**Laboratoare ISS**

**25 FEBRUARIE 2018**

**Lab 1**

**Prezentari**

1. Prezentare a problemei (aplicatie transfuzii sange) si a cerintelor pentru cele 7

laboratoare.

2. Familiarizarea cu github, bitbucket sau altceva, in functie si de ceea ce stiu studentii.

3. Instalarea StarUML, prezentarea DrawIO sau a altui tool.

4. Familiarizare cu Trello, Asana, etc.

**Organizare**

5. Formarea echipelor (6-8 persoane) - 30-60 min.

6. Alegerea unui Team Leader(TL) pentru fiecare echipa. Acesta are rolul de a aloca fiecarui

membru sarcini de lucru de-a lungul semestrului, de a urmari (alaturi de noi) activitatea fiecarui membru din echipa, iar la final de a acorda notele astfel incat media notelor (intregi) sa fie egala cu nota proiectului. Nota proiectului e acordata de noi.

7. TL-ul trebuie sa imparta sarcini astfel ca fiecare membru sa contribuie la fiecare etapa de

lucru.

8. Alegerea de catre echipe a tipului de aplicatie (web, desktop, eventual + mobile) si a

tehnologiilor pe care intentioneaza sa le foloseasca : Server BD, Limbaj de programare, ORM (Hibernate, Nhibernate, Entity Framework, Laravel, Symfony, etc, in functie de limbajul ales). In primele 4 saptamani pot sa-si schimbe . -20-30 min

**Lab 2**

**Diagrame, pe care trebuie sa le prezinte studentii**

9. Scenarii de utilizare.

10. Diagrama(ele) cazurilor de utilizare.

**Analiza cerintelor**

11. Formularea de catre studenti a unor cerinte nonfunctionale.

12. Raspuns la intrebarile studentilor legate de cerintele aplicatiei.

**2**

**Implementare**

13. Familiarizarea cu git, StarUML, Trello, etc - continuare, pt ca in primul lab unii abia au auzit

de ele.

14. Incercari (nereusite de cele mai multe ori) de folosire a ORM-ului ales.

**Lab 3**

**Diagrame, pe care trebuie sa le prezinte studentii**

15. Diagrama de clase.

16. Diagrama de arhitectura.

17. Diagrama bazei de date.

**Implementare**

18. ORM - implementare completa.

19. GUI - eventual, optional o diagrama a ferestrelor folosite, unde e cazul.

20. Implementarea, la alegerea lor a unor functionalitati (ex: operatii CRUD pentru evidenta

clientilor (clinici), a donatorilor, a drepturilor specifice fiecarui actor, ect)

**Lab 4**

**Diagrame, pe care trebuie sa le prezinte studentii**

21. Diagrame de secventa.

22. Diagrame de comunicare/colaborare.

**Implementare**

23. Continuare implementare functionalitati (ex: “matching” intre cerere (de sange) si oferta

donatorilor, etc.).

**Lab 5**

**Diagrame, pe care trebuie sa le prezinte studentii**

24. Finisarea, corectarea diagramelor, eventual crearea ultimelor diagrame.

**Implementare**

25. Finalizare aplicatie.

**3**

26. Prezentarea succinta a aplicatiei. (Scopul acestei prezentari este ca noi sa observam daca

ceva nu e in regula si sa ii ajutam sa depaseasca problemele).

**Lab 6**

**Diagrame, pe care trebuie sa le prezinte studentii**

27. Crearea documentatiei. Un dosar folie care cuprinde:

○ Coperta (titlu proiect, nume echipa, componenta echipa, coordonator).

○ Prezentarea cerintei aplicatiei pe scurt (cel mult o pagina, nu trebuie reluata cerinta lui Paul cu copy-paste), a functionalitatilor, etapele proiectarii aplicatiei (concret, pt aplicatia lor, nu teorie luata din curs).

○ Prezentarea tehnologiilor folosite (server BD, lb de prg, ORM, creare diagrame, versionare, management task-uri, creare GUI (unde e cazul, ex: Scene Builder), modul de comunicare client-sever, daca e cazul, testare - manual + automata).

○ Scenarii de utilizare.

○ Diagrama(ele) cazurilor de utilizare.

○ Diagrama(ele) de secventa.

○ Diagrama (e) de comunicare (colaborare)

○ Diagrama (e) de activitati

○ Diagrama de arhitectura (client-server; n-tier; ect)

○ Diagrama de clase.

○ Diagrama bazei de date.

○ Testarea aplicatiei

○ O scurta prezentare si un mic Help al aplicatiei. (Gandita pentru a fi prezentata clientilor).

○ Prezentarea

PS. Acest proiect poate fi considerat o pre-licenta.

Nu trebuie create toate diagramele aplicatiei, ci doar 1-2 din fiecare tip, pe care le considera ei mai importante.

Daca credeti ca-s prea multe diagrame (sau prea putine) astept alte propuneri.

**Implementare**

**4**

28. Ultimele retusuri aduse aplicatiei.

**Lab 7**

**PREZENTARE PROIECTE**

**ADAUGIRI**

E evident ca la fiecare laborator trebuie reluata discutia de la zero. Normal ei ar trebui sa stie totul despre artifacte (diagrame), sa proiecteze aplicatia si abia apoi sa o implemneteze. Partea grea e (si pentru ei, si pt noi) ca ei proiecteaza si implementeaza din mers, in paralel. (Si nu inteleg scopul diagramelor). Si de multe ori trebuie sa vina cu diagrame pe care, din cauza orarului, inca nu le-au facut nici la curs, nici la seminar. Aici, in functie de orarul fiecarei grupe, fie mai asteptam 2 saptamani si schimbam ordinea task-urilor, fie, introducem la laborator, daca ramane timp, notiunile despre diagramele cerute pentru urmatorul lab.

**Laboratoare ISS**

**Seminar 1**

Scopul SE, Ce este SE, Modelare, Ce este UML, Etapele realizarii sistemelor software, managementul task-urilor, Analiza cerintelor, Cerinte functionale, non-functionale, Tipuri de diagrame, Scenarii de utilizare, Diagr cazurilor de utilizare (Uses cases). Exercitii la diagrame.

**Seminar 2**

Diagrame de clase. Exercitii

**Seminar 3**

Diagrame de interactiune (secventa)

**5**

**Seminar 4**

Diagrame de interactiune (comunicare/colaborare)

**Seminar 5**

??? E posibil ca seminarul 1 sa fie prea mare si atunci se ajunge cu decalajul aici.

**Seminar 6**

Design Pattern (3-4 din cele pe care le fac la curs)

**Seminar 7**

Modele de testare