

Ingineria programării. Proiect: inferența prin enumerare

Nicolae Boca, Ștefan Ignătescu, Gabriel Răileanu
1409A

mai 2020

1 Introducere

Programul de față calculează probabilitățile de apariție ale unor afecțiuni medicale, pe baza datelor deja existente despre anumite comorbidități. Pe baza probabilităților deja cunoscute, se pot defini noduri observate și noduri evidentă.

Proiectul de față își propune respectarea unor standarde în ceea ce privește stilul codului, tratarea excepțiilor, dar și modalitățile în care se realizează testarea unităților.

De asemenea, s-au urmărit toate etapele prezente în dezvoltarea unui produs software (proiectare, implementare, testare și livrarea artefactului software).

2 Diagrame UML

Sunt prezentate diagrame UML pentru: cazuri de utilizare, clase, activități și secvențe.

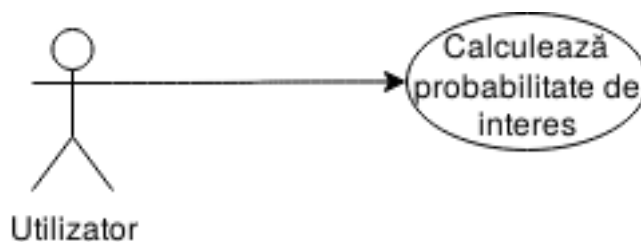


Figura 1: Diagramă de activitate

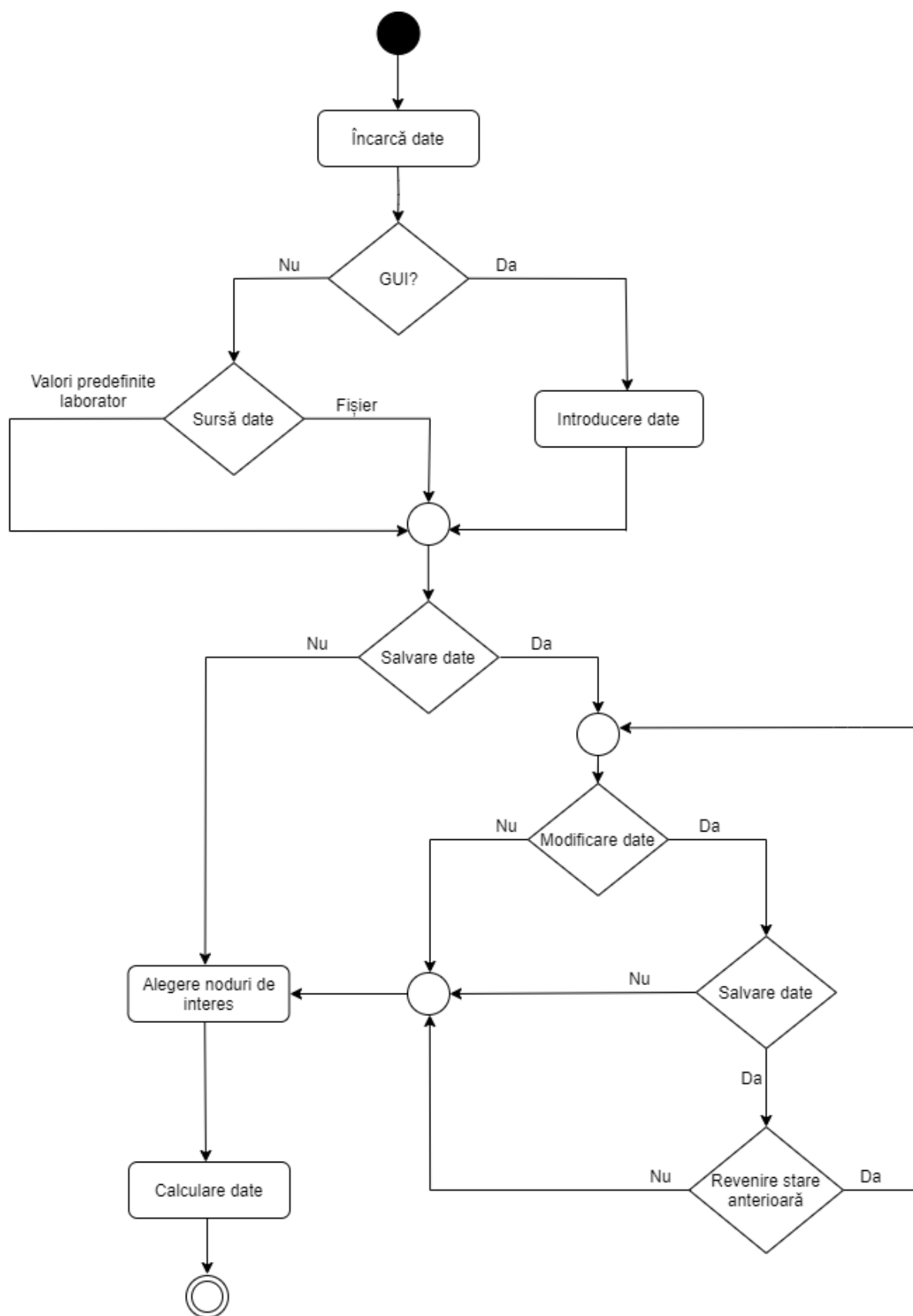


Figura 3: Diagramă de activitate

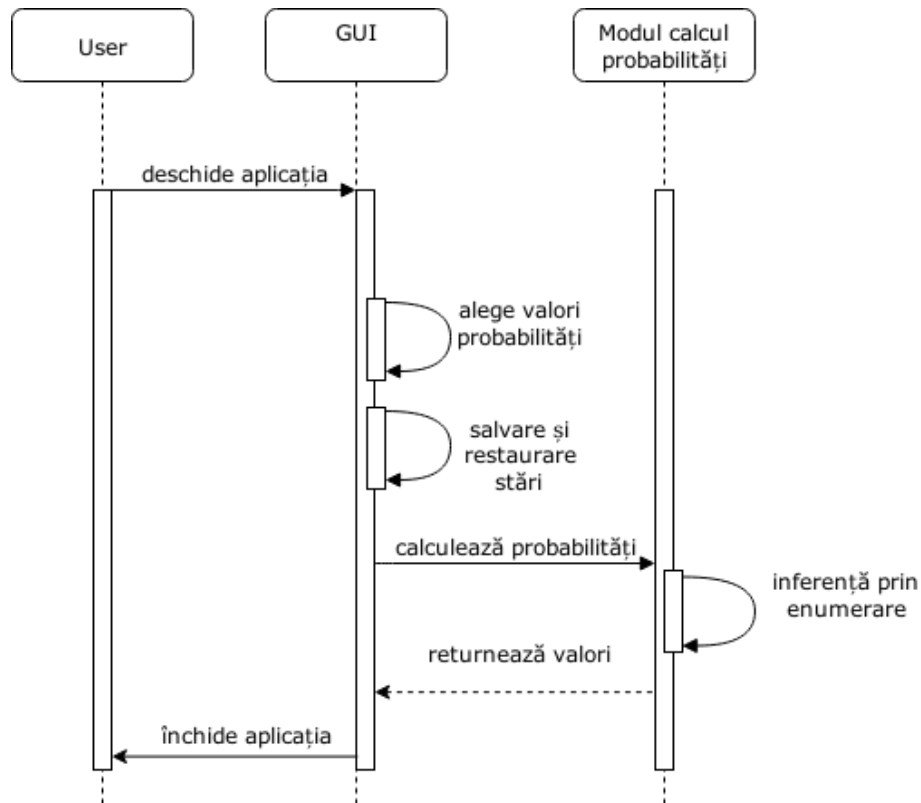


Figura 4: Diagramă de secvențe

3 Modul de utilizare al programului

Aplicația dispune de o interfață grafică din care utilizatorul poate seta valorile probabilităților pentru afecțiunile disponibile. De asemenea, există posibilitatea importării unui set predefinit de date (cele folosite în cadrul laboratorului de inteligență artificială), dar și opțiunea de a încărca datele dintr-un fișier, care trebuie să respecte un anumit format, ilustrat în tabelul 1.

Tabela 1: Exemplu de format pentru probabilități, în cazul unui nod cu doi părinți

first parent	second parent	val($P(x)=true$)
true	true	value
true	false	value
false	true	value
false	false	value

În cazul introducerii valorilor de probabilitate în mod manual, utilizatorul va fi împiedicat să scrie date eronate, iar câmpurile alăturate se vor completa automat pentru a ușura acest proces.

După alegerea valorilor pentru probabilități, utilizatorul poate alege din meniuri de tip radio sau drop-down nodurile de interes (noduri evidențe sau noduri observate).

4 Testarea aplicației

Pentru aplicația dezvoltată au fost testate atât funcționarea în parametri a interfeței grafice, cât și logica pentru calcularea probabilităților condiționate. Aceste cazuri de test sunt prezentate în figura 1.

Nume test	Date intrare	Rezultat așteptat	Rezultat obținut	Observații
ComputeEvidenceNodeProbabilityTest_1	Gripă: da Abces: nu Febră: nod evidență Oboseală: da Anorexie: nespecificat	87.5	87.5	corect
ComputeEvidenceNodeProbabilityTest_2	Gripă: nod evidență Abces: nespecificat Febră: nespecificat Oboseală: nespecificat Anorexie: nespecificat	10	10	corect
ComputeEvidenceNodeProbabilityTest_3	Gripă: nod evidență Abces: da Febră: nu Oboseală: da Anorexie: nu	2.877	2.877	corect
ComputeEvidenceNodeProbabilityTest_4	Gripă: nu Abces: da Febră: nespecificat Oboseală: nod evidență Anorexie: nespecificat	30	30	corect
ComputeEvidenceNodeProbabilityTest_5	Gripă: nod evidență Abces: da Febră: da Oboseală: da Anorexie: da	26.229	26.229	corect
ComputeEvidenceNodeProbabilityTest_6	Gripă: nu Abces: nod evidență Febră: nu Oboseală: nu Anorexie: nu	3.989	3.989	corect
ComputeEvidenceNodeProbabilityTest_7	Gripă: nespecificat Abces: nespecificat Febră: nespecificat Oboseală: nespecificat Anorexie: nod evidență	14.98	14.98	corect
CheckValidNumberTest_1	2	-1	-1	corect
CheckValidNumberTest_2	-2	-1	-1	corect
CheckValidNumberTest_3	~abc!	-1	-1	corect
CheckValidNumberTest_4	0.3d	-1	-1	corect
CheckValidNumberTest_5	0.5	0.5	0.5	corect
MementoTest_1	variabila = 1 SaveMemento() variabila = 10	10	10	corect
MementoTest_2	variabila = 1 SaveMemento() variabila = 10 RestoreMemento()	1	1	corect
MementoTest_3	variabila = 1 RestoreMemento()	Exception	Exception	corect
ComputeProbabilityConsideringParentsTest_1	nod: Gripă status: nespecificat	-1	-1	corect
ComputeProbabilityConsideringParentsTest_2	nod: Gripă status: true	0.1	0.1	corect
ComputeProbabilityConsideringParentsTest_3	nod: Gripă status: false	0.9	0.9	corect
ComputeProbabilityConsideringParentsTest_4	nod: Febră, status: true părinte1: Gripă, status: true părinte2: Abces, status: false	0.7	0.7	corect
ComputeProbabilityConsideringParentsTest_5	nod: Febră, status: false părinte1: Gripă, status: false părinte2: Abces, status: true	0.75	0.75	corect
ComputeProbabilityConsideringParentsTest_6	nod: Oboseală, status: true părinte: Febră, status: true	0.6	0.6	corect
ToDigitTest_1	true	1	1	corect
ToDigitTest_2	false	0	0	corect

Figura 5: Cazurile de test

5 Capturi de ecran

The screenshot shows a window titled "Bayesian Network Interface". It contains several input fields and buttons for configuring a Bayesian network query.

Variabila de interogare: Oboseala

Salvare stare (button)

Revenire stare anterioara (button)

Variabile observate:

- Gripă: ☐ Da ☐ Nu ☒ Necunoscut
- Abces: ☐ Da ☐ Nu ☒ Necunoscut
- Febră: ☐ Da ☐ Nu ☒ Necunoscut
- Oboseală: ☐ Da ☐ Nu ☒ Necunoscut
- Anorexie: ☐ Da ☒ Nu ☐ Necunoscut

Calculază (button) **Valori laborator** (button)

Help (button) **Incarca date** (button)

Rezultat

Variabila de interogare: Oboseala
Probabilitate: 22.9287226534933%

Conditional Probability Tables:

P (Gripă = Da)		P (Gripă = Nu)	
	0.1		0.9

P (Abces = Da)		P (Abces = Nu)	
	0.05		0.95

Gripă	Abces	P (Febră = Da)	P (Febră = Nu)
Da	Da	0.05	0.95
Da	Nu	0.25	0.75
Nu	Da	0.7	0.3
Nu	Nu	0.8	0.2

Febră	P (Oboselă = Da)	P (Oboselă = Nu)
Da	0.2	0.8
Nu	0.6	0.4

Febră	P (Anorexie = Da)	P (Anorexie = Nu)
Da	0.1	0.9
Nu	0.5	0.5

Figura 6: Interfața aplicației

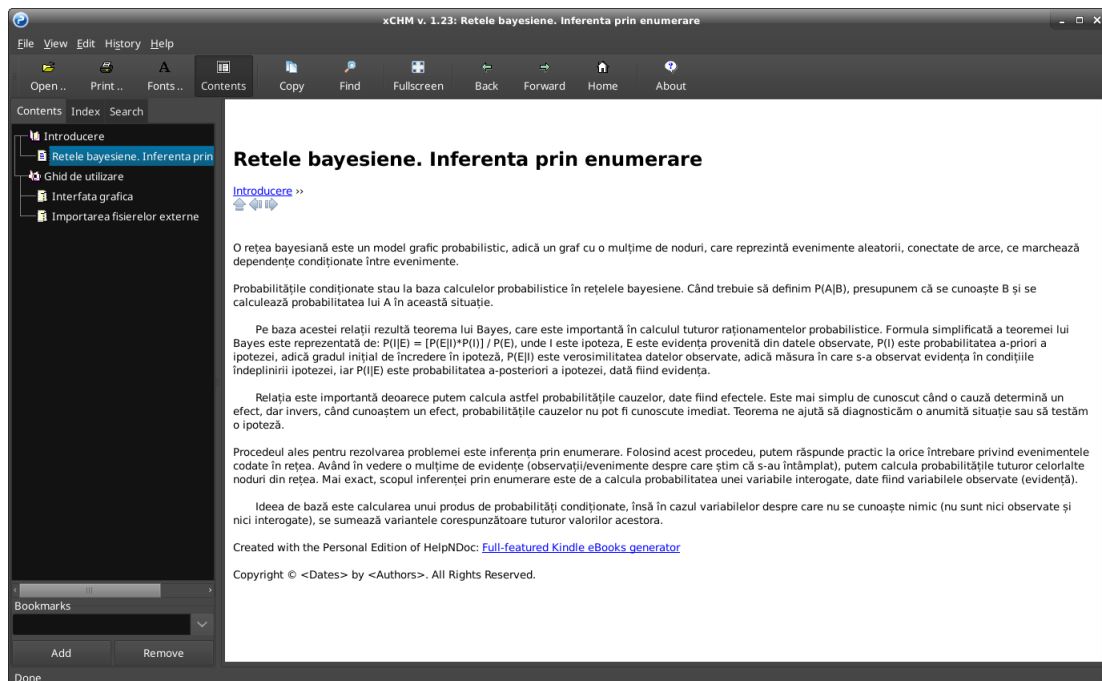


Figura 7: Captură de ecran din meniul de *help* al aplicației

6 Contribuții individuale

- **Nicolae Boca:** creare meniu *help*, design cazuri de test și testarea unităților
- **Ștefan Ignătescu:** refactorizare, implementare șablon de proiectare, respectarea stilului de codare
- **Gabriel Răileanu:** documentație, document SRS, coordonare echipă

DOCUMENT DE SPECIFICARE A CERINȚELOR SOFTWARE

pentru

Proiect - inferența prin enumerare

Versiunea 1.0

Autori: N. Boca, Ș. Ignătescu, G. Răileanu

Universitatea Tehnică „Gheorghe Asachi” Iași

26 mai 2020

Cuprins

1	Introducere	3
1.1	Scopul documentului	3
1.2	Convențiile documentului	3
1.3	Publicul destinat	3
1.4	Scopul produsului	3
1.5	Referințe	3
2	Descrierea generală	4
2.1	Perspectivile produsului	4
2.2	Funcțiile produsului	4
2.3	Clase de utilizatori și caracteristici	4
2.4	Mediul de utilizare	4
2.5	Proiectarea și implementarea constrângerilor	4
2.6	Documentația utilizatorului	5
2.7	Ipoteze și dependențe	5
3	Cerințe externe de interfață	6
3.1	Interfața cu utilizatorul	6
3.2	Interfața hardware	7
3.3	Interfața software	7
3.4	Interfața de comunicare	7
4	Cerințe funcționale	8
4.1	Cerințe de performanță	8
4.1.1	Cerințe de fiabilitate	8
4.1.2	Cerințe de securitate	8

1 Introducere

1.1 Scopul documentului

Scopul acestui document este descrierea cu exactitate a funcționalităților pe care produsul software *Bayesian Network Medical Application* le oferă utilizatorilor săi finali, specificând cerințele funcționale și non-funcționale pe care aplicația le implementează.

1.2 Convențiile documentului

Documentul urmărește formatarea standard IEEE pentru dezvoltarea de software. Standardul definește o formatare obișnuită a documentului care urmează, incluzând scrierea la persoana a III-a, vocea pasivă, precum și text coerent și corect din punct de vedere gramatical.

1.3 Publicul destinat

Documentul este destinat publicului tehnic, care deține cunoștințe de bază ale unor limbaje de programare orientate-obiect, alături de noțiuni elementare de statistică, necesare pentru înțelegerea problematicei expuse.

1.4 Scopul produsului

Bayesian Network Medical Application este o aplicație dezvoltată în limbajul C#. Cu ajutorul aplicației se poate determina probabilitatea existenței unei anumite afecțiuni cunoscând simptomele pe care aceasta le manifestă.

1.5 Referințe

Programul descris a avut ca referință aplicația din laboratorul numărul 11 cu tema „Rețele bayesiene” a cursului de Inteligență Artificială (prof. Florin Leon, Facultatea de Automatică și Calculatoare din Iași). De asemenea, implementarea are la bază limbajul C#, dezvoltat de Microsoft.

2 Descrierea generală

2.1 Perspectivile produsului

Deși există numeroase sisteme expert sau motoare de inferență care pot calcula probabilități dându-se date cunoscute, aplicația de față se folosește în calcul de metoda inferenței prin enumerare, caracteristică rețelelor bayesiene. Faptul că astfel de rețele sunt deseori reprezentate prin intermediul grafurilor ajută în procesul de înțelegere, acest lucru constituind un avantaj față de alte metode existente.

2.2 Funcțiile produsului

- încărcarea valorilor probabilistice din interfață sau dintr-o sursă externă
- calcul de probabilități folosind noduri observate și noduri evidență

2.3 Clase de utilizatori și caracteristici

Întrucât aplicația este una de uz general, nu există o categorie de utilizatori aparte. Ea este accesibilă oricui și poate fi folosită de un public larg.

2.4 Mediul de utilizare

Pentru executarea aplicației este nevoie de un sistem de operare Windows. Ca alternativă, pentru utilizatorii sistemelor UNIX/Linux se pot folosi soluții precum *Wine* sau *Mono*, care permit rularea aplicației. Pentru dezvoltarea altor funcționalități, se recomandă medii de dezvoltare precum Visual Studio, respectiv JetBrains Rider pentru sisteme Linux.

2.5 Proiectarea și implementarea constrângerilor

Aplicația dispune de o licențiere GNU GPL, dând dreptul oricărui utilizator de a copia, modifica și redistribui programe și coduri sursă ale programatorilor care își licențiază munca sub tutela GPL.

2.6 Documentația utilizatorului

Utilizatorul poate folosi ghidul aplicației, pus la dispoziție de către dezvoltatori în format **.chm** odată cu executabilul/codul sursă. În plus, există un meniu contextual care poate fi accesat de utilizatori în cadrul aplicației.

2.7 Ipoteze și dependențe

Ipotezele de la care pornește aplicația sunt factorii externi de care nu depinde în mod direct aplicația, cum ar fi funcționarea în parametri a calculatorului, inexistența unor erori majore în operațiile de calcul în virgulă mobilă etc.

3 Cerințe externe de interfață

3.1 Interfața cu utilizatorul

Pentru introducerea ușoară a datelor de intrare și pentru a vedea rezultatul final, aplicația dispune de o interfață grafică, prezentată în figura de mai jos:

The screenshot shows a software window titled "Bayesian Network Interface". It contains several input sections and a results area.

Variabila de interogare: A dropdown menu with "Oboseala" selected.

Variabile observate: Five rows of radio buttons for "Gripă", "Abces", "Febră", "Oboselă", and "Anorexie". Each row has three options: "Da", "Nu", and "Necunoscut".

Buttons: "Salvare stare", "Revenire stare anterioara", "Calculează", "Valori laborator", "Help", and "Incarca date".

Probability Tables:

- Gripă:**

	P (Gripă = Da)	P (Gripă = Nu)
	0.1	0.9
- Abces:**

	P (Abces = Da)	P (Abces = Nu)
	0.05	0.95
- Febră (contingency table):**

Gripă \ Abces	P (Febră = Da)	P (Febră = Nu)
Da Da	0.05	0.95
Da Nu	0.25	0.75
Nu Da	0.7	0.3
Nu Nu	0.8	0.2
- Febră (contingency table):**

Febră	P (Oboselă = Da)	P (Oboselă = Nu)
Da	0.2	0.8
Nu	0.6	0.4
- Febră (contingency table):**

Febră	P (Anorexie = Da)	P (Anorexie = Nu)
Da	0.1	0.9
Nu	0.5	0.5

Rezultat: A text box showing "Variabila de interogare: Oboseala" and "Probabilitate: 22.9287226534933%".

Figura 3.1: Interfața aplicației

Această interfață pune la dispoziția utilizatorului posibilitatea de a selecta variabila de interogare dintr-o listă de simptome și variabilele de evidență împreună cu datele cunoscute despre apariția acestora.

Probabilitățile de apariție ale simptomelor pot fi introduse atât manual prin câmpurile de text marcate, cât și încărcate din fișiere externe. După apăsarea butonului *Calculează*,

programul analizează toate datele primite și afișează rezultatul în câmpul de text din stânga jos.

Programul pune la dispoziție valorile probabilităților utilizate în laborator, iar la apăsarea butonului *Valori laborator* acestea vor popula câmpurile de text. Pentru a importa date din alte fișiere externe, acestea trebuie să aibă denumirea de forma *simptom.txt* și să conțină lista de probabilități ale cazurilor de adevăr. Fișierele în formatul specificat anterior vor fi salvate în folderul *probabilities/date* din cadrul soluției, iar datele vor fi încărcate în program la apăsarea butonului *Încarcă date*.

În cazul introducerii valorilor de probabilitate în mod manual, utilizatorul va fi împiedicat să scrie date eronate, iar câmpurile alăturate se vor completa automat pentru a ușura acest proces.

3.2 Interfața hardware

Fiind un produs software, acesta nu dispune de o interfață hardware dedicată.

3.3 Interfața software

Aplicația de față nu necesită o conexiune cu baza de date, toate datele necesare fiind cele inițiale. Acestea sunt stocate în fișiere, neexistând o justificare pentru a folosi o bază de date. Nu există o comunicare a aplicației cu alte componente ale sistemului de operare/elemente hardware.

3.4 Interfața de comunicare

Aplicația este livrată ca software independent și nu necesită alte componente pentru comunicare, fiind gândită să fie *self-contained*.

4 Cerințe funcționale

4.1 Cerințe de performanță

Aplicația va oferi răspunsul cerut de utilizator în timp real și o aproximare cât mai exactă, aceasta depinzând de precizia calculatorului pe care este rulată.

4.1.1 Cerințe de fiabilitate

Întrucât aplicația este una software, fiabilitatea ei este determinată de cerințele sistemului pe care rulează. Astfel, pot apărea erori la migrarea aplicației pe o altă platformă cu o configurație diferită, deși acest lucru este puțin probabil în cazul migrării către configurații cu aceeași familie de sisteme de operare.

4.1.2 Cerințe de securitate

Sistemul nu necesită măsuri de securitate, altele în afara celor recomandate pentru sistemul dumneavoastră de operare, neavând acces la componente esențiale din nucleul sistemului de operare și nici la alte fișiere din sistem.