International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:11, 2013

Web Application for University Internship Program Management

Prasanth Sabarish Nair, Thomas Binu, Madiajagan Muthaiyan

***Abstract*—**This paper discusses a software application to aid in the smooth functioning of a university internship program, including a student, faculty and an administration module. The software can also calculate the most apt combination of students to stations and allocate them respectively.

***Keywords*—**Academic evaluation, administration monitoring, automatic allocation system, internship, student preferences.

I. Introduction

THIS internship program, which is a part of the curriculum of any leading university, immensely contributes to the growth of the student's applicative, technical and economical awareness along with their personal advancement. Conducting and administering this process without the aid of a fully functional software application to keep track of the stations, students, faculties, academic progress, etc., could get greatly arduous and cumbersome. The functionalities of the modules are discussed below.

II. Application Software

The software, developed using PHP, MySQL, JavaScript, HTML and CSS [1], [2], helps the university to execute the internship program in a more effective and efficient manner. The different modules such as administration, faculty and student allow the application to greatly aid the department in managing the internship semester. The automatic allocation system used in this web application allocates a student to a station of their preference depending on their academic qualification and other factors which are elaborated below.

A. Administration Module

The administration department has the responsibility of making certain that all the different segments during an internship program in a university works smoothly and systematically. This software is capable of handling all of its different sections entirely through the administration module thus making the department's tasks much simpler. The different features of the portal have been explained below:

- Create, Read, Update & Delete (CRUD) [3] station details for each of the different internship semesters separately. The information such as whether or not the station is to be marked as 'invisible' thus hiding the station from all the students except those who have been shortlisted to that station and if the station requires a student with a valid driving license and a vehicle.
- CRUD faculty details including their contact information.
- CRUD student details that were provided during their preference submission.
- Execute the automated allocation system.
- Manually allocate a student to a different station if deemed necessary.
- Assign faculties to stations.
- Monitor student's academic progress by viewing their mid-term and final-term grades along with the end of semester transcript.
- Review a student internship transcript submitted by a faculty.
- View the history of stations and students of those who have participated in the internship program.
- Manage the availability of the different student portals, which are discussed below.
- Maintain backend database.

1. Automatic Student Allocation Algorithm

The script when executed allocates all the students of a particular internship semester to stations of their preference based on their academic qualification and available vacancies for stations [4], [5]. In case there is a mismatch between the number of students and the number of vacancies for stations that are accepting students from a stream, it is possible to combine similar streams so that the students belonging to one stream may be allocated to a station accepting students from a different but related stream thus minimizing the number of unallocated students. For example, in case of a shortage of vacancies for a particular stream and at the same time, if there is a surplus of requested students for an associated stream, then a combined station list could be utilized by the algorithm during its execution.

The system takes into account the students' academic qualification (CPGA, overall percentages, etc), whether or not the student was shortlisted by any station, if the student has introduced any station, the order of preference for the available stations, the number of available vacancies for that particular station, and the distance from the students' residence to the station. The distance factor would only come into play if the script fails to allocate the student to any of their initial preferences. As the algorithm traverses further along the preference list, the importance of the location of the stations and the student would gradually increase. The algorithm also checks if the student has been shortlisted by any station and allocates him/her to that station only if the station was also

Prasanth Sabarish Nair is with Best Cargo Shipping LLC (e-mail: prstsn@gmail.com).
Thomas Binu is with Gulf Survey and Engineering Services - GISTEC (e-mail: mailthomasb@gmail.com).
Madiajagan Muthaiyan is working as Professor In-Charge Info. Comm. Technology at BITS Pilani, Dubai (e-mail: jagan@bits-dubai.ac.ae).

their first preference. If the station has specifically requested students having a valid driving license and a car, then the algorithm only allocates students to these stations if they have both, a driving license and a car. The automatic allocation system has been pictorially represented in Fig. 1.

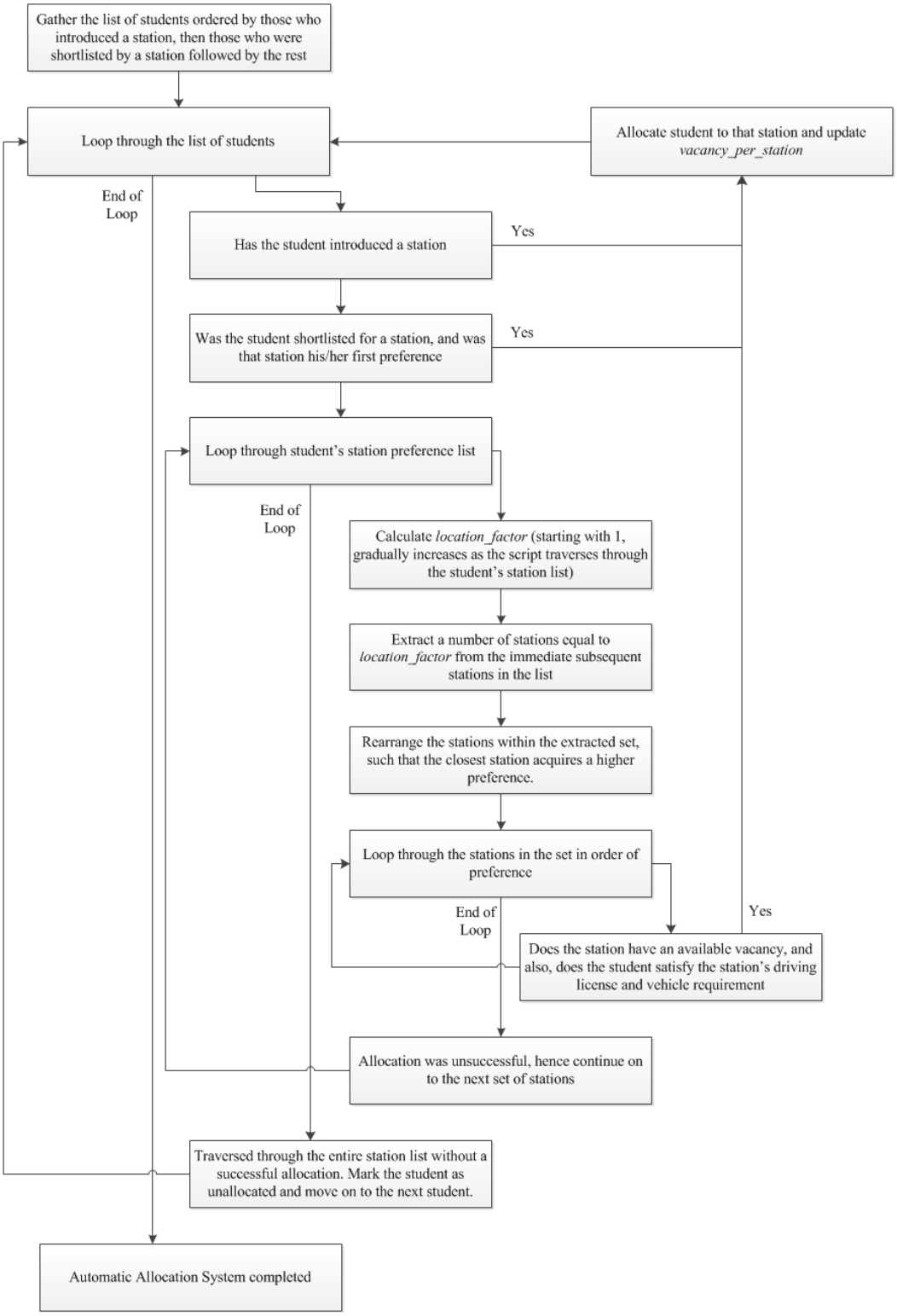

Fig. 1 Automatic Allocation System

Algorithm: Allocate Students (stream)

Input Variable:

streams (array): The combination of streams for that particular semester, for which the students have to be allocated to the appropriate industry for the internship program.

Main Internal Variables:

a. *student_list* (array): The list of students belonging to the selected stream(s) retrieved from the database contains information of the student, the station they have introduced (if any), the station they were shortlisted to (if any), their academic eligibility, if the student has a valid driving license and a vehicle, the location of residence, and the preference list of stations. This student list is arranged in such a way that those who have introduced a station are retrieved first, then those who have been shortlisted by a station, then followed by rest. Within each grouping, the students are arranged according to their academic proficiency.
b. *station_list* (array): The list of stations accepting students belonging to the selected stream(s) contains station information, number of vacancies for each selected stream, the station's vehicular requirement, and the location of the station.
c. *vacancy_per_station* (array): This variable stores the total number of available vacancies per station for the selected stream(s), and will be decremented when a student has been allocated to a station.

Pseudo-code

1. Generate a custom database query from *streams* to be used for retrieving student and station information and assign the value to *stream_query*.
2. Using *stream_query*retrieve the information from the database and populate *student_list* and *station_list*.
3. Iterate through *station_list*
   3.1 For each station, calculate total number of available vacancy for students and assign the value to *vacancy_per_station* along with the station id.
4. Iterate through *student_list*
   4.1 If the current student has introduced a station, then allocate him/her to that station and decrement *vacancy_per_station*for the station. After allocation, move on to the next student.
   4.2 If the current student has been shortlisted for a station and if the student has given the same station as his/her first preference, the student is allocated to the station, *vacancy_per_station*is updated and moves onto next student.
   4.3 If the student has neither introduced a station, nor was shortlisted for any, then iterate through their station preference list.
      4.3.1 Starting with 1, *location_factor* is incremented by 1 each time the algorithm traverses through a given percentage of number of stations in the preference list. For example, if the given percentage has the value of 10, initially the *location_factor* being 1, the algorithm upon reaching the 11th percentile station, the *location_factor* will be incremented to 2. When the algorithm reaches the 21st percentile station, it will be updated to 3.
      4.3.2 Extract a number of stations equal to *location_factor* from the immediate subsequent stations in the preference list.
      4.3.3 Rearrange the stations within the extracted set, such that the closest station acquires a higher preference.
      4.3.4 Iterate through each station within the set
         4.3.4.1. If the station has an available vacancy and if the station's vehicular requirement matches with that of the student, allocate the student, update *vacancy_per_station* for the station and move on to the next student.

B. Faculty Module

A university may have various types of internship programs with different evaluation criteria. When a faculty accesses to the application, they are required to select the pertaining internship semester. The application will then retrieve the appropriate information and functions.

It is essential that the faculties constantly and periodically evaluate their students on their performance in their respective practice school stations during the course of the internship. Using this module, all these observations can be recorded and be forwarded to the concerned authorities after satisfying the necessary validations criteria [6]. Monthly and End of semester progress reports can also be generated from the module based on the information from the database. The reports can only be generated if all of the necessary evaluation components have been recorded. If not, then the faculty would be alerted as to the cause of the failure and upon clearing the requirement they may continue to grade the student. All the evaluations can only be submitted once, although it is possible for the administration module to override them, thus enabling a faculty to reconsider their assessment.

An internship transcript details every aspect of the students' personality both on and off the station. The application provides an intuitive interface to generate the final transcript based on the overall performance of the student over the course of the semester. Once the document is completed, it can be reviewed and approved by the administration department. If, for some reason, the document was found to be unacceptable in any regard, the administration may reject the transcript which would require the faculty to reevaluate and resend the information.

C. Student Module

This module consists of two states, viz. pre and post preference submission, depending on which the functionality of the software will differ.

1. Pre-Preference Submission

Pre-allocation state has two states within itself, pre and post station preference submission deadline.

Pre-Deadline

If a student accesses the software before the administration department has closed the window for station preference submission, then they will be first asked to enter their personal and contact information for reference and to yield better results from the automatic allocation script. Information like if the student has a valid driving license and also a vehicle for easier commuting is also gathered. Upon filling this information, the student will then be required to carefully examine each station for its job requirements, relevance to one's career interests, whether or not the station is willing to absorb the student after graduation, convenience and practicality of travel to a preferred station, etc. and subsequently fill up the station preference list. Only stations accepting students from that particular student's branch will be available to them.

The software has a feature wherein student(s) maybe short-listed for a given station. If the said station is only accepting those student(s), then it is possible to mark the station as 'invisible' rendering it unavailable to other students. This station would be treated as a non-existent station to all students except to those specifically mentioned by the station.

After successfully choosing the preferred stations, they may put forward the information for administration's consideration. The student may review the information after the process has completed, and edit the information if found necessary.

Post-Deadline

Once the deadline of station preference submission has been reached, the admin may close the pre-deadline portal so that the students accessing the software henceforth will only be able to view the information that they had presented beforehand. No modification would be allowed by the software post-deadline.

2. Post-Preference Submission

Once the submission window closes, or if all the students have submitted their respective preferences, the admin may start the student allocation process by executing the allocation algorithm, which has been explained above.

Once every eligible student has been allocated to a station and a faculty has been assigned to each station, the administration can enable the students of that particular semester to enter the post-submission portal and thus continue with their internship responsibilities. Through the student-faculty portal in the software, the student will be able to track their progress during the semester and interact with respective faculties for their feedback. They may also retrieve station specific files that the faculty has uploaded onto the application and submit their periodic progress and project reports for faculty's assessment.

D. DatabaseModel Used by the Software System

The application is based on a database model which allows maximum flexibility and scalability by branching out the different components of the software and using those primary keys to link with its related tables [7]-[9]. This model facilitates most of the requirements with minimal code changes. Each *evaluation_component* belongs to an *evaluation_group* so that multiple groups can co-exist simultaneously thus allowing different type of internship evaluations for different types of internship semesters. The student details are stored in *students* and their individual evaluations are stored in the *student_evaluations*. The *faculty* table comprises of the faculties' details. The *stations* and *station_streams* table holds the station's information and also those that are stream specific. The *streams* table contains information pertaining to the different available streams in the institution. The database model is illustrated in Fig. 2.

III. Process Overview of the Software System

Fig. 3 represents the major features available in the application over the course of an internship semester.

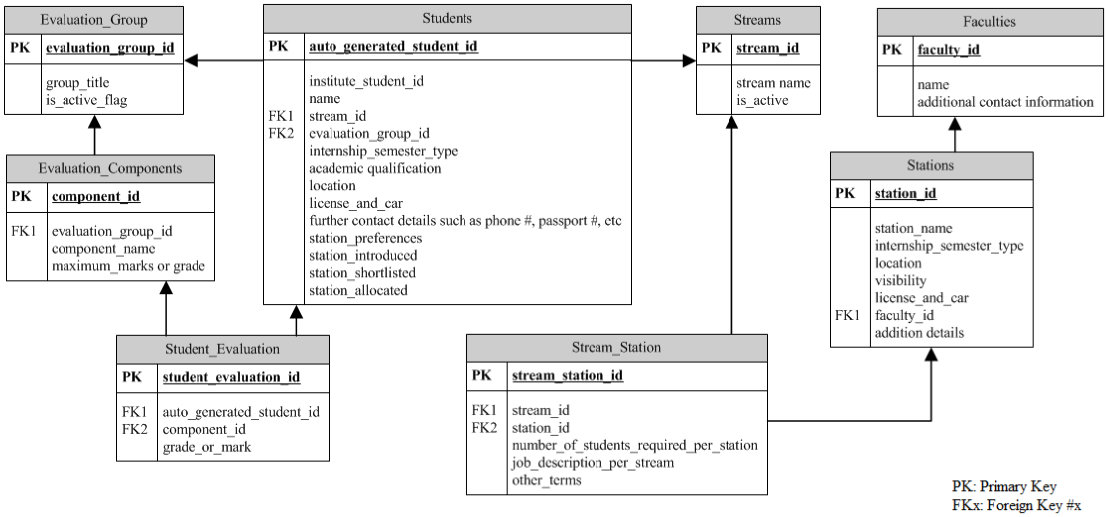

Fig. 2 Database Model used by the software system

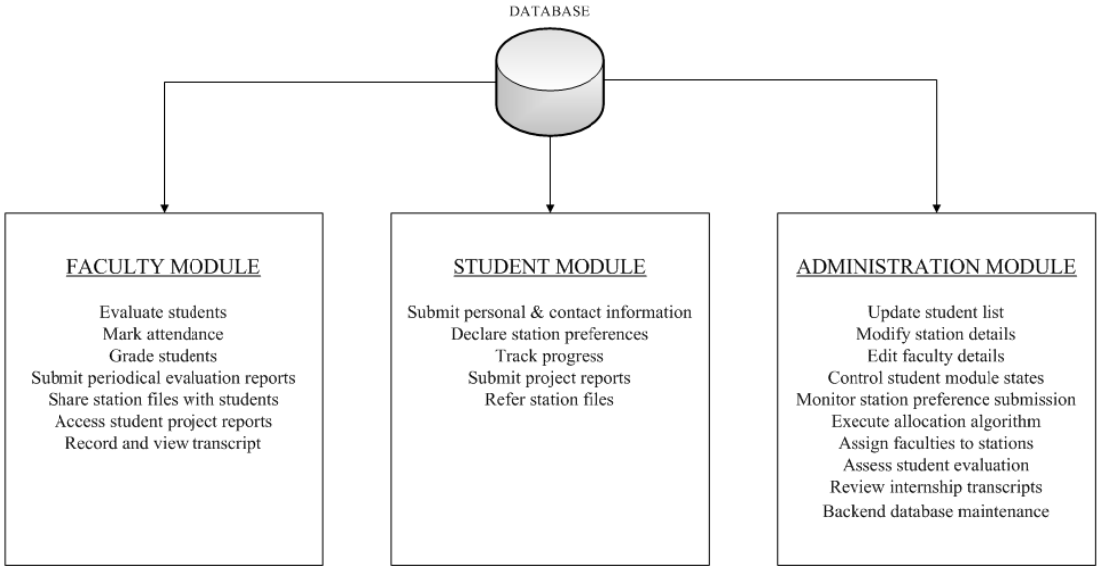

Fig. 3 Process Overview of the software system

IV. Conclusion

Students interested in pursuing a career in any competitive industry consider internships provided by universities as a vital opportunity to get a "sneak-peak" into the industry. Organizing all the different cogs of an internship program can be greatly simplified by employing a well thought-out standalone software application. The administration department could be given access to all the different aspects of the application for stricter control and easier management, and the student and faculties a different set of functions as required by them. In this new era of technology, where absolutely every major process is being replaced by a software counterpart, managing an internship semester by a university or institution should be no different.

Acknowledgment

The authors would like to thankProf. Dr. R.K. Mittal, Director of BITS-Pilani, Dubai, and Dr. Tanmay Panda, Dean of BITS-Pilani, Dubai Internship Program. The present work also benefited from the input of Mr. ArjunMuraleedharan.

References

[1] Robin Nixon, “Learning PHP, MySQL, Javascript and CSS” ISBN: 978-1-449-31926-7. O’Reilly Media, Inc. 2012.
[2] Jeremy Keith. DOM Scripting: Web Design with JavaScript and the Document Object Model. ISBN: 978-1590595336. FRIENDS OF ED, Inc. 2005.
[3] Brian Fioca. “Managing Sessions and State with PHP”. Available online: http://www.onlamp.com/pub/a/php/2006/05/18/managing-sessions-and-state.html.

[4] Michael T Goodrich, Roberto Tamassia. Algorithm Design: Foundations, Analysis, and Internet Examples. ISBN: 978- 0471383659. Wiley Publications. 2006.
[5] Jonathan Cutrell. "Understanding the Principles of Algorithm Design". Available online: http://net.tutsplus.com/tutorials/tools-and-tips/understanding-the-principles-of-algorithm-design/.
[6] Jonathan Chaffer, Karl Swedberg. Learning jQuery, Third Edition. ISBN-13: 978-1849516549. Packt Publishing, Inc. 2011.
[7] Thomas M. Connolly. Database Systems: A Practical Approach to Design, Implementation and Management. ISBN: 978-0321523068. Addison Wesley, Inc. 2009.
[8] C.J Date. Database Design and Relational Theory. ISBN: 978-1-449-32801-6. O'Reilly Media, Inc. 2012.
[9] Barney Desmond. "Introductions to SQL query optimization". Available online: http://www.onlamp.com/pub/a/php/2006/05/18/managing-sessions-and-state.html.