**Software Requirements Specification (SRS)**

## 1. Introduction

### 1.1 Purpose

The purpose of the Digital Complaint and Maintenance Management System (DCMMS) is to provide an efficient, transparent, and centralized platform for managing maintenance-related complaints in institutional environments such as universities, colleges, hostels, and corporate organizations. The system allows students and staff to submit complaints, track their status, receive updates, and view progress. Administrators can monitor complaints, assign them to maintenance staff, track delays, and issue warnings. Maintenance staff can update the work status, view assigned tasks, and respond to admin feedback.

This document outlines the functional and non-functional requirements of the system, ensuring clarity for developers, testers, and stakeholders.

### 1.2 Document Conventions

* **Abbreviations**
  + **UI**: User Interface
  + **Admin**: System role responsible for monitoring, assigning, and managing complaints.
  + **Student User**: End-user (student) submitting complaints.
  + **Maintenance Staff User**: End-user responsible for resolving assigned complaints.
* **Standards**
  + This document follows **IEEE standards** for SRS documentation.



### 1.3 Intended Audience and Reading Suggestions

* **Project Developers** → For design and coding reference.
* **Testers/QA Team** → To understand test cases and validation requirements.
* **Institutional Authorities/Admins** → To ensure requirements align with operational needs.
* **Maintenance Staff** → To understand system interactions for task updates.

It is recommended that **developers and testers** focus on Sections 2, 3, and 4, while **administrators** review Sections 1 and 2 for understanding system scope and functionality.

### 1.4 Project Scope

The DCMMS simplifies maintenance complaint handling by providing:

* A **student panel** to file complaints, track progress, and receive updates.
* An **admin panel** to view, assign, monitor, and manage complaints.
* A **maintenance staff panel** to view assigned work, update progress, and receive admin warnings.

Key benefits include:

* Transparency in complaint resolution.
* Faster task assignment and updates.
* Tracking overdue complaints (>7 days).
* Reduced paperwork and manual errors.

### 1.5 References

* IEEE SRS Standard Documentation Template.
* Internal project discussions and UI design documents.
* Previous Complaint Management workflows.

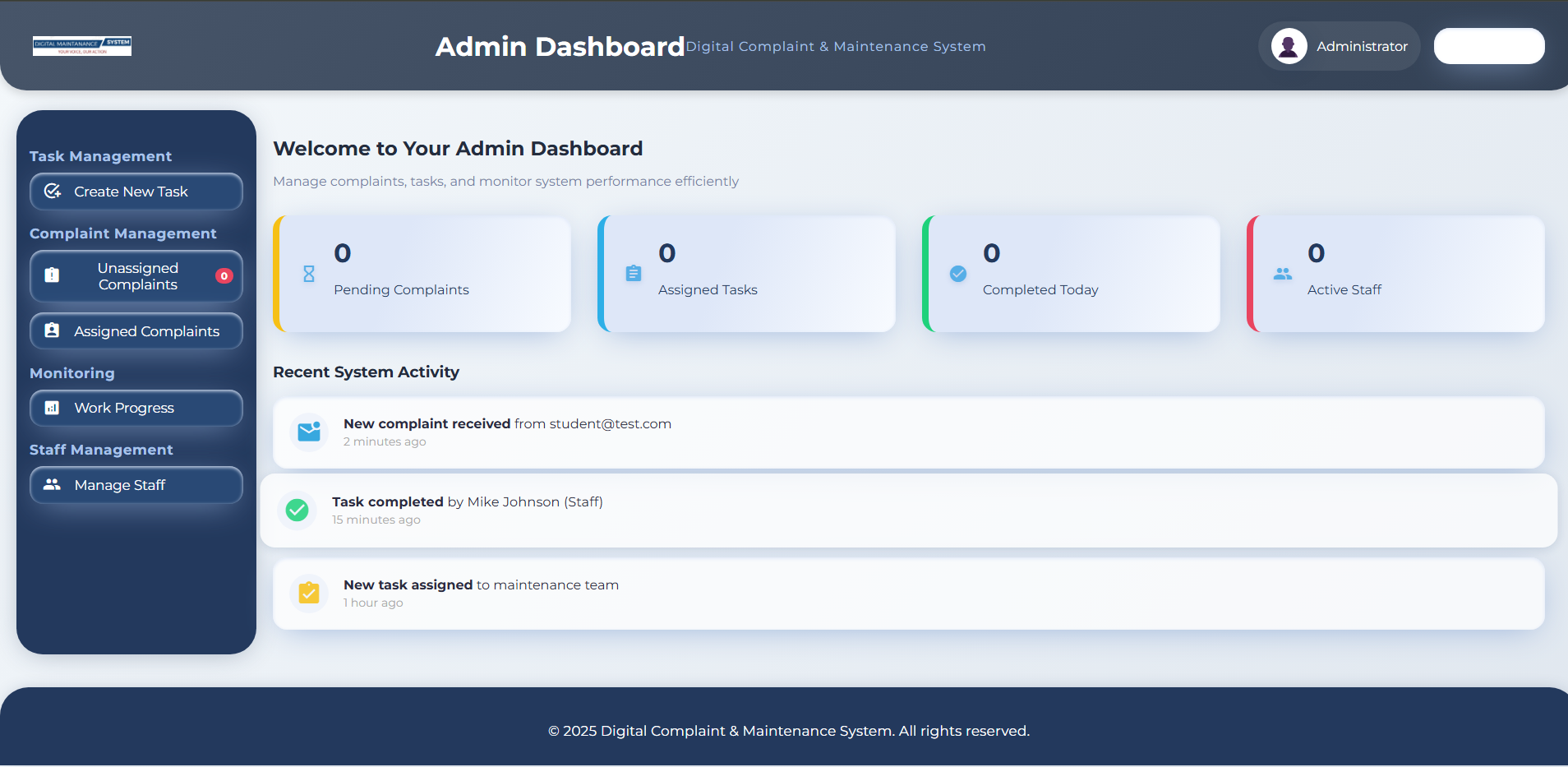
## 2. Overall Description

### 2.1 Product Perspective

The system will be a **web-based application** accessible via desktop and mobile browsers. It consists of three user panels (Admin, Student, Maintenance Staff). It integrates with a backend database (MySQL) and uses PHP for server-side operations.

### 2.2 Product Features

* **User Authentication** (Login/Logout for students, admin, staff).
* **Complaint Management** (File, Assign, Update, Resolve complaints).
* **Progress Tracking** (Students track complaint progress).
* **Delayed Complaint Warning** (Admin gets notified for tasks pending >7 days).
* **Admin Notifications/Warnings** (Admin can send warnings to staff).
* **Reports & History** (Complaint history for admin, staff, and students).



### 2.3 User Classes and Characteristics

* **Students**: Submit complaints, view status, receive notifications. Basic computer literacy assumed.
* **Admin**: Assign complaints, manage delays, issue warnings, view reports. Moderate technical knowledge.
* **Maintenance Staff**: View assigned tasks, update status, respond to warnings. Basic system navigation skills required.

### 2.4 Operating Environment

* **Frontend**: HTML, CSS, JavaScript
* **Backend**: PHP
* **Database**: MySQL (via phpMyAdmin)
* **Server Environment**: Apache/XAMPP/WAMP stack
* **Supported Devices**: Desktop, Laptop, Mobile (Responsive UI)

### 2.5 Design and Implementation Constraints

* Must support **role-based authentication**.
* Complaints must be updated in **real-time**.
* Internet access required for cloud/server deployment.
* Limited to institutional complaint workflows only.

### 2.6 User Documentation

* User Guide for Students (How to file and track complaints).
* Admin Guide (Assigning, monitoring, sending warnings).
* Staff Guide (Updating complaint progress, replying to warnings).

### 2.7 Assumptions and Dependencies

* Users have basic internet and device access.
* Server must be operational 24/7 for real-time tracking.
* Dependencies on institutional IT infrastructure.

## 3. System Features

### 3.1 User Authentication

* **Description**: Users (Students, Admin, Staff) must log in using role-specific credentials.
* **Functional Requirements**:  
  + FR-1: The system must validate credentials before granting access.
  + FR-2: Different dashboards based on role (Student/Admin/Staff).

### 3.2 Complaint Management (Student Panel)

* **Description**: Students file complaints and track progress.
* **Functional Requirements**:  
  + FR-3: Students can submit complaints with description, category, and optional image.
  + FR-4: Students can view status (Pending, In-Progress, Resolved).

### 3.3 Complaint Assignment (Admin Panel)

* **Description**: Admin assigns complaints to maintenance staff.
* **Functional Requirements**:  
  + FR-5: Admin can view unassigned complaints.
  + FR-6: Admin assigns complaints to specific staff.

### 3.4 Work Updates (Staff Panel)

* **Description**: Maintenance staff update assigned complaint status.
* **Functional Requirements**:  
  + FR-7: Staff can mark work as “In Progress” or “Resolved.”
  + FR-8: Staff can reply to admin warnings.

### 3.5 Delayed Complaint Alerts

* **Description**: Admin is notified when tasks exceed 7 days.
* **Functional Requirements**:  
  + FR-9: The system automatically flags complaints unresolved >7 days.
  + FR-10: Admin can send warnings to staff.

## 4. External Interface Requirements

### 4.1 User Interfaces

* **Students**: Dashboard, File Complaint, My Complaints, View Progress.
* **Admin**: Dashboard, Unassigned Complaints, Work Progress, Staff Warnings.
* **Maintenance Staff**: Dashboard, Assigned Work, Update Work, View Warnings.

### 4.2 Hardware Interfaces

* Supports desktops, laptops, and mobile devices with modern browsers.

### 4.3 Software Interfaces

* PHP–MySQL integration for backend.
* No third-party API dependencies at present.

### 4.4 Communication Interfaces

* HTTPS (for secure data transmission).

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

* Page load time < 3 seconds.
* Real-time complaint update within 5 seconds.

### 5.2 Safety Requirements

* Daily database backups.
* Prevent data loss on server crash.

### 5.3 Security Requirements

* Role-based access control.
* Password encryption (MD5/SHA).
* Session management to prevent unauthorized access.

### 5.4 Software Quality Attributes

* Usability: Intuitive, responsive design.
* Reliability: 99% uptime on institutional server.
* Maintainability: Modular code for easy updates.
* Scalability: Can support multiple institutions.

## 6. Other Requirements

* The system must support both **English and Bangla UI text (future expansion)**.
* Reports should be exportable in PDF/CSV formats.

## Appendices

**A. Acronyms**

* UI: User Interface
* DB: Database

**B. Analysis Models**

* Use Case Diagrams (Admin, Student, Staff interaction flows).
* Entity-Relationship (ER) Diagram for complaints management.

**C. Issues List**

* Future integration of mobile app version.
* Voice command support (Bangla) as a future feature.