International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor (2018): 7.426

Flexible Student Performance Analysis System

Vigilson Prem M[1], Joel Kingsley R[2], Mohammed Ashik S[3], Manjunath M[4]

[1] Professor, Department of CSE, R.M.K. College of Engineering and Technology, Chennai, Tamil Nadu, India

[2, 3, 4] UG Student, Department of CSE, R.M.K. College of Engineering and Technology, Chennai, Tamil Nadu, India

Abstract: ***All educational institutions across the world have a management system for record keeping of information about students, exams, and results. Most of these systems are manual meaning each record has to be manually entered by a staff member. Some universities and schools do have computerized systems but the problem is that the system design is rigid therefore not being able to make changes when the education board or university changes its regulations. Also, these systems are not designed with the structure and vision to add data analysis to them in the future. This project proposes a SPAS system that can be used in any Anna University affiliated college in Tamil Nadu, and possibly in any other university with similar structure, and is flexible to be able to accommodate any changes in course structure and add data analysis to it in the future.***

Keywords: Student performance; result analysis; flexible system design; report generation

1. Introduction

Tests and assessments in schools and universities are intended to measure a test taker's knowledge of a subject in a course. Testing often results in a grade or test score. Before wide adoption of computers by educational institutes, all of the record keeping was done in books. This was found to be not safe as any physical item can be damaged, lost or stolen. With the internet boom, computers started to get widespread adoption by many schools and universities. Now, instead of physical spreadsheets, they started using spreadsheet software to create and maintain virtual spreadsheets. But the problem of accessibility which already existed with physical record keeping couldn't be overcome using this method. The staff still had to hold copies of the spreadsheets with their updated data and share it among their peers.

This is when student information systems came into play. Many organizations started to develop these systems that store all the student records and information about exams and results. The data could be centralized to a single server thereby enabling access of the latest version of the records to every user. The problem with many of these systems is that the system design is neither scalable nor flexible to handle changes in the course structure. If a new regulation is put forth, then there might be a discrepancy between the batches under the new regulation and the existing ones.

Currently, colleges use a log book for each student which would contain all the marks and attendance information of the student[2]. Along with this, it would also contain the feedback which was given to the ward in each consulting session given by the student's counselor. After every set of exams, the student's counselor would talk with the student with the logbook in hand, which the student would then verify and sign. The problem here is that the logbooks can only be used for logging previous grades and marks. There is no possibility of actually making any analysis or conclusion based only on the static data.

On the other side, staffs use a spreadsheet to store the same information digitally. With apps like Google Spreadsheets, multiple staffs could edit the same spreadsheet without having to worry about conflicts that might occur during simultaneous updates. Although there are some ways to calculate the cumulative GPA of a student by using spreadsheet formulas, it is fairly difficult to make complex formulas for more advanced computations. Also when considering data integrity, more than 86% of spreadsheets have human error[1].

That is why we came up with a system design including a database design which would be effective in this scenario. There are a few objectives that were found during the course of the research behind this system:

1) To develop a system for student's performance analysis.
2) To assist the lecturers in keeping track of the students' progress throughout the semester.
3) To generate student's report from the existing data available.
4) To make the entire system flexible to future additions.
5) To make the system deployable in any host machine of the institute's choice.

2. Literature Review

A background study was performed to review similar existing student performance analysis systems in the market. We found that there are truly a large number of student performance analysis systems. The problem was that in every single one, they were commercialized and sold for a yearly fee. It was also found that the few systems that were available for free of cost didn't have a solid system design.

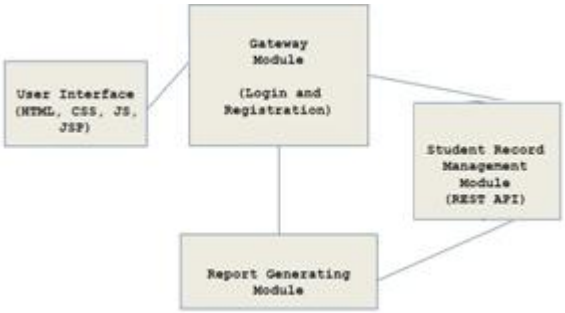
A major reason why a decision was made to develop a flexible student performance analysis system in the first place is that in the case of all existing systems, the data is stored in the software provider's server. This is not favorable in the case where colleges and universities have a privacy issue and they don't want to put their data in the open. The system proposed will overcome this issue by enabling the system to be deployed directly in the institute's infrastructure. This ensures that the data remains private no matter what because of the physical barrier. Also, the system composes of microservices which could be run in a docker container and replicated as it scales.

3. Proposed System

The main concerns that the proposed system overcame were:
a) Security
b) Flexibility
c) Automation

In order to do this, a few features from the existing systems were employed during the design and implementation of the proposed system. These features include adding and retrieving student's results, report generation which existed in many of the commercialized systems[3].

The following are the four modules in the system:
a) User Interface (HTML, CSS, JS, JSP)
b) Gateway Module (Login and Registration)
c) Student Record Management Module (REST API)
d) Report Generating Module (Google Charts API)



a) User Interface

It was necessary to be able to provide a user interface module that could easily be replaced if necessary. There are many framework options to consider including single-page application using Angular, and multi-page application using Servlets and JSP. The decision was made to go with the Java route but the same can equally be done using Angular as well. There are three set of web pages that were created for the three default user types (Administrator, Department Moderator, Managed User)

- Administrator: The admin account of the entire system. Granted access to all CRUD operations to all database tables.
- Department Moderator: The department's moderator account granted to the Head of the Department (HOD) or any other professor who should be able to add exams and update student information.
- Managed User: The rest of the professors who should be able to add exam results of the exams that pertain to their department.

b) Gateway Module

Authentication and authorization is a major part of this system, as there has to be inter-department privacy and only the staff of the department should be able to manipulate data that pertains to that department. Therefore registration and login have to be done for all the users of the system. All new users would be registered by the administrator through the admin portal. Registered users can log in to the system by providing the credentials in the login page which is handled by the gateway servlet.

c) Student Record Management Module

Record management is done by using a MySQL database and a REST service that is invoked by the User Interface module to interact with the database. The REST API is built using Spring Boot, which provides defaults for code and annotation configuration. Thereby it is possible to quickstart new projects within no time.

The following is the database schema for the record management module:

Table: **user_types**
user_type_id
name

Table: **users**
user_id
name
email
password
fk_departments_department_id fk_user_types_user_type_id

Table: **subjects**
subject_id
subject_code
subject_name
credit

Table: **batches**
batch_id
batch_start_year

Table: **departments**
department_id
department_name
abbreviation

Table: **students**
student_id
register_number
student_name
gender
is_hosteler
fk_batches_batch_id
fk_departments_department_id

Table: **semesters**
semester_id
semester_number
fk_batches_batch_id
fk_departments_department_id

Table: **exams**
exam_id
fk_exam_types_exam_type_id
fk_subjects_subject_id
fk_semesters_semester_id

Table: **exam_types**
exam_type_id
exam_type_name
abbreviation

Table: **exam_results**
exam_result_id
fk_exams_exam_id
fk_students_student_id
marks
grade

This system design would enable the institution to add in new exam types apart from the default types such as unit tests, internal exams, and final exams. It also provides different user types in order to provide different users with different access rights. And if the institute introduces a new department of study in the future, then the system can simply add a new department.

Access to these critical features would be only available to the super admin user which comes as default. Features that would not affect the critical parts of the system such as updating student information and adding exams are accessible to the staffs with the highest authority in each department(Department Moderator). These staffs are usually the Deans, Head of the Departments, Principal, and so forth. Since all professors would need to add their class' exam results to the system, all of the professors are given access to add and manipulate exam results of students in their own department.

D. Report Generating Module

Generating the report dynamically is the main objective of this system. The data had to be fetched from the database and given as parameters to a chart API. There were many options when it came to chart libraries from Highcharts, Chart.js, Google Chart API and CanvasJS.

Google Chart API seemed to be the best option as it was for the main part free, and also had a wide range of chart types.

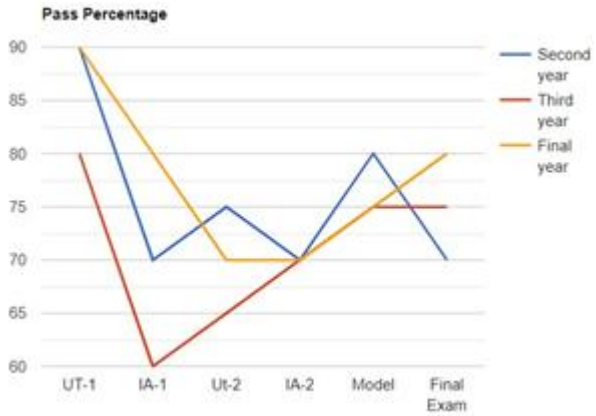


Figure 2: Cumulative Performance of each batch in a department in a semester

But a problem that was faced is that the API was really slow when it came to generating charts. As there were no other chart APIs that were available for free, the decision was made to stick with it for now, although a better alternative could be used.

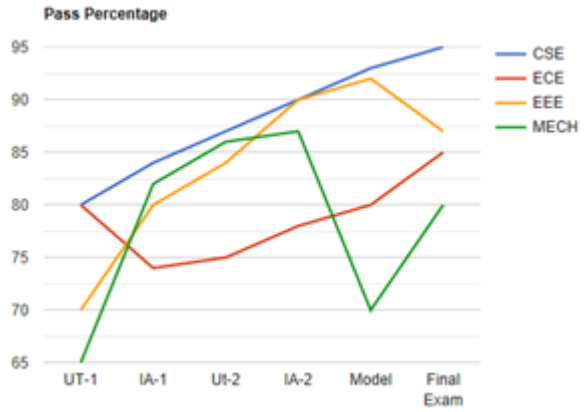


Figure 3: Cumulative Performance of all departments in a batch in a semester

4. Conclusion

In this paper, we have presented a system to store, retrieve and analyze the performance of the students in a university. We have also developed it in such a way that the system is flexible enough to accommodate changes in the university and course structure. The main conclusion is that this project is just the first step to build an open source platform that could be used by any educational institute in the world.

References

[1] Panko, Raymond. (2008). Spreadsheet Errors: What We Know. What We Think We Can Do.
[2] Lars Bollen, Andreas Harrer, H. Ulrich Hoppe. (2004). An Integrated Approach for Analysis-Based Report Generation
[3] Peter F. Tarassenko, Marina S. Bukharova. (2001). System for database report generating