IITG Validation Team's Interim Report on Joint Seat Allocation Software

29th May, 2015 (11:15)

Contents

This document reports only the errors and concerns with regard to joint seat allocation software. We have omitted the observations where we have not observed any problem.

1 Introduction

IIT Guwahati team attended a meeting held at NIC office on 14-May-2015 whose objective was to fetch the *Joint Seat Allocation* software developed at NIC, New Delhi. An informal writeup about the meeting was shared to IIT Guwahati JEE Vice Chairman on 16-May-2015. For a quick reference the same is reproduced here. The reader may skip this particular thread and can directly move to Section 2.

(i) *Question:* Did your teams get the software yesterday?

Response: IITG team did get latest version of the software from NIC on 14-May-2015.

(ii) *Question:* Did they run some trial runs on the local installations?

Response: Yes. IITG team ran the software installed on local machine. Test run was performed on the sample dataset provided to us by NIC which contains 1026 candidates details, seat matrix information and candidate choices.

(iii) *Question:* Do they have sample input/output files i.e., if they use a given sample input, they should get the same sample output as is getting generated on the NIC machine. This basic test is essential and lets us know that the installation on the local machine is the same as that on the NIC machine.

Response: NIC team provided sample input; However, corresponding sample output was not explicitly provided. By sample output we mean expected output corresponding to the sample input.

When run the software on local machine with the input data provided by NIC we did get the same output as that generated by NIC team. The method of comparison of the output produced is to compare the allotment table as well as the summary sheet provided by the JoSAA software. These two methods are in agreement.

Validation modules available in the NIC software were also executed with a success.

(iv) *Question:* According to your teams, are the formats for the various input and output databases frozen?

Response: Input to the JoSAA software comprises of three elements namely:

- `Eligible_candidate_JoSAA`: This table has 62 columns (in the version that was shipped to us) corresponding to candidate details. Details of 62 columns can be found in the `Master-Directory-JoSAA` excel file. During the 14-May-2015 meeting this particular input format has been frozen. IITG team is testing JoSAA software based on 62 columns input.

There is a discussion on a particular column namely `MarksChange` and its need to split into two columns namely `MarksChangeEng` and `MarksChangeArc`. This is due to the business rule III (4) which elaborates the eligibility for a candidate to participate in joint seat allocation. Note that the computation varies for engineering and architecture. This particular aspect need to be captured by the column in the table.

- `Seat Matrix:` This matrix consists of 16 columns corresponding to each institute's available seats in every branch. Available seats are further detailed according to every category. Details of each of the column is described in the `Master-Directory-JoSAA` excel file. This input element format also has been frozen.

  Follow up question: Seat matrix table: we have been asking the limitations / restrictions [e.g., female candidates not allowed, etc.] for academic program to be read dynamically. However, I do recall seeing these hard-coded in the seat matrix table. What is the current status?

  Response: Limitations or Restrictions are taken out of the seat matrix. We are told by the NIC team that these restrictions are taken care by another `software` which is responsible for collecting candidate's preferences. This software, at the time of preference collection **filter out** institutes and/or branches as per the limitations/restrictions. This software is responsible for displaying only those institutions and branches that a candidate is *eligible*.

  As per the IITG team understanding, choice matrix does not include any choice which potentially violates limitations/restrictions. NIC team has walked through a presentation but working prototype was not demonstrated during the meeting.

- `Choice Matrix:` This matrix consists of candidate's choices described in five columns. Once again, details of each column can be found in the `Master-Directory-JoSAA` excel file. This table format is also frozen as of 14-May-2015.

(v) *Question:* Does the software work for all four rounds, can handle prep course allocation and DS category preferential allocation?

Response: We were provided with a demonstration on single round allocation. When asked about the allocation for subsequent rounds, we were told that changes to Eligible_candidate_JoSAA table will be necessary to handle subsequent rounds allocation. However, no explicit test cases were demonstrated during the meeting.

Follow up question: What is preventing changing the eligible_candidate_JoSAA table? Are the synthetic data generating teams ready with data for 2^{nd} and subsequent rounds i.e., including freezing / floating / sliding options?

Response: As per the IITG validation teams understanding the only place where allocation in rounds 2, 3 and 4 can be realized by *changing the decision* column of the

`Eligible_candidate_JoSAA` table. We are not aware of the recent efforts from the data generation teams for these rounds.

(vi) *Question:* Does the software work for all four rounds, can handle prep course allocation and DS category preferential allocation?

Response: An elaborate discussion is provided in Section 5 titled **Tested Business Rules**.

2 Preliminaries

(i) **Fresh Installation:** First step in running this software is meeting system requirements for installing this software. As per the NIC team, this software require:

- A 4 GB RAM computer.
- Windows 7.0 or higher version Operating System.
- MS SQL software with version 8.1 or higher.
- .NET software with version 4.0 or higher.

The `setup` executable file in the `JoSAA` folder guides the installation process.

Remark: We have faced some issues even after meeting the above configuration requirements. Inspite of multiple attempts we could not install the software on *freshly* installed Windows 7.0 operating system. We could only make progress after installing Windows 8.1 operating system. A closer look at the MS SQL software's internal configuration parameters did solve the installation problems. These are described Appendix.

(ii) **Steps in the joint seat allocation software:** There are three steps to be performed for completing allocation. These are:

(a) **Step 1 - Creating Database:** This step creates a fresh database and is ready to load the three elements into this database. This is a straight forward step and no issues found in this step.

(b) **Step 2 - Import Data: Input File Format:** This software expect an `mdb` file consisting of three input elements (tables) discussed in section 1 point iv. To begin with, we were provided with this mdb file for execution of this software. Once the data is imported into the database created in step 1, step 3 is ready to be performed.

(c) **Step 3 Allotment:** Once the data is loaded into the freshly created (or existing) database, allotment step involve a single operation namely clicking perform allotment button.

Our observations on these steps are as follows:

(i) **Step 1:** No issues found in this step.

(ii) **Step 2: Intermediate Messages:** At the time of data import (step 2 of the software), the software does not provide any user centric messages. In particular how much time is left to load the data, how much data has already been loaded etc. As the input elements are expected to be very large, this software should be more user centric. Only way to know the status of data import is to examine the respective database tables such as `candidate, choice` or `PSeat` tables. However, examining database tables should be avoided at any cost.

(iii) **Step 3:** Dashboard needs *substantial* improvements from user perspective. For example, when one does "Perform Allotment", this software does not show:

- The *round* of allotment.
- *Output of the allotment:* It is not explicitly clear from this particular step where is the output of "perform allotment". One need to open MS SQL, write an SQL query on `Allotment` table.
- *Stale Summary:* After performing every round of allocation, the summary does not change in the allotment page. Another case where stale summary is found when we load different database and/or import different data and proceed with allotment the summary page points to the *first* allotment that took place.
- *Incomplete Dashboard:* Users cannot scroll down the "Allotment" page even though some information is available.
- *Which Database?:* While performing allocation it is not clear which database is in use. This information needs to be explicitly mentioned by the software at any point of time.
- *Analysis of the allotment:* Except for summary and category wise allotment, the allotment step does not allow user to probe further about final allotment. It is not possible in this version of the software to know seat matrix status *after* allotment in the current round is complete or not. As mentioned earlier, one needs to go into database `PSeat_1` table and write an SQL query. Log messages about unallocated seats needs to be available.
- **Is this correct?** *Fixed Number of Rounds:* There are only four rounds of allotment in the entire allotment cycle. However one can potentially disobey this business rule and software does not consider this restriction into account. One can perform any number of allocation rounds.
- *Preparatory Allocation:* As per the business rule #26, only preparatory allotment has to happen in the fourth round and no other allotment. This version do not take into account this situation.
- *Undo Allotment:* For some unforeseen reason if a candidate or a set of candidate's allotment needs to be reversed (undo allocation) the present software does not allow such an option. A work around for undo is to change the

input elements and re-run the "perform allotment". However, re-run is an expensive operation and round numbers change with every run of perform allotment.

- *Parallel Execution of the Software:* When two instances of this software are executing in parallel on the *same database* with conflicting input elements, the allocation is proceeding in both instances. In our test case we have the following numbers:

# candidates	2500	# candidates	5000
# choices	837133	# choices	1664709
# seat	44195	# seat	3552

Our observation is that seat matrix is overwritten so does the candidate matrix and choice matrix. Allocation is proceeding till certain extent and operations in both instances are getting aborted not from the application software but by the database system. This behavior is undesirable.
One can also modify the candidate, choice and seat matrix *while the allocation algorithm* is in execution. There by producing incorrect allocation results.

3 Input and Output Validation

(i) **Inputs:** In the menu `Validation` an option is provided to Validate Candidate Details and Seat Information. This feature checks correctness of candidate details and seat information. In case of errors in any of the tables, software prompts - with a red color - associated reason for the errors. Example error messages are:

(a) Failed: AI_Eng_CRL_Rank. AI_Eng_CRL_PD_Rank. AI_Eng_OBC_NCL_Rank and AI_Eng_OBC_NCL_PD_Rank must be available.

(b) Failed. Institute-Branch in Seat Table not available in Master Directory.

(c) Failed. Quota Must be AI or HS.

Note: Even in the presence of some of the above errors one can proceed with allotment step. Allocation is completed with success. This raises several questions. Important one among them is: Are these errors creeps into final allocation table? In most of the cases we are unable to interpret the errors. There is no indication about which row number(s) or column entries are incorrect.

(ii) **Outputs:** After the allocation is completed, one can verify the allocation for potential merit violations through "Validation → Validate Allotment" option. We have identified series of problems with this particular option. In particular, this validation is required but it does not necessarily ensures adherence to business rules. We will elaborate these shortly in Section 5.

4 File Format Issues

This software expects an `mdb` format input file. However key questions are

(i) *How the three input elements (tables) are created?* This is important as every input element is generated at different stages by variety of methods. If these input elements exists in different formats (for example, comma separated values (CSV), excel) conversion tools are the real source where errors gets introduced into the data.

(ii) *Are these tables generated at the same source?* For example, is it the case that eligible_candidate_JoSAA and seat matrix are available at IIT Bombay, and choice matrix is generated at NIC? Who owns the data?

Answers to these questions are crucial to the validation team as the input elements are foundation in obtaining error free allocation. However, if input elements are generated at different sources, handling diverse formats becomes bottleneck.

To elaborate on this, we have a created three CSV one for each input element. As this software accepts *only one* format, generated CSV files need to be imported to the `mdb` file format. Conversion of CSV to mdb formats have pitfalls. When the candidate choice matrix has rows more than six crores (which is reasonable to estimate) conversion fails.

(iii) **Different Sources:** If these input elements are generated at different sources, will the format be adhere to the `mdb`?

(iv) **Data Import Failure Cases:** For some reason if data import fails, this software does not provide reason for import failure. It only shows technical messages. These are often understood only by seasoned programmers. One such example error message is: *String or binary data would be truncated. The statement has been terminated.* To elaborate, we had generated a particular column `Adv_RegNo` in the Eligible_candidate_JoSAA with more than 10 characters (by design). The software does not recognize this at the time of database import. Instead the software throws technical error messages.

(v) **Erroneous input** (in a particular row or a candidate data entry) was partially loaded into the database without indicating errors in the input elements and ignoring rest of the contents.

(vi) **Blank/NULL Value Issue:** In particular input elements in seat matrix when blank values are inserted the import data step throws errors and exists.

(vii) **Issues Creating mdb File:** When an mdb file is created with other methods such as importing from CSV or excel file the notion of type becomes a prominent issue. This particular point will take us to point i discussed above.

(viii) **Provision for correcting input elements:** As of now, there is no handle to correct entries in the input elements after importing data and before proceeding with allocation.

(ix) The field names are not consistent across various representations. For example, both PwD and PD are used to represent the same information. This lead to confusion in interpreting results.

(x) The ordering of columns is not consistent between external and internal representation. This is a serious issue if the data is loaded from CSV files. Note that the first row cannot be column names in CSV format as it is not accepted by Access software.

(xi) **Redundant Information in Data:** The seat matrix, which is the part of input, has a total column. Its use is not clear and we believe that it is redundant.

Fixed file format suggest the **rigidity** of this software.

5 Tested Business Rules

Following test cases were developed to validated adherence to business rules.

(i) **Multiple Students with the Same Rank:** Four candidates having *identical rank* with identical first preference and competing single seat.

Remark: The expected output should be 3 supernumerary seats to be created. However we observed that supernumerary seats were not created. In the process of allocation the seat matrix got corrupted with number of seats reading as six instead of one.

(ii) **Zero Seats in Seat Matrix:** Two candidates having CRL rank 1 and 2 competing for a branch, as first preference, in an institute where there are no seats available.

Remark: Their subsequent preferences should be allocated. However, allocation was performed and one candidate having better rank was allocated that branch. Where as the other candidate got his second preference.

(iii) **Zero Rank:** Two candidates having zero rank in every category and competing for a single seat.

Remark: We do not know how zero is being interpreted by the software. We observed that these two candidates were allocated to their first preference by supernumerary rule.

(iv) **Seat Allocation for DS Candidates #1:** There are a total of four candidates and all of them happen to be DS candidates. One of them is an OBC and can get admission through Open/OBC category. All of them happen to provide identical choices for institute/branch. Only two seats are available in Open category and 1 seat from OBC category.

Remark: Aim of this test is to check whether DS candidates are tread as regular candidates or not in the case of allocation. This test case FAILS by allocating every one in the *open category* for the same institute, the same branch even though only two seats are available. The category DS is not considered at all.

(v) **Seat Allocation for DS Candidates #2:** There are two candidates; one of them belong to GN category. This candidate has higher rank and DS candidate has lower rank. Only one seat is available in the institute and branch for which these two are competing.

Remark: DS candidate should get the seat. This test case FAILS by allocating GN candidate a seat.

(vi) **Seat Allocation for DS Candidates #3:** All the four candidates happen to be DS. Two of the DS candidates have the same rank. One of them belong to OBC and should get through OBC category. All of them competing for the same institute and branch.

Remark: In this case supernumerary seat should be created. This test case FAILS.

(vii) **Seat Allocation for DS Candidates #4:** Only two DS candidates are present and there are no seats available in open category in the institute and branch they opted.

Remark: None of them should get the seat. This test case FAILS.

(viii) **Seat Allocation for DS Candidates #5:** All the DS seats are already allocated. DS candidate rank is lower. They should not get the seat.

Remark: There is no way to code this particular test case

6 Multiround and Deferred Allotment

We could not find how software handles multiround admission process. In fact, every time the data is reloaded the software increments the round number. It is unclear how multiround data can be given as an input. Therefore, we conclude the software cannot handle multiround data.

We also could not find how software uses the "Decision" column where a candidate's *Freeze*, *Float*, *Slide* preferences are stored. It is unclear how this information can be loaded after the first round of counselling. We again conclude that the software currently ignores the "Decision" column.

7 Large Data Experiment

We have generated the three input elements namely candidate, choice and seat matrices with 10 lakh, 33 crores and 831 institute-branches respectively.

(i) *Correctness:* We observed anomalies in the allocation. For instance *excessive* supernumerary seats are created (more than 100). Note that by data generation program design, there are no candidates with the same rank. Zero rank is being interpreted differently than what we think it should be.

(ii) *Time Estimation - Importing Data:* The software took four minutes to load 1 crore choice data. Note that data loading takes considerable amount of time. During this phase if any errors are detected in the data, one needs to attempt data loading process from scratch. Please refer to point viii.

(iii) *Time Estimation - allocation:* Aim of this experiment is to test the time required for allocating 10 lakh candidates with each candidate providing 334 preferences on an average. We could successfully load candidate and seat matrices. However we could not load more than 1 crore choices. This is due to the point discussed in ii. Time taken to allocate for all candidates with 1 crore preferences is 14 minutes on a machine having configuration: Dell Precision T3600, Intel Xeon processor E5-1603 with four cores, 16 GB RAM with 1 TB SATA 7200 RPM hard drive.

8 Input Data Sanity Check

The current version cannot validate the following.

(i) Field types, and

(ii) Correlation between columns. For example ranking order of OBC_NCL and OBC_NCL_PD should be the the same.

9 Security and Data Integrity

We have concerns about the data integrity and security especially when a large amount of data is stored and manipulated by the software. The obvious problem is that candidate details, seat matrix and candidate choices are open for editing. It is unclear how this software used in the entire process which includes data collection, formatting, loading, generation of results and export in various formats for different purpose. Is it possible to verify that data is not changed directly in database tables just before allocation? How do we know that software has no malicious code that interferes with the allocation process only in certain situations?

10 Usability and Testability

Based on our experience, we believe that it is almost impossible to use this software independently without any support. The problems faced are mentioned in the other sections. The GUI displays incorrect information, gives no indication about the progress and fails to update information when an operation is done multiple times.

In the current form the software is extremely hard to test. The GUI based interaction makes it impossible to automate the testing process. It is unclear why a GUI is required.

A command line interface would have been much better which could take the three input matrices in CSV format and generate an allocation in CSV format which then can be opened using various software such as libreoffice, openoffice and MS-excel.

11 Other Issues Beyond Validation

We are concerned about the non-availability of important information about the software such as, how were business rules "mapped" to software requirement specifications, software design and methodology used, programming language used, number of lines of code, details of unit/component, integration and system testing. How many defects were found during internal testing phase? What regression tests were developed after correction of these defects? What was the duration of development and testing phases?

This information is extremely important to access the quality of the software as no amount of "external" validation can ensure coverage of all internal details. It should be mentioned that industrial software development cycles has more than 60% of the time spent on the testing phase.

12 Conclusion

We have reported our observations and concerns about verification, validation, performance, usability, data integrity, security and the role of JoSSA software in the entire allotment process. Based on our evaluation we cannot recommend JoSAA software to be used for the seat allocation in this year. The problems, that we have reported in this report, can be corrected and the software can be improved, however, this will require considerable time. We believe this is not feasible in a month's time.