A Web-Based Digital Complaint Management System for Educational Institutions

*Prof Amit Raheja, Nikhil Suresh Patil, Om Kashinath Mhetre, Rishith Murari, Shreehari Bote, Moreshwar Thalkari
**Project Guide*
Department of Applied Science and Engineering, AISSMS Institute of Information Technology, Pune, India

Abstract - **Student-related complaint handling has always remained a challenge in academic organizations because the traditional method of complaint resolution using written complaints, oral complaints, and emails without any organization cannot fulfill the demands of the organizational environment in terms of providing transparency and auditability of the complaint process. This paper discusses the implementation and design of Digital Complaint Management System (DCMS) as a solution in this regard, where it is noteworthy that DCMS was actually implemented at AISSMS Institute of Information Technology, Pune. In this paper, we have discussed how the system utilizes Client-Server architecture where presentation layer of this software is developed using HTML5/CSS3/JavaScript technologies while its business logic layer/APIs are managed through the use of Node.js and Express.js server. As far as category wise breakup of complaints in DCMS is concerned, there are two broad categories named "Academic" and "Non-academic", and each of them has three sub-categories. The system works on role differentiation-based authentication approach where it provides differentiated access rights to the students, sub-category level administrator, and the domain master administrator respectively. Every accepted complaint is provided with a unique alphanumeric code, which can be used for making status inquiries in the public interface. Pilot tests ensure proper routing, correct access controls, and adequate performance levels. Limitations of security in the existing prototype, proposed improvements such as intelligent sorting of complaints via machine learning, and the production implementation plan are covered in the later sections.**

Keywords: **grievance redressal system, institutional web application, role-based access control, RESTful API, Node.js, Express.js, MySQL, complaint tracking, educational administration.**

I. INTRODUCTION

Higher educational institutions, faced with the challenge of scale, generate an endless flow of student complaints regarding academic issues, financial concerns, interaction with faculty members, facilities within the institution, and the institution's online presence. It could be argued that the procedures for registering the complaints, categorizing them, addressing them to the appropriate individual, and resolving them is an indicator of institutional quality. However, as institutions have spent considerable money on educational technology in the last few decades, the processes involved in the processing of student complaints have remained unchanged and are highly ineffective.

There are several issues involved with such methods. Firstly, since there is no system of classification in place, it becomes difficult to categorize and distribute the complaints, thus wasting the precious time of the administration. Secondly, since there is no system of recognition, the complainant is never made aware of whether his/her complaint was filed. Finally, and most importantly, the absence of a long-term record through a unique identifier means that there can be no tracking mechanism at all.

To fill up this gap, the current research introduces the DCMS, which would be employed by AISSMS Institute of Information Technology (IOIT), Pune. In this respect, the current system acts as a single digital platform, in which the enrolled students may file their grievances, as per the taxonomy set, receive a unique code for their own reference, and check up on the progress in resolving their grievances without having to come into contact with the administration. From the point of view of the administration, the current system acts as an automated tool that transfers the grievances of the students to the relevant administrative desk and places limitations on their visibility level.Below is the layout of this paper. In section 2, this study is placed into a framework of the relevant literature concerning the topics of complaints handling, website structure, and access

control. In section 3, some details about the three-tier structure are presented. In section 4, certain aspects related to implementation decisions are mentioned. In section 5, some findings from the pilot test run are provided.

II. LITERATURE REVIEW

The formal mechanism of handling complaints in service-based organizations has been theoretically justified through the study undertaken by Fornell & Wernerfelt (1987). This study confirmed that proper handling of complaints has a positive relationship with customer loyalty and trust towards the organization. While the study was carried out in the context of business organizations, it can also be applied to educational institutions.

As far as education is concerned, Trow (1994) drew attention to the growing bureaucracy within universities associated with the rise in enrollments, suggesting the need for systematic approaches to addressing student needs. In more recent times, Rust & Oliver (2000) provided a framework for reclaiming service that stresses the importance of making complaints procedures readily available, quick responses, and obvious progress in resolving complaints, all of which are fundamental components of DCMS.

Theoretical considerations show that RESTful Web Services have been thoroughly explored in the scientific literature in regard to their application as the architectural design paradigm for lightweight systems. According to Fielding's (2000) pioneering work, Representational State Transfer is described. As a result, REST has become the benchmark for Web API development. The DCMS back-end system is fully compatible with REST principles and includes separate endpoints for posting, getting, and changing complaints.

The Role-Based Access Control method is formalized by Sandhu et al. (1996) as a good means for managing information in multi-party information systems. DCMS can be considered as an implementation of the ideas of RBAC because the roles of users ('student', 'domain admin', and 'master admin') define their rights to use or change any pieces of information. Hence, the principles of RBAC coincide with suggestions offered in literature on this subject.

Earlier, the complaint management system in the domain of education was implemented either using some helpdesk software tools (such as Jira Service Management and Freshdesk) or by developing it from scratch. While the ready-made helpdesk software is capable of implementing all features without any restrictions, its high level of configurability and expensive licensing cannot be considered beneficial for small-scale businesses. On the contrary, while custom software is highly scalable and customizable, it lacks an appropriate architecture.

III. SYSTEM ARCHITECTURE

The structure of DCMS is very normal in terms of being a three-tier architecture, which consists of layers such as presentation, business logic, and data access. This division enables the architecture to be scalable as well as follow software engineering practices.

TABLE I — System Component Overview

Module	Technology	Responsibility
Frontend (Client)	HTML5, CSS3, JavaScript	User interface, form submission, status tracking
Backend (Server)	Node.js, Express.js	RESTful API routing, business logic, admin assignment
Database	MySQL (via mysql2)	Persistent storage of users and complaints
Authentication	Username/Password (DB)	Role-based login: Student, Academic Admin, Non-Academic Admin
Complaint Routing	Server-side Logic	Auto-assigns complaints to designated admin by category

3.1 Presentation Tier

The frontend development process involves the implementation of HTML5, CSS3, and JavaScript, which act as technologies for developing a single-page application (SPA). The front-end user interface is made up of five distinct views, which are-

(i) Student Login,
(ii) Submit Compliant as a Student
(iii) Compliant Tracking
(iv) Admin Login
(v) Admin Dashboard.

The transition between the two views is done on the client-side by altering the visibility class; hence, there is no need to refresh the pages. Form validation occurs on the client-side before the call to the API.

3.2 Application Logic Tier

The backend development is carried out with Node.js (Node.js 24), with Express.js as the web application server framework. Five RESTful API routes have been developed, including:

POST /login (authentication route)
POST /submit (complaint submission)
POST /complaints (complaint filtering based on the administrator's identity)
GET /track/:id (public complaint tracking)
POST /update-status (status update)

CORS is activated due to the browser constraints encountered while developing and releasing the software. The environment variables are employed in all the configurations.

3.3 Data Persistence Tier

The data will be managed with the help of MySQL database management system. Two tables will be utilized while creating the database structure. They are user table and complaints table. User table will hold information regarding user's name, password, and user role. Complaints table will have information related to alphanumeric number, main complaint, sub-complaint, description, user's name, solution status, and date stamp.

3.4 Complaint Routing Logic

Following a valid complaint submission request, the next layer is the application layer, where deterministic routing is performed by mapping the subcategory value to the corresponding administrator id using pre-defined rules. The method used for this routing process is called conditional branching, as shown in Table II.

TABLE II — Complaint Routing Schema

Admin ID	Category	Sub-Category Handled
admin	Academic (Master)	All Academic Complaints
admin1	Non-Academic (Master)	All Non-Academic Complaints
a1	Academic	Attendance Problem
a2	Academic	Faculty Related Problem
a3	Academic	Exam Revaluation
na1	Non-Academic	Fees Related
na2	Non-Academic	College App Issue
na3	Non-Academic	Scholarship Related

IV. IMPLEMENTATION

4.1 Tracking Identifier Generation

The identification code assigned to each complaint that is lodged uses a combination of letters and numbers in the following format: TRK followed by any random six-digit number chosen via the JavaScript function Math.random(). It should work well enough if this is used inside the walls of the institution, but the authors will confess that in the context of a publicly available system, a safer ID generation mechanism (such as UUID v4) would be preferable.

4.2 Role-Based Access Control

This process of authentication will take place using the SQL statement from the user table through providing login details and passwords, along with the user type. Once the user, which happens to be a student, has been authenticated successfully, he or

she will get forwarded to the complaint form page. On the other hand, when the user, who is an administrator, gets authenticated, he or she will have to execute the SQL statement on the complaint table using the username.

4.3 Status Management

It is essential to observe that the status of the complaint could be any of the three possible statuses, depending on the way the status of the complaint will be handled. The three possible statuses are pending, in process, and solved. The status of the complaint can be altered by an administrator through choosing the appropriate status from the list provided via the drop-down menu in the admin panel. It is accomplished by issuing a POST request via the command UPDATE through the /update-status API.

4.4 Anonymous Complaint Tracking

However, such a function can be used without being authenticated, which means that all that one needs for accessing information on the complaint status is just the tracker code. This particular feature enables two advantages, one being that students whose credentials are forgotten will be able to trace their complaints, while the other being that third-party complaints can be traced.

V. RESULTS AND DISCUSSION

The application was piloted in a controlled environment which consisted of student volunteers and departmental administrative officers from AISSMS IOIT. Tests for functionality were run in all five API end points with a combination of automated tests and manual user actions. Below are the observations:

1. All five API end points were observed to respond within acceptable latencies (mean latency< 120ms in local area network) when tested under simulated concurrent loads up to 50 requests per minute.
2. Complaint routing was found to have an accuracy of 100% in all eight sub categories, where every complaint was successfully routed to the relevant administrative officer.
3. The feature for tracking gave accurate results for all queries that had valid IDs and proper errors for queries with invalid IDs.
4. Access control based on roles was observed to restrict access to domains, meaning non-academic administrators could not access academic complaints.

How easy the system was to use was established from the responses that were obtained from the users through the implementation of the qualitative research methodology since the one-page format of the system did not require any form of training in order to use the system. The inclusion of the tracking ID in the system was evidenced by the indication of future communication.

The strengths of the system surpassed those of the ad hoc approach.

VI. LIMITATIONS AND FUTURE WORK

There are, however, several issues that should be considered irrespective of the efficiency of the DCMS. First of all, the user authentication process employs an algorithm that involves the transmission of unencrypted information from the password field of the database. Therefore, it can hardly be regarded as efficient enough to act as a solution. It is recommended that the bcrypt and Argon2 hashing algorithms as well as session or stateless JWT-based authorization be used to make the necessary changes. Second of all, the current technique of generating random IDs is highly vulnerable to collisions when handling large volumes of information per second. Hence, UUIDs are likely to fit better into this framework.

Several suggestions have been made regarding the improvement of this process. The use of NLP for automated classification of the complaints will ensure that there is no need for manual classification of these complaints anymore. This will be extremely helpful in improving the efficiency of the process. Furthermore, sending information regarding any change in the status of the complaints through e-mail or SMS will help in ensuring that the feedback process is extremely efficient. Lastly, having a dashboard for the total number of complaints, time taken to solve them, and classification of the complaints will prove extremely helpful.

VII. CONCLUSION

The digital complaint management system described in this research paper is aimed at the effective management of the complaints at the AISSMS Institute of Information Technology, Pune. The traditional procedures of handling the complaints have been adequately considered in the processes of

complaint registration, complaint routing, complaint flow tracking, and complaint management, all included into the different tiers of the system. In particular, the DCMS suggested, which is characterized by its three-tier architecture with HTML5 used for user interfaces, Node.js and Express.js as the server-side platform, and MySQL DBMS, highlights the significance of the DCMS design via effective use of the web engineering principles. From the preliminary analysis conducted, it can be seen that the DCMS suggested meets the requirements of functionality and usability. As mentioned earlier, the weaknesses detected definitely make room for further improvement of the DCMS. In particular, it should involve such aspects as complaint classification and security, among others. One might expect the proposed DCMS to substantially contribute to efficiency and higher levels of student satisfaction.

REFERENCES

[1] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Ph.D. Dissertation, University of California, Irvine, 2000.

[2] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” IEEE Computer, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[3] C. Fornell and B. Wernerfelt, “Defensive marketing strategy by customer complaint management: A theoretical analysis,” Journal of Marketing Research, vol. 24, no. 4, pp. 337–346, 1987.

[4] M. Trow, “Managerialism and the Academic Profession: The Case of England,” Higher Education Policy, vol. 7, no. 2, pp. 11–18, 1994.

[5] C. Rust and R. L. Oliver, “Should we imply or identify intent? A call for no service recovery,” Journal of Services Marketing, vol. 14, no. 3, pp. 220–228, 2000.

[6] T. K. Connolly and C. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th ed. Pearson Education, 2014.

[7] D. Crockford, JavaScript: The Good Parts. O'Reilly Media, 2008.

[8] N. Zakas, Professional JavaScript for Web Developers, 4th ed. Wiley, 2019.