

Prezentare

Aplicații ale problemelor de tip Stable Matching

LUCRARE DE LICENȚĂ

Boca Ioan-Bogdan

UNIVERSITATEA ALEXANDRU IOAN-CUZA DIN IAȘI
Facultatea de Informatică

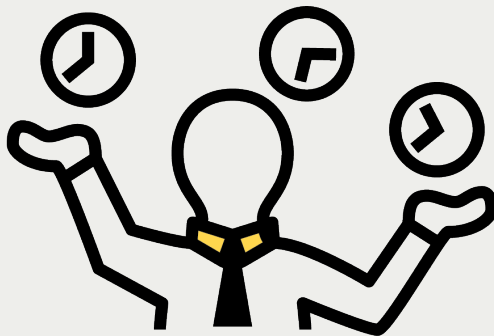


Coordonator științific: Lect. Dr. , Frăsinaru Cristian

4 Iulie, 2018

- 1 Motivație
- 2 Stable-Matching
- 3 Arhitectură și implementare
- 4 Demo
- 5 Concluzii și direcții de viitor

Motivație #1



Stable-Matching #2

- Definiție
- Clasificare:
 - One-to-One: (ex: Stable-Marriage)
 - One-to-Many: (ex: Student-Project Allocation, Hospital-Residents, Stable-Admissions)
 - Many-to-Many: (ex: Stable Allocations, Teachers-Repartitions)
- Gale și Shapley propun un algoritm încă din 1962



Definirea problemei

Student-Project Allocation

Input

- $S = \{s_1, s_2, s_3, \dots, s_n\}$
- $L = \{l_1, l_2, l_3, \dots, l_k\}$
- $P = \{p_1, p_2, p_3, \dots, p_m\}$

Output

M submulțime a $S \times P$ a.î.

- $(s_i, p_j) \in M$ a.î. $p_j \in A_i$
- $\forall s_i \in S$ este valabilă $|\{(s_i, p_j) \in M : p_j \in P\}| \leq 1$

Exemplu

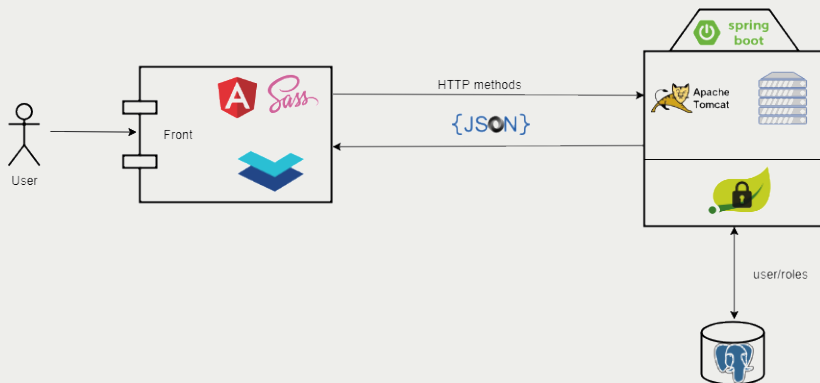
Preferințe studenți	Preferințe profesori
$s_1 : p_1 p_3$	$l_1 : s_1 s_2 s_3 s_4$
$s_2 : p_1 p_2 p_3$	$l_2 : s_2 s_1 s_3$
$s_3 : p_3 p_4$	
$s_4 : p_3$	

 l_1 ofera p_1, p_2 l_2 ofera p_3, p_4 Capacități proiecte: $c_i = 1$ Capacități profesori: $d_1 = 1, d_2 = 2$ $M = \{(s_1, p_1), (s_2, p_3), (s_3, p_4), (s_4, \emptyset)\}$

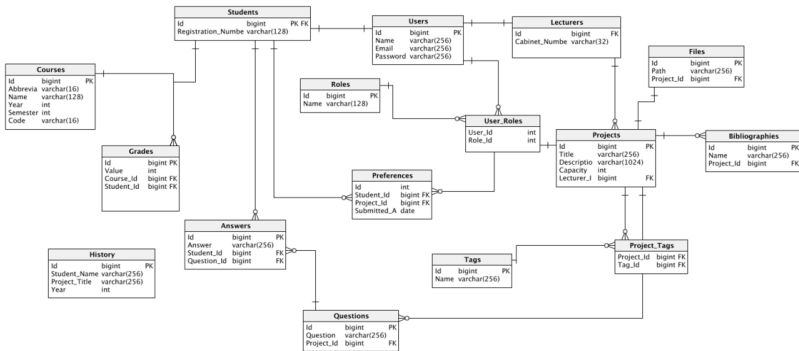
Figure: Instanta Student Project Allocation



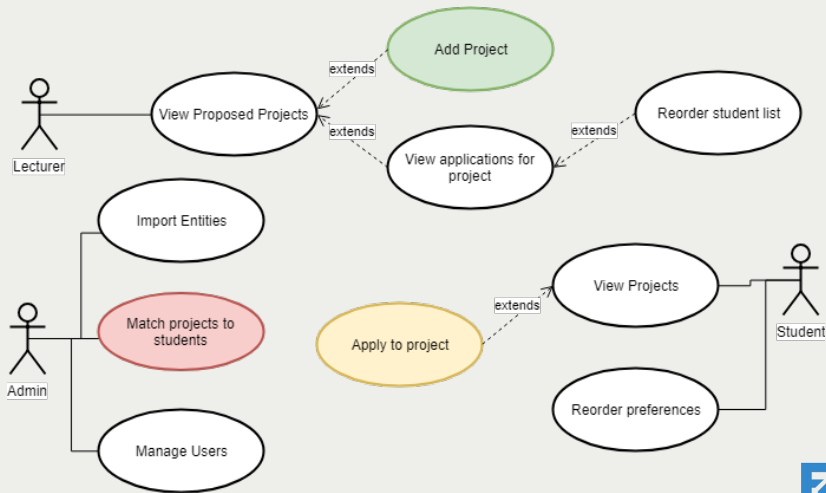
Arhitectura generală a proiectului #3



Arhitectura bazei de date



Scenariu de utilizare



Cum?



Demo #4



Concluzii și direcții de viitor #5

Beneficii

- Sistem centralizat de repartizare
- Transparență
- Accesibilitate

Direcții de viitor

- Ghid complet în obținerea diplomei de licență
- Echilibrarea supervizării proiectelor în rândul profesorilor
- Propuneri din partea studenților

Final

Vă mulțumesc!

