Findpair

Final project
Ranin Nassra 312354327
Yuosif Hanna 205914401

Background

FindPair is an android application for finding a studying partner. Its hard to find a partner for a course, project or preparing for exams that is compatible to the student in so many criterions, especially nowadays, when the lessons are held online due to the pandemic.
We decided to build a platform that will make the life of the students easier, and pair them with each other in the most efficient and optimal way.

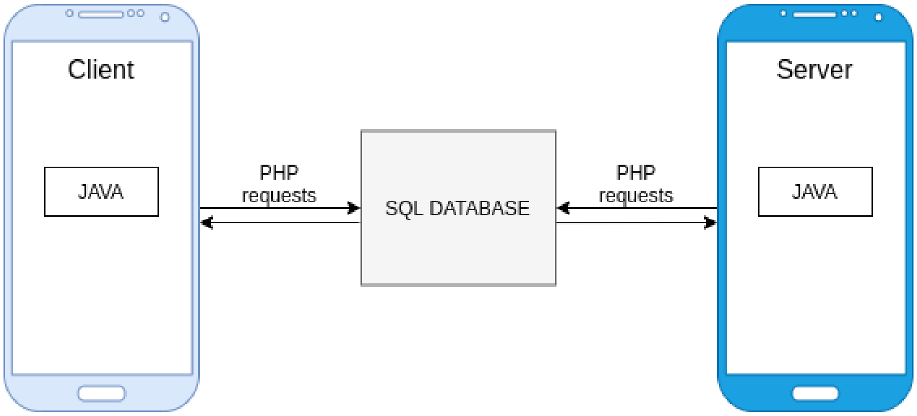
Motivation

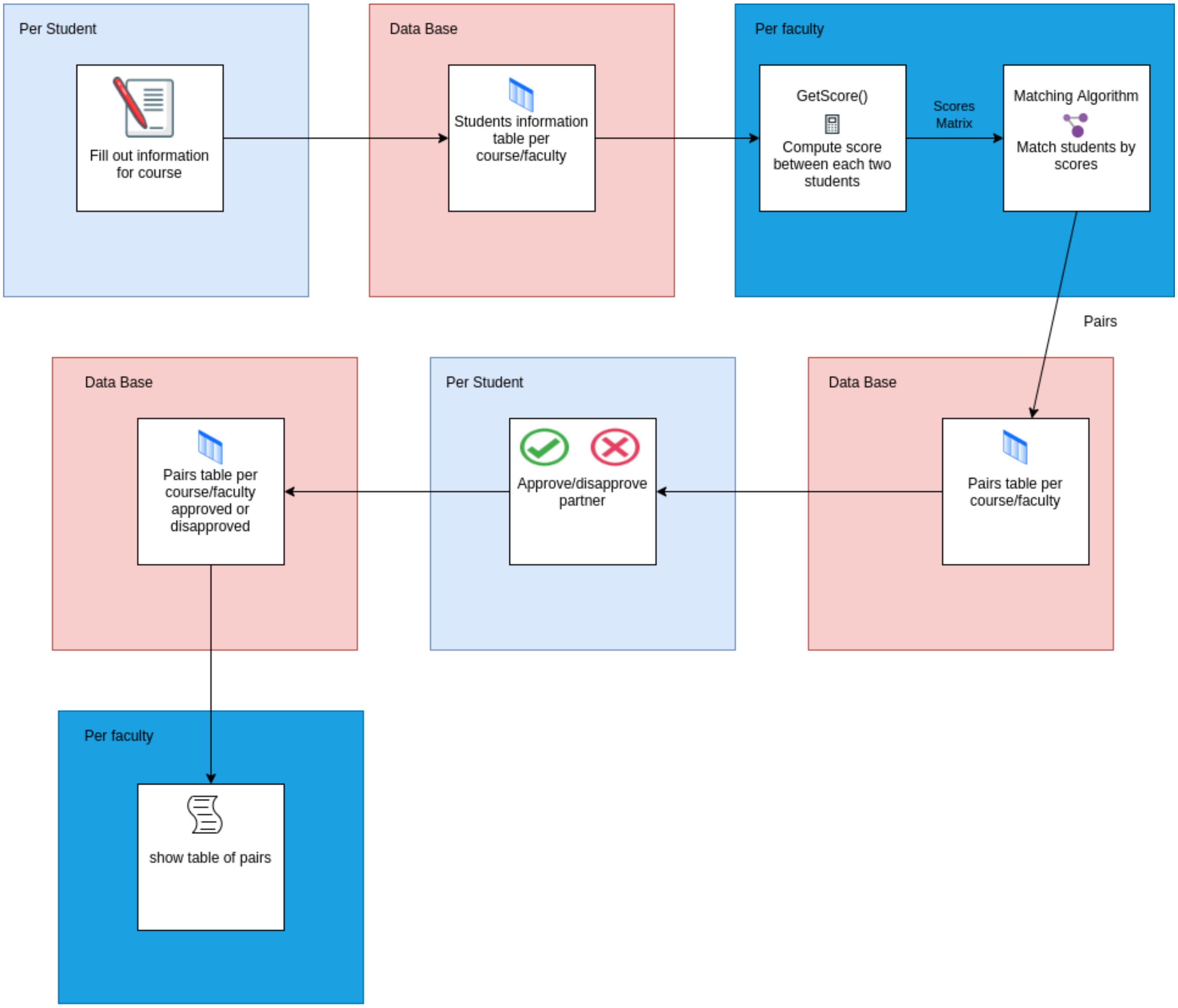
- Students are partnered with each other based on personalized profile, preferences and criterion filled out when registering to each course/project.
- Students spend a lot of time in searching for the right person, the application contains an algorithm that will find a pair for each student, and save a lot of time.
- Complete and arranged data of paired students for the management/faculty to view.

Overview

- ➔ Automated pair generation per faculty and course through the matching of students.
  + show paired students
  + ability to merge new entered students
  + ability to delete and manage the resulted pairs.
- ➔ Students can easily register to each course, manage their data and accept or decline the proposed partner.

Tools and architecture



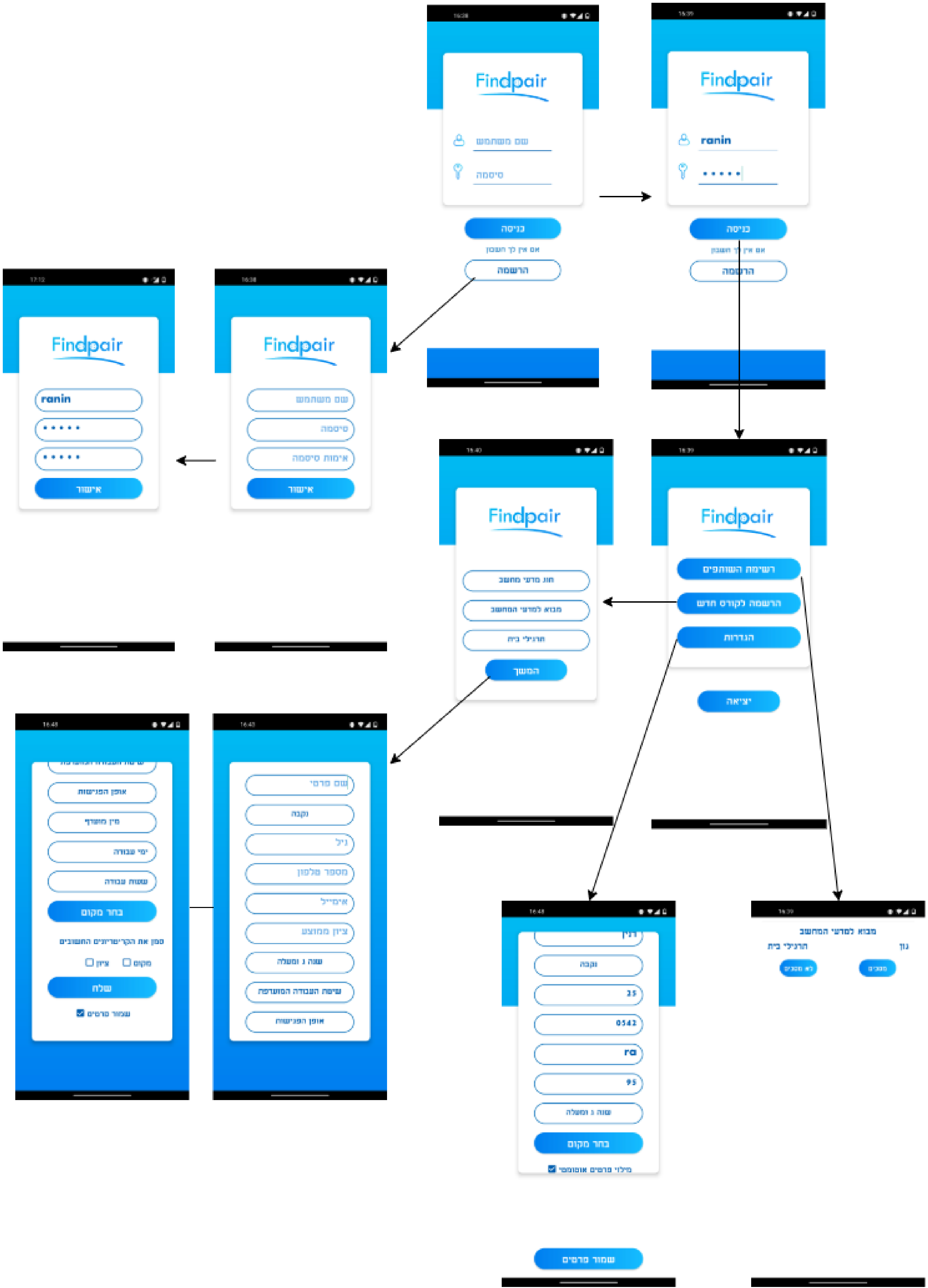

Per Student
Fill out information for course
Data Base
Students information table per course/faculty
Per faculty
GetScore()
Compute score between each two students
Scores Matrix
Matching Algorithm
Match students by scores
Pairs
Data Base
Pairs table per course/faculty
Per Student
Approve/disapprove partner
Data Base
Pairs table per course/faculty approved or disapproved
Per faculty
show table of pairs

GetScore: A function that calculates the similarity between every two students, either for all the students that registered or in each course and work type. It builds a matrix of size (nxn) when n is the number of the students that registered. In each (i,j) in the matrix it shows the score between student i and student j.
The similarity between each two students is calculated by these criterion:
Location: the smaller the distance the bigger the score of this criterion, the distance is relative to a fixed maximum distance that we determined. Calculated when the importance of this criterion is checked.
Working way: when students chose similar type of working way this criterions score is 100, otherwise its 0. When the option is “doesn’t matter” this criterion gets 100.
Preferred gender: when students chose similar preferred gender this criterions score is 100, otherwise its 0. When the option is “doesn’t matter” this criterion gets 100.
Meetings: when students chose similar preferred meeting way this criterions score is 100, otherwise its 0. When the option is “doesn’t matter” this criterion gets 100.
Working days/hours: the score of this criterion is based on the similarity of the chosen days, and hours of work.
GPA: the higher the gpa the higher the score, if the importance of this criterions is checked then the score is higher.
Based on all this criterions and their scores the function calculates the total score of the similarity between each two students, when the criterions is on “doesn’t matter” the weight of it is smaller.
Time complexity: O(n^2)

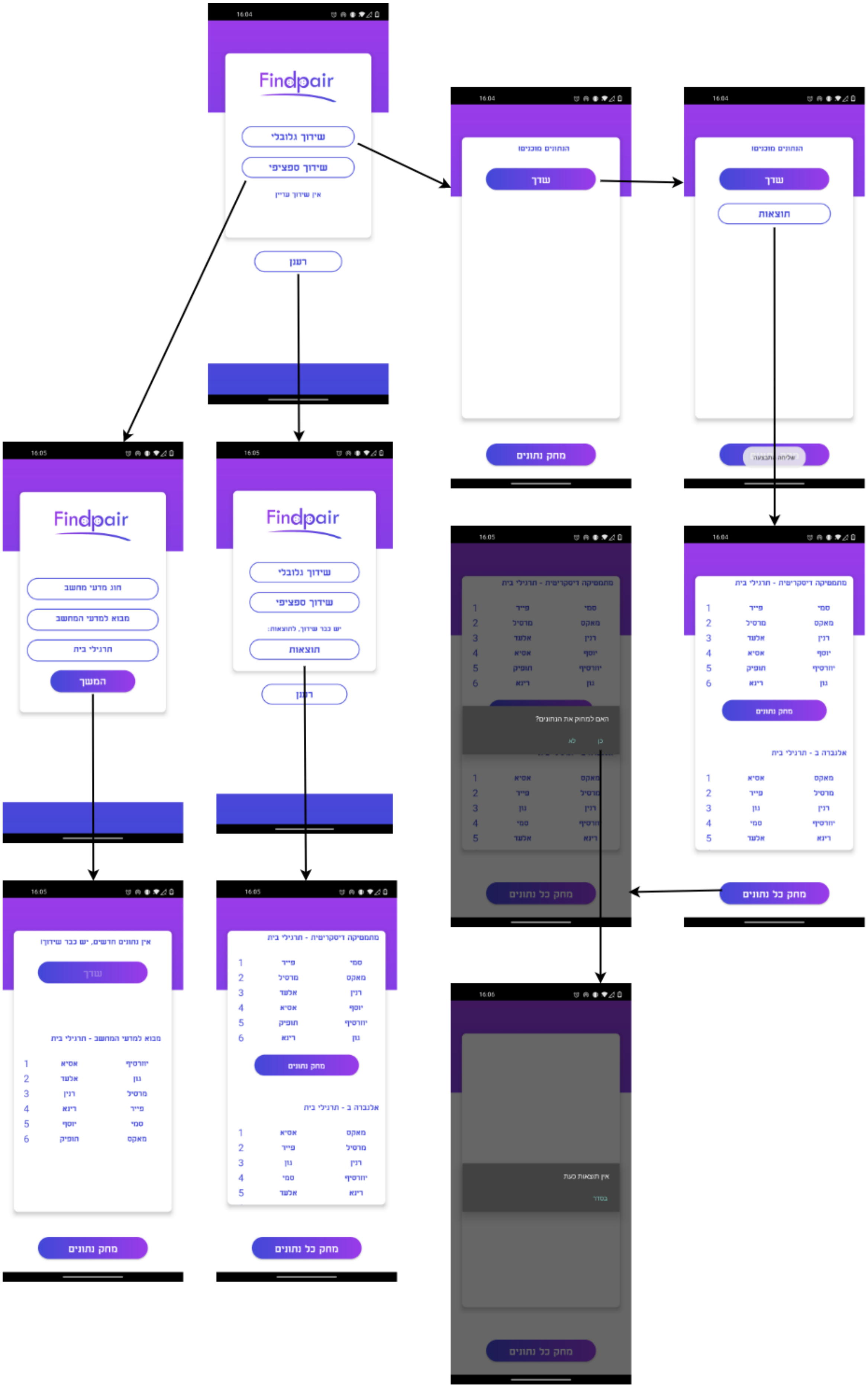
Matching Algorithm: We looked into different current algorithms that could be a good fit to our matching problem, and decided to use **Hungarian Algorithm**:
https://en.wikipedia.org/wiki/Hungarian_algorithm
We used an already implemented code of this algorithm, that takes the matrix of the scores we calculated before between all the students, and returns the matched pairs. Because the algorithm returns pairs with the least number of scores , we made the matrix negative before using it.
implementation of the algo:
https://github.com/aalmi/HungarianAlgorithm
Time complexity: O(n^3)

User Interface

Client/student



Server/faculty



מתמטיקה דיסקריטית - תרגילי בית

1	פייר	סמי
2	מרסיל	מאקס
3	אלעד	רנין
4	אסיא	יוסף
5	תופיק	יורסיף
6	רינא	נון

אלגברה ב - תרגילי בית

1	אסיא	מאקס
2	פייר	מרסיל
3	נון	רנין
4	סמי	יורסיף
5	אלעד	רינא

האים למחוק את הנתונים?

אין נתונים חדשים, יש כבר שידוך!

מבוא למדעי המחשב - תרגילי בית

1	אסיא	יורסיף
2	אלעד	נון
3	רנין	מרסיל
4	רינא	פייר
5	יוסף	סמי
6	חופיק	מאקס

מתמטיקה דיסקריטית - תרגילי בית

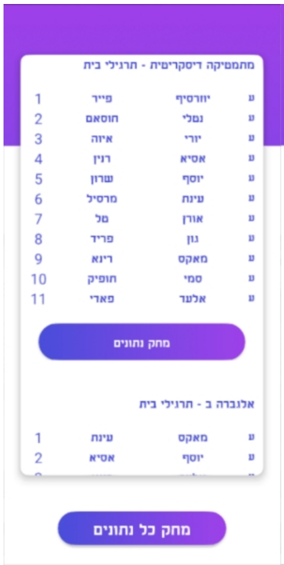
1	פייר	סמי
2	מרסיל	מאקס
3	אלעד	רנין
4	אסיא	יוסף
5	תופיק	יורסיף
6	רינא	נון

אלגברה ב - תרגילי בית

1	אסיא	מאקס
2	פייר	מרסיל
3	נון	רנין
4	סמי	יורסיף
5	אלעד	רינא

אין תוצאות כעת

Scenario from the server side: 24 students that sent 115 requests for 5 courses. The matching will give us:

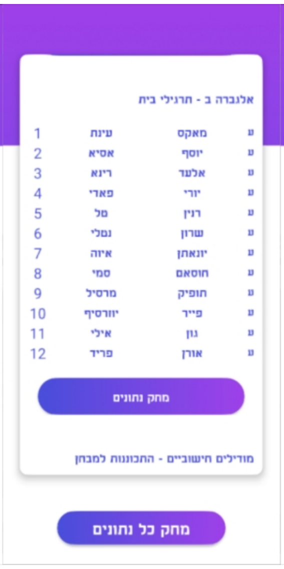


מתמטיקה דיסקריטית - תרגילי בית

ע	יוורסיף	פייר	1
ע	נמלי	חוסאם	2
ע	יורי	איוה	3
ע	אסיא	רנין	4
ע	יוסף	שרון	5
ע	עינת	מרסיל	6
ע	אורן	טל	7
ע	גון	פריד	8
ע	מאקס	רינא	9
ע	סמי	תופיק	10
ע	אלעד	פאדי	11

אלגברה ב - תרגילי בית

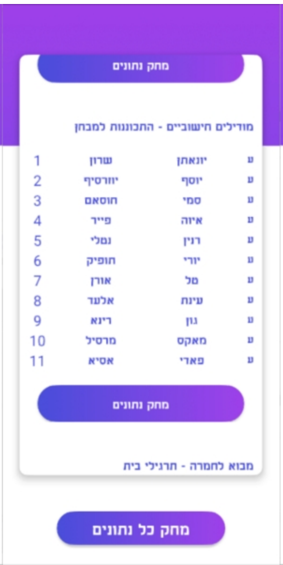
ע	מאקס	עינת	1
ע	יוסף	אסיא	2



אלגברה ב - תרגילי בית

ע	מאקס	עינת	1
ע	יוסף	אסיא	2
ע	אלעד	רינא	3
ע	יורי	פאדי	4
ע	רנין	טל	5
ע	שרון	נמלי	6
ע	יונאתן	איוה	7
ע	חוסאם	סמי	8
ע	תופיק	מרסיל	9
ע	פייר	יוורסיף	10
ע	גון	אילי	11
ע	אורן	פריד	12

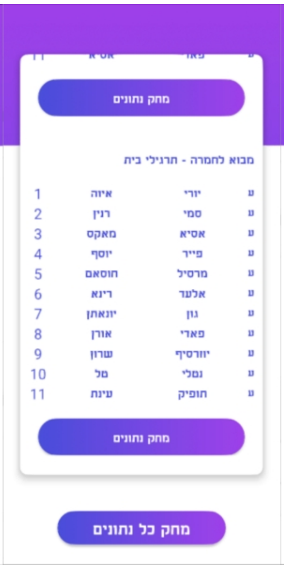
מודילים חישוביים - התכוננות למבחן



מודילים חישוביים - התכוננות למבחן

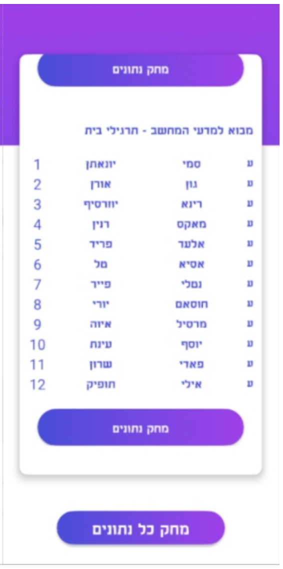
ע	יונאתן	שרון	1
ע	יוסף	יוורסיף	2
ע	סמי	חוסאם	3
ע	איוה	פייר	4
ע	רנין	נמלי	5
ע	יורי	תופיק	6
ע	טל	אורן	7
ע	עינת	אלעד	8
ע	גון	רינא	9
ע	מאקס	מרסיל	10
ע	פאדי	אסיא	11

מבוא לחמרה - תרגילי בית



מבוא לחמרה - תרגילי בית

ע	יורי	איוה	1
ע	סמי	רנין	2
ע	אסיא	מאקס	3
ע	פייר	יוסף	4
ע	מרסיל	חוסאם	5
ע	אלעד	רינא	6
ע	גון	יונאתן	7
ע	פאדי	אורן	8
ע	יוורסיף	שרון	9
ע	נמלי	טל	10
ע	תופיק	עינת	11



מבוא למדעי המחשב - תרגילי בית

ע	סמי	יונאתן	1
ע	גון	אורן	2
ע	רינא	יוורסיף	3
ע	מאקס	רנין	4
ע	אלעד	פריד	5
ע	אסיא	טל	6
ע	נמלי	פייר	7
ע	חוסאם	יורי	8
ע	מרסיל	איוה	9
ע	יוסף	עינת	10
ע	פאדי	שרון	11
ע	אילי	תופיק	12