



ORIGINAL RESEARCH ARTICLE

DEPARTMENT MANAGEMENT SYSTEM: A WEB-BASED APPLICATION FOR ACADEMIC WORKFLOW AUTOMATION

Sobia Fayyaz*Department of Computer Science
University of Agriculture, Faisalabad*Corresponding Author Email: sobiafayyazthaheem@gmail.com**ABSTRACT**

Department Management System (DMS) is a web application designed to reduce paperwork and help manage academic operations digitally for a university department. Students typically fill out numerous undergraduate (UG) forms — such as Transcript Requests, FYP Proposals, and Leave Applications — on paper, which are often misplaced, slow to approve, and lack transparency. DMS provides Students, Teachers, and Administrators with their own role-based portal for performing these operations. Students can sign up, log in, update their profiles, and submit UG forms online; Teachers can track student activity; and Administrators have full authorization over system operations. The application was built using React.js and Tailwind CSS on the frontend, and Node.js, Express.js, and MongoDB on the backend. As a web application, DMS is platform-independent, mobile-responsive, and deployable across multiple departments or institutions. Functional testing covered all defined test cases without error, supporting DMS as a practical, hassle-free solution to departmental paperwork.

Keywords: *Department Management System; Web Application; UG Forms; Role-Based Access Control; React.js; Node.js; MongoDB; Academic Digitization.*

1. INTRODUCTION

Universities typically generate large amounts of paper-based work for academic administration, spanning student registration, profile creation for students and teachers, and the creation, filling, and approval of forms. Workflows conducted on paper often lead to disorganization, lost records, slow turnaround times, and a lack of transparency.

The primary objective of the Department Management System (DMS) is to minimize these pain points by creating an automated system for use at the University of Agriculture, Faisalabad. As UAF continues to enroll more students each semester, administrative responsibilities on faculty and department staff increase accordingly. Every student must fill out multiple UG forms — such as Transcript Request forms, FYP Proposals, and Leave Applications — throughout their university life, and managing these manually creates unnecessary delays (Agarwal and Singh, 2020).

DMS provides a single web portal through which Students, Teachers, and Administrators from a university can operate. It reduces paperwork by digitizing form workflows, maintaining digital records, sending instant notifications, and enforcing role-specific access. DMS automates form-filling, approval, and record management, and was built using popular, open-source technologies — React.js, Node.js, Express.js, and MongoDB — to ensure it remains scalable and cost-effective for university-wide deployment.



This paper discusses the drawbacks of current paper-based departmental processes, the proposed DMS along with its architecture and design, its implementation, and its testing (*Bharamagoudar, Geeta and Totad, 2013*). Finally, it discusses how DMS improves overall efficiency, transparency, and user experience.

1.1 Existing System

Most administrative academic processes at the University of Agriculture, Faisalabad — as at most public universities in Pakistan — are handled manually and on paper. To avail a given service, students must collect a hardcopy form from a department office, fill it out by hand, obtain the required signatures, and submit it, then wait days or weeks to learn whether it has been accepted or denied. This process poses several problems (*Rajmane, Mathpati and Dawle, 2016*).

Professors and administrative staff carry a high workload each semester due to increasing student numbers. Forms can be misplaced, torn, lost, or need to be resubmitted, leading to delays, and students have no way of knowing whether their form has even been seen. Notices from the administration are also handled inefficiently, typically posted on physical bulletin boards or communicated by word of mouth, and students lack a proper channel through which to submit complaints to the department.

Limitations of the Existing System

- Absence of an electronic repository
- No audit trail for uploaded forms
- Paper-intensive processes
- Opaque approval process
- High chances of losing documents
- No appropriate role-based access

1.2 Proposed System

The Department Management System converts academic administrative workflows into an electronic process at the department level. DMS provides a three-tier user role model — Student, Teacher, and Admin — each with a distinct permission level. Student users can sign up with their email ID, log in, edit their profile, and apply for multiple UG forms from their dashboard. Teachers can access their profile and view students' activities. Admin users can register students and teachers, view all users and their information, approve or reject student-submitted forms, post notices, manage complaints, and view complete logs of system activity. From the student's dashboard, all applied form details are sent to the Admin panel for approval, with the response conveyed to students through DMS's integrated notification system (*Rohini et al., 2024*).

Advantages of the Proposed System

- No paperwork hassle or file maintenance
- Role-based access control — students, teachers, and administrators have secure, distinct access
- Faster submission and approval of forms

Centralized location for storing all data in a single digital location (*Bansal and Chawla, 2016*)

- Improved communication through announcement and notification systems
- Minimal chances of document loss and administrative errors



- Real-time tracking of applications and requests
- Access from any browser, with no additional software required
- Responsive interface accessible from mobile and multiple device types
- Deployable across multiple departments and universities

2. METHODOLOGY

2.1 System Design and Architecture

DMS uses a three-tier (N-tier) architecture. The Presentation, Application, and Data logical layers make DMS highly modular, easy to maintain and update, and easy to scale, since each layer is completely separate from the others (Botta-Genoulaz, Millet and Grabot, 2005). This architecture is described using both a Level 0 and a Level 1 Data Flow Diagram, illustrating how data is processed by the system.

Tier	Layer	Description
Tier 1	Presentation Tier	The interface the user interacts with, built with React.js and styled with Tailwind CSS.
Tier 2	Application Tier	Node.js and Express.js power the application's data manipulation logic in this tier.
Tier 3	Data Tier	A MongoDB database serves as the repository for all application data.

Table 1. Three-tier system architecture.

Technologies used in DMS span both the frontend and backend. Table 2 summarizes each technology and its purpose within the system (Rajmane, Mathpati and Dawle, 2016).

Technology	Purpose
React.js	Front-end, component-based JavaScript library
Tailwind CSS	Responsive front-end CSS framework
Node.js	Backend JavaScript runtime environment
Express.js	Backend framework for API routes and middleware
MongoDB	Database for storing users, forms, and system data
VS Code	Integrated development environment for debugging, snippets, and refactoring
GitHub	Tool used for collaboration and version control

Table 2. Technical specifications.

2.2 User Roles

1. Admin

- Adds and edits student and faculty information
- Adds and edits courses



-
- Uploads notices and announcements
 - Secures and maintains system accounts

2. Teachers

- Can enroll students
- View student information
- Notify students

3. Student

- Submit UG forms
- Submit transcript requests
- Submit complaints

2.3 Data Flow Diagrams

The following diagrams show how data flows between the user and the system. The main data flows are: the Admin enters student and faculty data into the database; faculty can view enrolled students; students fill forms (UG, Undertaking, etc.); and all data is stored in the database.

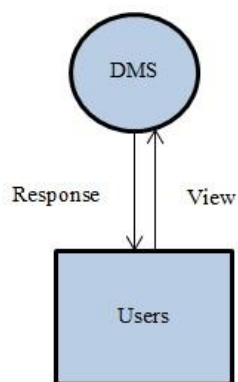


Figure 1. DFD Level 0.

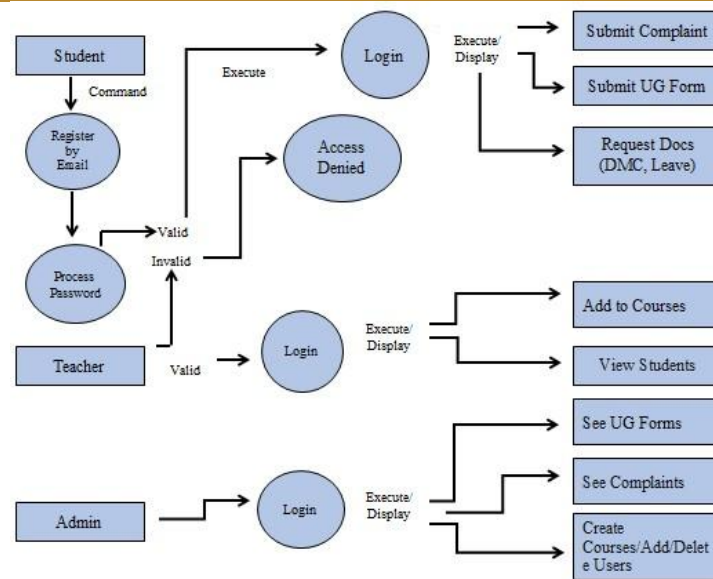


Figure 2. DFD Level 1.

2.4 Functional Requirements

ID	Requirement Category	Key Features
FR01	User Registration (Sign Up)	Users can create an account using a valid email address.
FR02	User Login (Authentication)	Users can log in via username, email, and password, with role-based access (admin/student/faculty).
FR03	User Profile Management	Users can edit and save their profile information.
FR04	UG Form Submission (Student)	Students can submit UG forms, Transcript Requests, FYP Proposals, and Leave Applications.
FR05	Form Approval (Admin)	Admin can view all registered users and approve or reject submitted forms.
FR06	Admin Controls (User Management)	Admin can view all activity within the system for each user.
FR07	Announcements Tab	All users can view announcements.
FR08	Complaints Tab	Students can submit complaints.
FR09	Notifications Tab	Students are notified about their forms and new announcements.

Table 3. Functional requirements.

2.5 Non-Functional Requirements

Non-functional requirements are summarized below (Bharamagoudar, Geeta and Totad, 2013).

- Availability — the system remains available to students, teachers, and admin during university working hours.



- Security — the system enforces role-based access control, preventing students from accessing admin features and vice versa.
- Usability — the UI, built with Tailwind CSS, is clean and simple to use.
- Performance — user authentication and dashboard visibility complete within 3 seconds of a login attempt.
- Data Integrity — all forms are stored in and retrieved from MongoDB with their respective approval status.
- Reliability — the system maintains functionality without interruption when multiple users are logged in.
- Maintainability — the system's codebase is structured for straightforward future maintenance and improvement.

2.6 Modules

DMS consists of three main modules, each defining the capabilities and authorizations available to a given role in the department (*Ferrari and Hurst, 2021*).

Student Module: Students register and log in via a secure process, arriving at their own dashboard. The module supports profile creation and editing (name, contact, address, academic details); UG form submission (Transcript Request, Change of Degree, FYP Proposal, Leave Application, Under-Taking Form); viewing submission history and real-time approval status; receiving notifications when forms are approved or rejected; and accessing the complaints and announcements tabs.

Teacher Module: Teachers log in with their credentials and are directed to a dashboard showing their profile information. Teachers can check their personal and academic history, track student activity relevant to their role, and view course information, with a UI that surfaces only the information teachers need without overloading the screen.

Admin Module: The Admin has full control of the system and can monitor every action taken within it — viewing all registered students and teachers along with their profile information, approving or rejecting UG forms with optional comments, posting, editing, or deleting announcements, viewing and deleting student complaints, and receiving a notification of every system action as an audit trail. The Admin dashboard presents pending approvals and all users in clean, searchable tables.

Module	Description
Student Registration & Login	Secure role-based authentication for students using university email credentials.
Profile Management	Students and teachers can view and update their profile information.
UG Form Submission	Students submit forms (transcript, projects, and leave) from their dashboard.
Admin Form Approval	Admin can view submitted forms and approve or reject them along with optional notes.
Announcements	Admin can create department-wide announcements viewable by all users.
Complaints	Students raise complaints; Admin tracks and updates their resolution status.
Notifications	The system notifies students of form decisions and new announcements in real time.

Table 4. Modules.



2.7 Development Methodology

The Agile Scrum model was used, as its iterative and incremental approach allows the DMS product to grow alongside a deeper understanding of the multi-user academic setting. Major modules — Student Registration, Manage Complaint, Approve UG Form, Create Courses, and Enrollment — were developed in parallel as standalone mini-projects, allowing work to be time-boxed into 1–2 week sprints, each ending with a shippable prototype ready for review.

3. RESULTS

3.1 Testing Methodology

Functional testing was executed on the Department Management System with each type of user and each functionality (Ricca and Tonella, 2001). Five test cases were used, covering all roles and functions; all five passed on the deployed application, meeting the goal of a successful pass rate. Sample test data was drawn from real-life students.

All test cases passed successfully, resulting in a full success rate. No functional errors were identified; remaining observations were suggestions to speed up API calls (Kitchenham, Linkman and Law, 1997).

The student dashboard supports simple identification of features using a green and blue color scheme with large, clickable buttons for primary actions (Submit UG Form, File a Complaint, Request Documents) (Martins and Kellermanns, 2004). Information is laid out cleanly on the teacher dashboard and admin panel using tables, and toast notifications confirm that user actions have been registered by the application, reducing user error and building confidence in the system.

3.2 System Performance

- The system correctly processed all five types of forms (UG Forms, Transcript Request, FYP Proposals, Leave Application).
- Admin users were able to approve and reject these forms.
- Each action pushed a notification to the student's notification tab and updated the corresponding status in the database.
- Profile edits for both students and teachers were successfully saved to MongoDB and correctly displayed on page refresh.
- Login credentials were properly restricted by role.

Test Case	Description	Expected Result	Actual Result	Status
User Sign Up (TC-1)	Verify new account registration	Successful registration, redirect to login page	As expected	PASS
User Login (TC-2)	Verify existing user authentication	Successful login, redirect to home page	As expected	PASS



Test Case	Description	Expected Result	Actual Result	Status
Profile Management (TC-3)	Verify profile view and edit	Save profile information of student/teacher successfully	As expected	PASS
UG Form Submission (TC-4)	Verify submission of UG forms	Students can fill and submit forms successfully	As expected	PASS
Admin View Users (TC-5)	View all registered users	Successful login, redirect to admin page	As expected	PASS

Table 5. Test case results summary.

4. DISCUSSION

4.1 Significance of Findings

The testing team successfully passed functional tests across all defined test cases, exercising each main workflow — including form creation, role-based access, and approval routing — without failure.

This system builds on the document management approach of (Rohini *et al.*, 2024) by adding real-time notifications and admin access to activity logs. (Thoring, Rudolph and Vogl, 2017) implemented a similar web-based department system, but did not disclose their testing results.

4.2 Current Limitations

The application is currently usable for only one department and has not been stress-tested for large-scale traffic. Security features such as two-factor authentication and session timeout are planned for future updates.

As of now, DMS is deployed only in a single department and has been tested only within the development environment, not under massive load. Nonetheless, DMS represents a step toward making public universities in Pakistan paperless: as a platform-independent, browser-based application, it requires no additional installation.

4.3 Future Enhancements

The following enhancements are proposed for future versions of the Department Management System (Terras, 2011):

- Cloud computing integration
- Support for mobile applications
- Biometric attendance
- Artificial intelligence-based analytics and reports
- SMS and email alert system
- Chat support for faculty and students
- Multiple department management



5. CONCLUSION

The Department Management System successfully digitizes core departmental processes that were previously operated manually. Paperwork was reduced, transparency was increased, approval processes were accelerated, and centralized academic management was made available under one roof on a secured web-based platform. Filled-out forms are now stored electronically, routed efficiently based on the user's role, and displayed on intuitive dashboards for Students, Teachers, and Administrators (Bansal and Chawla, 2016). Recurring problems in academic administration — lost documents, slow approvals, and a lack of communication and transparency — can be addressed through DMS as a unified, department-wide web application. DMS passed functional testing across all defined cases, validating it as a working product built with React.js, Node.js, Express.js, and MongoDB, with long-term maintenance and scalability in mind. As a mobile-responsive web application, it can be used by anyone with browser access and requires no additional prerequisites.

DMS represents only the beginning of digitalization for the department. Future work includes interfacing DMS with the university's central student information system, and eventually offering mobile app support, administrative analytics dashboards, and expanded form support as departmental needs evolve. DMS moves the department further toward a fully paperless operation and, with minimal changes, could serve as a template for other departments and universities.

REFERENCES

- Agarwal, J. and Singh, R. (2020) 'Department management system: web-based application', *International Journal of Engineering and Applied Science Technology*, 5, pp. 356–361.
- Bansal, R. and Chawla, S. (2016) 'Design and development of semantic web-based system for computer science domain-specific information retrieval', *Perspectives in Science*, 8, pp. 330–333.
- Bharamagoudar, S., Geeta, R. and Totad, S. (2013) 'Web based student information management system', *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6), pp. 2342–2348.
- Botta-Genoulaz, V., Millet, P. and Grabot, B. (2005) 'A survey on the recent research literature on ERP systems', *Computers in Industry*, 56(6), pp. 510–522.
- Ferrari, C. and Hurst, A. (2021) 'Accessible web development: opportunities to improve the education and practice of web development with a screen reader', *ACM Transactions on Accessible Computing*, 14(2), pp. 1–32.
- Kitchenham, B., Linkman, S. and Law, D. (1997) 'DESMET: a methodology for evaluating software engineering methods and tools', *Computing and Control Engineering*, 8(3), pp. 120–126.
- Martins, L. and Kellermanns, F. (2004) 'A model of business school students' acceptance of a web-based course management system', *Academy of Management Learning & Education*, 3(1), pp. 7–26.
- Rajmane, S., Mathpati, S. and Dawle, J. (2016) 'Digitalization of management system for college and student information', *Research Journal of Science and Technology*, 8(4), p. 179.
- Ricca, F. and Tonella, P. (2001) *Analysis and testing of web applications*. IEEE, pp. 25–34.
- Rohini, K. et al. (2024) 'A web-based department management system for academic institutions', *International Journal of Advanced Engineering Research and Science*, 11, p. 5.
- Terras, M. (2011) 'The rise of digitization', in *Digitisation Perspectives*. pp. 3–20.
- Thoring, A., Rudolph, D. and Vogl, R. (2017) 'Digitalization of higher education from a student's point of view', *European Journal of Higher Education IT*, 47, pp. 1–10.