Group 05: Mavericks

Seat Allocation

System Requirements Specification

Prepared for
Software Systems Lab Project
Instructor: Sharat Chandran

1 All tools applied

1) Java, Javadoc
2) Python, Django, html
3) Javascript in django (seat.html)
4) makefile
5) shell script
6) top, time (use explained in readme)
7) Latex, Bibtext
8) git repo

2 Introduction

The project tries to implement two algorithms for college admissions namely Gale-Shapley algorithm (modified to cater reservation) and Merit Order Admission. [3]

3 Requirements

JVM must be installed in the system
Javadoc must be installed in the system
python version 3.4.2 and Django version 1.7 must be installed in the system.
4 Package Structure for Lab 10

4.1 Packages

4.1.1 common
Candidate.java VirtualProgramme.java Main.java

4.1.2 GS
GaleShapleyAdmission.java Meritlist.java (For GaleShapley)

4.1.3 MeritOrder
MeritOrderAdmission.java Meritlist.java (For merit order)

5 Limitations of Algorithm
The algorithm we have been asked to implement has following drawbacks: 1) Both the algorithms happen to be unfair (not optimal) for the DS candidate who might have got a better seat through normal allocations (without considering DS allocations separately). 2) In MeritOrderAdmission candidate not featuring in multiple lists can suffer disadvantage even with a better rank in category list. 3) MO: A candidate’s seat won’t be upgraded, even when better options are available after dereservation.

6 Lab 11

6.1 Files in Lab 11

6.2 ./seatalloc
_init_.py admin.py migrations tests.py views.py __pycache__ forms.py models.py urls.py

6.3 ./templates/seatalloc
add_preference.html enterrank.html seat.html candidate.html freeze_preference.html detail.html index.html
6.4 ./seatalloc/migrations/
It contains migrations.

6.5 populatedb.py
populates data (data_u-2012.csv) Database:db.sqlite3

6.6 Models

6.7 Views
1. add_candidate: adds a candidate details to database if a candidate already exists, displays error msg [2]
2. add_preference: adds preferences of a candidate one at a time to database if a candidate’s tries to add same preference gives an error (a django error, all other errors are displayed by us as text on html)
3. candidate: displays all preferences of candidate gives addpreference and freeze preference feature displays error if candidate does not exist
4.freeze:writes preferences to mydata.csv
5. enterrank: user enters rank here
6. seatfinder: finds all seats a (rank X category) is eligible to as well as the seats which have been more preferred

6.8 HTML
1. open page http://127.0.0.1:8000/seatalloc/ (index.html)
2. To add your preferences go to http://127.0.0.1:8000/seatalloc/detail/your_rollno/ (this link is not directly reachable by any other page) (add_preference.html)
4.seat.html: seatfinder
5.freeze_preference.html:frozen preference
6.enterrank.html

6.9 Files produced
mydata.csv:contains preferences of candidates
7 Features

1. Unique value of roll no
2. Candidate can’t add same preference twice
3. Freeze Preference option: freeze the preference and written in mydata.csv in the form of choices.csv Preferences once frozen should not be changed again. Since this causes overwriting data in mydata.csv
4. Seat finder gives a descriptive evaluation of seats available in each institute. It also provides 2 lists in each institute, one that is more preferred and other the eligible ones
5. Url system: each candidate’s details and add preference option are provided at its unique url (Considering only the candidate is aware of his/her username, only he can access his details)
6. Admin interface: http://127.0.0.1:8000/admin

8 Limitations

1) We have not checked for input validation in roll no. field (any string length ≠ 9 permitted), and rank (any integer permitted). So please enter valid values

References