|  |  |
| --- | --- |
| Autor(i) : Oscar Gal  Maria Florică  Doriana Mureșan | Referinta : Caiet de sarcini |
| Statut : Valid | Versiune : B |
| Data : 13/12/2020 | Modificari : Definire task-uri |

|  |
| --- |
| Caiet de sarcini |

CUVINTE CHEIE

UML, Diagrame UML, StarUML.

|  |  |  |  |
| --- | --- | --- | --- |
| **Istoricul actiunilor efecutate asupra documentului** | | | |
| **Nume autor** | **Data** | **Versiune** | **Actiune(i)** |
| Oscar Gal | 05/11/2020 | A | Creare |
| Doriana Muresan  Maria Florica  Toti | 10/11/2020  12/11/2020  18/11/2020 | A  A  A | Inserare descriere  Inserare descriere  Recitire si corectare |
| Toti  Doriana Muresan  Toti  Gal Oscar | 23/11/2020  4/12/2020  13/12/2020  30/12/2020 | A  B  B  B | Finalizare & Editare  Adaugare modificari  Corectare & Finalizare  Adaugare diagrame lipsa + adaugare generare cod |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Cuprins |

[1. Cerinte 4](#_Toc60219538)

[1.1. Context 4](#_Toc60219539)

[1.2. Obiectivele generale ale proiectului 4](#_Toc60219540)

[1.3. Discutie pe baza cerintelor 4](#_Toc60219541)

[1.3.1. Rezumat al exigentelor 4](#_Toc60219542)

[2. Propunere tehnica 5](#_Toc60219543)

[2.1. Diagrame UML 5](#_Toc60219544)

[2.1.1. Diagrama Bazei de Date 5](#_Toc60219545)

[2.1.2. Diagrama Use Case 6](#_Toc60219546)

[2.1.3. Diagrama de clasa 8](#_Toc60219547)

[2.1.4. Diagrama de secvența 9](#_Toc60219548)

[2.2. Interfața grafica 15](#_Toc60219549)

[2.2.1. Portabilitatea aplicatiei 18](#_Toc60219550)

[2.3. Descriere tehnica 18](#_Toc60219551)

[2.3.1. Numele aplicatiei 18](#_Toc60219552)

[2.3.2. Limbaje si unelte de programare 18](#_Toc60219553)

[2.4. Planul propus (general) 18](#_Toc60219554)

[2.5. Plan de calitate 19](#_Toc60219555)

[2.6. Plan de management 19](#_Toc60219556)

[2.6.1. Organizarea proiectului si a echipei de proiect 19](#_Toc60219557)

[2.6.2. Gestiunea resurselor 19](#_Toc60219558)

[2.6.3. Contact client 19](#_Toc60219559)

[3. De furnizat la client 19](#_Toc60219560)

# Cerinte

## Context

Cerința proiectului nostru a fost să descriem proiecte software folosind diagrame UML (reverse engineering). Însă pentru realizarea acestor diagrame an ales utilizarea limbajului UML (Unified Modeling Language) și a aplicației bloodDonation ca suport.

## Obiectivele generale ale proiectului

Obiectivul proiectului este de a realiza o descriere a aplicației bloodDonation prin intermediul diagramelor UML, mai exact prin crearea celor mai sugestive diagrame. Aceste diagrame vor fi folosite mai apoi pentru a putea genera cod, putând fi ales orice limbaj de programare pentru a fi generat.

## Discutie pe baza cerintelor

Obiectivul proiectului este de a realiza o descriere a aplicației bloodDonation prin intermediul diagramelor UML. În modelarea acestei aplicații este utilizat limbajul UML (Unified Modeling Language), care ne permite crearea următoarelor tipuri de diagrame :

* Diagrama use-case
* Diagrama de clase
* Diagrama de baza de date
* Diagrama de secvență

Aceste diagrame vor fi mai apoi folosite pentru a genera cod. Codul generat va fi utilizat in dezvoltarea aplicației.

### Rezumat al exigentelor

In partea de dezvoltare, va trebui sa luam in evidenta următoarele aspecte: aplicația trebuie sa fie una ușor scalabila, iar de asemenea aceasta trebuie sa fie independenta de tehnologia si limbajul de programare folosit. Limbajul de programare va putea fi ales mai târziu, deoarece utilizând diagrame ULM pentru a genera cod, nu depindem de un limbaj de programare.

Ca si funcționalitate aplicația va fi una ușor de folosit, structurata si cat mai intuitiva. Aplicația fiind una care trebuie folosite de persoane de toate vârstele si din toate mediile, va avea un design ușor de folosit, fiind o aplicație intuitiva

# Propunere tehnica

Am împărțit aceasta secțiune in doua mari subsecțiuni. Prima se va ocupa de prezentarea diagramelor UML a aplicație, iar in partea a doua se va prezenta manualul de utilizare a aplicației.

## Diagrame UML

### Diagrama Bazei de Date

Aceste tipuri de diagrame structurate sunt utilizate in vederea conceperii bazei de date. Acestea conțin diferite simboluri si conectori pentru vizualizarea a doua informații importante: **entitățile din scopul sistemului** si **relația intre aceste entități.**

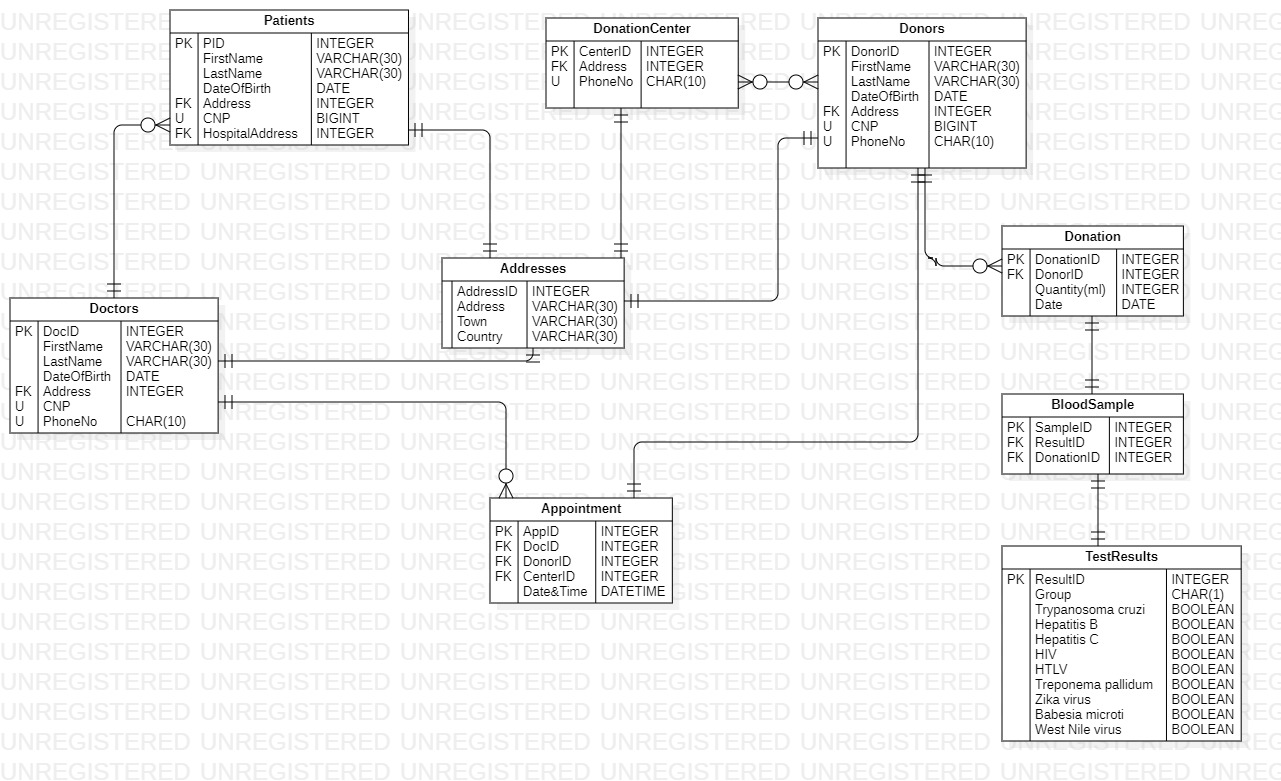


Figura 1: Diagrama bazei de date

Entitățile sunt lucruri sau concepte dintr-un sistem, care pot fi împărțite in diferite categorii, cum ar fi: persoane - *Patients, Doctors, Donors*; obiecte – *BloodSample;* concepte: *Appointment, Donation*.

O entitate este reprezentata in acest tip de diagrama sub forma unui dreptunghi, aceasta fiind împărțita in 2 părți, cea sus unde se va pune numele entității, iar in cea de jos unde sunt coloane care reprezintă atributele acestor entități. Pe lângă înșirarea atributelor in cadrul acestei diagrame trebuie de asemenea sa specificam si tipul acestui atribut si daca este cheie primara sau cheie secundara.

O realtie intre doua entitati inseamna ca cele doua entitati sunt asociate unele cu altele cumva. De exemplu, un doctor este desemnat uneia sau mai multor programari. Entitatea *Doctors* este legata de *Appointment* print-o cheiestraina *DocID* trecuta in tabela *Appointment* ca si atribut.

### Diagrama Use Case

Diagrama use-case, ilustrată în Figura2, reprezintă forma primară de reprezentare a unui produs software care este în curs de dezvoltare. Aceasta prezintă o colecție de cazuri de utilizare și actori (entități sau utilizatori) care:

* oferă o descriere generală a modului în care va fi utilizat sistemul
* furnizează o privire de ansamblu a funcționalităților ce se doresc a fi oferite de sistem
* arată cum interacționează sistemul cu unul sau mai mulți actori
* asigură faptul că sistemul va produce ceea ce s-a dorit.

Diagram

Description automatically generated

Figura 2: Diagrama Use-Case valabila pentru toate tipurile de utilizatori

Diagram

Description automatically generated

Figura 3: Diagrama Use-Case valabila pentru un utilizator Donator

Diagram

Description automatically generated

Figura 4: Diagrama Use-Case valabila pentru un utilizator Doctor

Diagram, text

Description automatically generated

Figura 5: Diagrama Use-Case valabila pentru un utilizator din cadrul personalului centrului de donare

### Diagrama de clasa

Modelarea unui sistem software presupune identificarea elementelor importante din punctul de vedere al celui care modelează, elemente care au o mulțime de proprietăți.

Diagrama de clase este folosită pentru a modela structura(viziunea statică asupra) unui sistem, astfel că aceasta conține clase/interfețe, obiecte și relații care se stabilesc între acestea.

Diagram

Description automatically generated

Figura 6: Diagrama de clasa valabila pentru elementele din domeniu

Diagram

Description automatically generated

Figura 6: Diagrama de clasa valabila pentru elementele din controller si repository

### Diagrama de secvența

O diagrama UML de secvența arata modul in care un set de obiecte interacționează unul cu altul intr-un proces in timp. In cadrul acestei diagrame se afișează mesajele care trec intre participanți si obiectele dintr-un sistem si ordinea in care apar.

Mai jos sunt reprezentate diagramele de secventa pentru doctor si pentru personalul din cadrul centrului de donare de sange.

Chart, box and whisker chart

Description automatically generated

Figura 7: Inregistrarea doctorului

A picture containing diagram

Description automatically generated

Figura 8: Log-in doctor

Calendar

Description automatically generated with medium confidence

Figura 8: Meniul principal al doctorului

Diagram

Description automatically generated

Figura 9: Log out doctor

Diagram

Description automatically generated with medium confidence

Figura 10: Inregistrarea donator

A picture containing box and whisker chart

Description automatically generated

Figura 11: Log in donator

Diagram

Description automatically generated

Figura 12: Donare sange donator

Table

Description automatically generated

Figura 13: Meniu donator

Diagram

Description automatically generated

Figura 14: Log out donator

## Interfața grafica

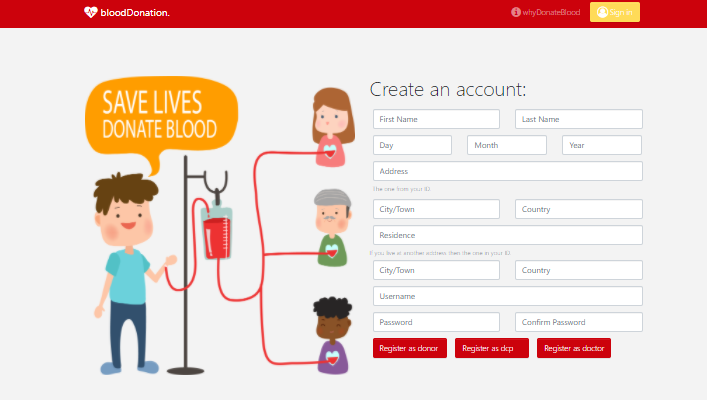


Figura 14: Interfața grafica a aplicației bloodDonation (homepage)

Interfața aplicației de bază bloodDonation conține elemente interactive precum imagini, câmpuri de introducere a diferitelor date personale, zone de text, butoane. Aceastea poate fi observate în Figururile 14-20.

Graphical user interface, text, application, email

Description automatically generated

Figura 15: Interfața grafica a aplicației bloodDonation (whyDonateBlood)

Graphical user interface, text, application, email

Description automatically generated

Figura 16: Interfața grafica a aplicației bloodDonation (login)

Graphical user interface, text, application

Description automatically generated

Figura 17: Interfața grafica a aplicației bloodDonation (register)

Graphical user interface, application, website

Description automatically generated

Figura 18: Interfața grafica a aplicației bloodDonation (homepage after login dcp)

### 2.2.1. Portabilitatea aplicatiei

Aplicata poate rula pe orice sistem de operare (Windows, MacOS, Linux) cat si pe orice device care suporta o aplicatie de browser, intru-un aceasta aplicatie este una web. Diagramele au fost create folosind aplicatia StarUML, iar din acestea vom genera cod pentru dezvoltarea aplicatiei.

## Generare Cod

Din cauza ca diagramele au fost scrise folosind un limbaj precum UML, am reusit sa generam tot codul necesar pentru clase in limbajul Java. De asemenea acest proces se poate repeta pentru orice limbaj de programare. In figura 19 se va putea observa o bucata de cod generate, restul de cod generat fiind atasat acestui document.

Graphical user interface, text, application

Description automatically generated

Figura 19: Cod generat dupa diagrame

## Descriere tehnica

### Numele aplicatiei

Numele aplicației este „Descrierea proiectelor software folosind diagrame UML”, deoarece are ca și scop descrierea aplicațiilor software prin modelarea vizuală a acestora, utilizând diagrame UML.

### Limbaje si unelte de programare

Limbajul utilizat este UML, limbaj destinat vizualizării, specificării, construirii și documentării sistemelor de aplicaţii, nu este dependent de tehnologia sau de limbajul folosit.

Unealta pentru generarea diagramlor UML utilizată este StarUML: freeware, independentă de mediul de dezvoltare, rulează în Windows/MacOS/Linux.

## Planul propus (general)

Împărțire task-uri intre autori:

Definire milestone-uri (milestone = data/ optiunile/documente disponibile pentru distribuire)

* intern = 05/11/2020 aplicația distribuită

23/11/2020 prezentată în interiorul echipei

* extern = 25/11/2020 aplicația distribuită

16/12/2020 intalnire intermediara

13/01/2021 prezentată clientului

## Plan de calitate

Documentele furnizate sunt caietul de sarcini, caietul de specificații, extrasele de reuniune.

## Plan de management

### Organizarea proiectului si a echipei de proiect

Șef de proiect: Oscar Gal

Împărțire responsabilități:

* + Oscar Gal – întocmire diagrame,interfața aplicație, documentație
  + Maria Florică – întocmire diagrame, caiet specificații, documentație
  + Doriana Mureșan – întocmire diagrame, caiet sarcini, documentație

### Gestiunea resurselor

Suportul, mijloacele logistice și tehnice la dispoziția echipei: 3 calculatoare, tutoriale despre realizarea diagramelor UML, suportul de curs dispus pe Helios.

### Contact client

Frecvența întâlnirilor și modul de contact: săptămânal prin conferințe Zoom și e-mail.

# De furnizat la client

Versiune: la client este furnizată o descriere a proiectelor software folosind diagramele UML pe baza descrierii aplicației bloodDonation.

Suportul furnizat clientului: documente ce conțin descrierea aplicației în format electronic.

Documentele furnizare clientului sunt: un dosar cu documentele « Caiet de sarcini», « Caiet de specificatii».

**Semnatura client: Semnatura echipa :**