SeAN

# Cerinta proiectului

Se cere realizarea unei aplicatii Web capabila sa notifice o persoana sau grup de utilizatori autentificati asupra aparitiei unei alergii sezoniere: la polen, la expunere directa la soare sau la praf etc. – eventual, in conjunctie cu anumite fructe sau compusi alimentari. Notificarile vor fi semnalate – inclusiv cu o anumita prioritate/frecventa – atat in cadrul aplicatiei, cat si prin posta electronica. Se vor pune la dispozitie, de asemenea, sfaturi referitoare la preintampinarea unei/unor alergii, plus tratamentul si/sau medicatia. Utilizatorii vor putea adauga propriile alergii, cu posibilitatea de a fi partajate la nivel de grup, inclusiv la nivelul unei retele sociale. Se vor genera rapoarte in formatele HTML si PDF referitoare la statistici de interes.

# Tehnologii uzate

### Frontend

Pentru partea de frontend s-au folosit :

* HTML5
* CSS3
* Javascript
  + AJAX pentru a asigura comunicarea dintre frontend si backend

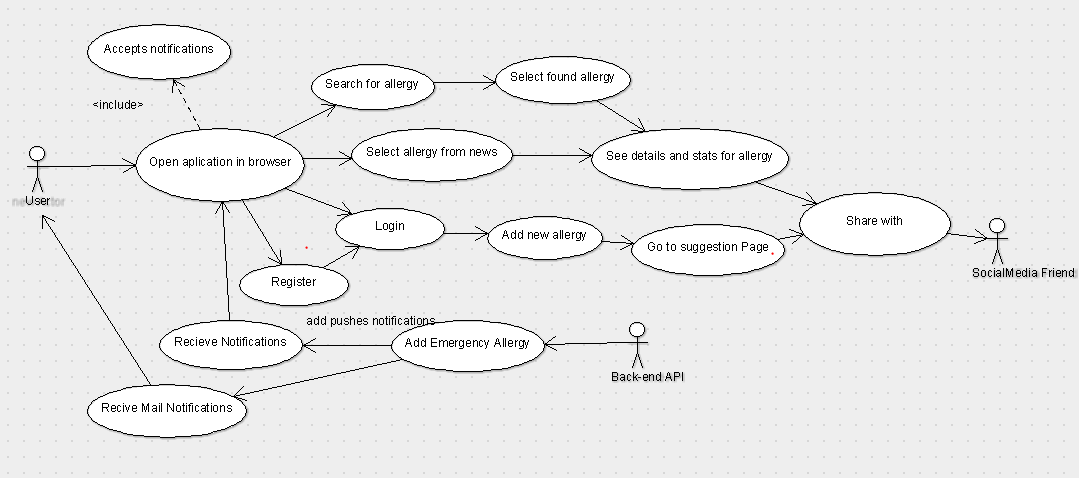
### Backend

Pentru partea de backend s-au folosit:

* Python 3.7, tot backend-ul este realizat in Python, serverul principal fiind unul CGI, din motive de usurinta de utilizare
* JSON pentru trimiterea usoara a datelor intre frontend si backend
* MySQL 8.0.15

# Client steps

### Diagrama use-case



### Principalii actori

* Utilizatorul neautentificat: un utilizator ce viziteaza pentru prima data site-ul si nu este logat sau inregistrat in baza de date dar care are la dispozitie urmatoarele functii:
  + Sa vizualizeze newsfeed-ul cu ultimele alergii introduce
  + Sa vizualizeze detalii despre o anumita alergie
* Utilizator autentificat: un utilizator ce este inregistrat in baza de date si care s-a logat la accesarea site-ului si care are la dispozitie urmatoarele functionalitati, pe langa cele mentionate anterior:
  + Sa adauge o alergie in sistem
  + Sa aleaga daca primeste sau nu notificari atat pe aplicatie cat si pe mail legat de anumite tipuri de alergii introduse in system

### Cazuri de utilizare

* + 1. Userul deschide aplicatia, selecteaza din newsfeed o alergie dand click pe ea si este redirectionat pe pagina cu detalii despre alergie.
    2. Userul deschide aplicatia si cauta prin inputul de la search o alergie apoi o selecteaza pe cea dorita dand click pe ea si este redirectionat pe pagina cu detalii despre alergie.
    3. Userul deschide aplicatia si se inregistreaza, apoi se logheaza in aplicatie si selecteaza din newsfeed o alergie dand click pe ea si este redirectionat pe pagina cu detalii despre alergie.
    4. Userul deschide aplicatia, se logheaza in aplicatie si selecteaza din newsfeed o alergie dand click pe ea si este redirectionat pe pagina cu detalii despre alergie.
    5. Userul se se inregistreaza, apoi se logheaza, si cauta prin inputul de la search o alergie apoi o selecteaza pe cea dorita dand click pe ea si este redirectionat pe pagina cu detalii despre alergie.
    6. Userul se logheaza si cauta prin inputul de la search o alergie apoi o selecteaza pe cea dorita dand click pe ea si este redirectionat pe pagina cu detalii despre alergie.
    7. Userul se logheaza, selecteaza din bara de meniu ruta catre “Add Allergy” si este redirectionat catre pagina de adaugare a unei alergii.
    8. Cazul de la punctual 7 la care se mai adauga inregistrarea unei noi alergii care va fi stocata in baza de date.
    9. Userul se afla pe pagina alergiei si o partajeaza cu prietenii de pe retelele sociale.
    10. Userul intra in aplicatie si selecteaza din ce categorie de alergii doreste sa primeasca notificari.
    11. Userul primeste pe mail notificarea aparitiei unei alergii sezoniere.
    12. Userul deschide aplicatia, selecteaza din newsfeed o alergie dand click pe ea si este redirectionat pe pagina cu detalii despre alergie. Da click pe HTMLReport si descarca un raport in format HTML.
    13. Userul deschide aplicatia, selecteaza din newsfeed o alergie dand click pe ea si este redirectionat pe pagina cu detalii despre alergie. Da click pe PDFstatistics si descarca un raport de statistici in format PDF.

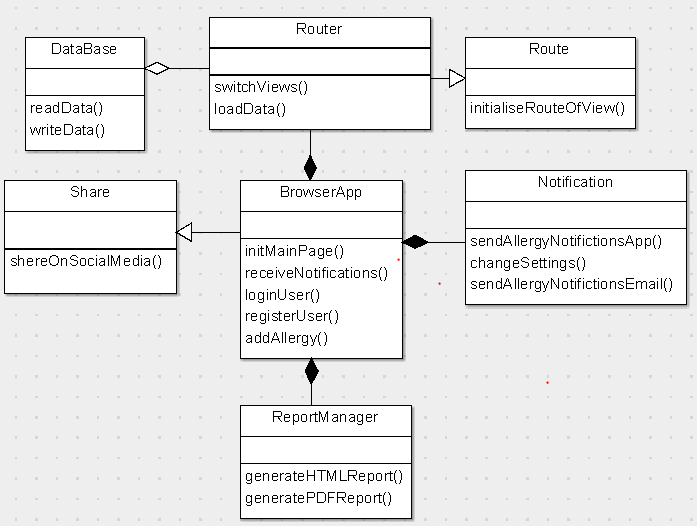
# Arhitectura

### Descriere

Arhitectura este de tip MVC(Model-View-Controller) in care View-ul este reprezentat de paginile html cu scriputile de stil CSS3 aferente si functionalitatea in JS , Model este reprezentat de stocarea datelor folosind baza de date MySQL iar Controller de scripturile backend python. Server-ul principal este de tip CGI-BIN, realizarea schimbului de informatii intre partea de front-end si partea de backend se realizeaza folosind obiecte serializate intr-un JSON ce contine atat informatia necesara realizarii unei anumite operatii la un moment dat cat si de implementarea metodelor necesare, adica : POST, GET, PUT, DELETE, toate fiind abstractizate prin request-uri generice initial iar apoi individualizate pentru a se plia nevoilor noastre la un moment dat.

Vor fi generate rapoarte in format : HTML si PDF. Stocarea informatiilor sensibile cum ar fi parolele utilizatorilor se vor realiza in mod criptat folosind MD5. Notificarea utilizatorului pe email se va face folosind protocolul SMTP din Python cu o adresa de mail dummy pe un domeniu deja existent(ex: gmail, yahoo etc ).

### Digrama arhitecturii



# Bibliografie

<https://teaching.alexcoman.com/laboratoare/web/proiecte>

<https://profs.info.uaic.ro/~busaco/teach/courses/web/>

<https://docs.python.org/3/>

<https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp>

<https://developer.mozilla.org/en-US/>

<https://ro.wikipedia.org/wiki/Model-view-controller>