimita mai departe daca este valida.
-model: care se ocupa de comportamentul programului independent de user si are acces direct la baza de date.

In aplicatia noastra nu sunt foarte bine definite diferentele dintre aceste 3 nivele pentru a scrie mai usor codul.

Servicii:
Management cont (login/register):

Permite utilizatorilor noi sa isi creeze un cont, iar celor existenti sa si-l acceseze sau sa isi schimbe parola.

Validarea pentru autentificare se va face astfel: se iau datele introduse de utilizator, se cripteaza cu ajutorul unui sistem de securitate si se verifica daca respectiva entitate criptata este in baza de date. Daca este, se va intoarce un id_utilizator ce va putea fi folosit pentru a accesa datele de pe conturile personale ale utilizatorului, daca nu, nu se va putea accesa nimic.

Inregistrarea in aplicatie se va face astfel: se va completa un formular ce va fi trimis catre un serviciu ce se ocupa cu managementul conturilor. Dupa validarea datelor(sa nu fi aparut inca o data in bd acelasi mail, acelasi nume de user, ..), acesta va trimite utilizatorului un mail prin protocolul SMTP de confirmare cu un cod random valabil pentru o perioada scurta de timp,.Utilizatorul va putea folosi codul pentru a activa noul cont. Daca acesta activeaza contul, va fi transmisa o noua inregistrare de user ce va fi introdusa in baza de date(criptata). Daca nu se activeaza contul in 24h (sa zicem), se va dezactiva respectivul cod de inregistrare.

Resetarea parolei se va face in maniera asemanatoare inregistrarii, doar ca asupra bazei de date se va face o operatie de update si nu de insert.

Exportator de date:
Ce date primeste: id_utilizator ce doreste sa faca exportul, si formatul in care doreste sa il faca.
Ce date ofera: fisierul corespunzator utilizatorului solicitant ce va contine lista datelor de pe site, in formatul droit.

Ce face: verifica inca o data daca acel utilizator e conectat si e in baza de date, apoi ii selecteaza datele personale.
Cand face: cand primeste aceasta cerere de la un utilizator conectat.
Cum face: primeste cererea, cripteaza utilizitorul si cauta entitatea corespunzatoare lui in BD, extrage datele sale criptate, le decripteaza, le pune intr-un fisier specific, le returneaza.

Importator de date:
Ce date primeste: id_utilizator ale carui date sunt si un fisier cu date, in diverse formate (txt, xml, ..)
Ce date ofera: o lista de entitati(ce reprezinta conturi cu datele asociate) ce vor putea fi puse in baza de date, intr-o categorie predefinita.
Cum: se ia fisierul si se parseaza. In functie de tipul acestuia(extensie) delimitatorul poate sa difere. Daca sunt informatii culese de pe alt site asemanator ce a exportat datele, parsarea se va face mai usor.

Generator de parole:
Ce date primeste: id_utilizator
Ce date ofera:o parola random puternica
Cum: un algoritm nedeterminist va face o selectie aleatorie de litere, cifre si simboluri de o lungime de 10-15 caractere.

Manager de conturi si categorii:
Ce date primeste: id_utilizator, operatia pe care doreste sa o faca si argumentele
Ce date ofera: face operatii CRUD in baza de date
Cum: primeste cererea, cripteaza utilizitorul si datele, iar apoi face operatii CRUD in BD conform cererii effectuate.

Sistem de securitate:
Componenta de criptare:
Ce date primeste: un plaintext si un id_utilizator
Ca date ofera: textul criptat
Cum: Construieste o cheie in functie de id_utilizator printr-un algoritm pseudoprobabilist. Face criptarea folosind cheia obtinuta. Putem alege o criptare de tip RSA sau Vigenere. Returneaza criptotextul.

Componenta de decriptare:
Ce date primeste: un criptotext si un id_utilizator
Ca date ofera: textul decriptat
Cum: Face decriptarea folosind cheia generata pseudoprobabilit corespunzatoare acelui utilizator. Returneaza textul in clar.

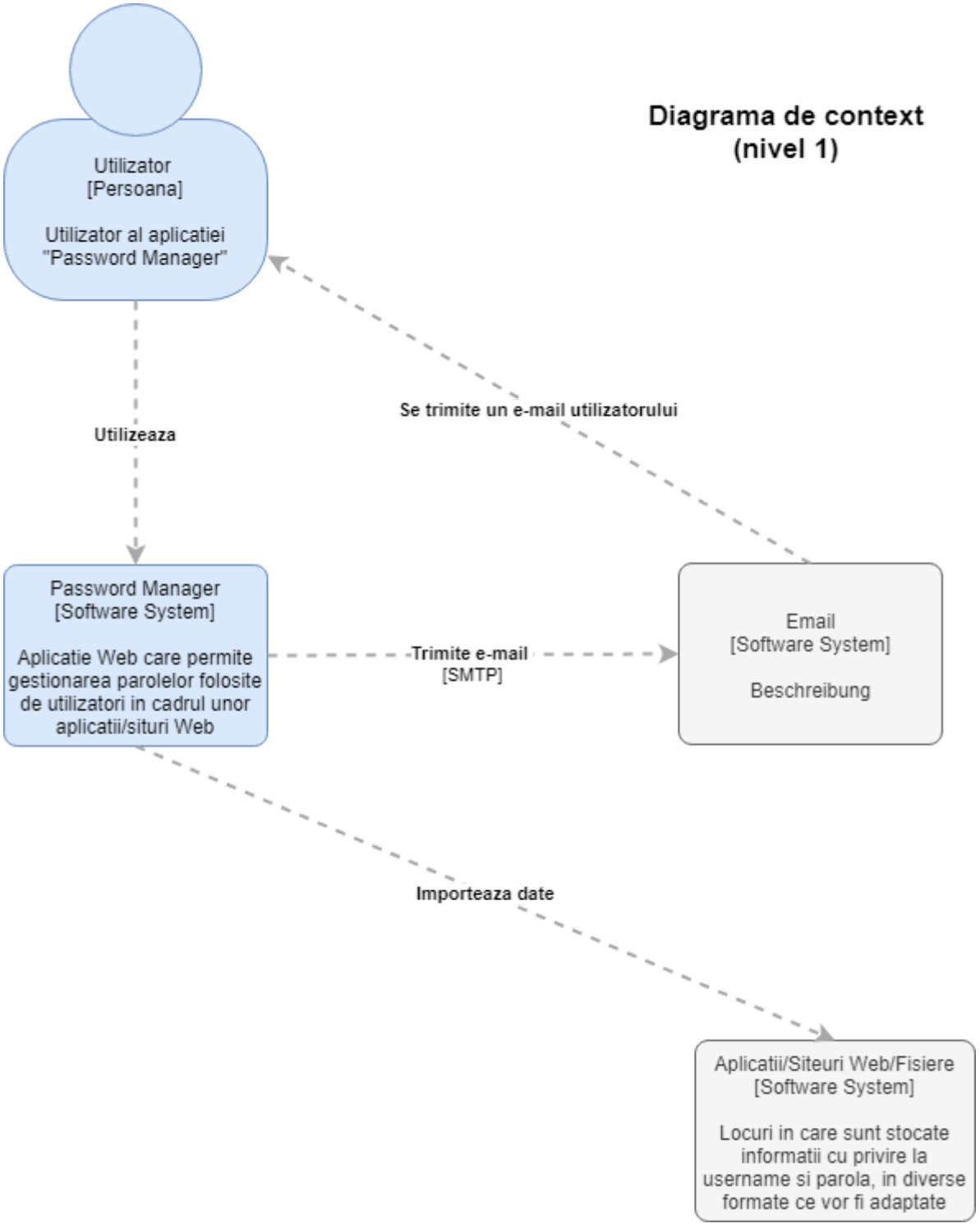


Diagrama de context
(nivel 1)
Utilizator
[Persoana]
Utilizator al aplicatiei
"Password Manager"
Utilizeaza
Se trimite un e-mail utilizatorului
Password Manager
[Software System]
Aplicatie Web care permite gestionarea parolelor folosite de utilizatori in cadrul unor aplicatii/situri Web
Trimite e-mail
[SMTP]
Email
[Software System]
Beschreibung
Importeaza date
Aplicatii/Siteuri Web/Fisiere
[Software System]
Locuri in care sunt stocate informatii cu privire la username si parola, in diverse formate ce vor fi adaptate

Container Diagram (nivelul 2)

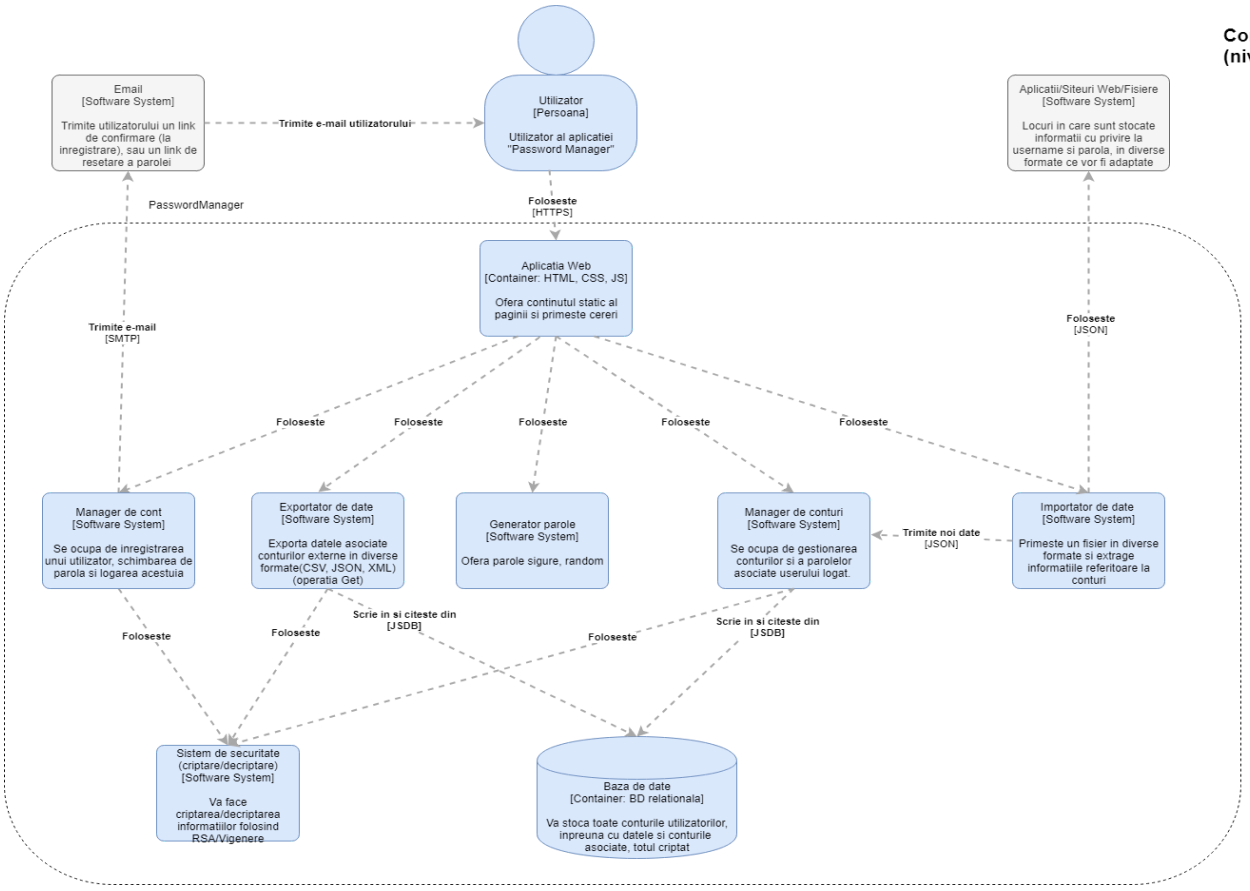


Diagrama componenta (nivelul 3)

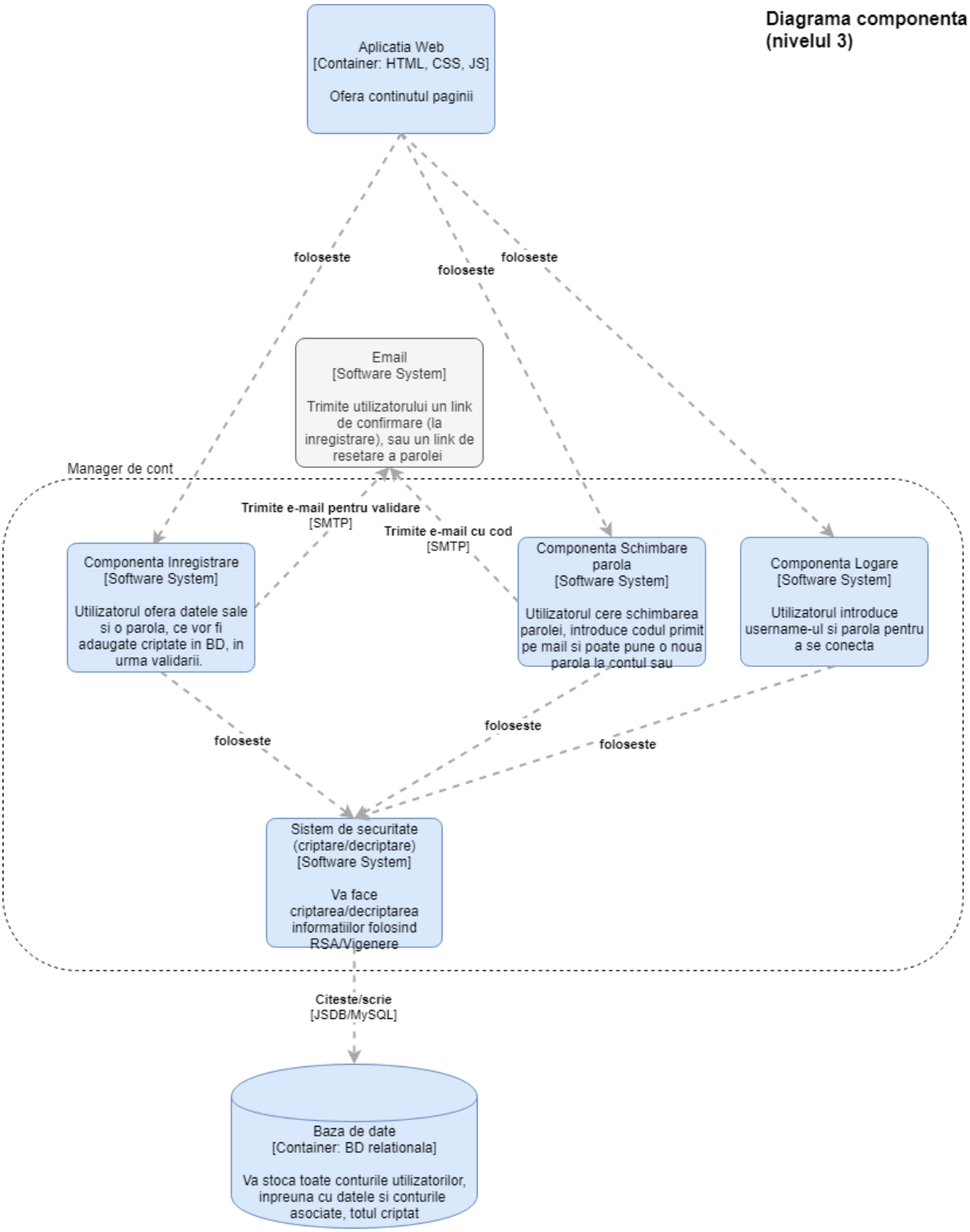


Diagrama componenta (nivelul 3)

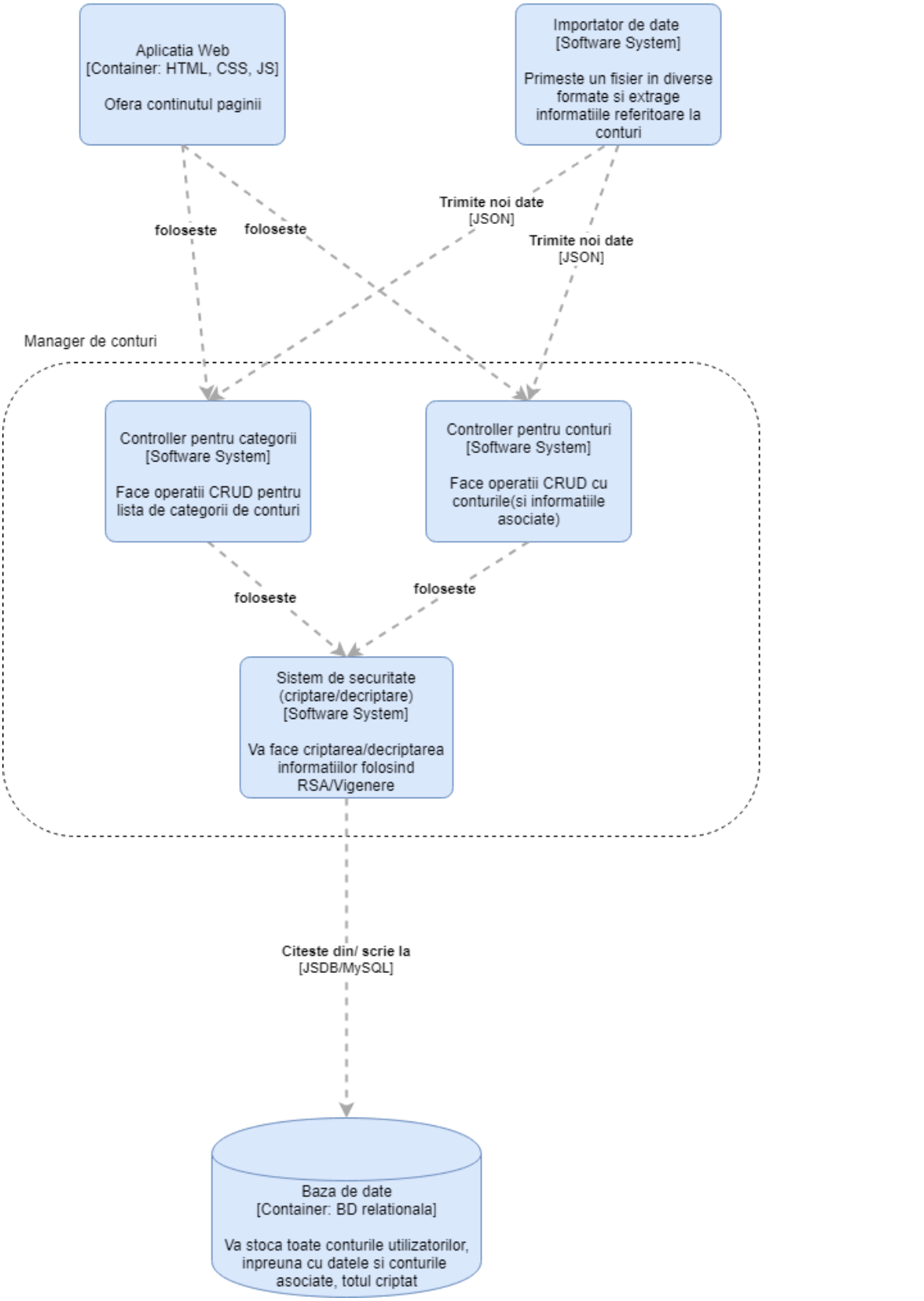
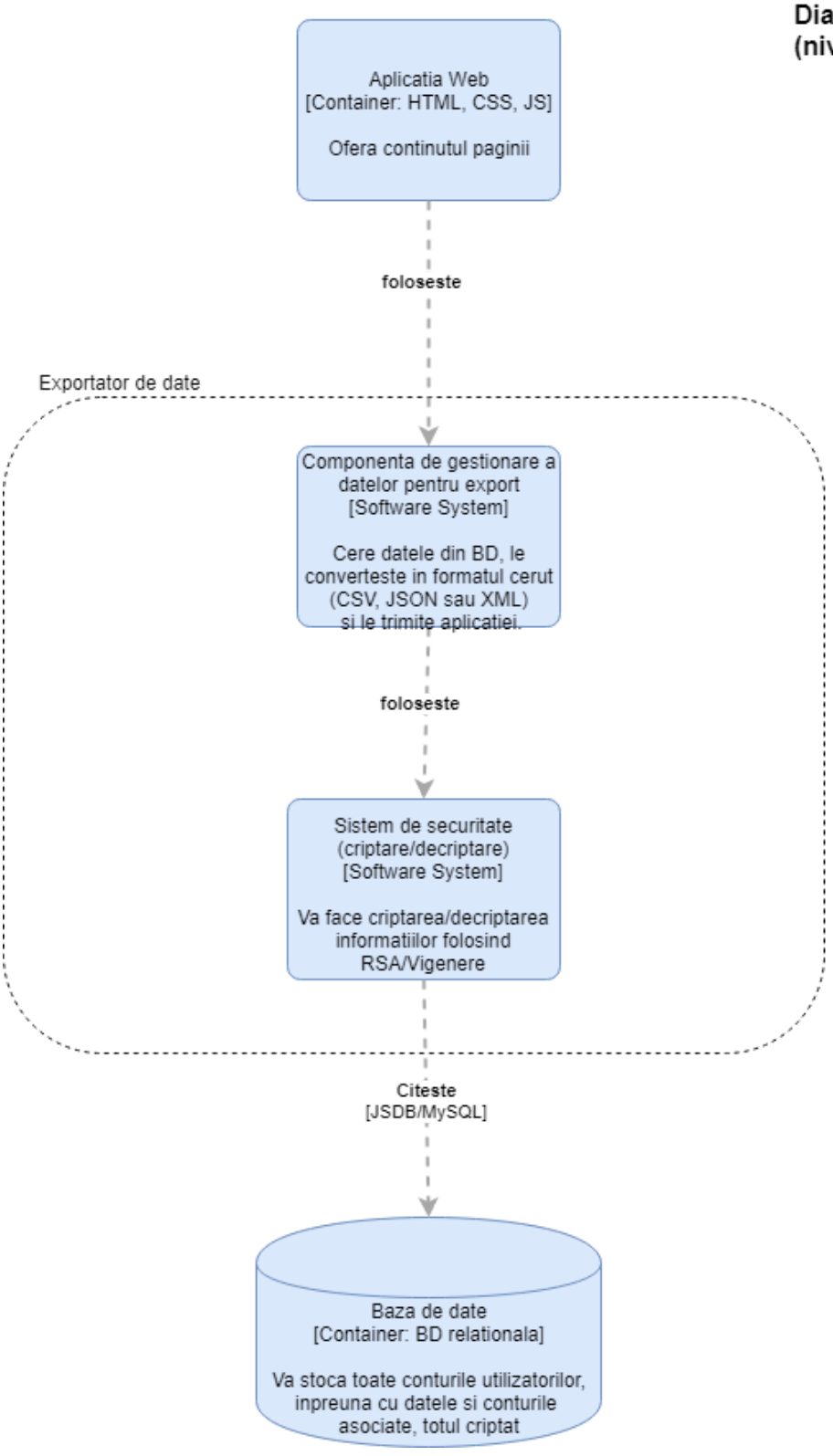


Diagrama componenta (nivelul 3)



5. Tipuri de utilizatori:

Utilizator neautentificat:
Va putea accesa doar fatada publica a aplicatiei. Mai exact: pagina de start si cea de inregistrare.

Utilizator autentificat:
Va avea un cont la care va avea acces doar el.Prin intermediul lui va putea accesa orice pagina a aplicatiei, dar doar la datele personale.

Serverul:
Se ocupa de furnizarea de servicii utilizatorului.

Diagrama use case pentru pagina de login:

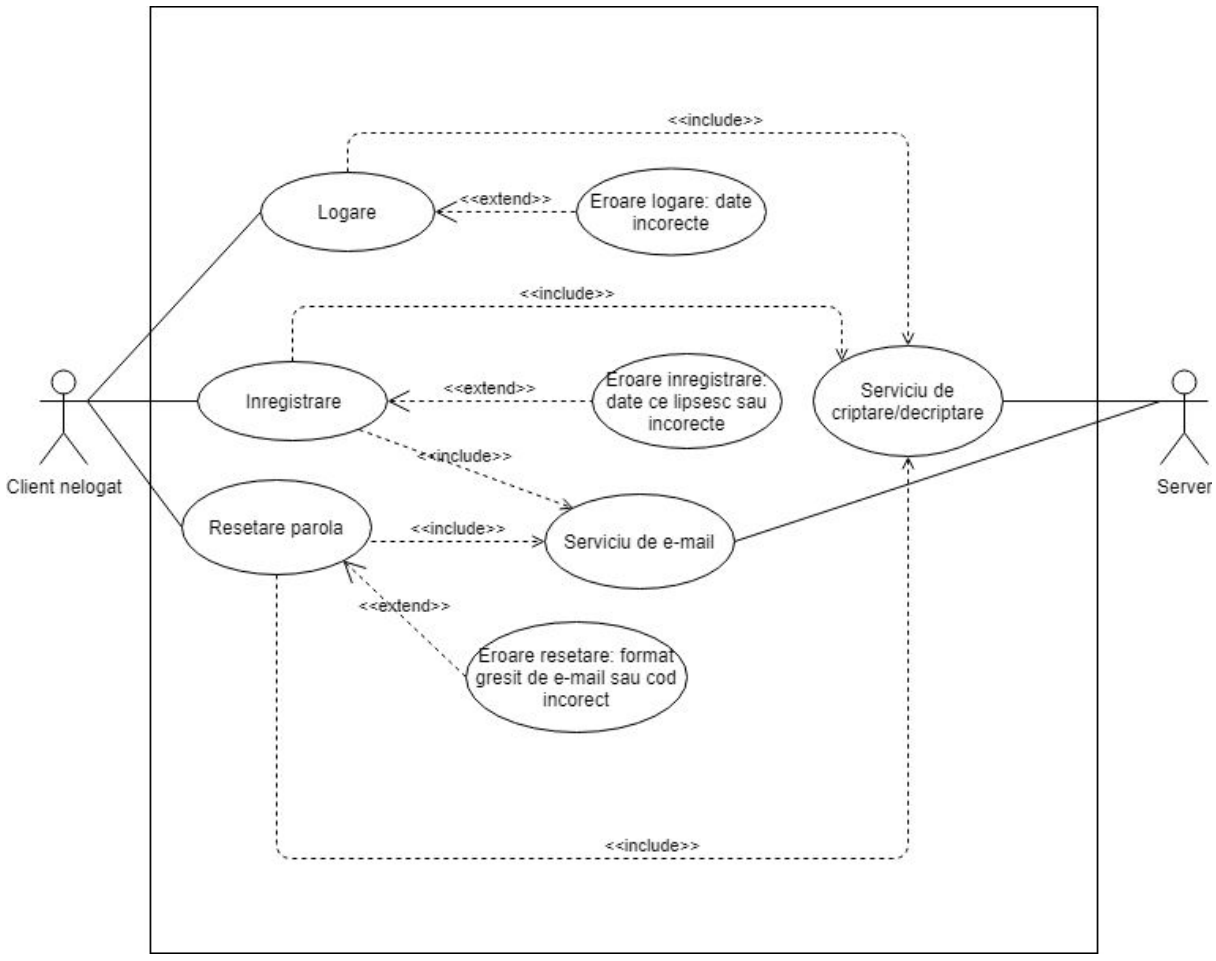
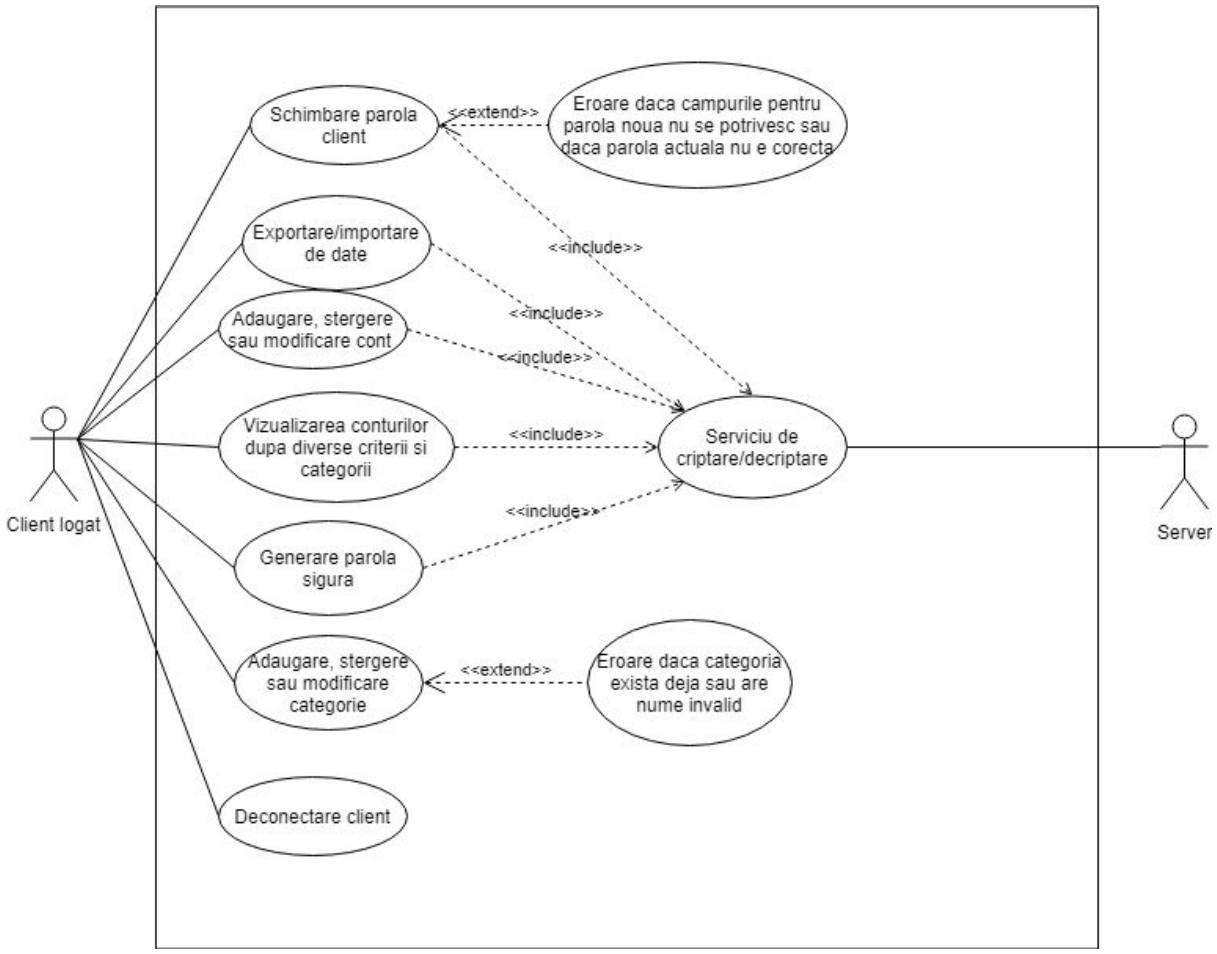


Diagrama use case pentru pagina principala:



5.5. Principalele scenarii de utilizare - detaliere:

Clientul adauga o noua categorie:
Descriere: Se adauga o noua categorie de conturi
Pasi:
-Userul logat va accesa pagina principala(cu meniul)
-Va apasa butonul “Add new category”.
-Se va deschide un field nou ce va cere un sir de caractere.
-Sirul va fi validat de server si introdus in baza de date.
Exceptii: Utilizatorul introduce un sir de caractere ce mai apare inca o data in baza de date -> se va afisa un mesaj de genul “already exists”
Utilizatorul introduce un sir vid -> se va afisa mesajul “invalid category name”.

Clientul adauga un nou cont:
Descriere: Se adauga un nou cont, impreuna cu detaliile asociate.
Pasi:
-Userul logat va accesa pagina principala(cu meniul)
-Va apasa butonul “Add new account”, din cadrul unei categorii(se poate alege si categoria “Others” in care vor fi puse conturile fara o categorie aume).
-Se va deschide o pagina de adaugare a contului ce va cere anumite informatii despre contul ce se doreste a fi adaugat..
-Utilizatorul va introduce de la tastatura date.
-Serverul va valida si le va introduce in baza de date.
Exceptii:
-Utilizatorul introduce un user pentru un cont ce mai apare inca o data in baza de date -> se va afisa un mesaj de genul “already exists”
Utilizatorul introduce un sir vid in campuri obligatorii -> se va afisa mesajul “invalid category name”.

Clientul se logheaza:
Descriere: se cere userul si parola. Daca se afla in baza de date, se face asocierea intre sesiunea de lucru si contul clientului.
Exceptii: daca userul sau parola este gresita, se va afisa un mesaj de eroare si se va resolicita autentificarea.

Clientul se inregistreaza:
Descriere: se vor cere informatiile referitoare la cont si daca sunt valide se va realiza crearea contului
Exceptii: username-ul este deja in uz, caz in care se va cere alegerea altui username; e-mail-ul este deja in uz, caz in care se va cere alegerea altui e-mail; parola nu se potriveste cu reinsararea ei, caz in care se va atentiona clientul; exista campuri care lipsesc, caz in care se va atentiona clientul

Clientul isi reseteaza parola:
Descriere: clientul isi poate reseta parola cu folosirea e-mail-ului.
Pasi:
-Clientul isi introduce e-mail-ul
-Clientul primeste in e-mail codul de resetare
-Clientul foloseste codul de resetare pentru a schimba parola cu una noua
Exceptii: daca codul este gresit se va afisa un mesaj de eroare; parola nu se potriveste cu reinsararea ei, caz in care se va atentiona clientul

Clientul cere vizualizarea conturilor dupa frecventa utilizarii:
Pasi:
- Utilizatorul va apasa butonul “Accounts by use frequency” pagina principala
- Se vor afisa in coloana a treia toate site-urile in functie de frecventa utilizarii.

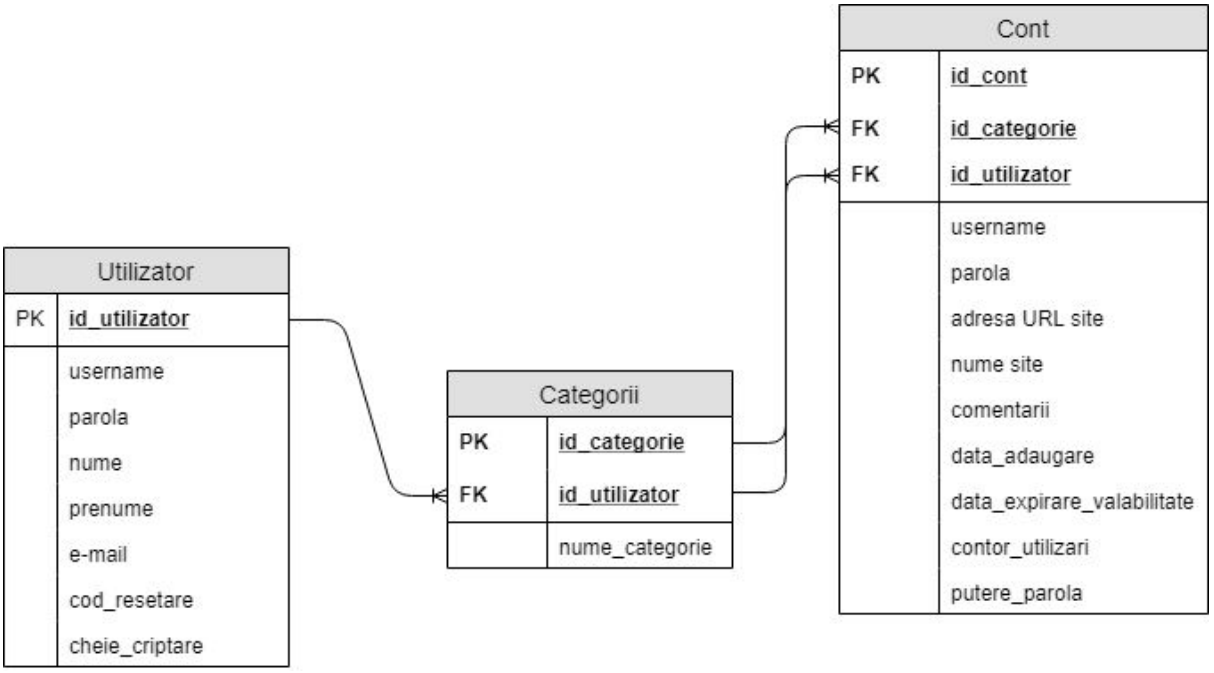
Clientul cere vizualizarea conturilor dupa taria parolei:
Pasi:
- Utilizatorul va apasa butonul “Accounts by use password strength” pagina principala
- Se vor afisa in coloana a treia toate site-urile in functie de frecventa utilizarii.

Clientul genereaza o parola noua:
Pasi:
-Utilizatorul selecteaza din bara de navigare pagina de genrerare de parole.
-Apasa butonul de generare.
-Parola este generate si afisata sub forma de buline/stelute
-Utilizatorul copiaza in clipboard parola.

Clientul importa date:
Pasi:
-Utilizatorul logat selecteaza o sursa externa (url sau incarca un fisier)
-Serverul extrage datele si le parseaza pentru a le integra in baza de date.
-Serverul afiseaza datele rezultate.
-Utilizatorul va confirma corectitudinea si integrarea acestora in baza de date.
Exceptii:
-Sursa externa este inaccesibila/ nu sunt drepturi de citire -> se va afisa un mesaj corespunzator
-Datele nu pot fi parsate de server sau sunt insuficiente cele extrase pentru a fi relevante- > se va afisa un mesaj corespunzator

Clientul exporta date:
Descriere: clientul isi poate exporta datele in formatul XML, JSON sau CSV.
Pasi:
-Utilizatorul alege unul dintre cele trei formate
-Utilizatorul va primi un fisier cu datele sale in formatul ales

6. Baze de date:



Primul tabel “Utilizator” va contine id-ul utilizatorului, username, parola, nume, prenume, e-mail si cod resetare. Username-ul si parola vor fi folosite la momentul logarii cu scopul autentificarii. Codul de resetare va fi folosit atunci cand se doreste resetarea parolei. Odata ce a fost resetata parola, codul se va schimba. Cheia de criptare va fi folosita pentru criptarea parolelor utilizatorului. Cheia va fi diferita de la un utilizator la altul. Toate datele vor fi preluate la momentul inregistrarii, cu exceptia codului de resetare si a cheii de criptare care vor fi generate intr-un mod aleatoriu. Daca se doreste resetarea parolei, se va folosi e-mail-ul pentru a trimite codul de resetare.

Cu ajutorul id-ului se vor cauta si obtine categoriile ce apartin utilizatorului din tabelul “Categorii”. Campul nume_categorie va fi folosit pentru a reda numele categoriilor odata ce sunt preluate. Datele din acest tabel vor fi obtinute din crearea de categorii de catre utilizatori si vor fi modificate atunci cand categoriile sunt modificate sau sterse.

In al treilea tabel, “Cont”, vor fi stocate conturile utilizatorului cu toate datele lor. La momentul redarii conturilor dintr-o categorie, datele ce apartin acelui utilizator si acelei categorii vor fi preluate si redate dupa decriptare. Campul “data_expirare_valabilitate” va contine data expirarii parolei. Dupa ce a trecut data respectiva, utilizatorul va fi anuntat cu scopul schimbarii aceseteia. Campul “contor_utilizari” va fi incrementat de fiecare data cand este folosita parola respectiva (copied to clipboard) cu scopul sortarii dupa frecventa utilizarii. Campul “putere_parola” va fi calculat folosind un algoritm aplicat pe parola si va fi folosit cu scopul sortarii dupa “taria” parolei. Datele din acest tabel vor fi obtinute din crearea de conturi de catre utilizatori si vor fi modificate atunci cand conturile sunt modificate sau sterse.

7. Entitati

Entitate cont utilizator - este folosita pentru inregistrare, informatiile din entitate vor fi adaugate in baza de date. Va fi formata din id_utilizator, username, parola, nume, prenume, e-mail.

Entitate nume-parola - este folosita pentru logare, informatiile din enitate vor fi comparate cu tabelul utilizatorilor cu scopul autentificarii. Va fi formata din username si parola.

Entitate resetare parola - este folosita pentru resetarea parolei unui utilizator, informatiile din entitate vor fi comparate cu datele din tabelul de utilizator. Va fi formata din codul de resetare, username-ul si parola noua.

Entitate categorie - este folosita pentru crearea si schimbarea unei categorii, informatiile din entitate vor fi adaugate sau comparate cu datele din tabelul de categorii. Va fi formata din id categorie, id utilizator si nume categorie.

Entitate cont - este folosita pentru crearea si schimbarea unui cont. Informatiile vor fi adaugate sau comparate cu datele din tabelul de cont. Va fi formata din cele trei id-uri si informatiile legate de cont.

8. Sincronizare date

Userul va face manual update cu datele pe care le-a modificat, accesand butonul de modificare din meniul principal.

9. Conditii ca un user sa nu poata modifica datele altui user

Un user nu va putea modifica datele altui user astfel incat controller-ul va asigura faptul ca user-ul isi va putea modifica numai conturile ce ii apartin lui. Un user nelogat nu va avea acces la functionalitatile paginii principale.

Referinte:

https://stackoverflow.com/questions/7381150/how-to-send-an-email-from-javascript
https://www.awwwards.com/build-a-simple-javascript-app-the-mvc-way.html
https://medium.freecodecamp.org/monolith-vs-microservices-which-architecture-is-right-for-your-team-bb840319d531

MVC:
In a conversation form:

View: "Hey, controller, the user just told me he wants item 4 deleted."

Controller: "Hmm, having checked his credentials, he is allowed to do that... Hey, model, I want you to get item 4 and do whatever you do to delete it."

Model: "Item 4... got it. It's deleted. Back to you, Controller."

Controller: "Here, I'll collect the new set of data. Back to you, view."

View: "Cool, I'll show the new set to the user now."