Report On

My College Roomie: Roommate Finder Application

Submitted in partial fulfillment of the requirements of the Course project in Semester IV of Second Year Computer Engineering

by

Vipul Naresh Bhoir (Roll No. 06)

Mrudul Parag Chaudhari (Roll No. 11)

Abhinav Bhimrao Desai (Roll No. 13)

Supervisor
Dr Megha Trivedi

University of Mumbai

Vidyavardhini's College of Engineering & Technology

(2022-23)
Department of Computer Engineering

Vidyavardhini's College of Engineering & Technology

Department of Computer Engineering

CERTIFICATE

This is to certify that the project entitled "**My College Roomie: Roommate Finder Application**" is a bonafide work of "Vipul Bhoir (Roll No. 06), Mrudul Chaudhari (Roll No. 11), Abhinav Desai (Roll No. 13)" " submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester IV of Second Year Computer Engineering.

Supervisor

Dr Megha Trivedi

Internal Examiner	External Examiner

Dr Megha Trivedi Head of Department	Dr. H.V. Vankudre Principal

Abstract :

In the modern world, lodging prices have been rising quickly. Finding a shelter that fits one's preferences, finances, interests, and location is also difficult. If the person seeking housing is a student, the issue is made considerably worse. For students, elements including cost, distance from the university, and similar

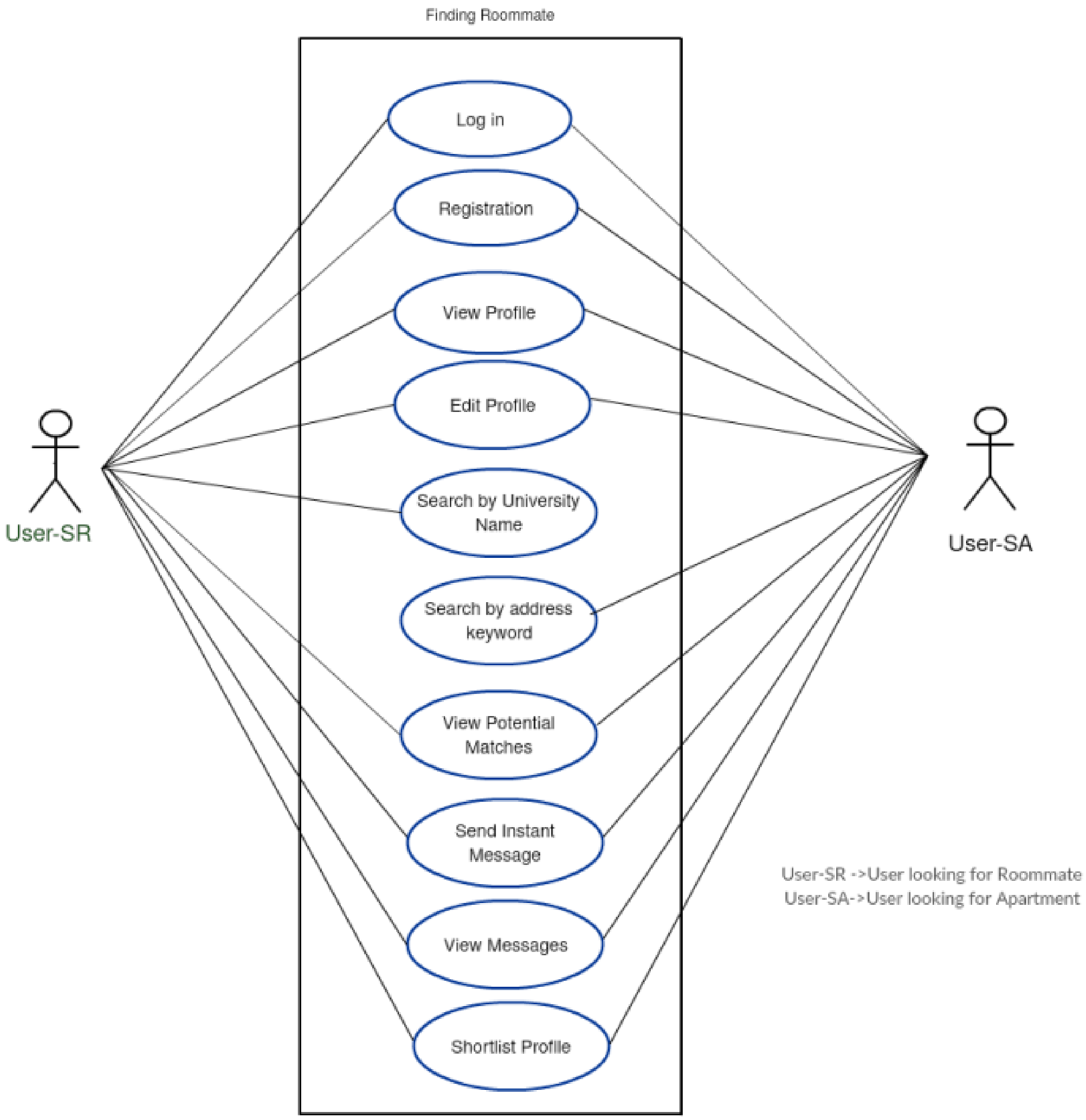
Company, etc., is most important. There are number of websites and mobile apps that provide facilities for finding suitable roommate and vacant apartment, but as of now, there is no such application that helps to find roommate or apartment for a specific College or University. This application is aimed at trying to solve the major accommodation problem for College or University students.This application provides seamless process of searching roommate and apartment. It has rich features like sending messages, searching based on university name and address, potential match based on user's preferences and shortlisting. For developing eye catching and interactive user interface.

Contents **Pg. No**

Problem Statement :

- This project is to create a roommate finder application so that student can find roommate as per convenient.
- To develop application by which student can find roommate easily.
- Due to some economic reasons many students or people must live with roommates. Often roommate issues arise from disputes over cleanliness in common areas.
- The proposed system will have two types of users:
  1. Users looking for an apartment close to their university
  2. Users having an apartment, but looking for a roommate, preferably from the same university.
- Once the user is successfully logged in, the app provides access to the user to view and update profile and preferences.

Work Flow and Working of Proposed System :



User-SR ->User looking for Roommate
User-SA->User looking for Apartment

Figure 1: Use Case Diagram

Module Description :

1] User Options: This app provides following features to the user:

- Once the user logs in the app, the type of the user is determined. The current user can be either looking for an apartment or a roommate.
- User can register for app and fill out preferences and information.
- User can view profile and preferences.
- User has access to update information through edit profile.
- User looking for roommate can search by selecting university name.
- User can view potential match with percentage based on their preferences, interests and university.
- User can shortlist other users

2] Login: After opening the website, login page will be displayed. By entering valid username and password, users will get access to respective app features according to their role. By entering wrong username or password, users will get error message.

3] Registration: After opening the website, user can sign up. For that, user has to provide purpose of registration. After that second step would be to enter basic information like name, contact details, date of birth, gender etc. Third step is university related information like university name, year of joining, year of graduation, intended major, an education level. Fourth step for user looking for apartment is to add apartment preferences like budget, sharing preference of room and apartment, type of home they are looking for and ready to move in date. Fourth step for user looking for roommate is to add apartment details like rent, no of roommates, apartment address, apartment website, apartment images and availability date. Fifth step is to add interests and activities. Sixth step is for general preferences like gender preference, eating preference, smoking preference and additional notes.

4] View profile: This page will display all the details that a user entered during registration process. Users can view basic information, university related information, apartment preference for user looking for apartment and apartment details for user looking for roommate, interests and activities and general preferences.

5] Edit profile: After logging in, user can view profile page and by clicking on edit button icon user can make changes to existing records.

6] Search for user looking for roommate: After logging in to website and clicking on Search from menu, user will be redirected to search page. By selecting university from list, list of users looking for apartment will be shown. By selecting any one of them, detailed information page will be shown. If user clicks on “Add to Shortlist” icon, that user will be added to shortlisted profiles. If user clicks on “Send Message” icon, one dialog box will be open, after writing message user can send message to other user by clicking “Send” icon.

7] Potential Match for user looking for roommate: After logging in to website and clicking on Potential match from menu, user will get all the preferred matches with percentage will be displayed on page. By selecting any one of them, detailed information page will be shown. If user clicks on “Add to Shortlist” icon, that user will be added to shortlisted profiles. If user clicks on “Send Message” icon, one dialog box will be open, after writing message user can send message to other user by clicking “Send” icon.

Brief Description of Software & Hardware used and its programming :

- **Front-end technologies**: This includes the user interface design, client-side scripting, and markup languages. Commonly used front-end technologies include HTML, CSS, JavaScript.

- **Back-end technologies**: This includes the server-side programming languages, databases, and web servers. Popular choices for back-end technologies include Node.js, Python, and Django. Databases like MySQL are commonly used to store and manage user data.

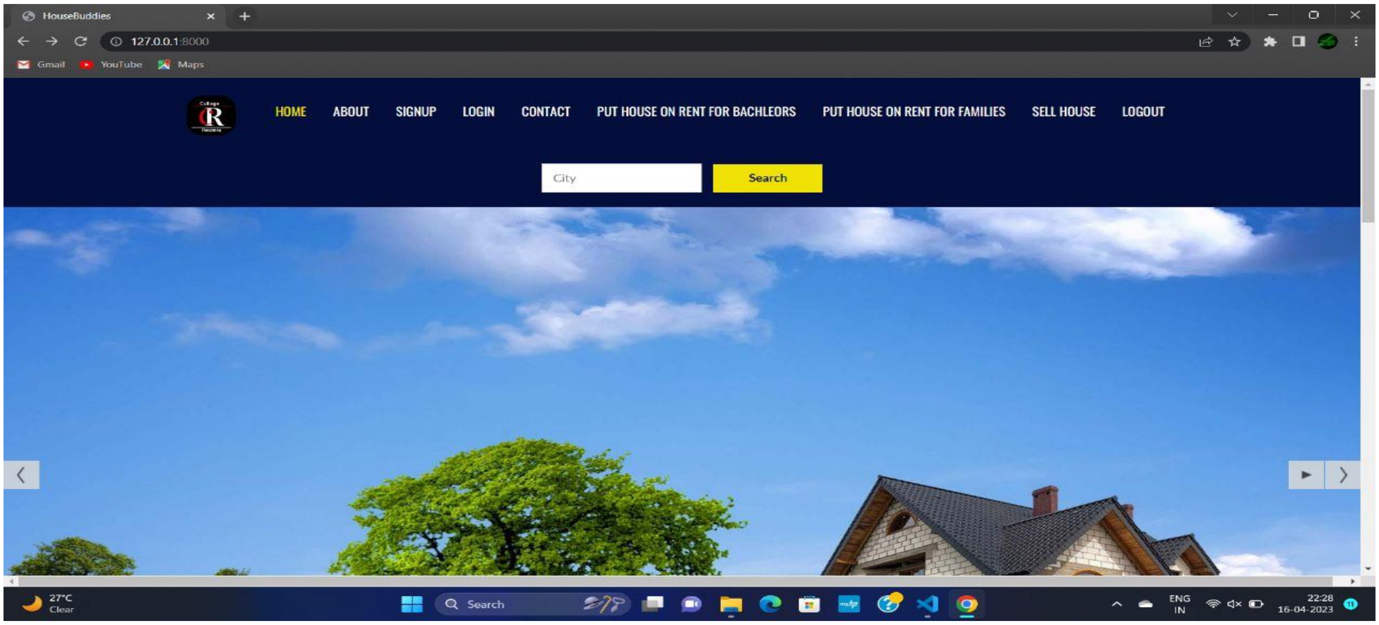
- **Mobile Application**: To extend the reach of the website, a mobile application can be developed that can use the API exposed by the website.

Code :

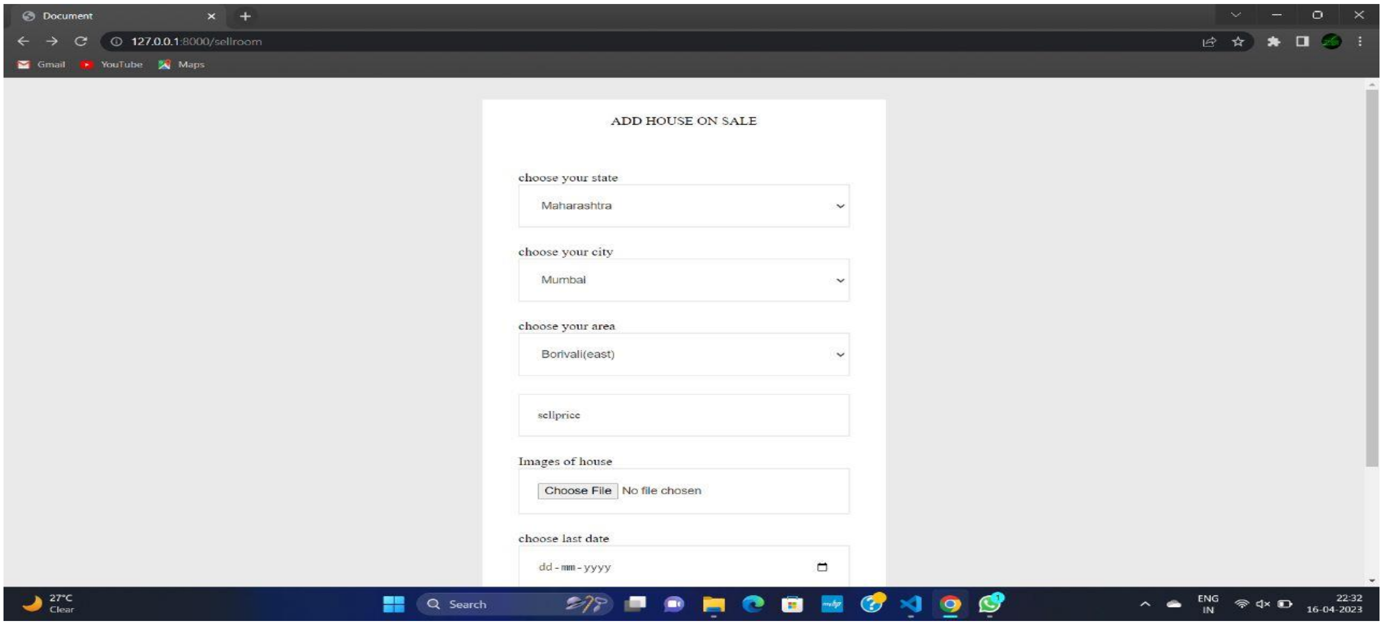
```
from django.db import models

# Create your models here.
class Contact(models.Model):
    name = models.TextField()
    email = models.EmailField()
    phone = models.CharField(max_length=10)
    desc = models.TextField()
    # date = models.DateField()

    def __str__(self):
        return self.name



gender_choice = (
            ("Male", "Male"),
            ("Female", "Female"),
            ("Others", "Others"),
)



class Register(models.Model):
    username = models.TextField()
    firstname = models.TextField()
    qualification = models.TextField(default = 0)
    password = models.TextField(default=0)
    lastname = models.TextField()
    birthdate = models.DateField()
    gender = models.TextField(choices=gender_choice, default = True)
    email = models.EmailField()
    phone = models.CharField(max_length=10)
    # upload = models.ImageField(upload_to ='uploads')
    file=models.FileField()

    def __str__(self):
        return self.username
```

```
class Roommate(models.Model):
    username = models.TextField()
    firstname = models.TextField()
    lastname = models.TextField()
    birthdate = models.DateField()
    email = models.EmailField()
    phone = models.CharField(max_length=10)
    # upload = models.ImageField(upload_to ='uploads')
    file=models.FileField()
    baseaddress = models.TextField()

    def __str__(self):
        return self.username



state_choice = (
            ("Maharashtra", "Maharashtra"),
)
city_choice = (
            ("Mumbai", "Mumbai"),
)
area_choice = (
            ("Borivali(east)", "Borivali(east)"),
            ("Borivali(West)", "Borivali(west)"),
            ("Kandivali(east)", "Kandivali(east)"),
            ("Kandivali(west)", "Kandivali(west)"),
            ("Dahisar(east)", "Dahisar(east)"),
            ("Dahisar(West)", "Dahisar(west)"),
            ("Miraroad(east)", "Miraroad(east)"),
            ("Miraroad(West)", "Miraroad(west)"),
            ("Bhaindar(east)", "Bhaindar(east)"),
            ("Bhaindar(West)", "Bhaindar(west)"),
            ("Naigoan(east)", "Naigoan(east)"),
            ("Naigoan(West)", "Naigoan(west)"),
            ("Vasairoad(east)", "Vasairoad(east)"),
            ("Vasairoad(West)", "Vasairoad(west)"),
            ("Nalasopara(east)", "Nalasopara(east)"),
            ("Nalasopara(West)", "Nalasopara(west)"),
            ("Virar(east)", "Virar(east)"),
            ("Virar(West)", "Virar(west)"),
)
no_choice = (
            ("0", "0"),
            ("1", "1"),
            ("2", "2"),
            ("3", "3"),
            ("4", "4"),
            ("5", "5"),
            ("6", "6"),
            ("7", "7"),
            ("8", "8"),
            ("9", "9"),
            ("10", "10"),
)
class Vacantroom(models.Model):
    username = models.TextField()
    address = models.TextField()
    state = models.TextField(choices = state_choice, default = 'Maharashtra')
    city = models.TextField(choices = city_choice, default = 'Mumbai')
    area = models.TextField(choices = area_choice, default = 'Kandivali(East)')
    room_mate_present = models.IntegerField(choices = no_choice, default = 0)
    room_mate_require = models.IntegerField(choices = no_choice, default = 0)
    rent = models.IntegerField()
    file=models.FileField()
    lastdate = models.DateField()

    def __str__(self):
        return self.address

class Vacanthouse(models.Model):
    username = models.TextField()
    address = models.TextField()
    state = models.TextField(choices = state_choice, default = 'Maharashtra')
    city = models.TextField(choices = city_choice, default = 'Mumbai')
    area = models.TextField(choices = area_choice, default = 'Kandivali(East)')
    rent = models.IntegerField()
    file=models.FileField()
    lastdate = models.DateField()
    def __str__(self):
        return self.address


class Sellroom(models.Model):
    username = models.TextField()
    address = models.TextField()
    state = models.TextField(choices = state_choice, default = 'Maharashtra')
    city = models.TextField(choices = city_choice, default = 'Mumbai')
    area = models.TextField(choices = area_choice, default = 'Kandivali(East)')
    sellprice = models.IntegerField()
    file=models.FileField()
    lastdate = models.DateField()

    def __str__(self):
        return self.address



class Req(models.Model):
    ownername = models.TextField(default = 0)
    buyername = models.TextField(default = 0)
    gender = models.TextField(default = 0)
    email = models.TextField(default = 0)
    rent = models.TextField(default = 0)
    address = models.TextField(default = 0)
    qualification = models.TextField(default = 0)
```

Output and User Interface :



This is the interface of My College Roomie after login .Firstly, You have to make account than using that account you have to do login and above screen will be displayed.



ADD HOUSE ON SALE

choose your state

Maharashtra

choose your city

Mumbai

choose your area

Borivali(east)

sellprice

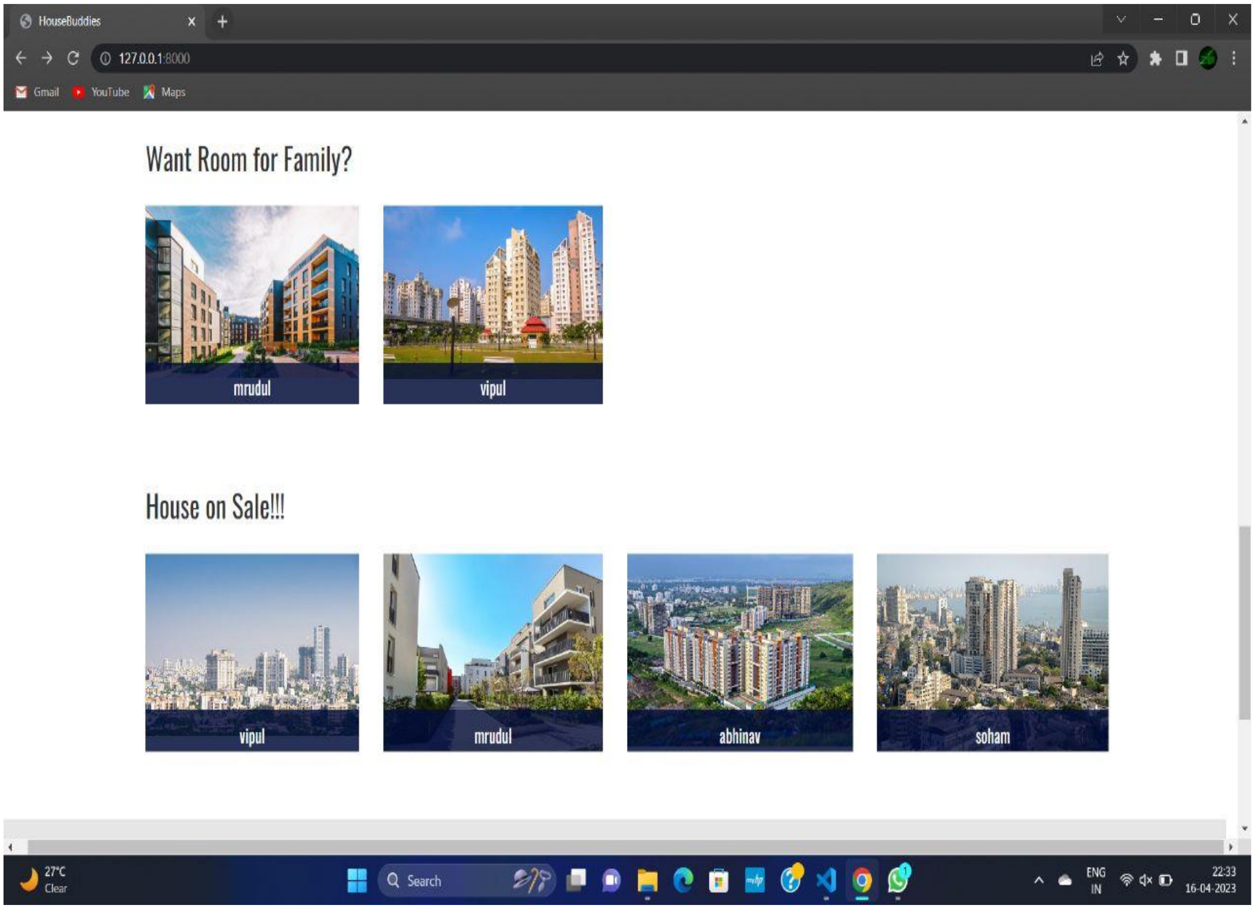
Images of house

Choose File No file chosen

choose last date

dd-mm-yyyy

Here we are going to add the house information for sale so that buyer can see all the information of house.These all features comes under profile section.



Want Room for Family?

mrudul

vipul

House on Sale!!!

vipul

mrudul

abhinav

soham

Here on this page we can see the house which are on sale.

Conclusion :

This website is extremely useful for finding roommate or vacant apartment around your College or University campus. It is easy to access and its features can help to search preferred matches. This application can be used by wide range of people as it satisfies need for two types of users. Moreover, it provides different type of communication to connect to two users like one-to-one messages, text message to other user"s phone, and direct email to other users.

References :

[1]. James Henry Behrens, “The utility of the FIRO-B for the study of college roommate compatibility”, Cornell University, 1976
[2]. R.W.Irving, D.F.Manlove, “The stable roommates problem with ties”, J.Algorithms, vol.43, no.1, 85-105, 2002.
[3]. M.Peski, “Large roommate problem with non-transferable random utility”, J.Econ. Theory, vol.168, pp.432-471, Mar.
[4]. E.Ronn, ”NP-complete stable matching problems”, J.Algorithms, vol.11, no.2, pp.285-304, 1990.
[5]. P.H.Chan, X.Huang, Z.Liu, C.Zhang, S.Zhang, “Assignment and pricing in roommate market”, Proc.13th AAAI Conf. Artif. Intell. (AAAI), pp.446-452, Feb.2016.
[6]. G.Huzhang, X.Huang, S.Zhang, X.Bei, “Online roommate allocation problem”, Proc.27th Int. Joint Conf. Artif.Intell.(IJCAI), pp.235-241, Aug.2017.
[7]. R.W.Irving, “An efficient algorithm for the „stable roommates" problem”, J.Algorithms, vol.6, no.4, pp.577-595, 1985.
[8]. K.Cechlarova, “On the complexity of exchange-stable roommates”, Discrete Apps.Math., vol.116, no.3, pp.279-287, 2002.
[9]. J.Barthoidi, M.A.Trick, “Stable matching with preference derived from a psychological model”, Oper. Res. Lett, vol.5, no.4, pp.165-169, 1986.