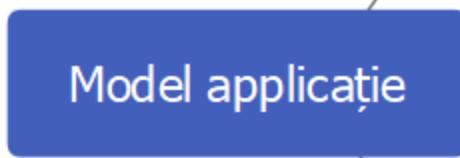

LUCRAREA DE LICENȚĂ

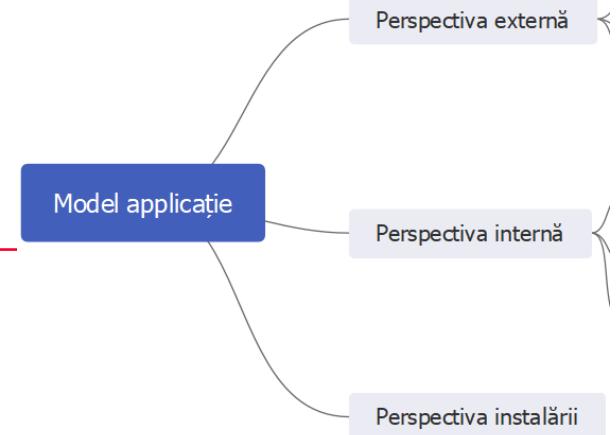
Modelarea aplicației



Model aplicatie

Conf. Dr. Cristina Mîndruță

PERSPECTIVELE asupra APlicației



Perspectiva externă

Reprezentarea aplicației (sistemu*m*ului software) în contextul său.

Perspectiva internă

Reprezentarea modelelor elementelor interne ale sistemului software.

Perspectiva instalării aplicației

Reprezentarea infrastructurii de instalare și a relațiilor dintre componentele aplicației și elementele de infrastructură.

PERSPECTIVELE asupra APLICAȚIEI

Perspectiva externă : Reprezentarea aplicației (sistemuil software) în contextual său.

Perspectiva internă : Reprezentarea modelelor interne ale sistemului software.

Perspectiva instalării aplicației : Reprezentarea infrastructurii de instalare și a relațiilor dintre componentele aplicației și elementele de infrastructură.

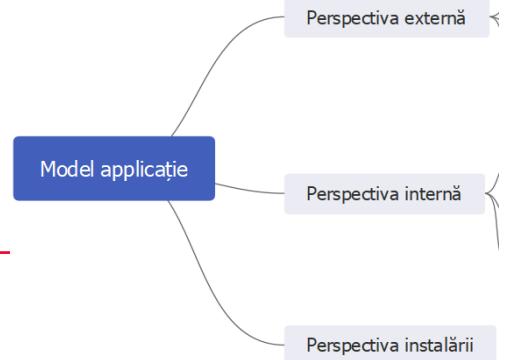
În lucrarea de licență ar trebui să existe modele ale aplicației văzută din fiecare perspectivă.

Fiecare lucrare de licență va conține modelele potrivite cu tipul de aplicație.

Acestea vor fi selectate din setul de modele recomandate în continuare.

Se recomandă alegerea modelelor corespunzătoare și folosirea majorității acestora în activitatea de proiectare a aplicației, înainte de dezvoltarea codului.

PERSPECTIVA EXTERNĂ

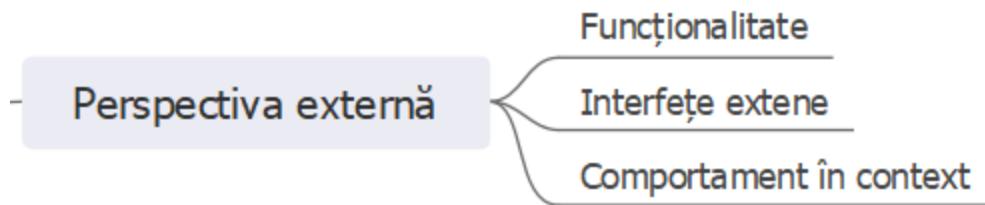


Reprezentarea aplicației (sistemului software) în contextul său.

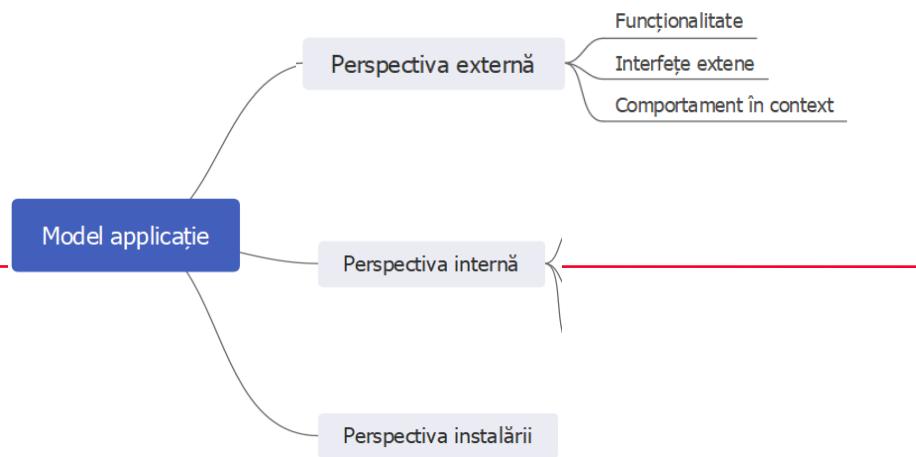
Modelare funcționalitate

Modelare interfețe externe

Modelarea comportamentului aplicației în context



PERSPECTIVA EXTERNĂ



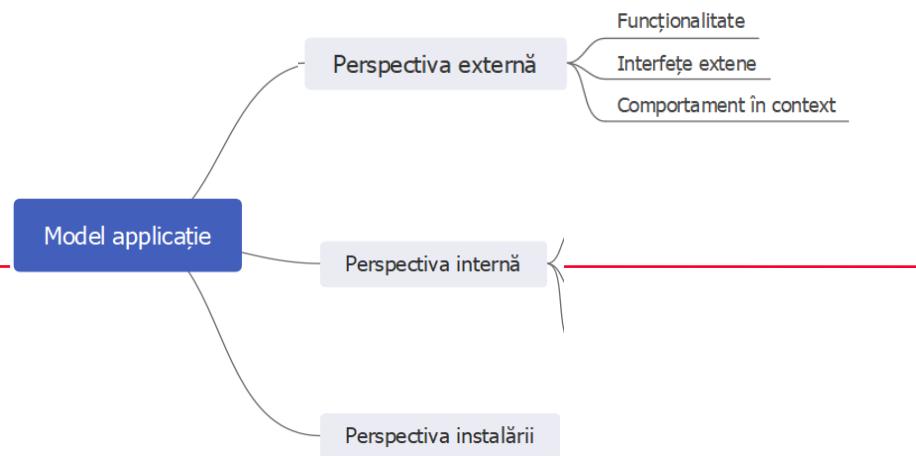
Reprezentarea aplicației (sistemului software) în contextul său.

Modelare funcționalitate

Cazuri de utilizare – diagrama UC

Interacțiuni – diagrama de secvențe la nivel de sistem

PERSPECTIVA EXTERNĂ



Reprezentarea aplicației (sistemului software) în contextul său.

Modelare interfețe externe

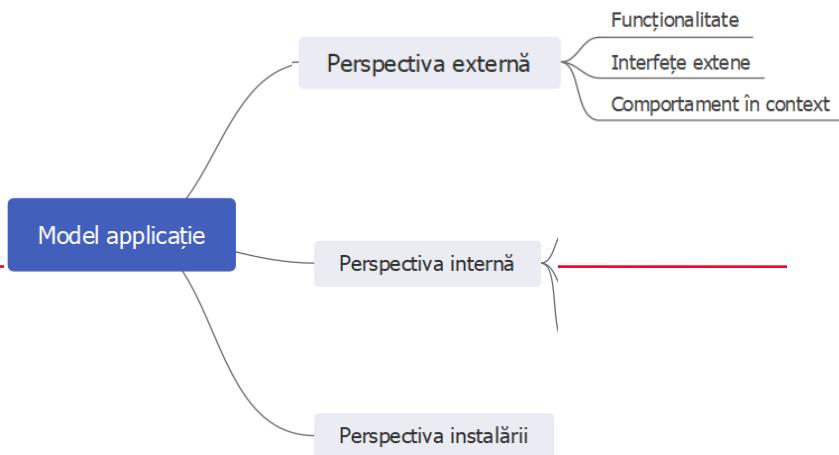
GUI (ecranele) :

conținut (capturi de ecran)
și flux de navigare – diagrama de stări și tranziții

API (funcțiile și structurile datelor transferate)

OpenAPI și instrumente (ex. *Stoplight Studio*)

PERSPECTIVA EXTERNĂ

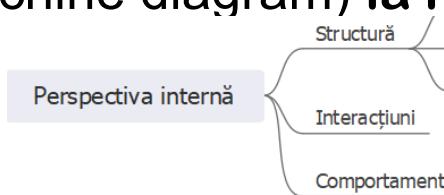


Reprezentarea aplicației (sistemului software) în contextul său.

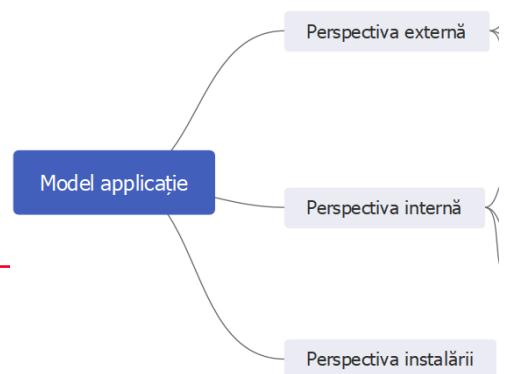
Modelare comportament în context

Reprezentarea fluxului de activități în utilizarea aplicației, cu **diagrama de activitate** (eventual **swimlane**).

Reprezentarea răspunsului aplicației la evenimente, cu **diagrama de stări și tranziții** (state machine diagram) **la nivel de sistem**.



PERSPECTIVA INTERNĂ

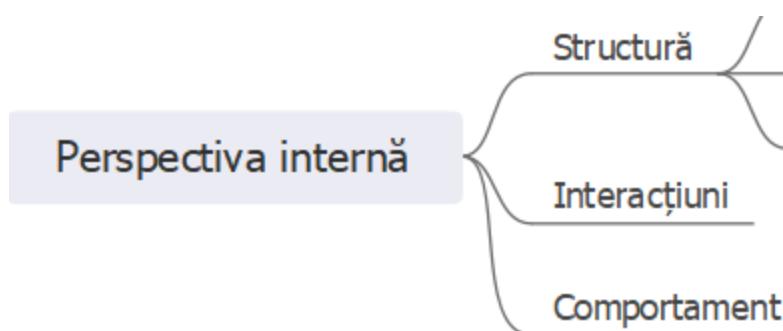


Reprezentarea modelelor elementelor interne ale sistemului software.

Structura internă a aplicației

Modelul interacțiunilor interne

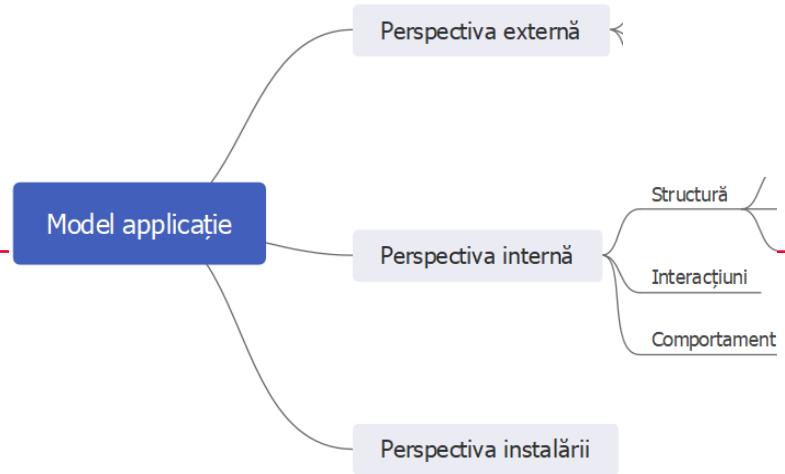
Modelul comportamentului intern relevant



PERSPECTIVA INTERNĂ

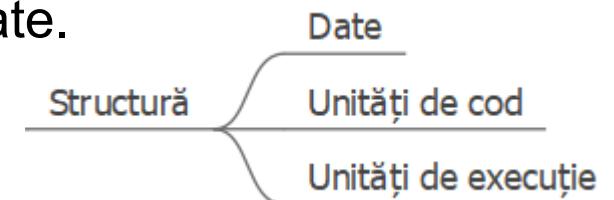
Reprezentarea modelelor elementelor interne ale sistemului software.

Structura internă a aplicației



Modelul datelor

Structuri ale datelor interne, structura bazei de date.



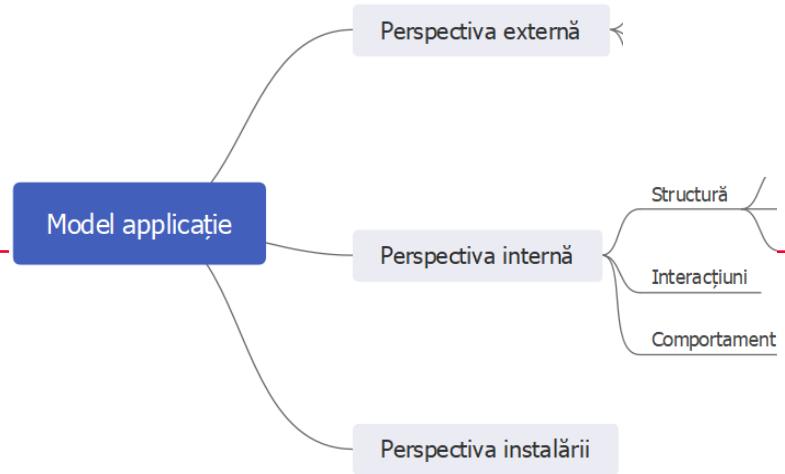
Modelul unităților de cod

Reprezentare clase (în abordarea OO) sau module de cod (în alte abordări) cu **diagrama de clase**.

Modelul unităților de execuție

Reprezentarea subsistemelor/componentelor și a interconectării acestora folosind **diagrama de componente**.

PERSPECTIVA INTERNĂ

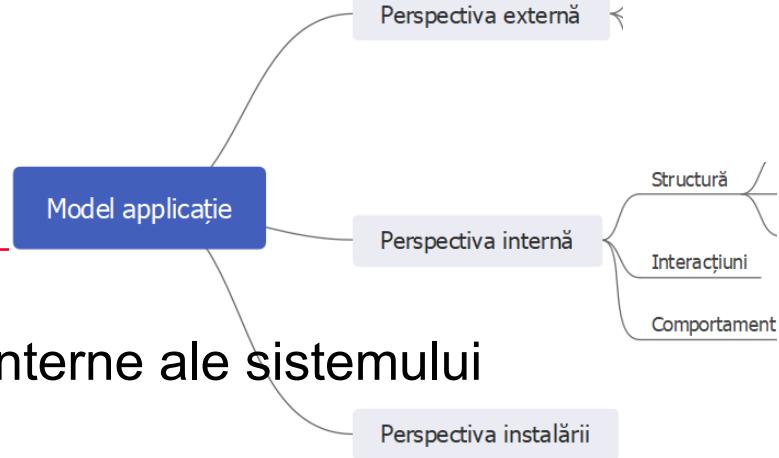


Reprezentarea modelelor elementelor interne ale sistemului software.

Modelul interacțiunilor interne

Reprezentare interacțiuni dintre subsisteme/componențe/obiecte folosind **diagrame de secvențe**.

PERSPECTIVA INTERNĂ



Reprezentarea modelelor elementelor interne ale sistemului software.

Modelul comportamentului intern relevant

Modelul proceselor interne

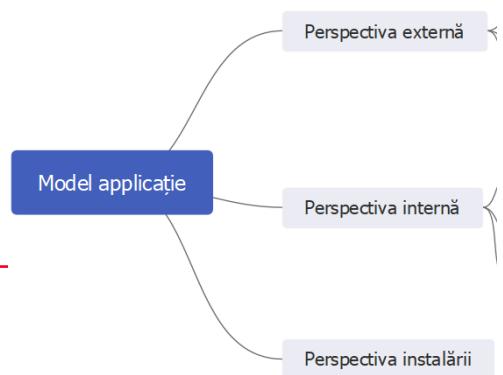
Descriere algoritmi/procese folosind **diagrame de activitate** sau **pseudocod**.



Modele de stări și tranziții

Descrierea stărilor și tranzițiilor subsistemelor ca răspuns la evenimente externe și interne, folosind **diagrame de stări și tranziții** (state machine diagrams).

PERSPECTIVA INSTALĂRII APLICAȚIEI

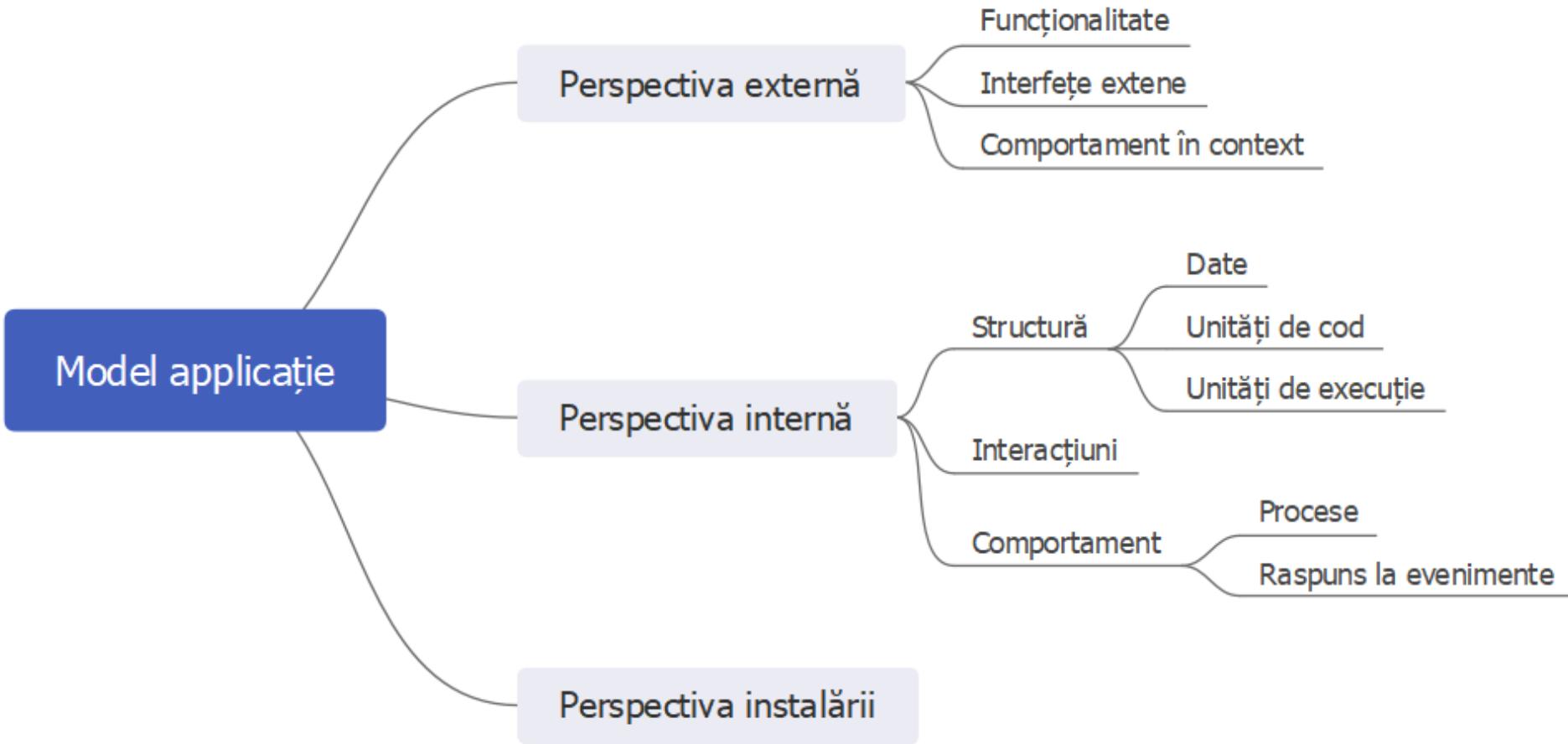


Reprezentare

- infrastructura de instalare
- relațiile dintre componentele aplicației și elementele de infrastructură

folosind **deployment diagram**.

Concluzie



Recomandări

În lucrarea de licență ar trebui să existe modele ale aplicației văzută din fiecare perspectivă.

Fiecare lucrare de licență va conține modelele potrivite cu tipul de aplicație.

Acstea vor fi selectate din setul de modele recomandate în continuare.

Se recomandă alegerea modelelor corespunzătoare și folosirea majorității acestora în activitatea de proiectare a aplicației, înainte de dezvoltarea codului.

PERSPECTIVA EXTERNĂ

PERSPECTIVA INTERNĂ

PERSPECTIVA INSTALĂRII APLICAȚIEI

PERSPECTIVA EXTERNĂ

MODELARE PROCES BUSINESS

Diagrama de activitate

Reprezentarea **procesului business** cu o diagramă de activitate.

Utilitate : identificare cazuri de utilizare.

DIAGRAMA SWIMLANE

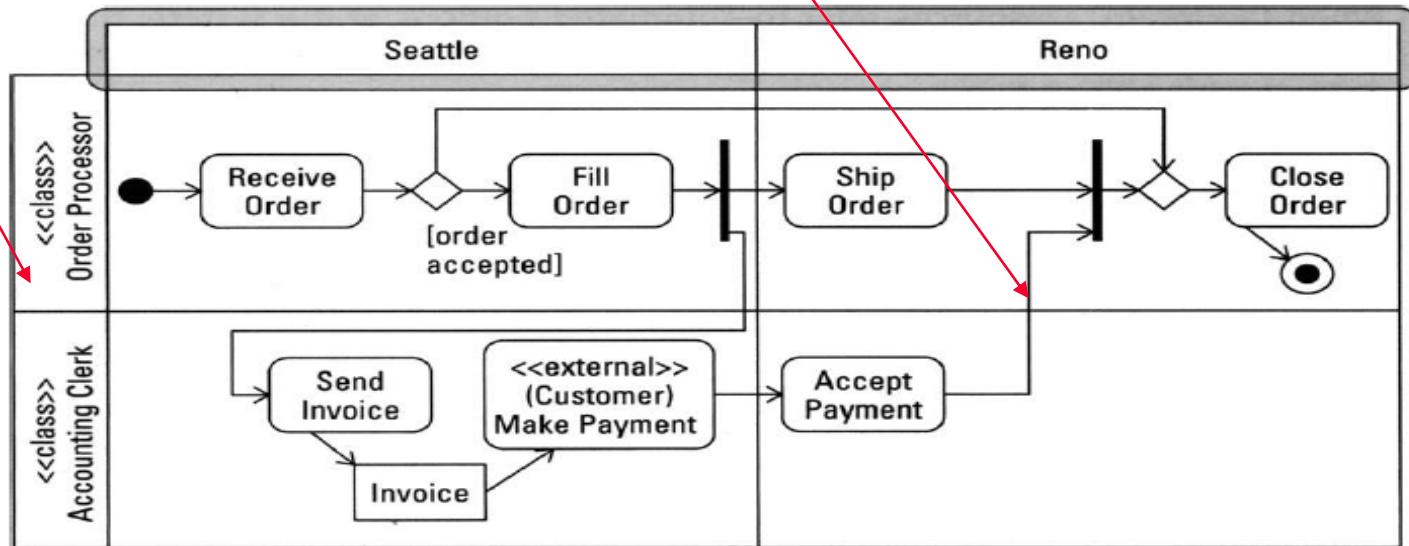
Perspectivă organizată a diagramei de activitate prin gruparea activităților în **partiții**.

Utilizată, de obicei, pentru reprezentare procese business.

Criteriu tipic de partiziionare: **Entitatea** (actor, clasă, etc.) sau
grupul de entități responsabile cu
realizarea fiecărei activități.

Diagrama poate avea 1 sau 2 dimensiuni.
Se pot reprezenta subpartiții.

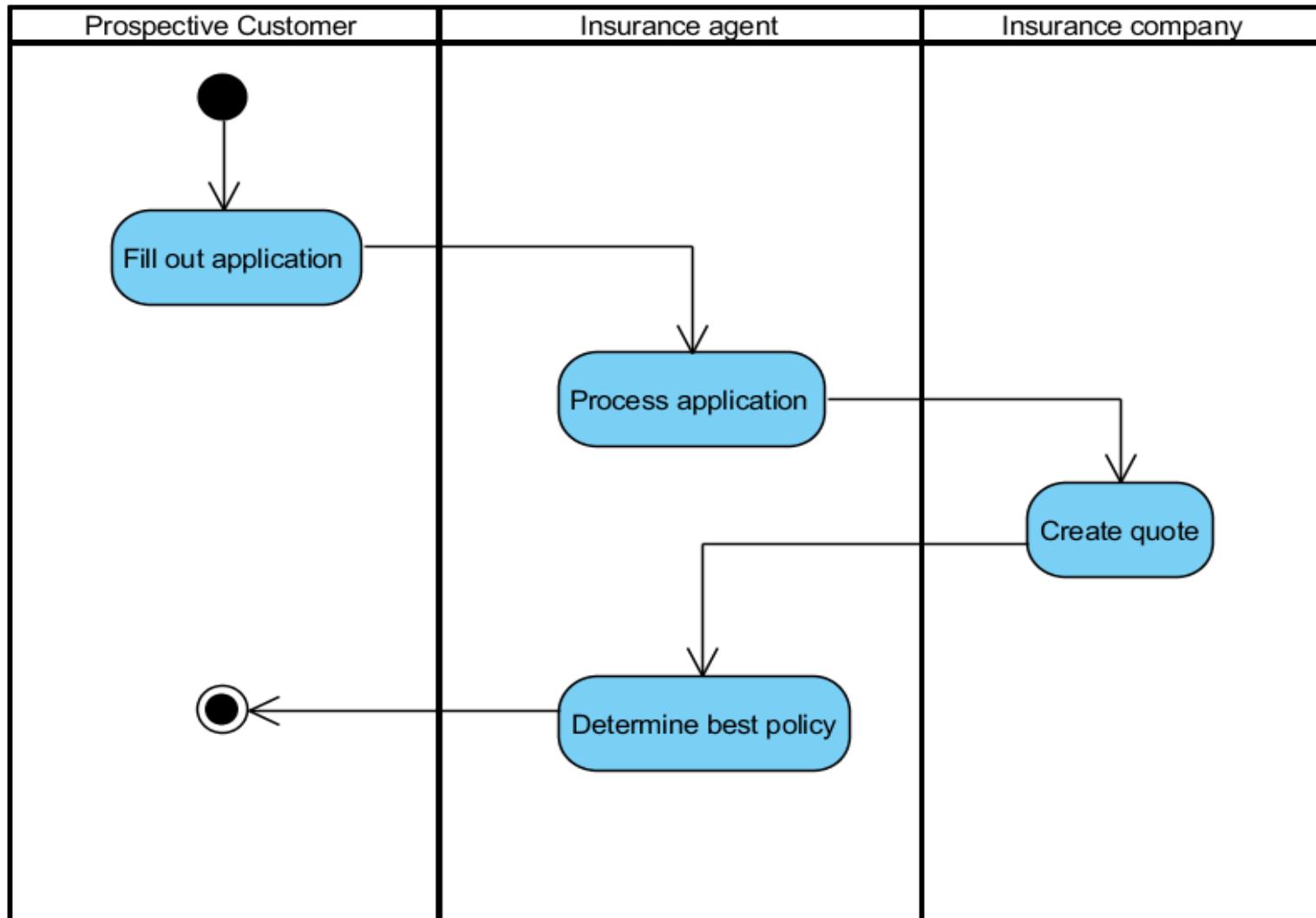
Este vizibil transferul responsabilității.



PERSPECTIVA EXTERNA MODELARE PROCES BUSINESS

Diagrama de activitate

Alte exemple :

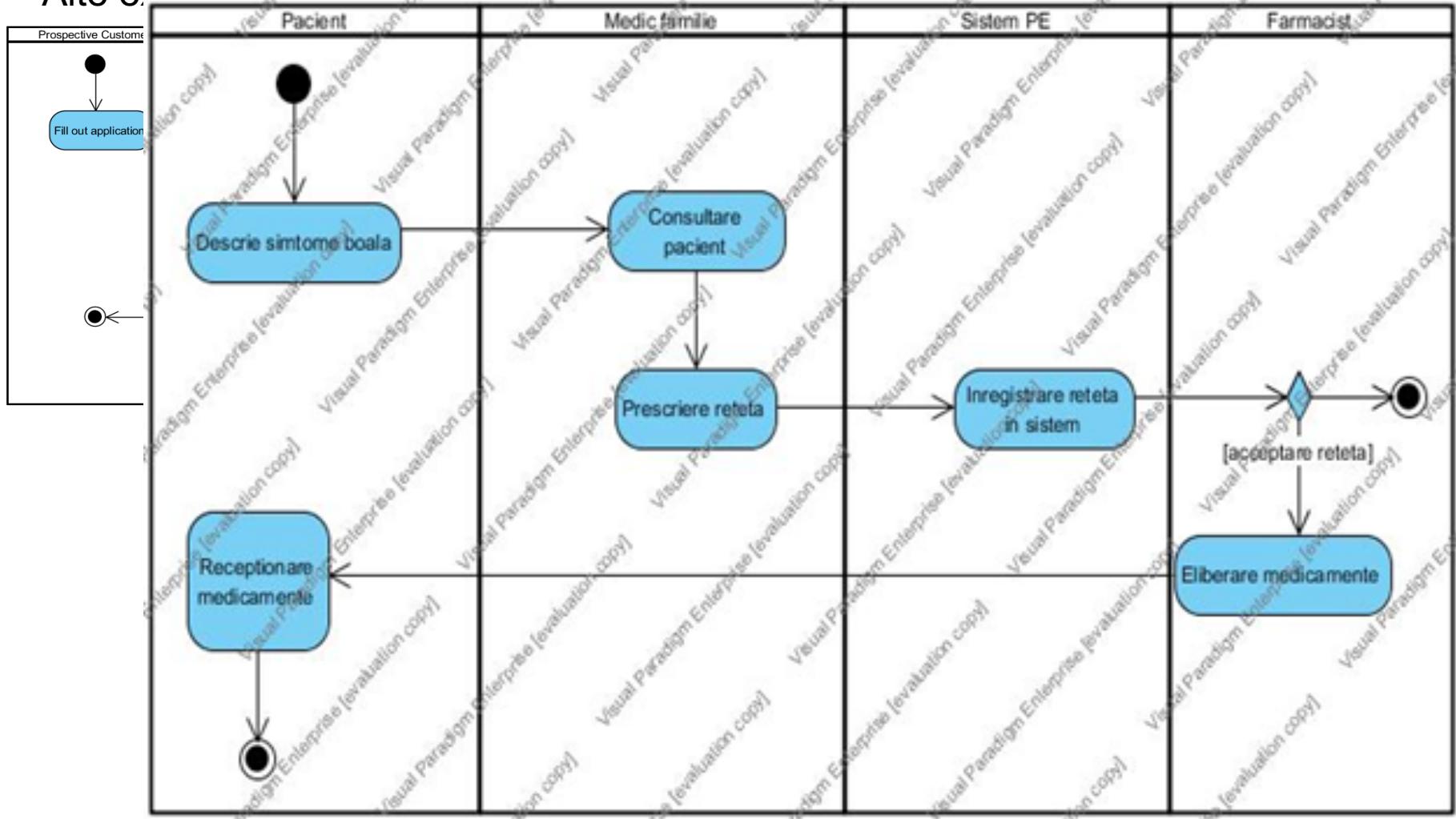


PERSPECTIVA EXTERNA

MODELARE PROCES BUSINESS

Diagrama de activitate

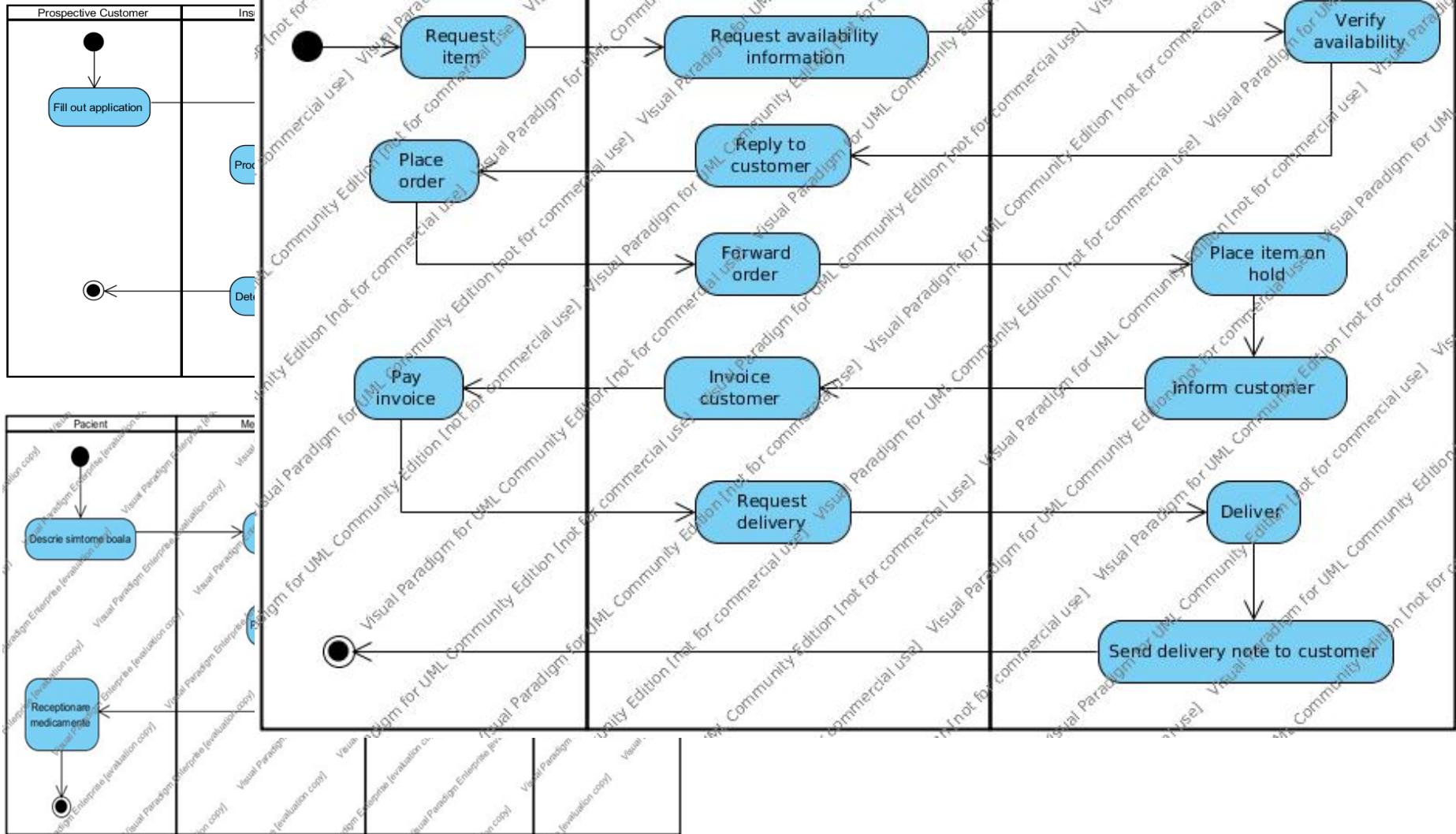
Alte exemple:



PERSPECTIVA EXTERNA MODELARE PROCES BUSINESS

Diagrama de activitate

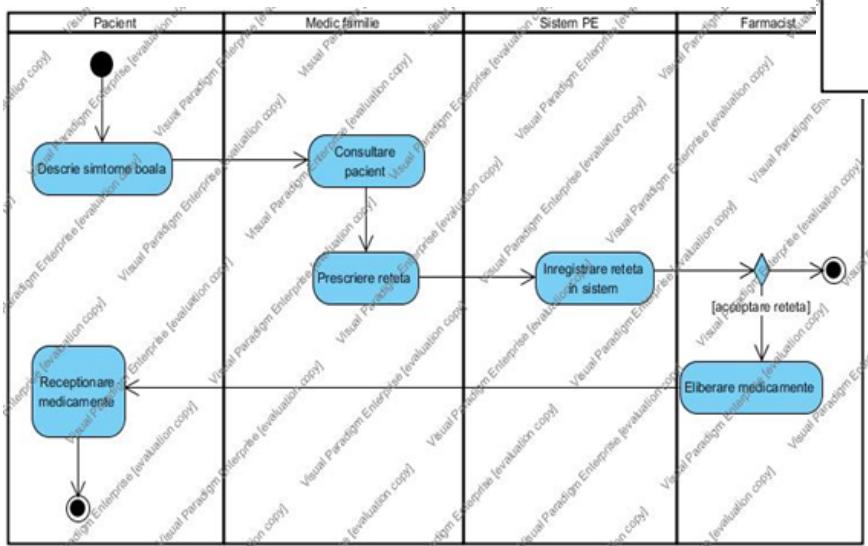
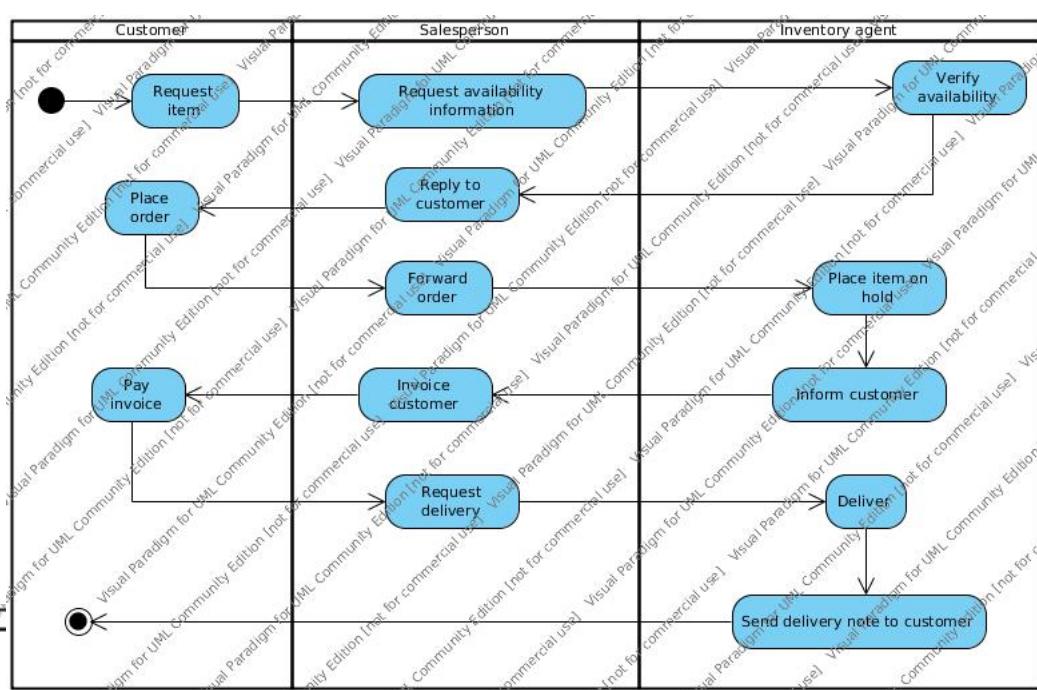
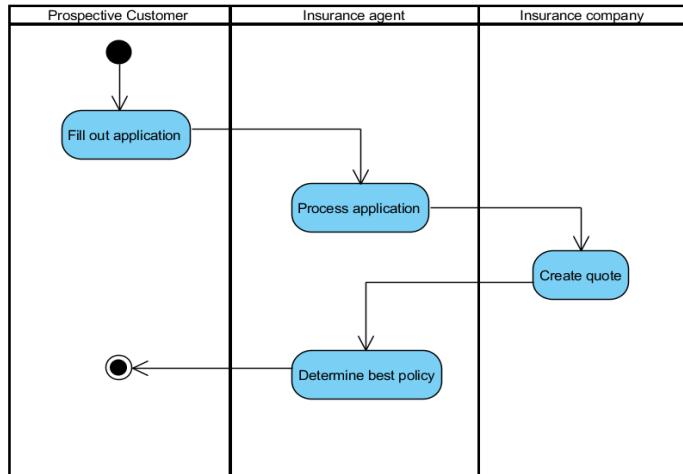
Alte exemple



PERSPECTIVA EXTERNA MODELARE PROCES BUSINESS

Diagrama de activitate

Alte exemple :



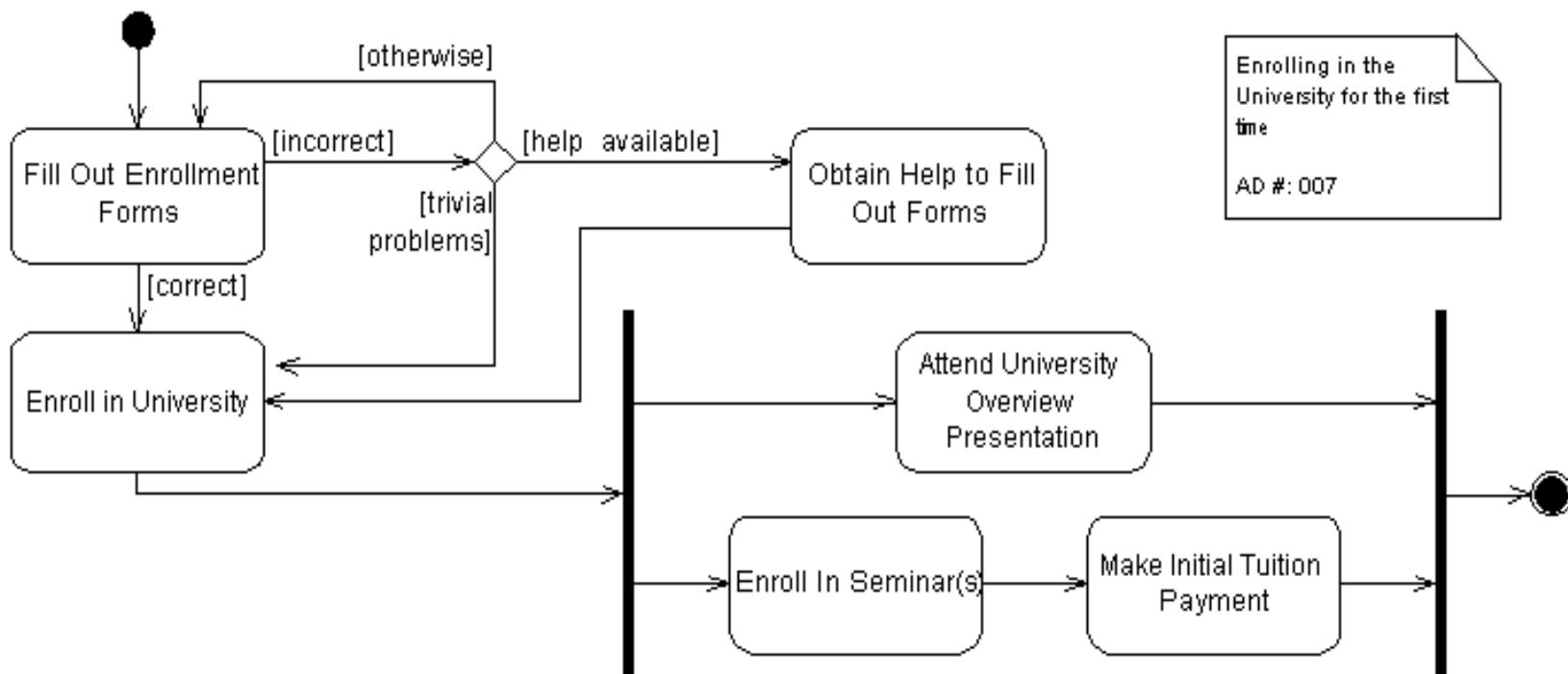
PERSPECTIVA EXTERNA MODELARE PROCES BUSINESS

Diagrama de activitate

Reprezentarea **procesului business** cu o diagramă de activitate.

Utilitate : identificare cazuri de utilizare.

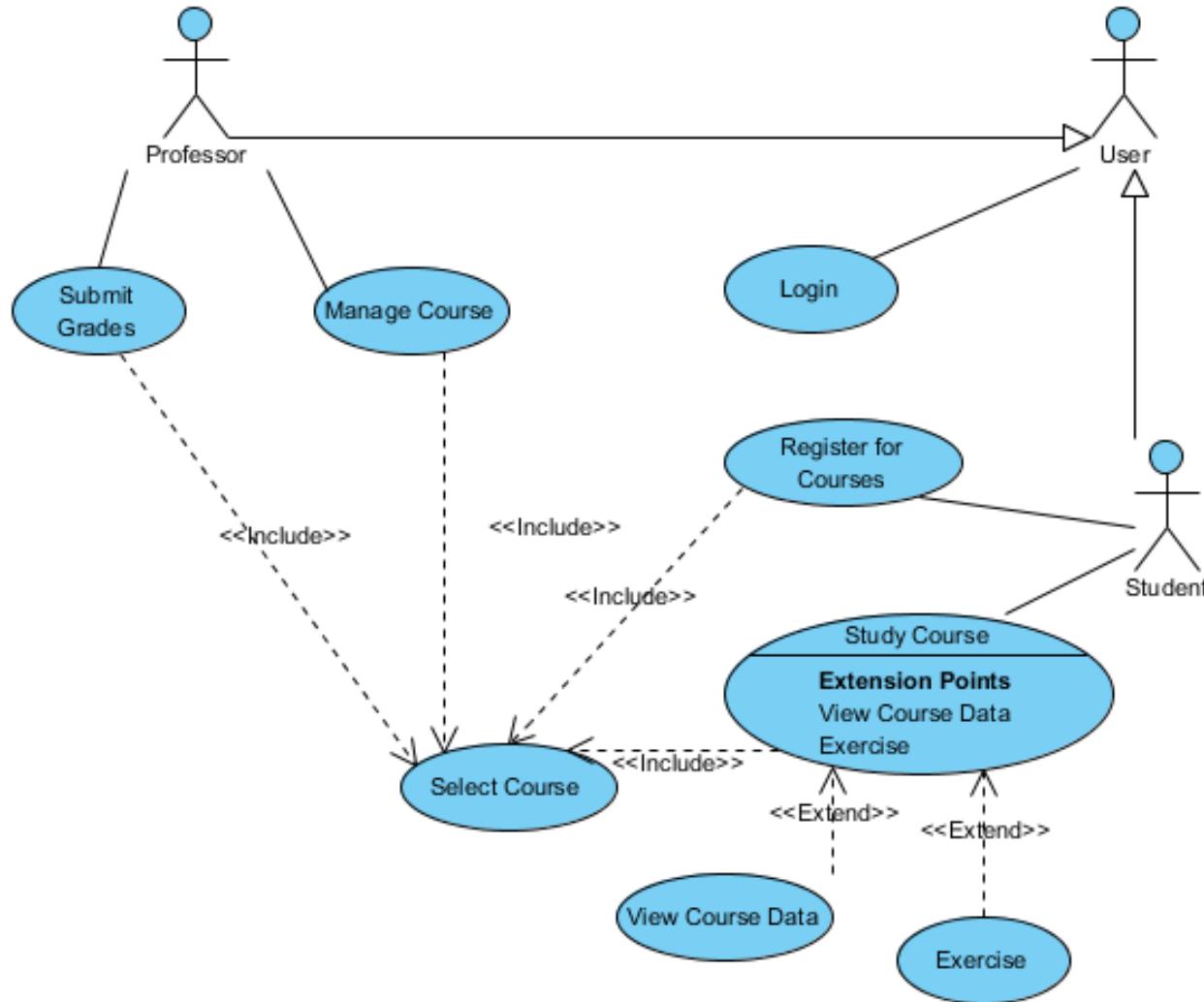
Exemplu de modelare a unui proces business, din punct de vedere al unui singur actor, cu o diagramă de activitate:



PERSPECTIVA EXTERNĂ

MODELARE FUNCȚIONALITATE

Diagrama cazurilor de utilizare

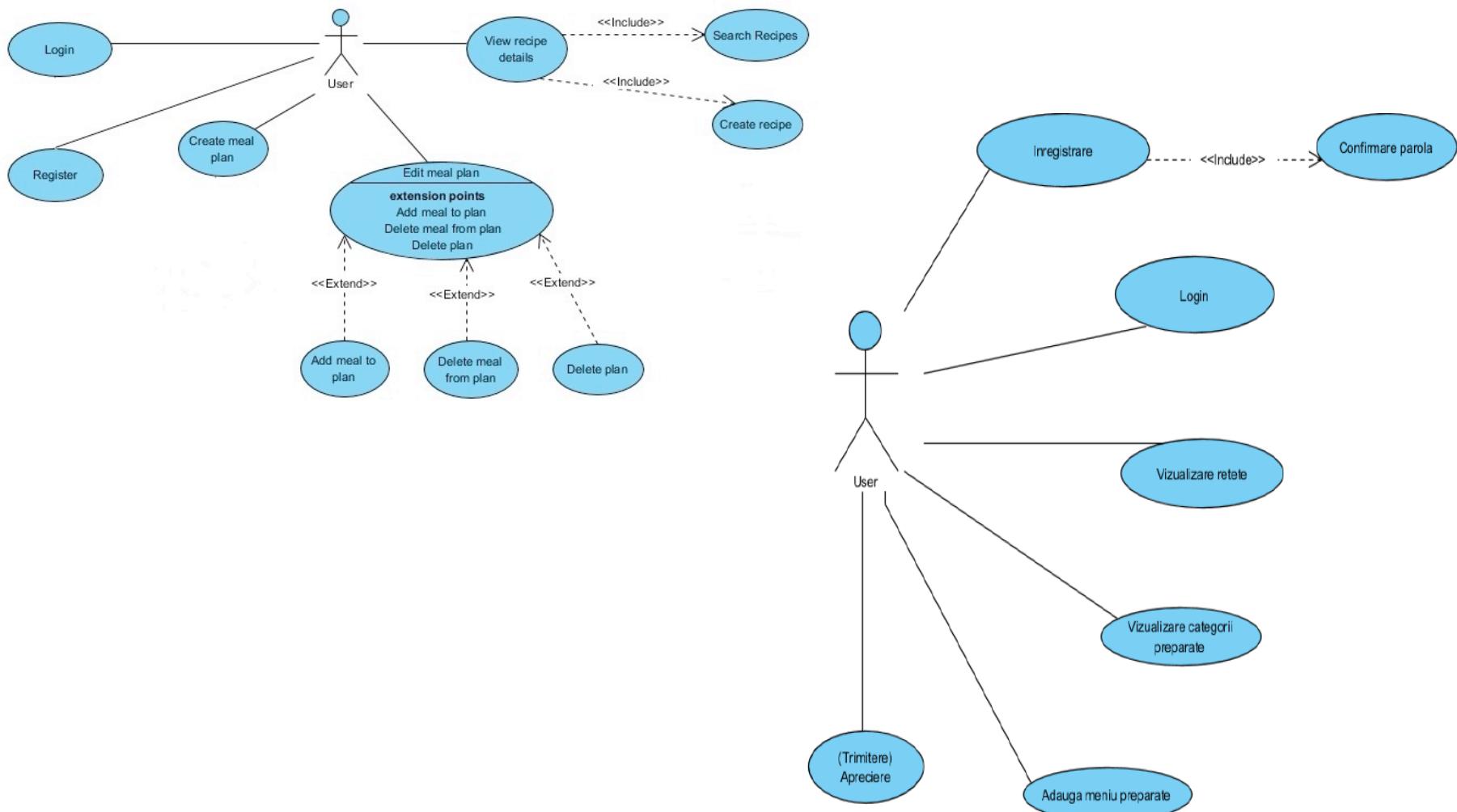


PERSPECTIVA EXTERNĂ

MODELARE FUNCȚIONALITATE

Diagrama cazurilor de utilizare

Alte diagrame UC.



PERSPECTIVA EXTERNA MODELARE FUNCȚIONALITATE

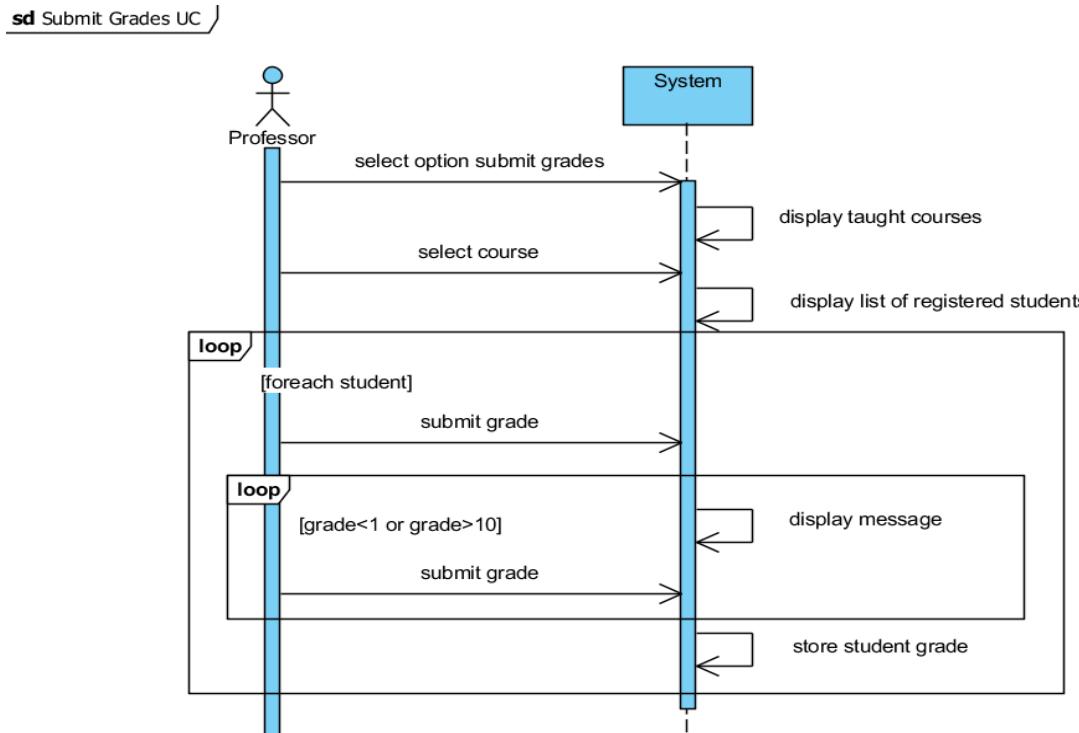
Detalii UC

UC name: Submit grades

Actor(s): Professor

Description: Professor registers the grades obtained by the students.

Precondition : The actor must be identified in the system as Professor.



Postcondition : The grades obtained by the students to the selected course exist stored in the system.

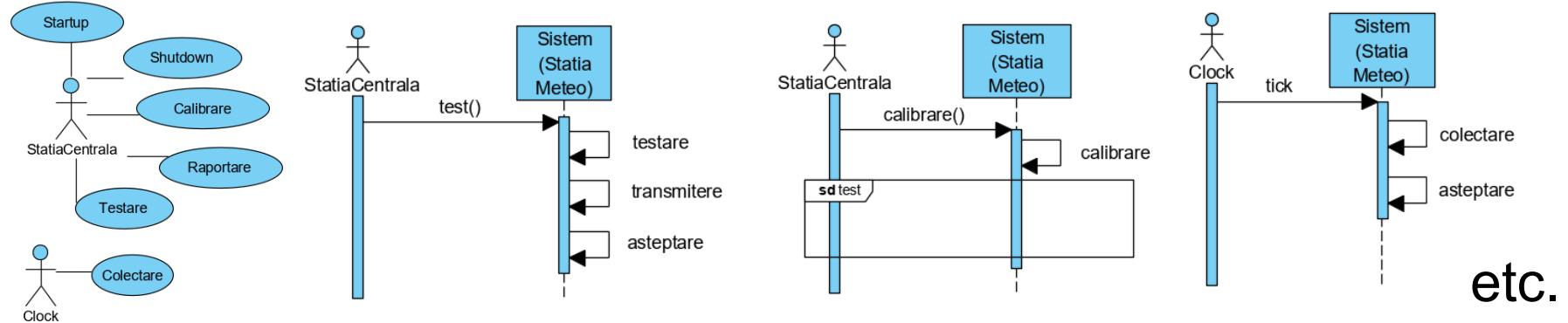
PERSPECTIVE EXTERNĂ

MODELAREA COMPORTAMENTULUI

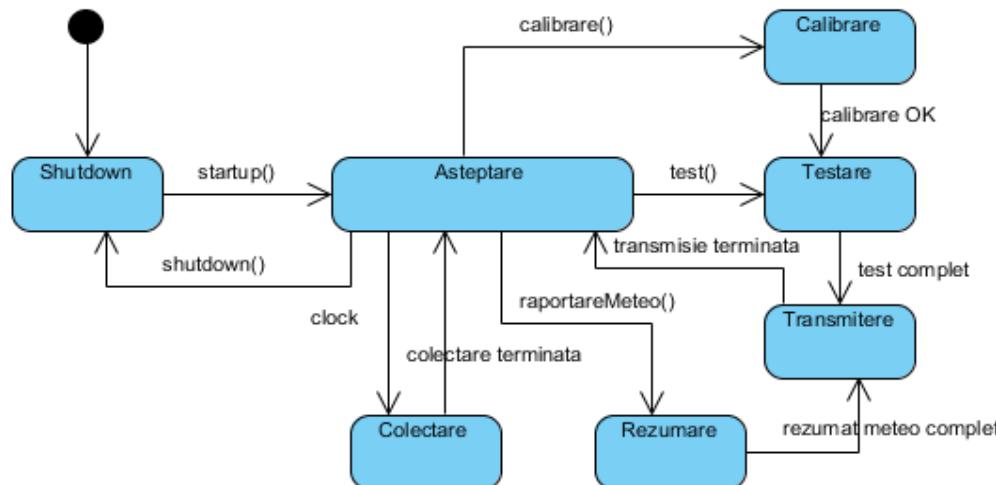
la nivelul SISTEMULUI

Exemplu: Stație meteo

Crearea *diagramei de secvențe la nivel de sistem*, pentru fiecare caz de utilizare.



Crearea *diagramei de stări și tranzitii* la nivelul sistemului.

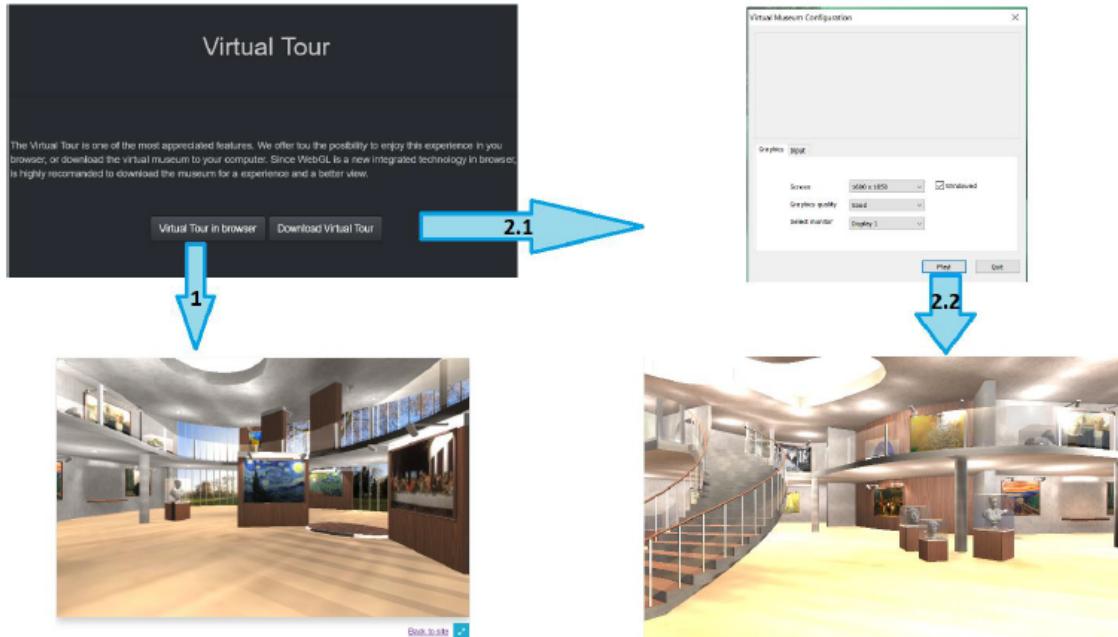


MODELARE INTERFEȚE EXTERNE

GUI (varianta 1)

GUI (ecranele) : conținut (capturi de ecran) și flux de navigare descris în text.

Interfața de accesare a muzeului virtual (pagina web Virtual Tour):



Fluxul evenimentelor:

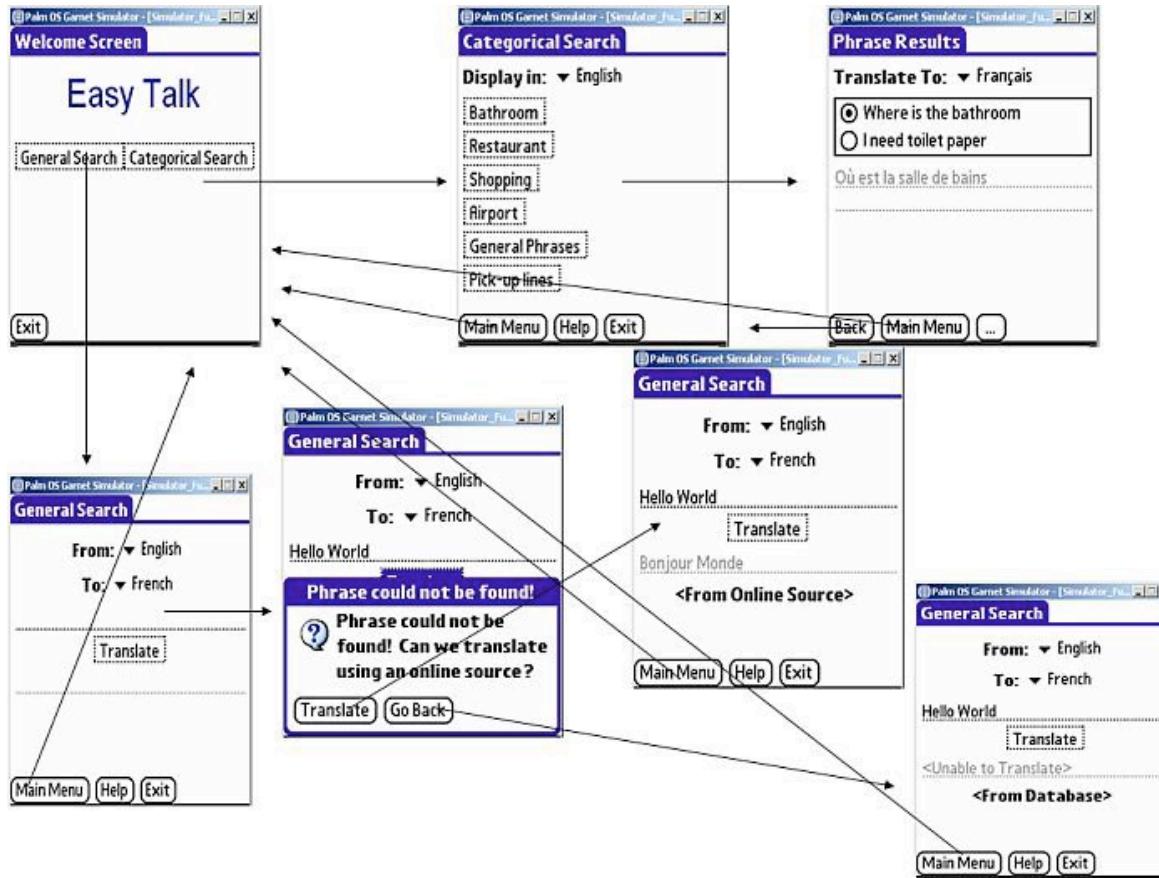
1. Optiunea de vizualizare a muzeului virtual in browser, va accesa pagina web a muzeului virtual;
2. Optiunea de a descarca muzeul virtual ofera un fisier executabil;
 - 2.1 La deschiderea fisierului executabil apare o fereastra de configurare;
 - 2.2 Prin apasarea butonului "Play", se deschide muzeul virtual.

Vezi [GUI.pdf](#)

MODELARE INTERFEȚE EXTERNE GUI (varianta 2)

GUI (ecranele) :

conținut (eventual capturi de ecran)
și flux de navigare – reprezentat cu săgeți



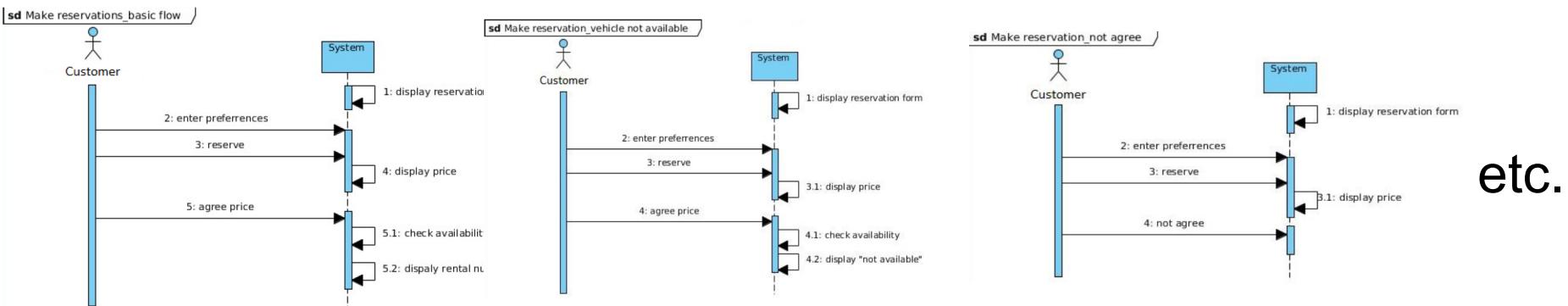
MODELARE INTERFEȚE EXTERNE

GUI (varianta 3)

GUI (ecranele) :

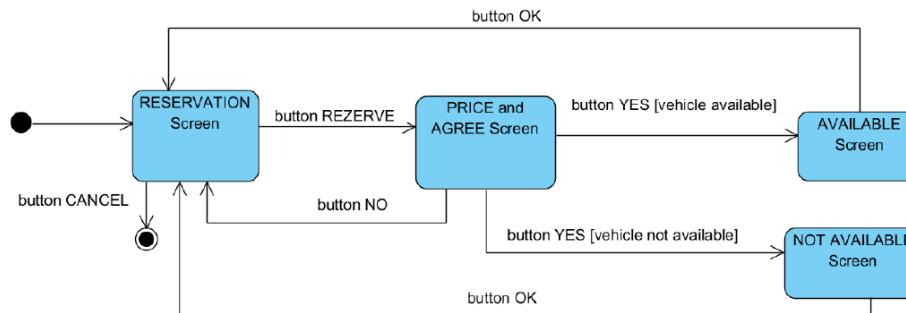
conținut (capturi de ecran) și flux de navigare – diagrama de stări și tranzitii

Pe baza *diagramelor de secvențe la nivel de sistem*, pentru fiecare caz de utilizare.



Crearea *diagramelor de stări și tranzitii* cu fluxul ecranelor.

Example: Make reservation UC



MODELARE INTERFEȚE EXTERNE GUI (varianta 3)

GUI (ecranele) : conținut (capturi de ecran) și fluxul ecranelor.

Alt exemplu : *Diagramă de stări și tranzitii* cu fluxul ecranelor.

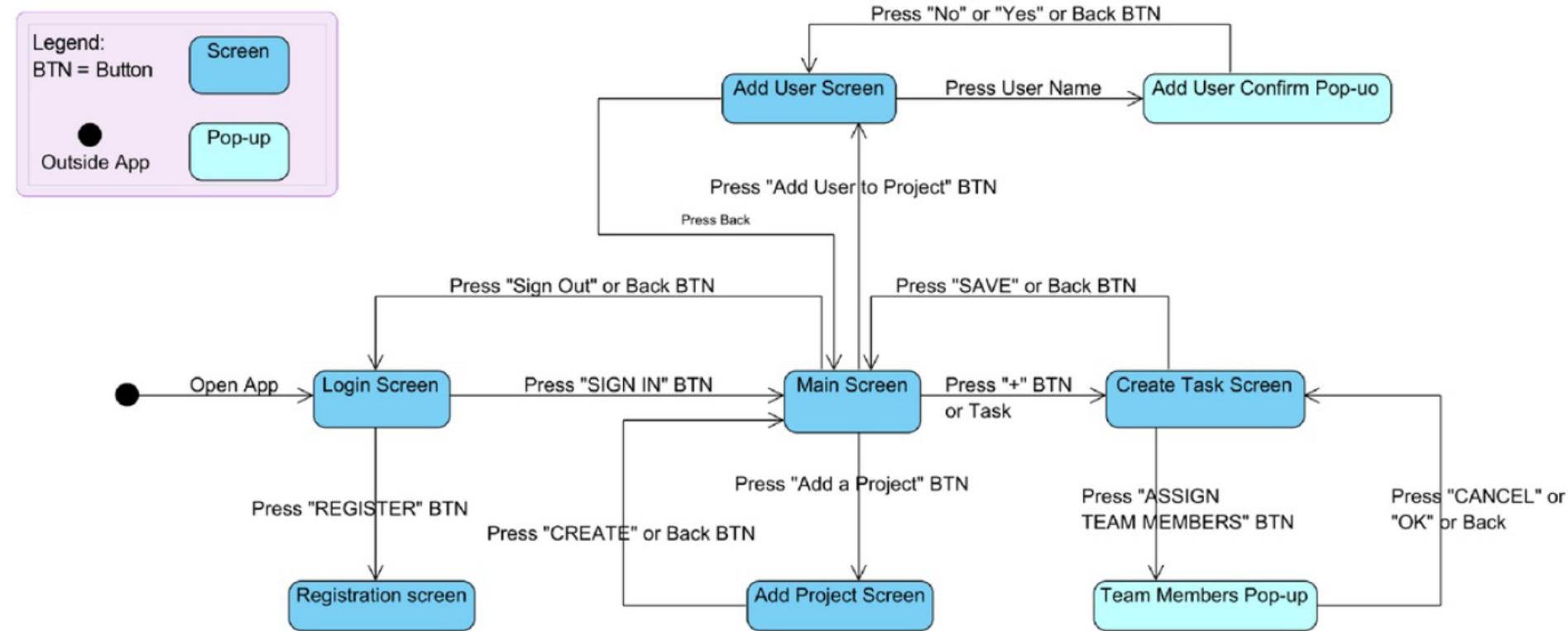


Fig.20 Diagrama de stări a ecranelor.

MODELARE INTERFEȚE EXTERNE API

API (funcțiile și structurile datelor transferate)

OpenAPI și instrumente (ex. Stoplight Studio)

| | | |
|--------|------------------------------|--|
| GET | /Accounts/{id}/tickets | Queries tickets of Account. |
| POST | /Accounts/{id}/tickets | Creates a new instance in tickets of this model. |
| DELETE | /Accounts/{id}/tickets | Deletes all tickets of this model. |
| GET | /Accounts/{id}/tickets/{fk} | Find a related item by id for tickets. |
| PUT | /Accounts/{id}/tickets/{fk} | Update a related item by id for tickets. |
| DELETE | /Accounts/{id}/tickets/{fk} | Delete a related item by id for tickets. |
| GET | /Accounts/{id}/tickets/count | Counts tickets of Account. |

Figura 15: Definiția API-ului pentru bilete

PERSPECTIVA EXTERNĂ

PERSPECTIVA INTERNĂ

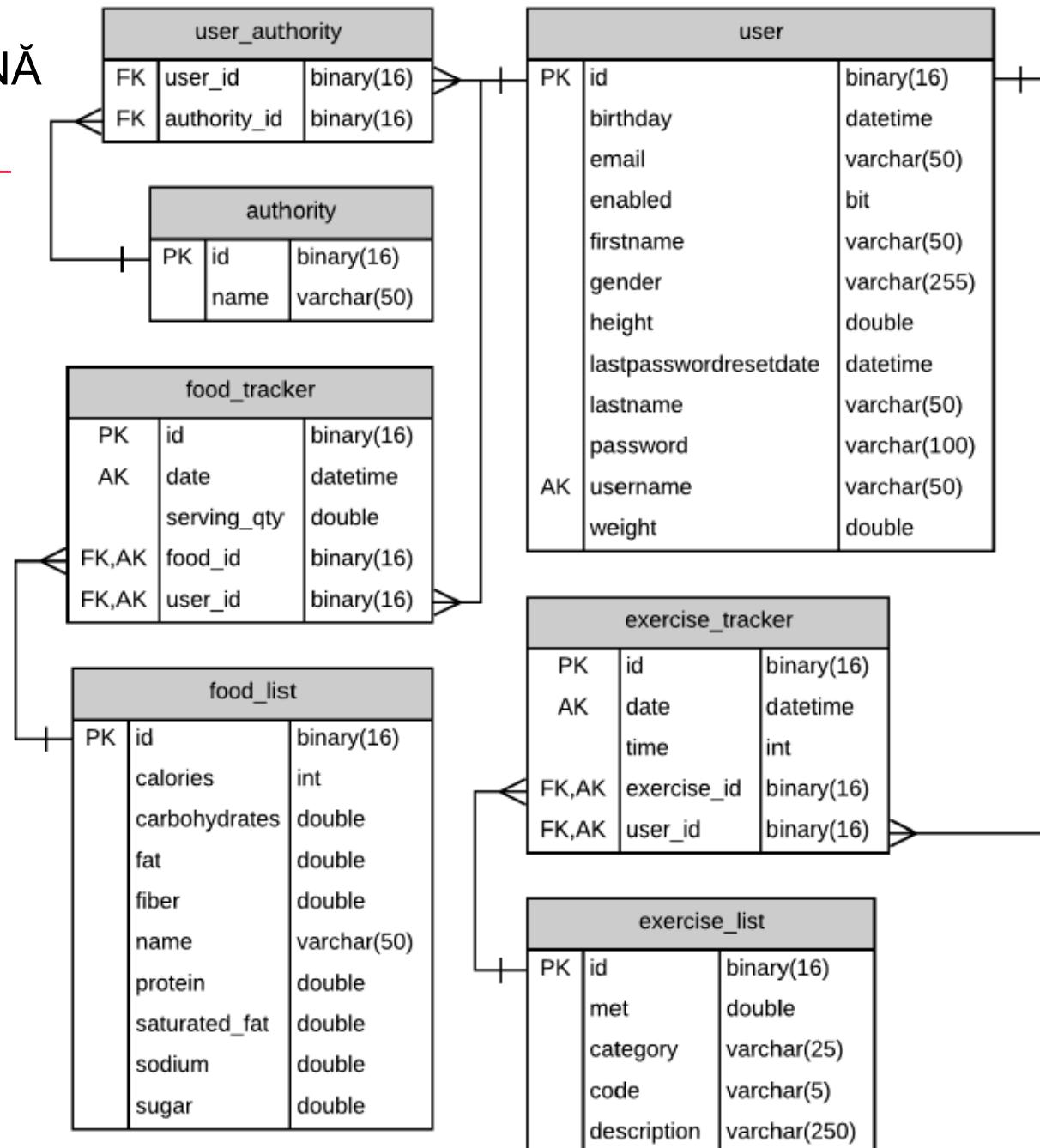
PERSPECTIVA INSTALĂRII APLICAȚIEI

PERSPECTIVA INTERNĂ STRUCTURĂ DATE

BD relatională.

Trebuie completată cu dicționarul datelor.

Ex. [BD.pdf](#)



PERSPECTIVA INTERNĂ

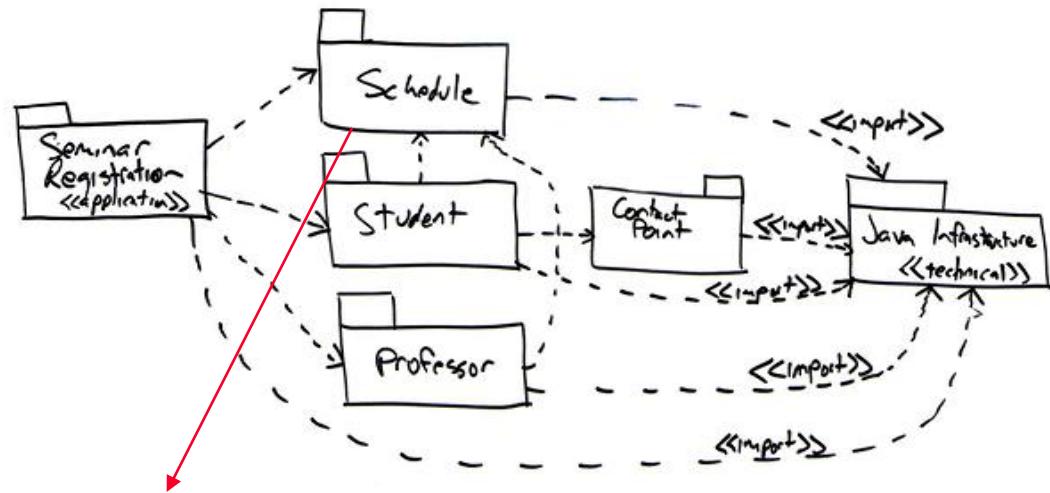
STRUCTURĂ UNITĂȚI DE COD

PACHETE și CLASE

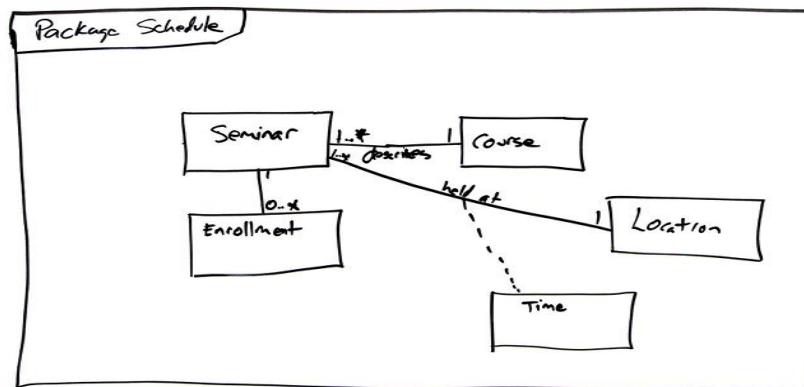
Organizare clase în pachete.

PACHETE. Construcții UML utilizate pentru organizarea elementelor unui model în grupuri.

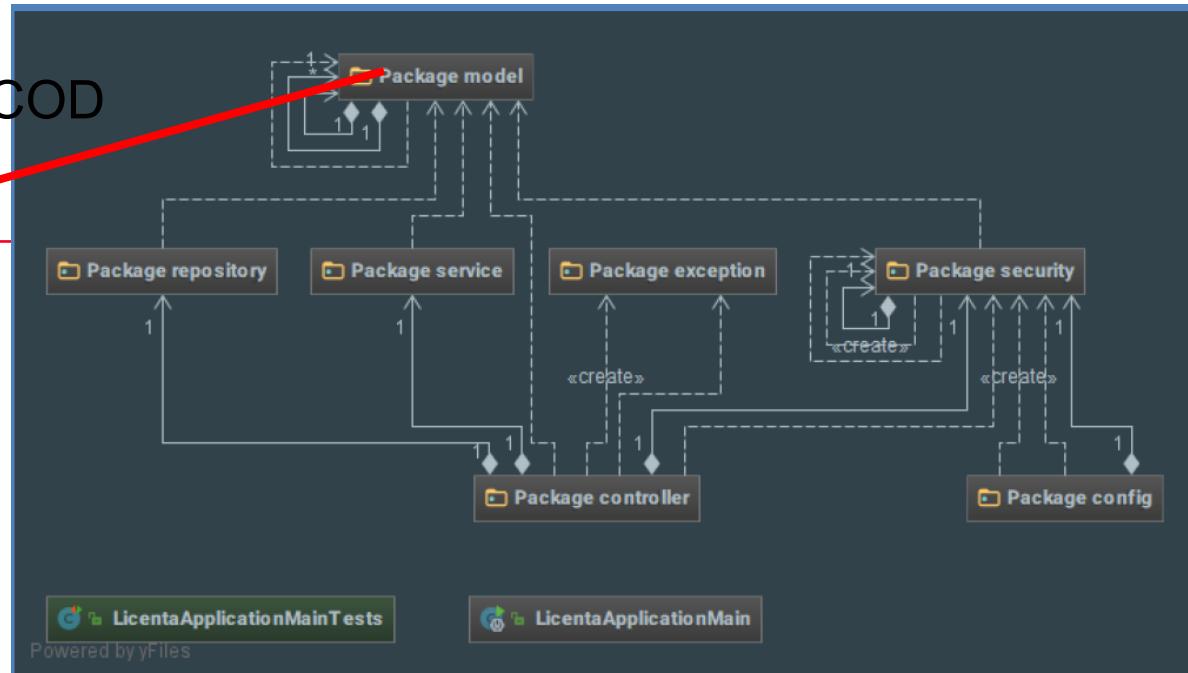
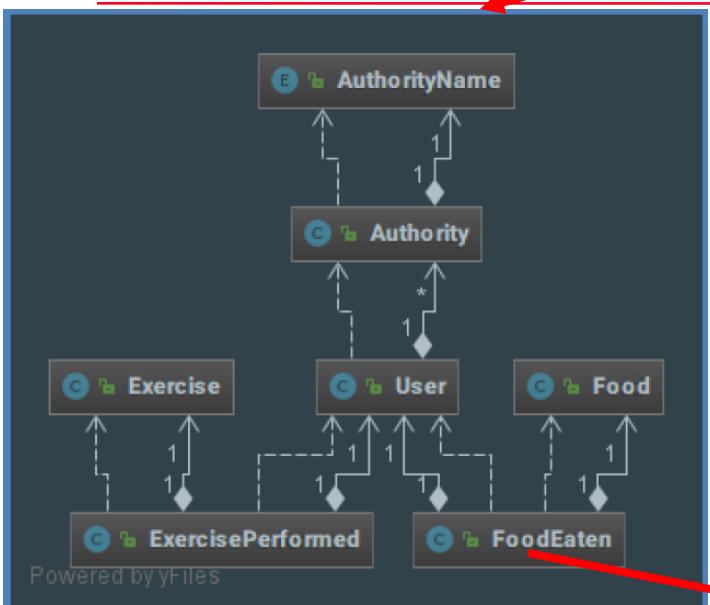
Diagramă de pachete:



Conținut pachet:



PERSPECTIVA INTERNĂ STRUCTURĂ UNITĂȚI DE COD



Powered by yFiles

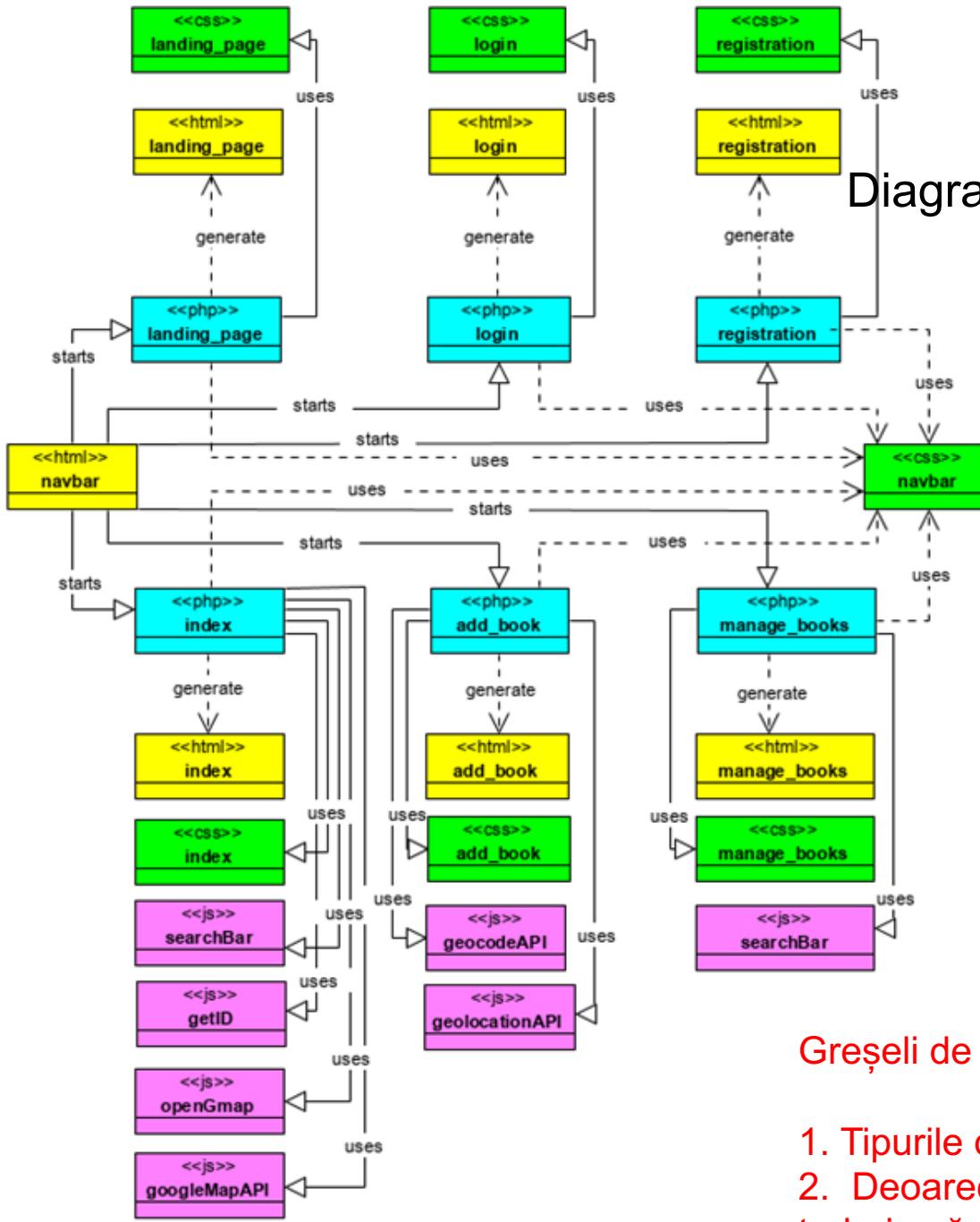
| | |
|--|--------|
| <code>FoodEaten</code> | |
| <code>f id</code> | UUID |
| <code>f user</code> | User |
| <code>f food</code> | Food |
| <code>f date</code> | Date |
| <code>f servingQty</code> | double |
| <code>m FoodEaten()</code> | |
| <code>m FoodEaten(User, Food, Date, double)</code> | |
| <code>m getid()</code> | UUID |
| <code>m setid(UUID)</code> | void |
| <code>m getUser()</code> | User |
| <code>m setUser(User)</code> | void |
| <code>m getFood()</code> | Food |
| <code>m setFood(Food)</code> | void |
| <code>m getDate()</code> | Date |
| <code>m setDate(Date)</code> | void |
| <code>m getServingQty()</code> | double |
| <code>m setServingQty(double)</code> | void |
| <code>m computeCaloriesEaten()</code> | int |

The FoodEaten class represents the meals dairy of each user. The user chooses the name of the food from a drop-down menu and inputs the quantity of said food, then the server autogenerated the next fields: the date is taken from server's current date in format("mm:HH dd-MM-yyyy") and the rest of the fields: the number of calories, the amount of fat, saturated fat, carbohydrates, fiber, sugar, protein, sodium are taken from the selected food but multiplied with the serving quantity ,be able to distinguish between users a unique id that is generated by the application. Additionally, it includes the usual getters, setters and constructors.

PERSPECTIVA INTERNĂ

STRUCTURĂ UNITĂȚI DE COD

Diagramă de clase cu clase stereotipate



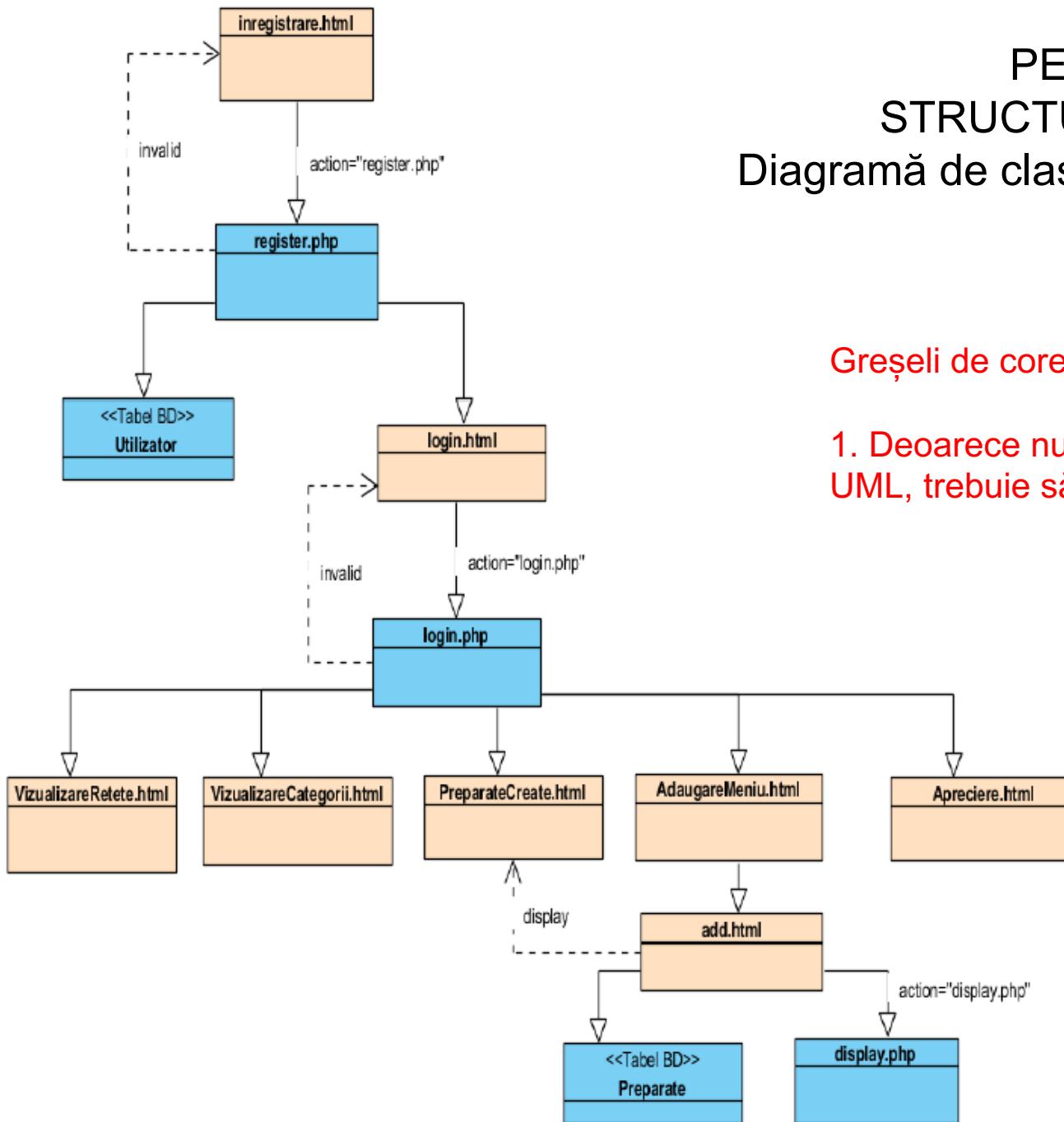
Greșeli de corectat:

1. Tipurile de săgeți nu sunt folosite în mod unitar.
2. Deoarece nu se folosesc relații standard UML, trebuie să existe o legendă.

PERSPECTIVA INTERNĂ

STRUCTURĂ UNITĂȚI DE COD

Diagramă de clase cu clase stereotipate



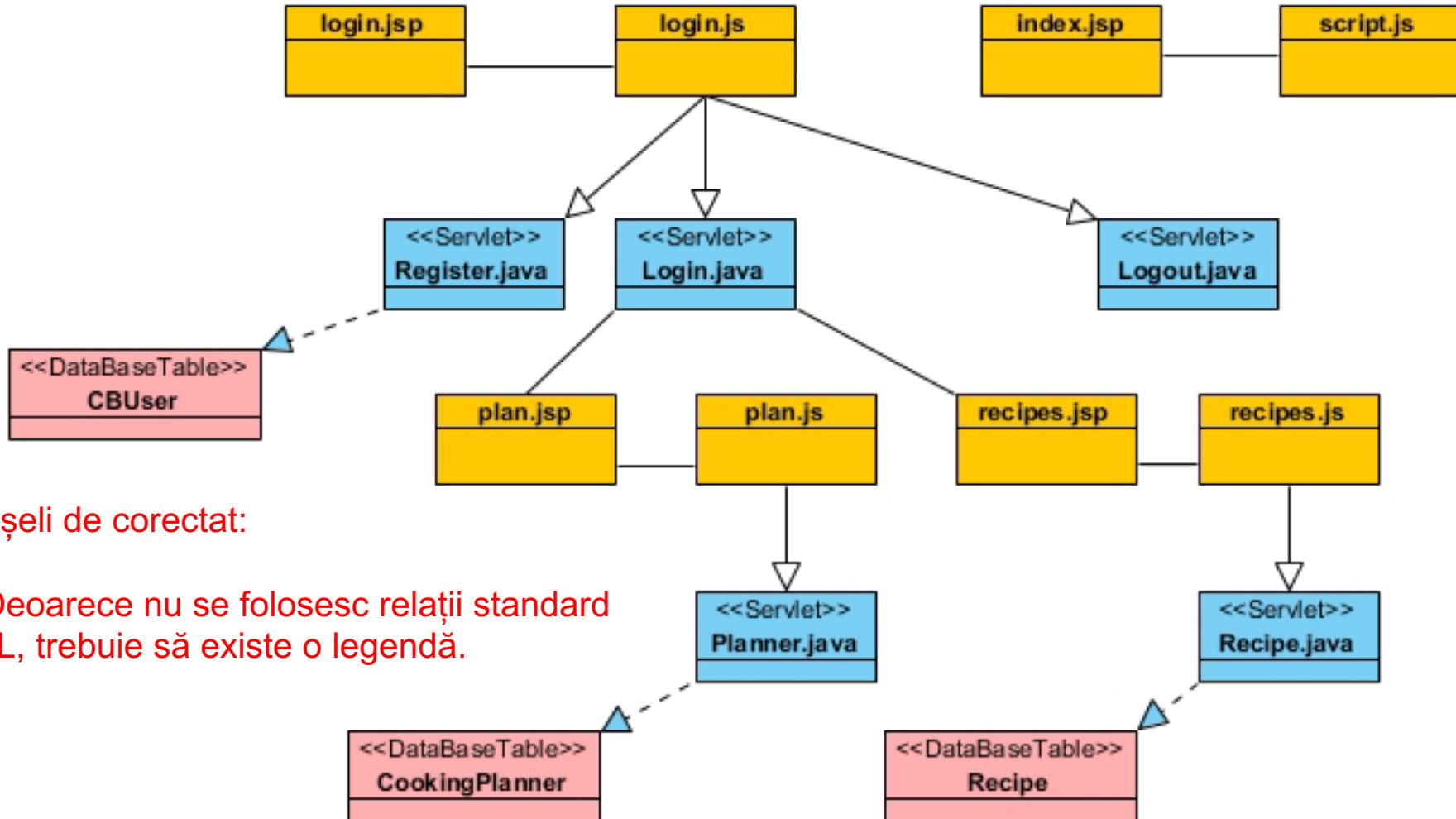
Greșeli de corectat:

1. Deoarece nu se folosesc relații standard UML, trebuie să existe o legendă.

PERSPECTIVA INTERNĂ

STRUCTURĂ UNITĂȚI DE COD și DATE

Diagramă de clase



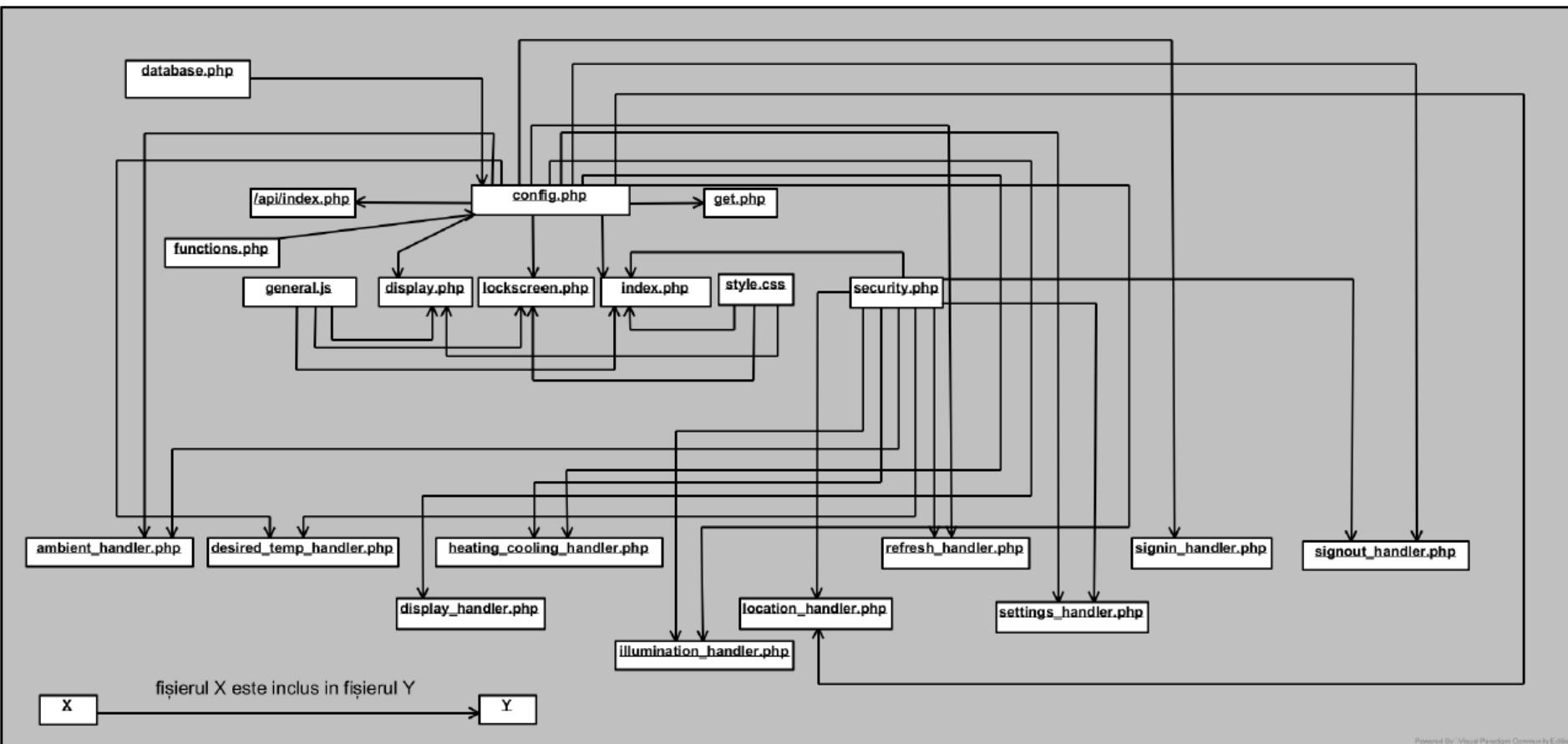
Greșeli de corectat:

1. Deoarece nu se folosesc relații standard UML, trebuie să existe o legendă.

PERSPECTIVA INTERNĂ

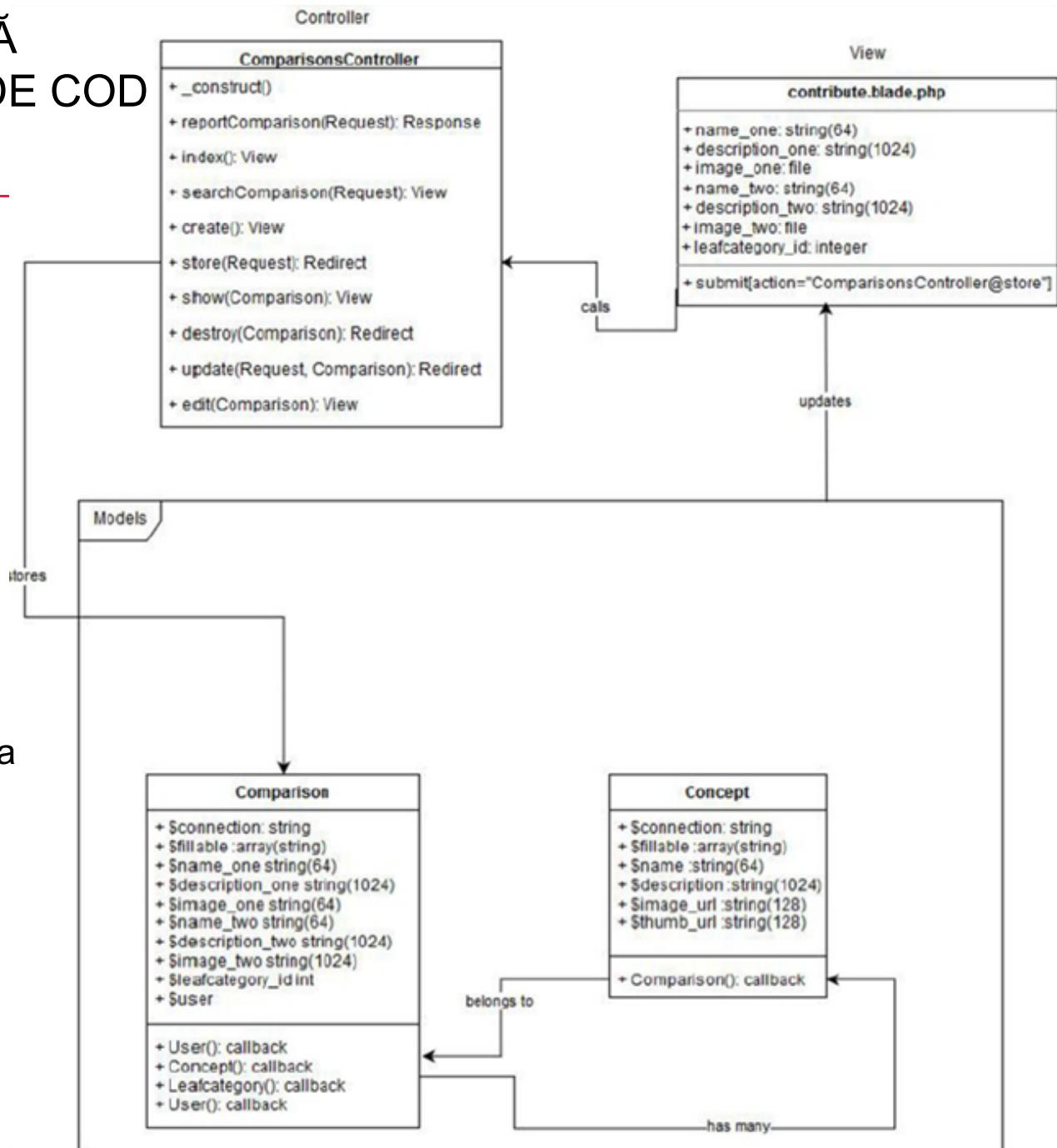
STRUCTURĂ UNITĂȚI DE COD

Structura de fișiere și relațiile de includere



PERSPECTIVA INTERNĂ STRUCTURĂ UNITĂȚI DE COD

Diagramă de clase

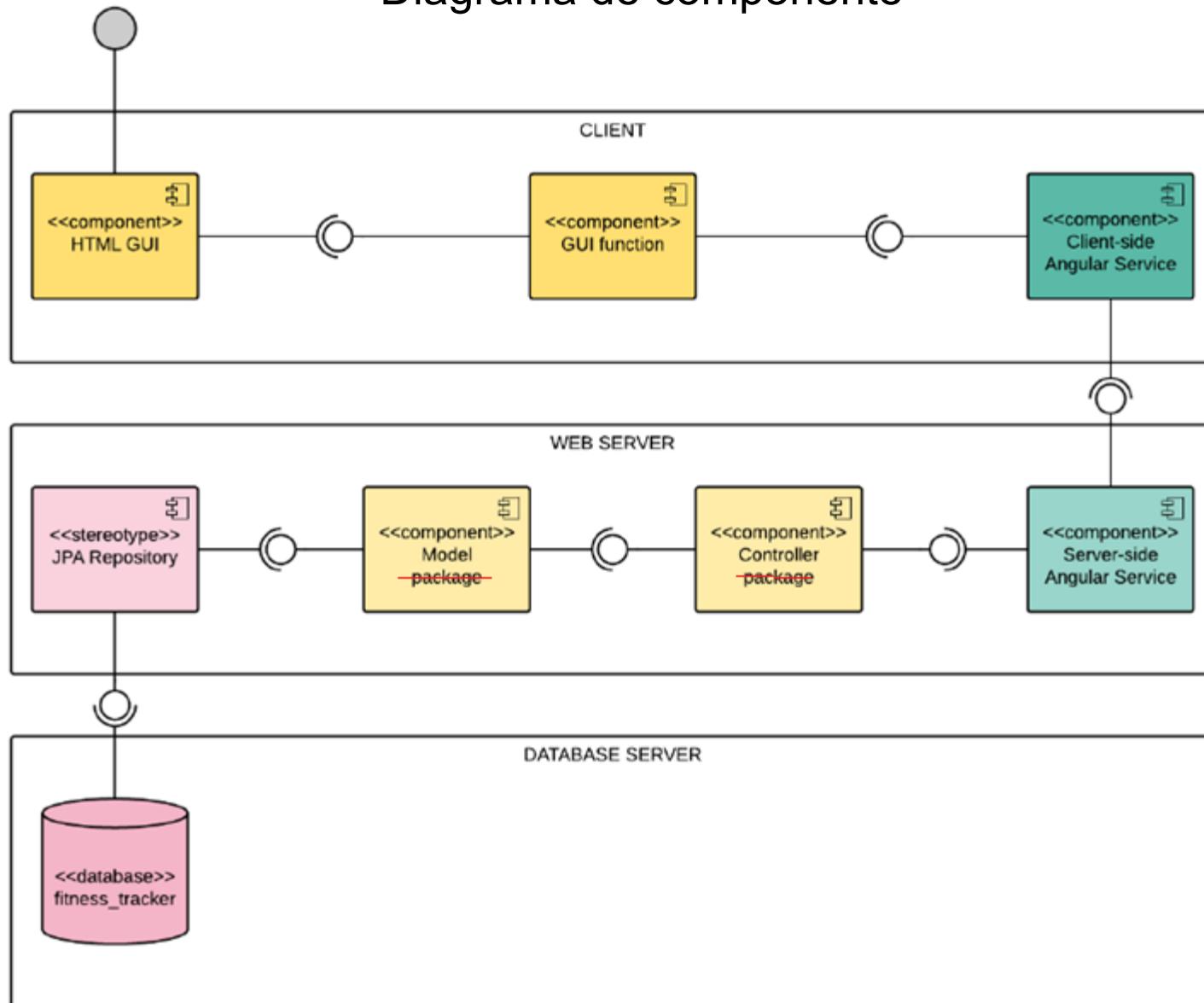


Extras din diagrama de clase a unei aplicații realizată folosind un framework MVC.

PERSPECTIVA INTERNĂ

STRUCTURĂ UNITĂȚI DE EXECUȚIE

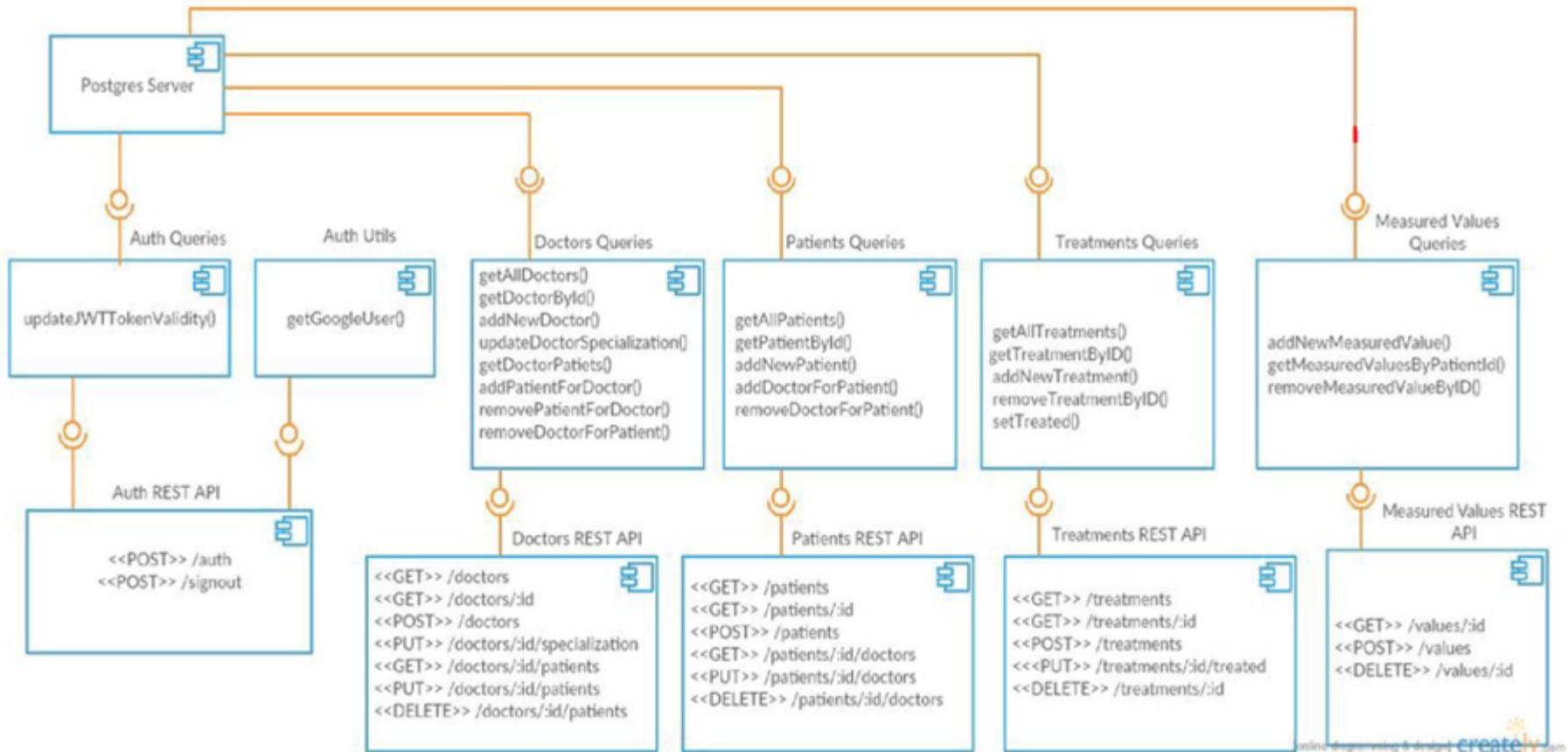
Diagrama de componente



PERSPECTIVA INTERNĂ

STRUCTURĂ UNITĂȚI DE EXECUȚIE

Diagrama de componente



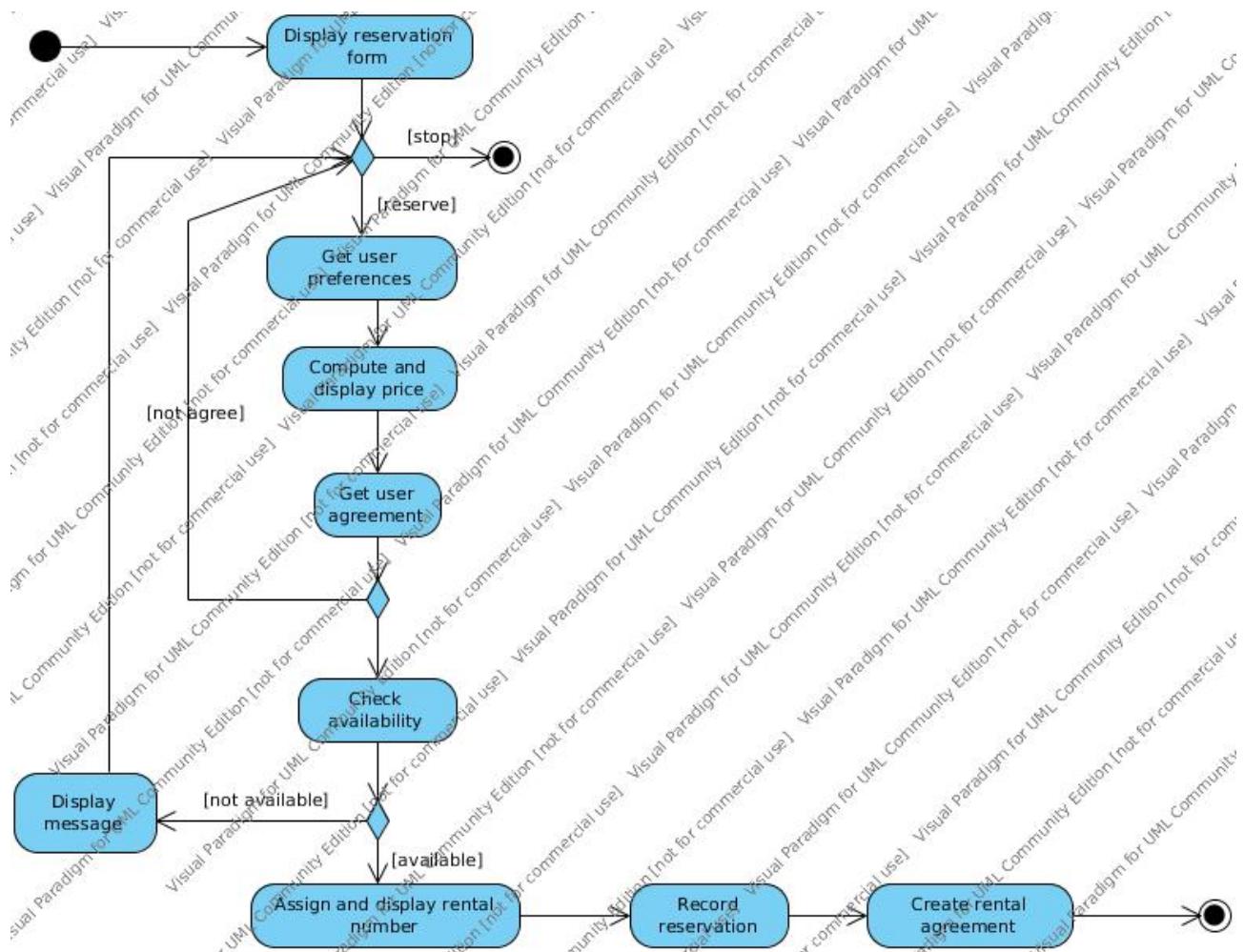
Greșeli de corectat:

1. Conectorii sunt inversați.

PERSPECTIVA INTERNĂ

MODELARE COMPORTAMENT la nivel de UC

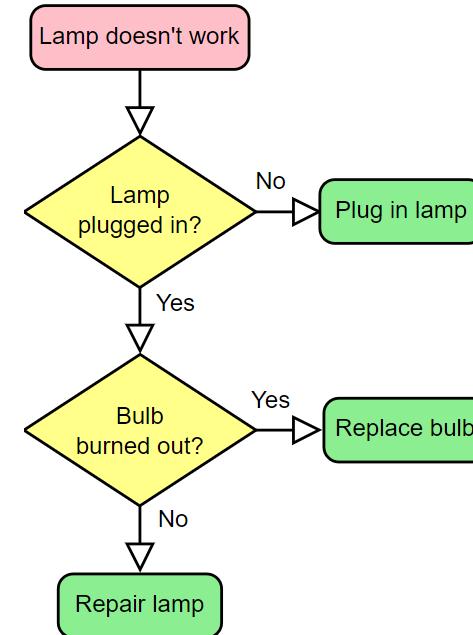
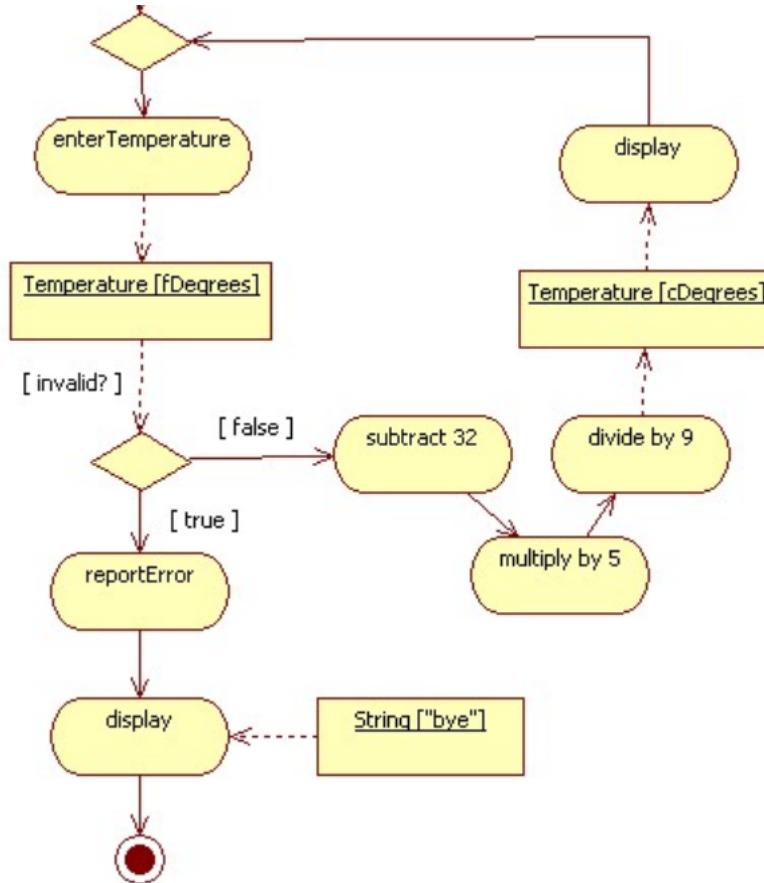
Diagrama de activitate



PERSPECTIVA INTERNĂ

MODELARE ALGORITM (comportament la nivel de funcție)

Diagrama de activitate



PERSPECTIVA INTERNĂ

MODELARE INTERACȚIUNI

Diagrama de secvențe (la nivel caz de utilizare)

Exemplu – extras din diagrama de secvențe.

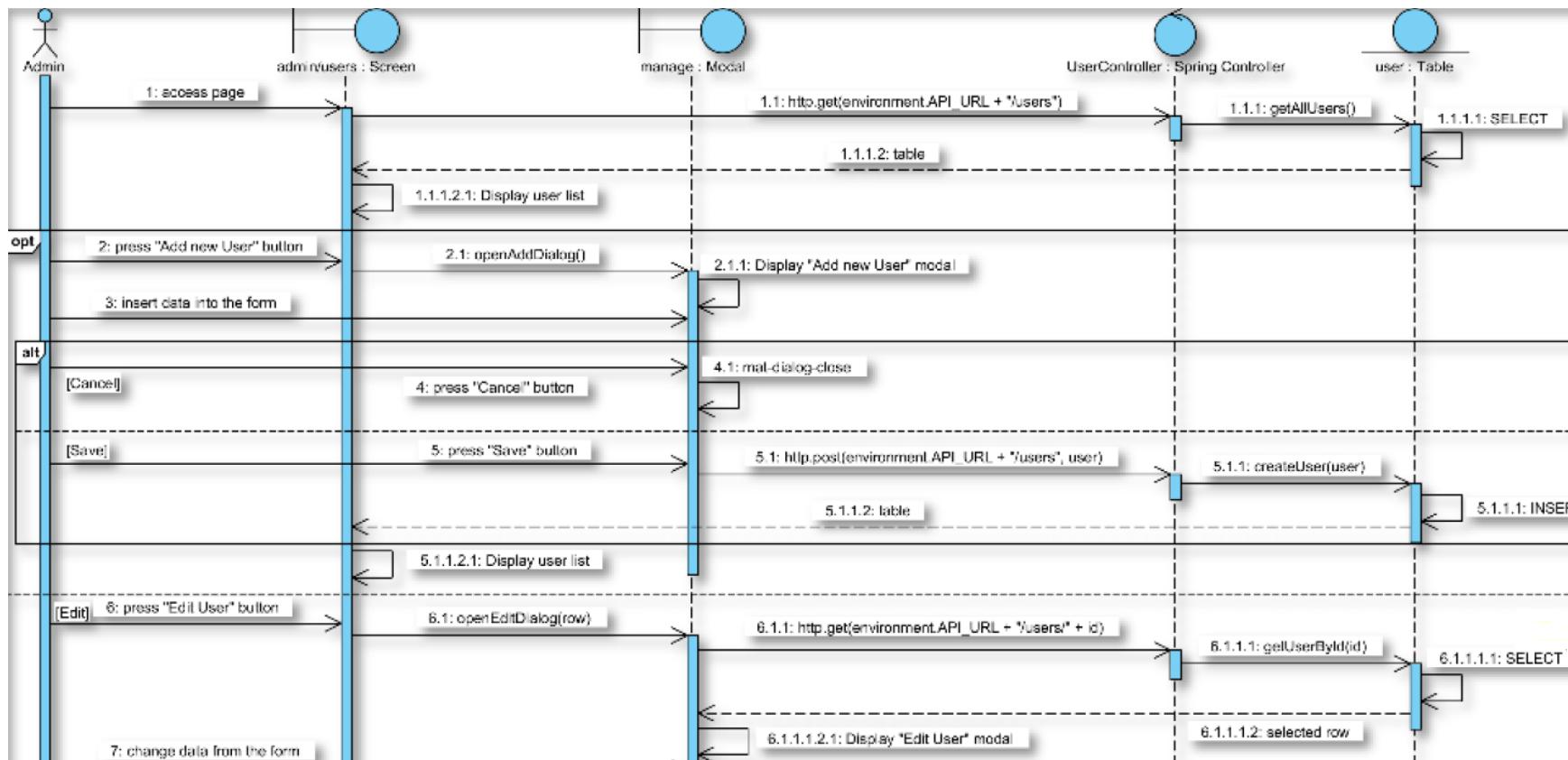
Aplicație dezvoltată pe framework Spring MVC.

UC Name: Manage users

Actor: Admin

Description: Admin manages the users list.

Pre-conditions: The actor must be authenticated as an admin.



etc... (diagrama nu este terminată și nici descrierea cazului de utilizare)

PERSPECTIVA INTERNĂ

MODELARE STĂRI ȘI TRANZIȚII (comportament subsistem)

Diagrama de stări și tranzitii

Exemplu: Subsistem Router

Aplicație de control de la distanță
a temperaturii cu telefon mobil.

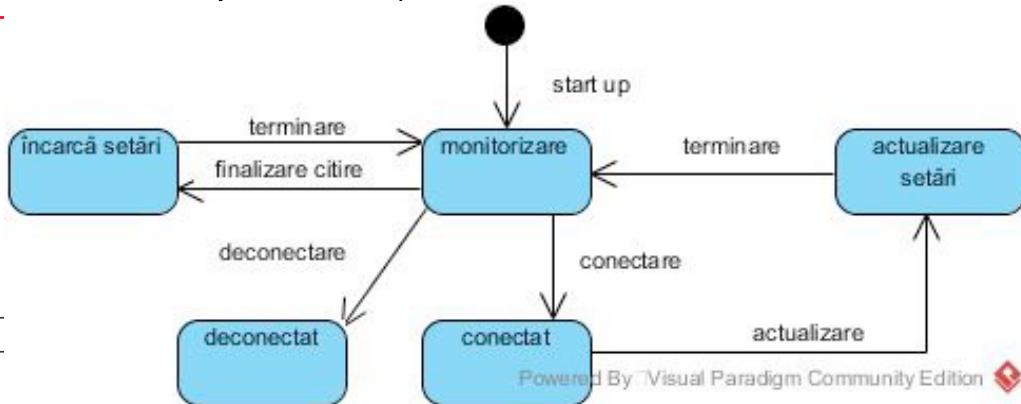


Tabela 4.3: Router- Descrierea stărilor

| Stare | Descriere |
|--------------------|---|
| monitorizare | -stare care se actualizează la fiecare 3 secunde -citire temperatură curentă de la Arduino -determină temperatură dorită -trimite temperatură curentă și cea dorită la toți clienții conectați -actualizare afișare Arduino(trimite date de la Arduino pentru afișare) |
| conectat | -trimite setări către mobil |
| deconectat | -utilizatorul nu este conectat la aplicație de pe telefonul mobil |
| încarcă setări | -citire date din fișier (modul operare, histereză, temperatură mod manual, programul zilnic) |
| actualizare setări | -citire setări noi de la socket(mod operare, histereză, temperatură mod manual, program zilnic) -salvare setări în fișier -trimitere setări către clienții conectați |

Tabela 4.4: Router- descrierea evenimentelor

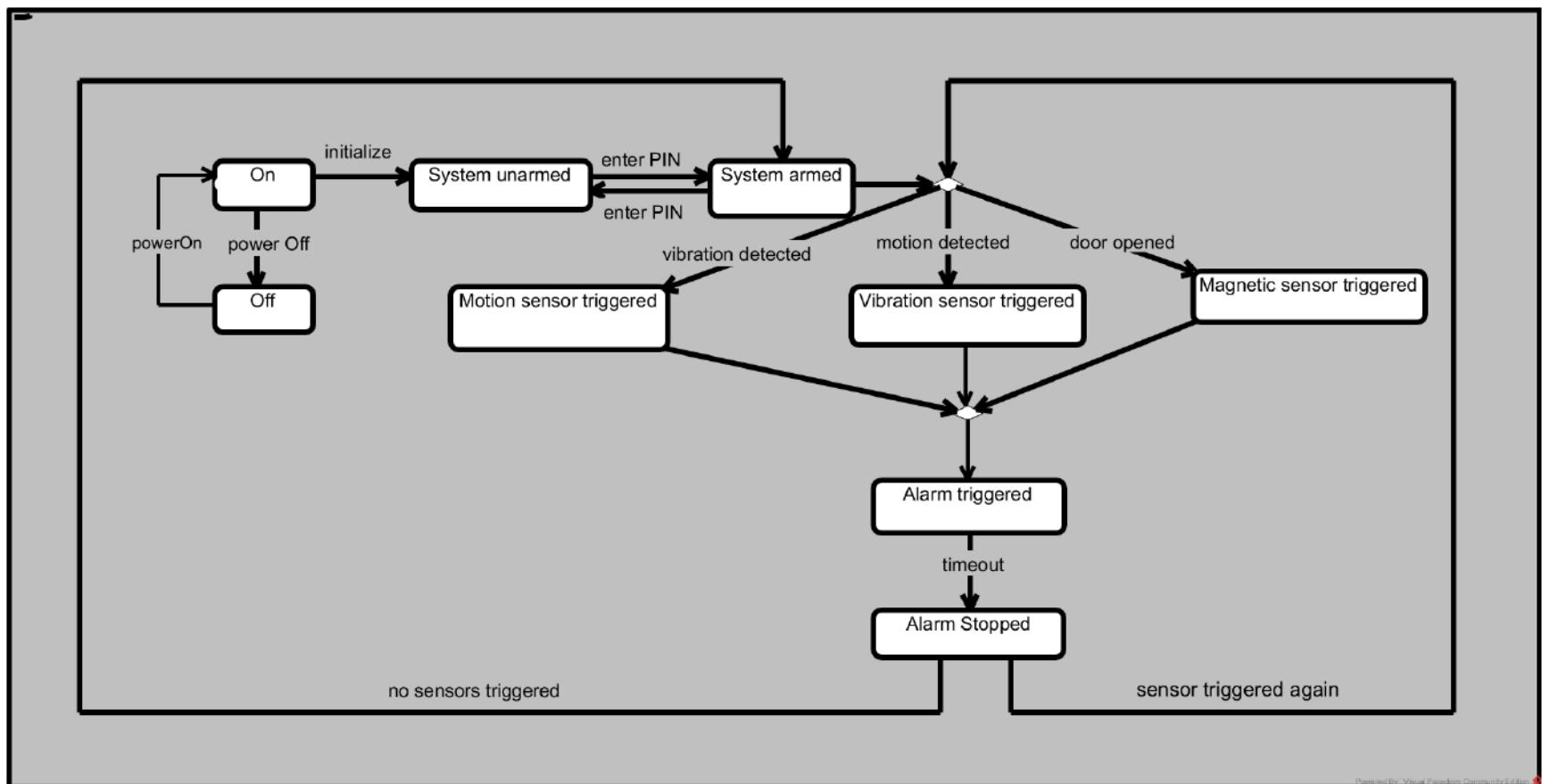
| Eveniment | Descriere |
|-------------------|--|
| start up | -pornirea aplicației server |
| finalizare citire | -setările noi au fost transmise către toți utilizatorii, citirea din fișier se încheie |
| conectare | -se apelează la conectarea unui utilizator |
| deconectare | -se apelează la deconectarea unui utilizator |
| actualizare | -se realizează când utilizatorul trimite noi setări de pe aplicația mobilă |
| terminare | -terminarea modificărilor setărilor după ce noile setări au fost salvare |

PERSPECTIVA INTERNĂ

MODELARE STĂRI ȘI TRANZIȚII (comportament subsistem)

Diagrama de stări și tranzitii

Exemplu: Subsistem de securitate
Sistem de automatizare și control a locuinței.



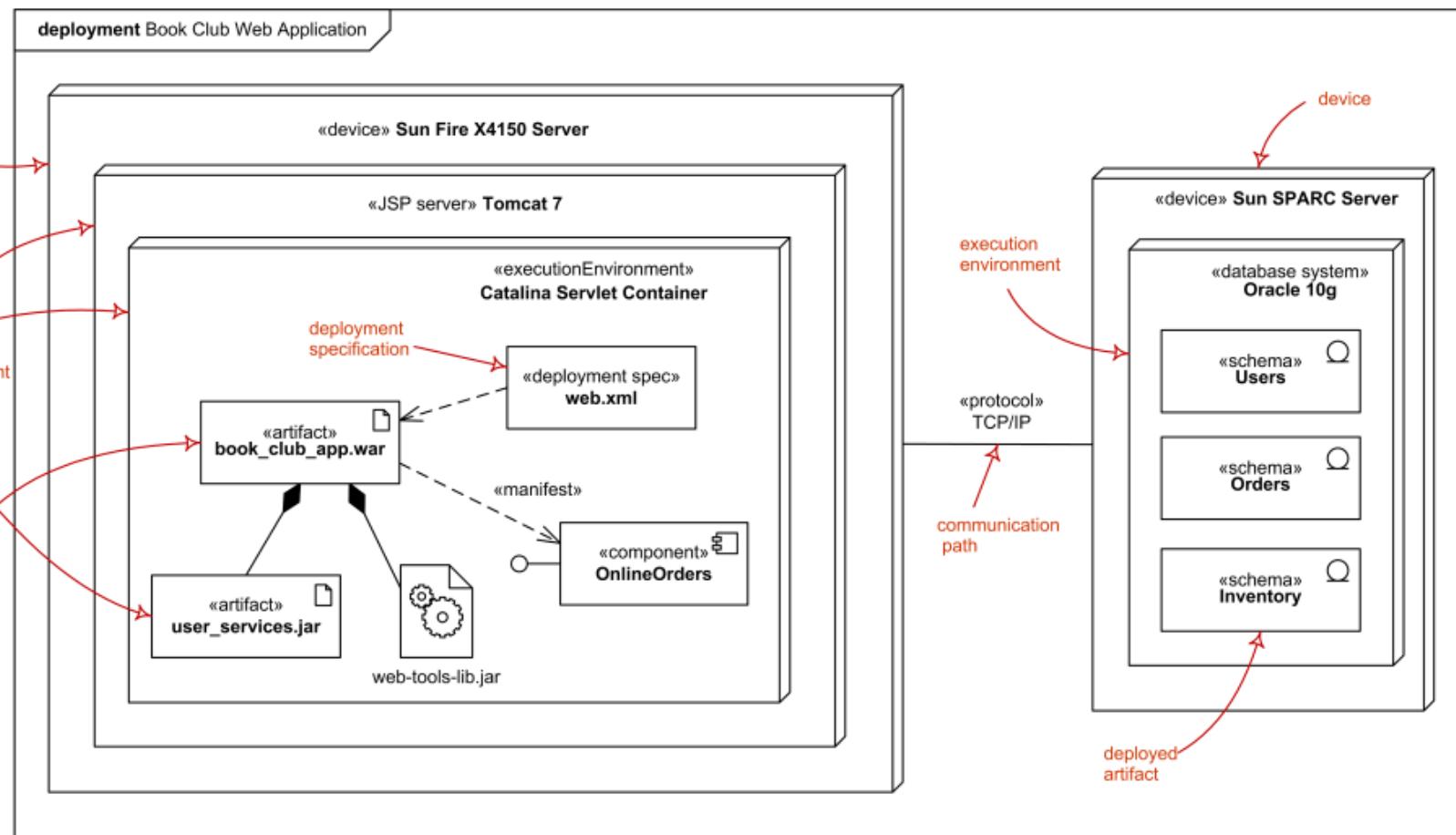
PERSPECTIVA EXTERNĂ

PERSPECTIVA INTERNĂ

PERSPECTIVA INSTALĂRII APLICAȚIEI

PERSPECTIVA INSTALĂRII

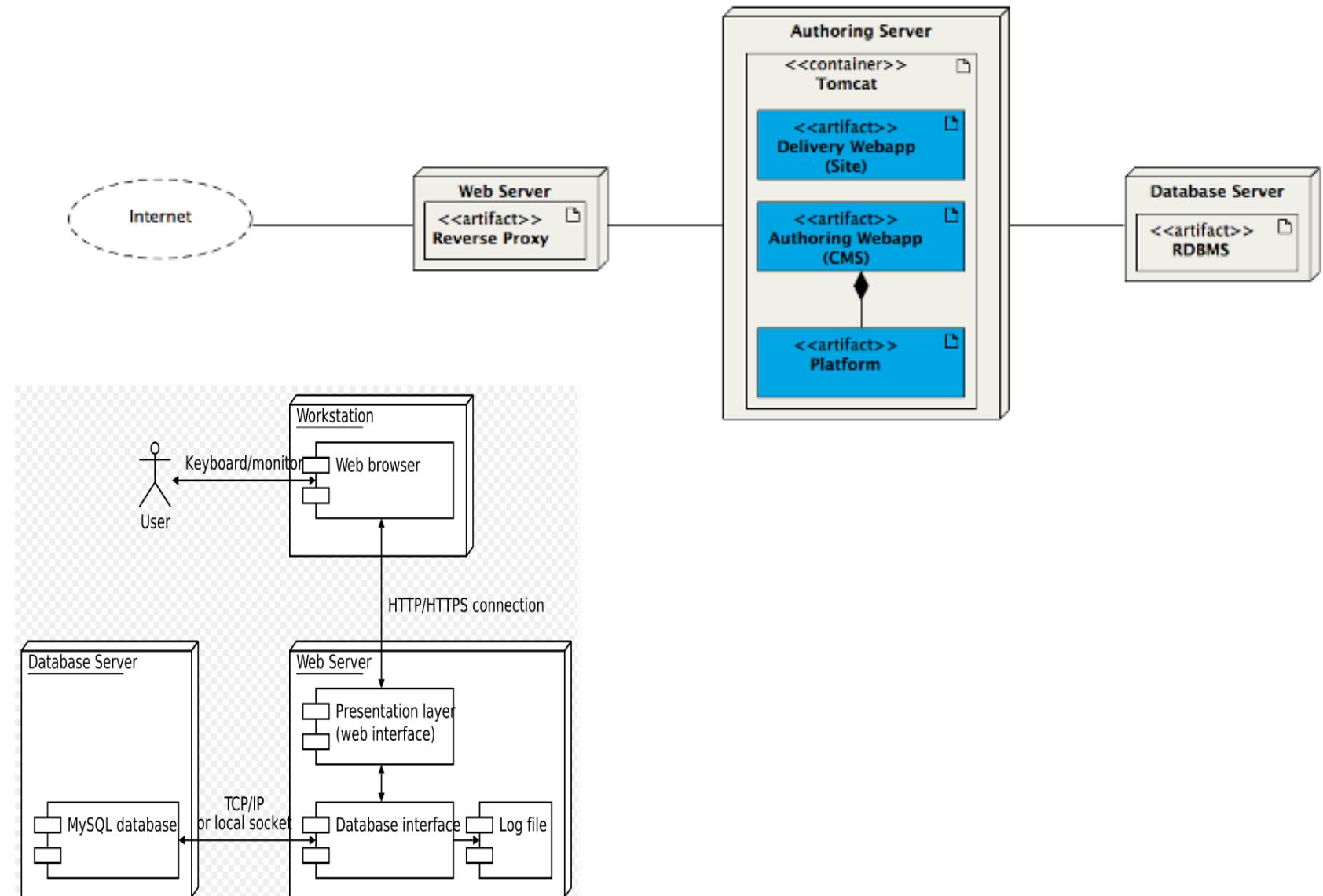
Diagrama de instalare (deployment)



<https://www.uml-diagrams.org/deployment-diagrams-overview.html>

PERSPECTIVA INSTALĂRII

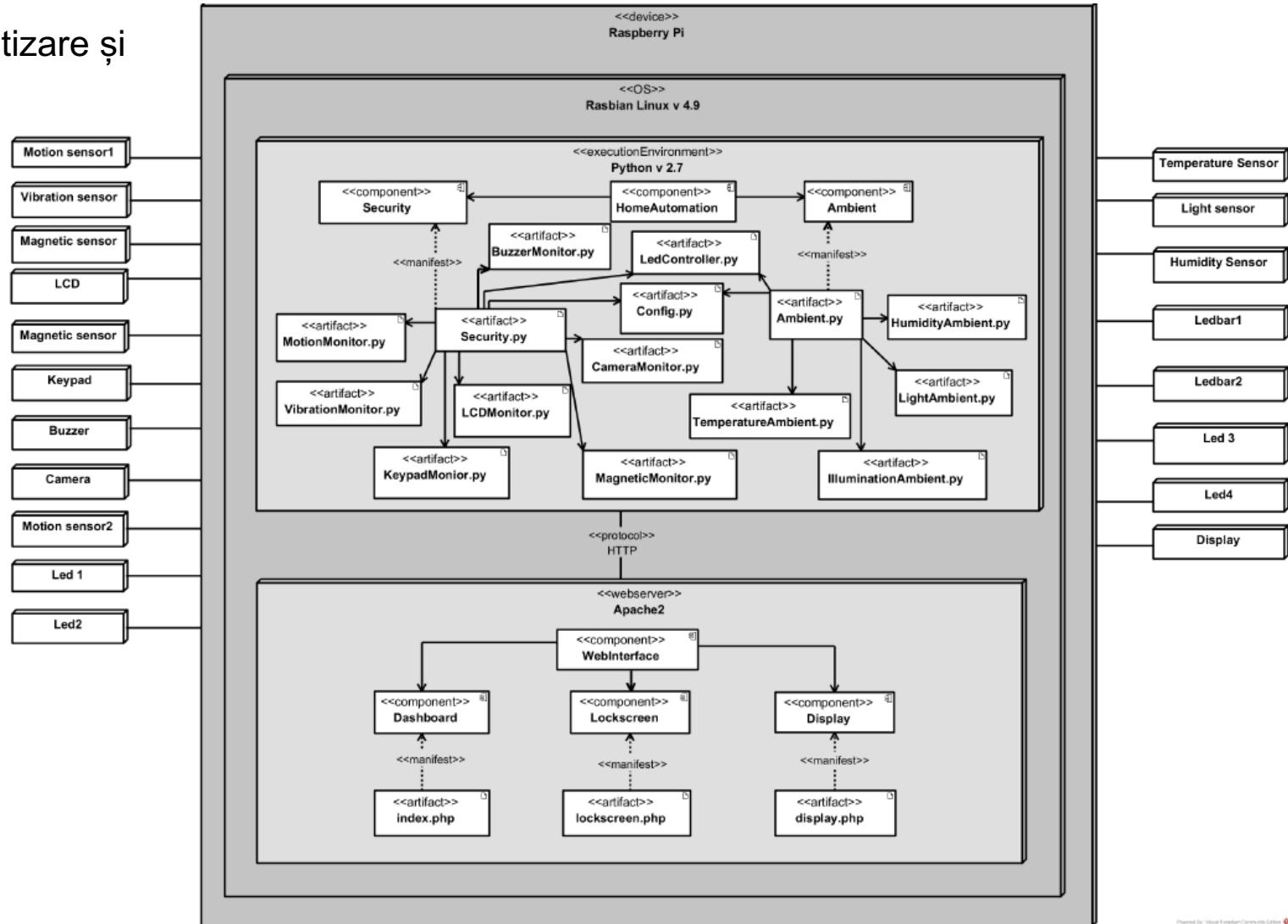
Diagrama de instalare (deployment)



PERSPECTIVA INSTALĂRII

Diagrama de instalare (deployment)

Exemplu:
Sistem de automatizare și
control a locuinței.



Recomandări

În lucrarea de licență ar trebui să existe modele ale aplicației văzută din fiecare perspectivă.

Fiecare lucrare de licență va conține modelele potrivite cu tipul de aplicație.

Acstea vor fi selectate din setul de modele recomandate în continuare.

Se recomandă alegerea modelelor corespunzătoare și folosirea majorității acestora în activitatea de proiectare a aplicației, înainte de dezvoltarea codului.