Cuprins

[1 Introducere 2](#_Toc11629610)

[1.1 Motivație 2](#_Toc11629611)

[1.2 Soluție existentă 3](#_Toc11629612)

[1.3 Soluție propusă 4](#_Toc11629613)

[1.3.1 Funcționalități 4](#_Toc11629614)

[1.3.2 Structura lucrării 7](#_Toc11629615)

[1.4 Contribuții 8](#_Toc11629616)

[2 Fundamentele teoretice ale aplicației 9](#_Toc11629617)

[2.1 Algoritmul „stable matching” clasic al lui „Gale–Shapley” (2) 9](#_Toc11629618)

[2.2 Variații ale algoritmului de „stable matching” : 11](#_Toc11629619)

[2.2.1 „Stable matching” cu liste incomplete de preferințe (3) 11](#_Toc11629620)

[2.2.2 „Stable matching” cu liste de preferințe de tip „ties” (3) 12](#_Toc11629621)

[2.3 Algoritmul de „stable matching” utilizat de aplicație 12](#_Toc11629622)

[3 Integrarea algoritmului de „stable matching” in aplicația web 14](#_Toc11629623)

[3.1 Tehnologii folosite 14](#_Toc11629624)

[3.2 „Flow”-ul 15](#_Toc11629625)

[3.3 Dificultăți întâlnite 17](#_Toc11629626)

[4 BE 17](#_Toc11629627)

[4.1 Tehnologii folosite 17](#_Toc11629628)

[4.2 Flow / descriere/ digrama bazei de date 17](#_Toc11629629)

[5 Fe 17](#_Toc11629630)

[5.1 Tehnologii folosite 17](#_Toc11629631)

[5.2 Flow/ descriere 17](#_Toc11629632)

[6 Manual de utilizare 17](#_Toc11629633)

[7 Concluzii 17](#_Toc11629634)

[8 Bibliografie 17](#_Toc11629635)

# Introducere

## Motivație

În aceasta lucrare am abordat una dintre cele mai frecvente probleme existente, din perspectiva unui cititor, la o bibliotecă. Inconveniențaeste reprezentată printr-o administrare ineficientă a împrumutului de cărți. Acest lucru este mai accentuat în cazul acelor cărți ce sunt într-un număr relativ redus în comparație cu cei ce doresc sa aibă acces la o anumită carte.

Un raport publicat de „Biblioteca Centrala Universitară” din Iași in anul 2018 evidențiază o scădere dramatică a numărului de cititori *„În ultimii zece ani, numărul utilizatorilor prezenți în bibliotecile universitare a scăzut constant și dramatic. Pe fondul declinului demografic care afectează România, scăderea efectivelor de studenți a însemnat și diminuarea cu 45% a numărului de utilizatori ai BCU „Mihai Eminescu”, de la 18.917, în anul 2009, la 10.444 utilizatori, în 2018*” (1). În urma acestui raport ar trebui să se încerce o diminuare al acestui efect, prin identificarea cauzelor care au dus la scăderea numărului de cititori.

Resursele tipărite ajung să fie din ce in ce mai neglijate, astfel încât cele de tip electronic sunt preferate datorită accesului facil la acestea. Există o concurență intensă între cărțile tipărite și cele virtuale, acestea din urmă ajungând sa câștige teren pe măsură ce tot mai multe resurse existente ajung sa aibă echivalent electronic. Cu toate acestea, colecțiile digitale nu sunt întotdeauna surse de încredere, informațiile transmise pot să difere de realitate, ajungând să dezinformeze.

Statul la rând, poate fi incomod pentru cititori, uneori chiar este ineficient în situațiile în care după perioada de așteptare ajung să afle că resursa dorită de persoanele în cauză, nu se află in colecția bibliotecii. Cu toate că resursa se află în catalog ea poate să nu fie disponibilă, fiind in acel moment împrumutată de alți cititori.

Unele dintre acestea încearcă să diminueze inconveniența prin existența unui catalog online, prin care se poate vedea dacă biblioteca dispune de carte respectivă, ba mai mult se poate realiza o rezervare on-line pentru cărțile disponibile în acel moment. Cu toate aceste funcționalități problema nu este rezolvată în totalitate, deoarece cartea poate fi indisponibilă neputând fi rezervată.

Astfel, bibliotecile dorind să rezolve aceste dificultăți ale cititorilor utilizează platformele on-line, prin intermediul cărora se facilitează accesul la cataloagele de cărți aflate in colecția acestora. În detrimentul caracteristicilor prezentate, platformele pot dispune de o optimizare aducând o eficientizare procesului de închiriere al cărților. Aceste schimbări pot fi reprezentate de accesul la statusul resurselor. Dacă sunt disponibile utilizatorul ar putea să le rezerve, pentru o perioada de timp, funcționalitate deja existentă, dar ar trebui automatizată, în sensul că după o perioadă de timp predefinită, cunoscută de cititor, ea sa devină invalidă. Astfel cititorul având siguranța că atunci când vine la bibliotecă în limita de timp specificată, cărțile mai sunt disponibile. Există de asemenea cazul în care resursa este indisponibilă din faza inițială, iar utilizatorul ar trebui să poată preciza dacă o dorește și să o primească atunci când devine accesibilă.

Noutățile prezentate mai sus aduc o experiența mai plăcută unui cititor în procesul de închiriere al unei cărți, determinând si o creștere al numărului de persoane ce doresc sa utilizeze resursele tipărite. Optimizările specificate reprezintă soluția mea găsită la această problemă și prezentate în 1.3.

## Soluție existentă

După cum am specificat mai sus, bibliotecile s-au gândit la o soluție pentru a îmbunătăți eficiența procesului de închiriere al unei cărți. În continuare voi prezenta una dintre aceste platforme online. Descrierea va fi una succintă axată mai mult pe funcționalitățile la care are acces un utilizator obișnuit, astfel prezentarea va fi din perspectiva unui cititor.

* Utilizator:
* Conectare ca și vizitator – Nu necesită existența unui cont cu o parolă validă. Permite doar căutarea unei cărți si vizualizarea existenței acesteia.
* Conectare – Utilizatorul se poate conecta cu un nume de cont valid și cu parola corespunzătoare acestuia. Astfel, utilizatorul are acces la fișa personală, poate rezerva on-line o carte etc.
* Căutare carte – Există funcționalitatea de căutare a unei cărți folosind mai multe criterii ( titlul cărții, autorul, seria, ediția, etc), peste care se pot aplica filtre ( căutare în interiorul unui cuvânt, alegerea bazei de date în care se face căutarea).
* Rezervare on-line – Rezultatele căutării pot fi vizualizate sub formă de tabel, ele fiind afișate detaliat, existând și informații referitoare la rezervare.
* Vizualizare fișă personala – Utilizatorul poate avea acces la profilul acestuia, ce conține informații despre nume, adresă, numărul de permis etc.

## Soluție propusă

Soluția propusă de mine pentru a optimiza procesul de închiriere al unei cărți la o bibliotecă, reducând inconveniența cititorilor, constă în realizarea unei platforme on-line ce conține cataloagele de cărți ale acesteia. Aplicația web în funcție de fiecare tip de utilizator în parte va permite realizarea de acțiuni specifice acestora.

### Funcționalități

În aplicația mea există două tipuri de utilizatori: primul este de tip bibliotecar având rol de administrator, dar are posibilitatea de a utiliza aplicația ca un utilizator obișnuit.

1. Bibliotecar :

* Conectare – Pentru această acțiune este necesară existența unui cont valid reprezentat prin adresă de email validă și parola corespunzătoare acestuia. La conectare bibliotecarul este redirecționat pe o pagină web, unde are acces la funcționalitățile unui utilizator cu rol obișnuit, dar are în pagina de navigație buton către partea de administrator.
* Vizualizare listă utilizatori – Toți utilizatorii sunt prezentați în detaliu(nume, adresă, număr permis bibliotecă, etc) sub formă de tabel, iar în dreptul fiecărui utilizator există posibilitatea de actualizare informație sau ștergere.
* Actualizare utilizator – Se poate actualiza toate informațiile specifice unui utilizator dorit, referitoare la profilul acestuia reprezentat prin nume, prenume, adresa, etc.
* Ștergere utilizator – Prin această acțiune se șterge definitiv utilizator în cauză.
* Căutare utilizator – Se căuta un utilizator după nume sau prenume, cu posibilitatea de căutare în interiorul cuvântului.
* Căutare avansata al unui utilizator – Funcționalitatea este reprezentată de posibilitatea de a căuta un utilizator după mai multe criterii dorite: nume, prenume, adresă de email, numărul permisului de la bibliotecă.
* Vizualizare cărți – Toate cărțile sunt afișate detaliat(nume, tip, nume autor, disponibilitate etc), într-un format tabelar ce facilitează posibilitatea de administrare a unei cărți.
* Actualizare cărți – Caracteristicile specifice unei cărți( nume, nume autor, tipul unei cărți, seria etc ) pot fi actualizate prin noi caracteristici furnizate.
* Căutare carte – Funcționalitatea permite căutarea unei cărți după nume, totodată există și posibilitatea căutării în interiorul unui cuvânt. Rezultatele sunt căutării sunt prezentate sub formă de tabel.
* Căutare avansată a unei cărți – Utilizatorul poate căuta o carte după mai multe criterii dorite: după nume sau tipul cărții, numele autorului sau prenumele acestuia. Se pot adăuga filtre la căutare: căutare în interiorul cuvântului, exclude cărți ce nu sunt returnate, afișează doar rezultatele ce au statusul disponibil.
* Ștergere cărți – Oferă posibilitatea de ștergere a unei cărți, în același timp se șterg si caracteristicile acesteia, precum autorul dacă el nu mai are in colecție și alte cărți.
* Deconectare – Utilizatorul are posibilitatea de deconectare, acesta fiind redirecționat către pagina de conectare.

1. Utilizator:

* Conectare – Utilizatorul pentru a se conecta la aplicație are nevoie de o adresă de email validă cu rol de utilizator obișnuit și parola corespunzătoare acesteia.
* Deconectare – Orice utilizator se poate deconecta atunci când dorește, iar ca rezultat el va fi redirecționat de către platformă spre pagina de conectare.
* Vizualizare profil: Utilizatorul poate accesa profilul său, unde sunt specificate informații referitoare la numele si prenumele acestuia, adresă, poză de profil, adresă de email.
* Schimbare parolă de conectare cont: Permite opțiunea de schimbare a parolei, dacă se dorește acest lucru, totodată trebuie cunoscută vechea parolă pentru validarea datelor.
* Căutare carte – Funcționalitatea permite căutarea unei cărți după nume, totodată există și posibilitatea căutării în interiorul unui cuvânt. Rezultatele sunt căutării sunt prezentate sub formă de tabel.
* Căutare avansată carte – Utilizatorul poate căuta o carte după mai multe criterii dorite: după nume sau tipul cărții, numele autorului sau prenumele acestuia. Se pot adăuga filtre la căutare: căutare în interiorul cuvântului, excludere cărți din lista de dorințe, exclude cărți ce nu sunt returnate, afișează doar rezultatele ce au statusul disponibil. Rezultatele sunt afișate detaliat sub formă de tabel, fiecare având opțiunea de rezervare sau adăugare in lista de dorințe.
* Rezervare carte disponibilă – Cărțile ce au statusul de disponibil pot fi rezervate, pentru o perioadă de 24 de ore, timp în care pot fi ridicate de la bibliotecă. După perioada specificată, rezervarea expiră.
* Adaugă carte indisponibilă la lista de dorințe – Acele cărți ce au statusul indisponibil pot fi adăugate in lista de dorințe, astfel utilizatorul specificând că o dorește atunci când cartea devine disponibilă.
* Vizualizare istoric cărți împrumutate – Funcționalitatea permite vizualizarea istoricului de cărți împrumutate până acum, dar și posibilitatea observării cărților ce au fost rezervate și nu au fost ridicate la timp.
* Vizualizare listă de dorințe curentă – Utilizatorul poate vizualiza lista de dorințe în carte sunt prezentate cărțile în ordine crescătoare a importanței pentru acesta. Lista este prezentată sub formă de tabel în care fiecare carte are informații specifice acesteia, dar și perioada pentru care se dorește să fie împrumutată.
* Actualizare listă de dorințe curentă – Utilizatorul poate actualiza informația referitoare la clasa de importanță a unei cărți și totodată poate schimba și perioada pentru care dorește să o închirieze.
* Vizualizare notificări – Notificările noi si necitite sunt disponibile în bara de meniu, dar utilizatorul are și posibilitatea vizualizării tuturor notificărilor în pagina de notificări. Notificările sunt generate în urma expirării unei rezervări, în urma asignării unei cărți din lista de dorințe sau ca efect al neacceptării cărții din „wishlist”, asignată în intervalul de timp specificat.
* Vizualizare statistici despre :
* tipul de carte preferat – Utilizatorul poate vedea un top trei al tipurilor de cărți preferate de acesta.
* coeficientul mediu de returnare al cărților – Este reprezentat printr-un procent și evidențiază timpul mediu de returnare al unei cărți închiriate. Dacă procentul are culoarea roșie înseamnă că utilizatorul obișnuiește să întârzie, iar în caz contrar are culoarea verde.
* numărul de cărți returnate la timp sau întârziate pe lună – Statisticile acestea au un rol informativ cu privire la istoricul cărților închiriate, totodată ele validează corectitudinea procentului de returnare.

### Structura lucrării

Lucrarea realizată este împărțită în următoarele capitole:

1. Fundamentele teoretice ale aplicației

Aplicația mea folosește pentru administrarea resurselor limitate un algoritm de „stable matching” și de aceea consider că mai întâi este firesc să vorbesc despre acesta. Astfel capitolul descrie noțiuni referitoare la acest algoritm, cât si algoritmul propriu-zis. Sunt evidențiate motivele pentru care algoritmul clasic nu este utilizat în practică, de asemenea sunt prezentate derivări ale acestuia, și tipul de algoritm ales de mine.

1. Integrarea algoritmului de „stable matching” în aplicația web

Capitolul acesta are rolul de a evidenția modul în care algoritmul este folosit într-o aplicație web. Sunt descrise tehnologiile folosite, modul în care este integrat astfel încât să nu împiedice funcționarea aplicației ce este asincronă, păstrând eficiența cu care răspunde la „request”-uri.

1. „Back-end”-ul aplicației web

Ca în orice aplicație web și în aceasta se face o distincție între modulul de „front-end” și cel de „back-end”. În capitol va fi prezentă o descriere a tehnologiilor folosite...

1. Fe

## Contribuții

Utilizatorul are acces la funcționalitatea de căutare a unei cărți dorite, iar în cazul existenței unui rezultat corespunzător acestuia îi sunt afișate detalii despre carte, totodată si statusul acesteia ce poate fi disponibil sau indisponibil.

Adițional aplicațiilor web actuale, oferite de biblioteci, platforma realizată de mine oferă o automatizare a procesului de închiriere al unei cărți oferind noi funcționalități cum ar fi rezervarea pentru o perioadă de timp a unei resurse disponibile sau posibilitatea de adăugare în „wishlist” a celor ce sunt indisponibile, reprezentate prin resurse momentan reduse.

Rezervarea unei cărți – După căutarea rezultatului dorit, utilizatorul poate opta pentru rezervarea acesteia, dacă are statusul de disponibil. Această acțiune este realizată automat de către platformă, iar în cazul în care resursa nu este revendicată într-un interval de timp de 24 de ore de la momentul rezervării, ea devine invalidă. Odată devenită invalidă, utilizatorul o poate rezerva din nou, dacă aceasta mai este disponibilă, altfel el poate opta pentru opțiunea de adăugare la lista proprie de dorințe. Prin intermediul funcționalități de rezervare se dorește evitarea situației întâlnite la platformele actuale, ce nu garantează că existența cărții in colecția bibliotecii este echivalent cu disponibilitatea acesteia.

Adăugare la lista proprie de dorințe a unei resurse indisponibile – O carte existentă în catalogul aplicației mele web, ce are la status indisponibil poate fi adăugată la „wishlist”. Statusul de indisponibil evidențiază că acea carte se încadrează în categoria de resurse limitate ( aceasta limitare se poate referi la una momentană, în sensul că resursa este tipărită într-o mulțime de exemplare, dar majoritatea sunt închiriate sau este echivalent cu un număr redus de exemplare ce trebuie administrate eficient). Decizia de asignare a unei cărți din lista de dorințe se realizează pe baza unui algoritm de „Stable matching”.

Odată la un interval de timp de 2 minute un „task scheduler” generează un „task” care este memorat într-un „message broker”, împreună cu momentul în care trebuie sa înceapă execuția acestuia. Un „worker” verifică „task”-urile memorate și le execută în „background” în funcție de timpul specificat. Astfel rutina este formată din execuția consecutivă a următoarelor etape:

* Ștergere rezervări ce nu au fost revendicate la timp
* Actualizează starea – Se interoghează baza de date si se face „update” la informațiile necesare algoritmului de „Stable matching”
* Executarea algoritmului
* Trecerea rezultatului in baza de date – Existând un „delay” între momentul actualizării cărții și cel de scriere al rezultatului, se face totodată și o verificare al acestuia, existând posibilitatea ca o carte asignată să nu mai fie dorită de utilizator. Utilizatorii ce nu au primit o carte în urma acestui algoritm și lista de dorințe nu este vidă, au posibilitatea de a le fi asignată o carte la următoare iterație a rutinei.

Algoritmul de „Stable matching” asignează unui utilizator una dintre cărțile dorite dacă este posibil. De menționat este că spre deosebire de algoritmul clasic, cele două mulțimi curente ( cărțile dintr-un „wishlist” și numărul total de cărți) au un număr de elemente diferite ce poate conduce la o incompatibilitate între preferințe. De aceea se consideră configurație stabilă, dacă în urma acestui algoritm un utilizator nu primește o carte, chiar dacă are o lista de preferințe, dar cărțile preferate de acesta sunt mult mai eficient asignate altui utilizator.

Criteriile de asignare ale algoritmului sunt: „rank”-ul unei cărți definit în „wishlist” perioada pentru care este dorită cartea, dar cel mai important pentru optimizarea împrumutului de cărți este coeficientul de încredere al unui utilizator. Acest coeficient este de fapt o medie între procentul din perioada rămasă de închiriere (sau întârziere) atunci când a fost predată cartea, împărțit la numărul de cărți.

# Fundamentele teoretice ale aplicației

## Algoritmul „stable matching” clasic al lui „Gale–Shapley” (2)

O problemă de tip „stable matching” presupune găsirea unui „stable matching” între două mulțimi ce conțin același număr de elemente. Această potrivire se face ținându-se cont de o listă ordonată de preferințe pentru fiecare element în parte. Astfel se încearcă o potrivire menită cu scopul de a satisface preferințele date.

O potrivire între două mulțimi distincte, A și B, fiecare având n elemente nu este stabilă dacă se întâmplă următoarea situație:

* există perechea formată din (a, b'), cu a ∈ A, b' ∈ B
* există perechea formată din (b, a'), cu b ∈ B, a' ∈ A
* a ≠ a', b ≠ b'
* a preferă mai mult pe b decât b'
* b preferă mai mult pe a decât a'

Astfel situația prezentată mai sus în care nu este formată perechea (a, b), deși aceasta ar fi trebuit să se realizeze reprezintă o stare instabilă a rezultatului.

* Formularea problemei : Dându-se un număr de n bărbați si n femei, fiecare dintre aceștia având o listă de preferințe de la 1 la n, cu o ordine totală peste persoanele de sex opus, să se cupleze un bărbat și o femeie astfel încât să nu existe 2 persoane de sex opus ce se preferă reciproc in detrimentul persoanei actuale din cuplu. Dacă nu există astfel de pereche precizată anterior, atunci există un „matching” stabil.
* Soluția acestei probleme este descrisă prin algoritmul următor :

|  |  |
| --- | --- |
|  | Inițial cele două mulțimi, a bărbaților reprezentată print „M” și cea a femeilor notată cu „W” nu conțin persoane ce au deja găsită o pereche finală sau momentană.  În prima iterație a algoritmului, se cuplează primul bărbat cu una dintre femei aflate prima pe lista lui |

de preferințe. Această cuplare nu este neapărat finală, ci este momentană, astfel încât pană la final poate să se schimbe.

În fiecare iterație a algoritmului de „stable matching”, se alege primul bărbat ce nu are încă o pereche și se parcurge pe rând preferințele acestuia. Dacă preferința curentă face parte din altă pereche, se verifică dacă nu cumva este mai convenabil atât pentru femeie cât si bărbatul în cauză să se cupleze. În cazul în care femeia îl preferă pe mai mult pe bărbatul pentru care se face cuplarea decât perechea ei, cei doi vor forma un nou cuplu, iar bărbatul din fostul cuplu va rămâne din nou fără femeie.

Algoritmul de clasic de „stable matching” are o complexitate de execuție polinomială, terminându-se în cel mai ineficient caz în O(), unde n reprezintă numărul de femei sau de bărbați, acesta fiind egal.

La sfârșitul execuției acestuia, algoritmul garantează că nu poate exista un bărbat ce a rămas fără pereche, deoarece dacă un bărbat nu a găsit în o femeie, la acesta se va reveni, până când toate opțiunile sale vor fi verificate. Astfel toate cele „n” opțiuni ale sale vor fi verificate și ținând cont că există un număr „n” bărbați și femei, fiecare bărbat / femeie va face parte dintr-o pereche.

În practică acest algoritm în forma lui clasică, care presupune existența a două mulțimi cu același număr de element, listele de preferință să fie complete, având numărul de elemente din acestea egal cu numărul de elemente din una dintre cele două mulțimi, nu este folosit. Aceste condiții reprezintă caracteristici ideale existând o șansă extrem de mică să fie îndeplinite, astfel s-a recurs la o relaxare a acestor condiții apărând unele variații ce sunt folosite în practică și enumerate mai jos.

## Variații ale algoritmului de „stable matching” :

### „Stable matching” cu liste incomplete de preferințe (3)

In această derivație al algoritmului se permite ca lista de preferințe, a unui bărbat sau a unei femei, să conțină un număr mai mic decât numărul de elemente al primei mulțimii, reprezentată de bărbați sau mai mic decât cea a femeilor.

Din cauza acestei relaxări, algoritmul nu garantează că la finalizarea acestuia toate persoanele sunt cuplate, în schimb ideea de bază al acestuia este de a realiza perechi convenabile pentru câte mai multe persoane. Astfel situația descrisă mai jos după finalizarea „stable matching”-ului reprezintă o stare validă:

* 2 mulțimi: B a bărbaților, si F a femeilor, fiecare având n elemente
* există perechea formată din (b, f), cu b ∈ B, f ∈ F
* b' nu are pereche, dar preferă pe f
* f nu preferă pe b' mai mult decât pe b

Algoritmul de „stable matching” cu liste de preferințe incomplete are o complexitate de execuție polinomială, terminându-se în cel mai ineficient caz în O(), unde n reprezintă numărul de femei sau de bărbați, acesta fiind egal.

### „Stable matching” cu liste de preferințe de tip „ties” (3)

Această formă a algoritmului presupune ca peste lista de preferințe să nu se aplice o ordonare totală, ea fiind înlocuită cu o ordonare parțială. Uneori poate lipsi de tot, fiind îndeajuns cuplarea cu o persoană indiferent de poziția acesteia în listă.

În acest tip de extensie cu liste de preferințe de tip „ties”, o stare instabilă este descrisă in următoare situație:

* 2 mulțimi: B a bărbaților, si F a femeilor, fiecare având n elemente
* nu există perechea formată din (b, f), cu b ∈ B, f ∈ F
* b preferă în mod strict pe f ' ( f și f ' nu au același nivel de preferință )
* f preferă în mod strict pe b' ( b și b ' nu au același nivel de preferință )

Dacă nu există o stare blocantă echivalentă cu cea descrisă mai sus, atunci rezultatul în urma executării algoritmului este de tip „weakly stable”.

Astfel aceste modificări aduse algoritmului clasic nu mărește timpul de execuție polinomial, acesta fiind tot O(), în cel mai ineficient caz, unde n reprezintă numărul de femei sau de bărbați, acesta fiind egal.

## Algoritmul de „stable matching” utilizat de aplicație

Pentru aplicația web realizată de mine, am ales să integrez algoritmul „stable matching” cu liste incomplete de preferințe, deoarece caracteristicile acestuia se pliază cu modul de funcționare al „wishlist”-ului unui utilizator.

Lista de cărți, pe care le preferă un utilizator și dorește să le închirieze atunci când ele vor fi disponibile, nu este obligatorie ca să fie distinctă cu a altui utilizator. Se poate întâmpla ca de cele mai multe ori ea să fie nulă sau să conțină un număr redus de elemente. Astfel modul de funcționare al unui „wishlist” este echivalentul unei liste de preferințe descris de algoritmul de „stable matching” cu liste incomplete.

Corespondentul celor două mulțimi pe care se realizează cuplarea din derivația algoritmul clasic ( reprezentată prin mulțimea de bărbați ce dorește sa își găsească o pereche din mulțimea de femei ) este reprezentat prin mulțimea de utilizatori ce au un „wishlist” ce nu este gol și mulțimea de cărți formată prin reuniunea conținutului fiecărui „wishlist”.

În situația platformei realizate definesc o stare stabilă ce respectă caracteristicile de mai jos:

* 2 mulțimi: U a utilizatorilor cu listă de preferințe ce nu este vidă, si C a cărților formată prin reuniunea cărților din fiecare „wishlist”
* există perechea formată din (u, c), cu u ∈ U, c ∈ C
* u' nu are pereche, dar preferă cartea c
* c nu preferă pe u' mai mult decât pe u

Spre deosebire de relația de preferință a unui utilizator ce este clar fixată de acesta, criteriul pe care îl urmează o carte pentru a alegere perechea sa este definit pe baza perioadei de închiriere la care se adaugă sau se scade un coeficient de încredere al acestuia.

Pentru a preciza condiția de decizie a cărți în vederea distingerii între doi utilizatori, voi defini următoarele notații :

* , perioda de timp până la returnarea cărții
* , unde n = nr de cărți returnate,

c = coeficient de returnare, este reprezentat ca o medie între diferențele de timp dintre momentul când expiră închirierea și momentul de timp când se returnează cartea.

Acest coeficient poate fi și negativ, semnul reprezentând că utilizatorul returnează de obicei cartea cu întârziere, iar valoarea acestuia evidențiază perioda în procente cu care întârzie.

Analog, în cazul în care procentul este pozitiv este echivalent cu o returnare mai devreme.

Ținând cont de notațiile definite mai sus decizia este reprezentată prin formula:

* D =

Astfel, o carte din motive de eficiență dintre doi utilizatori ce o doresc, logic ar fi ca aceasta să se grupeze cu acel „user” care are un „D” ( dat prin formula de mai sus ) cât mai mic.

„D”- ul din formulă reprezintă o predicție a perioadei de închiriere a utilizatorului, bazându-se pe istoricul de returnare al acestuia.

# Integrarea algoritmului de „stable matching” in aplicația web

## Tehnologii folosite

* Celery este o coadă distributivă de „task”-uri asincrone. Este folosit de obicei pentru execuția „task”-urilor consumatoare de resurse sau timp în aplicații web deoarece acesta permite realizarea lor în „background”. Astfel Celery nu suprimă timpul de răspuns al unui „request”, oferind o experiență plăcută utilizatorului menținându-se fluența aplicației.
* Celery beat Acesta este un „scheduler” ce generează periodic un „task”. Intervalul de timp la care se execută este setat din înainte, iar „task”-ul creat este executat de Celery prin intermediul unui „worker” disponibil în acel timp.
* RabbitMQ Un „message broker” îndeplinește rolul de „middleman” pentru un serviciu, de cele mai multe ori acel serviciu este o aplicație web. Acesta este folosită cu scopul de a reduce „load”-ul aplicației stocând „task”-urile și totodată se ocupă de „routing”-ul acestora către un „worker” din Celery.
* Python 3.6 Limbaj de programare interpretat, ușor de folosit fiind încadrat ca limbaj de „high-level”.
* PostgreSQL Este o bază de date relațională, este gratuită fiind „open-source”. De asemenea este stabilă și suportă lucrul cu un volum mare de informații.
* SQLAlchemy Acesta reprezintă un „ORM”, compatibil cu limbajul de programare Python și cu baza de date PostgreSQL. Am folosit acest „tool” ce reprezintă un „layer” intermediar între comunicarea bazei de date și aplicația mea web. SQLAlchemy este cel mai răspândit „ORM” din cele prezente în aplicațiile scrise în Python.

## „Flow”-ul

|  |
| --- |
|  |

1. „Scheduler”-ul generează o dată la un interval de 2 minute un „task”, acesta este redenumit de mine ca „routine”. El este realizează mai multe acțiuni printre care și execuția algoritmului de „stable matching”. Voi reveni ulterior la această „routine”, unde o voi prezenta în detaliu.
2. În această etapă este prezentat „exchange” ce primește în mod direct „task”-ul de la „beat worker scheduler”, el se ocupă de redirecționarea spre următoare etapă.
3. După decizia realizată de „exchange”, „task”-ul este asignat uneia dintre cozile „message broker”-ului
4. Un „worker”-ul disponibil preia „task”-ul și se ocupă de executarea acestuia.
5. „Worker”-ul însărcinat cu realizarea rutinei, preia informația din baza de date necesară algoritmului de „stable matching”
6. În această etapă se realizează salvarea rezultatului în baza de date, dar mai înainte se face o verificare a corectitudinii rezultatului, asigurându-se că utilizatorul mai dorește acea carte. Verificarea se face deoarece există un „delay” din momentul luării informației și momentul scrierii de rezultate în baza de date.

„Task”-ul numit „routine” conține etapele următoare:

* Invalidează cărți neacceptate Cărțile ce au fost asignate de către algoritmul de „stable matching”, în urma execuției acestuia și care nu au fost acceptate în intervalul precizat de 3 ore.
* Actualizarea timp cărți Se face „update” la timpul rămas pentru cărțile închiriate.
* Actualizare „input” pentru „stable matching” Se preia din baza de date informațiile necesare executării algoritmului ( utilizatori, liste de preferințe, număr de cărți).
* Execuția algoritmului propriu-zis Se încearcă găsirea unei perechi între utilizator și una dintre cărțile dorite din lista de preferințe incomplete.
* Scriere rezultate în baza de date Înainte de trecerea rezultatelor în baza de date se face o verificare a corectitudinii rezultatului, deoarece există un decalaj de timp între momentul actualizării „input”-ului pentru algoritmul de „stable matching” și scrierea efectivă a rezultatelor. Totodată verificarea este necesară deoarece „task”-ul se realizează asincron cu „request”-urile aplicației web, putând apărea modificări în „wishlist”-uri între timp.

## Dificultăți întâlnite

# BE

## Tehnologii folosite

## Flow / descriere/ digrama bazei de date

# Fe

## Tehnologii folosite

## Flow/ descriere

# Manual de utilizare

# Concluzii

# Bibliografie

1. ***Raport instituţional, Biblioteca Centrală Universitară „Mihai Eminescu”.* Iasi : s.n., 2018.**

**2. Shapley, D. Gale and L.S. *„College admissions and the stability of marriage".* s.l. : Amer. Math. Monthly, 1962.**

**3. Kazuo Iwama, Shuichi Miyazaki, Yasufumi Morita, David Manlove. *„Lecture Notes in Computer Science", capitol „Stable Marriage with Incomplete Lists and Ties".* s.l. : Springer, Berlin, Heidelberg, 2002. 978-3-540-66224-2.**