Functional Requirements Document (FRD)

Student Admission System

Document Information

- **Document Version**: 1.0
- **Date**: December 2024
- **Author**: Development Team
- **Project**: Student Admission System

---

1. Introduction

1.1 Purpose

This document outlines the functional requirements for the Student Admission System, a web-based application that enables students to submit admission applications and allows administrators to review, approve, or reject these applications.

1.2 Scope

The system covers the complete admission workflow from application submission to final decision and admission letter generation.

1.3 Definitions and Abbreviations

- **SAS**: Student Admission System
- **FRD**: Functional Requirements Document
- **PDF**: Portable Document Format
- **API**: Application Programming Interface
- **UI**: User Interface

---

2. System Overview

2.1 System Purpose

The Student Admission System streamlines the admission process by providing:

- Online application submission for students
- Document upload and management
- Administrative review interface
- Automated admission letter generation
- Application status tracking

2.2 System Users

1. **Students**: Submit applications and check status
2. **Administrators**: Review applications and make admission decisions

---

3. Functional Requirements

3.1 Student Application Module

3.1.1 Application Form Submission

FR-001: Student Application Form

- **Description**: Students must be able to submit admission applications through a web form
- **Priority**: High
- **Inputs**:
  - Personal information (name, email, phone, address, date of birth)
  - Academic information (course applied, previous qualification, CGPA)
  - Document uploads (degree certificate, ID proof)
- **Processing**:
  - Form validation for required fields
  - Email format validation
  - Phone number format validation
  - File format validation (PDF, JPG, PNG only)
  - Maximum file size limit (16MB)
- **Outputs**:
  - Unique application ID generation
  - Success confirmation message
  - Application stored in database
- **Business Rules**:
  - All mandatory fields must be completed
  - Valid email address required
  - Phone number must be 10-20 digits

    - Only specified file formats allowed
    - Application ID format: APP + YYYYMMDD + 8-character UUID

3.1.2 Document Upload

FR-002: Document Management

- **Description**: Students must upload required documents as part of their application
- **Priority**: High
- **Supported Formats**: PDF, JPG, JPEG, PNG
- **Maximum Size**: 16MB per file
- **Required Documents**:
    - Degree certificate
    - Government-issued ID proof
- **Security**: Files stored with unique names to prevent conflicts

3.1.3 Application Status Tracking

FR-003: Status Inquiry

- **Description**: Students can check their application status using their application ID
- **Priority**: Medium
- **Inputs**: Application ID
- **Outputs**:
    - Application status (Pending, Approved, Rejected)
    - Application details
    - Admin comments (if any)
    - Download link for admission letter (if approved)
- **Status Types**:
    - **Pending**: Application submitted, awaiting review
    - **Approved**: Application accepted, admission letter available
    - **Rejected**: Application declined with reason

3.2 Administrative Module

3.2.1 Admin Authentication

FR-004: Admin Login System

- **Description**: Secure authentication system for administrators
- **Priority**: High
- **Inputs**: Username and password
- **Processing**: Password hash verification
- **Outputs**: Authenticated session or error message
- **Security**: Password hashing using Werkzeug security

3.2.2 Application Review Interface

FR-005: Admin Dashboard

- **Description**: Centralized interface for reviewing all applications
- **Priority**: High
- **Features**:
  - List all applications with summary information
  - Filter by status (Pending, Approved, Rejected)
  - Application statistics display
  - Quick access to individual application reviews
- **Information Displayed**:
  - Application ID
  - Student name and email
  - Course applied
  - Application date
  - Current status

3.2.3 Application Review Process

FR-006: Individual Application Review

- **Description**: Detailed review interface for individual applications
- **Priority**: High
- **Features**:
  - Complete student information display
  - Document viewing capabilities
  - Status change functionality (Approve/Reject)
  - Comment system for feedback
- **Processing**:
  - Status update in database
  - Timestamp recording
  - Admin comment storage
- **Outputs**:
  - Updated application status
  - Automatic PDF generation (if approved)

3.3 Document Generation Module

3.3.1 Admission Letter Generation

FR-007: PDF Admission Letter

- **Description**: Automatic generation of PDF admission letters for approved applications
- **Priority**: High

- **Trigger**: Application status changed to "Approved"
- **Content**:
  - University letterhead
  - Student personal information
  - Course details
  - Application ID
  - Approval date
  - Instructions for next steps
- **Format**: Professional PDF document
- **Storage**: Server-side storage with secure access
- **Download**: Secure download link for students

3.4 API Module

3.4.1 RESTful API Endpoints

FR-008: API Interface

- **Description**: RESTful API for application data access
- **Priority**: Medium
- **Endpoints**:
  - `GET /api/applications` - List all applications
  - `GET /api/application/<id>` - Get specific application details
- **Authentication**: Admin session required
- **Response Format**: JSON
- **Use Case**: Integration with external systems or alternative front-ends

3.5 File Management Module

3.5.1 File Upload and Storage

FR-009: File Management System

- **Description**: Secure file upload and storage system
- **Priority**: High
- **Features**:
  - Unique filename generation
  - File type validation
  - Size limit enforcement
  - Secure file access
- **Storage Locations**:
  - `uploads/` - Student uploaded documents
  - `admission_letters/` - Generated PDF letters

4. Non-Functional Requirements

4.1 Performance Requirements

- **Response Time**: Web pages should load within 3 seconds
- **Concurrent Users**: System should handle 100 concurrent users
- **File Upload**: Maximum 16MB file size with progress indication

4.2 Security Requirements

- **Authentication**: Secure admin login with password hashing
- **File Security**: Uploaded files stored with unique names
- **Session Management**: Secure session handling
- **Input Validation**: All user inputs validated and sanitized

4.3 Usability Requirements

- **Interface**: Intuitive and user-friendly web interface
- **Responsiveness**: Mobile-friendly responsive design
- **Accessibility**: Bootstrap-based UI for consistency
- **Feedback**: Clear success and error messages

4.4 Reliability Requirements

- **Database**: Data consistency and integrity
- **Error Handling**: Graceful error handling with user-friendly messages
- **Backup**: Regular database backups (implementation dependent)

---

5. System Constraints

5.1 Technical Constraints

- **Framework**: Python Flask web framework
- **Database**: SQLAlchemy ORM with SQLite/PostgreSQL/MySQL
- **File Storage**: Local file system storage
- **PDF Generation**: ReportLab library

5.2 Business Constraints

- **File Types**: Only PDF, JPG, JPEG, PNG files accepted
- **Application ID**: Unique identifier required for status tracking

- **Admin Access**: Single admin role with full privileges

---

6. Assumptions and Dependencies

6.1 Assumptions

- Users have modern web browsers with JavaScript enabled
- Administrators have basic computer literacy
- Internet connectivity available for all users
- File storage space adequate for uploaded documents

6.2 Dependencies

- Python 3.7+ runtime environment
- Flask framework and related libraries
- Database system (SQLite for development, PostgreSQL/MySQL for production)
- ReportLab for PDF generation
- Web server for deployment (development server for testing)

---

7. Validation Criteria

7.1 Acceptance Criteria

- Students can successfully submit applications with all required information
- Administrators can review and process applications efficiently
- PDF admission letters are generated correctly for approved applications
- Application status tracking works accurately
- File uploads function properly with validation
- System handles errors gracefully

7.2 Testing Requirements

- All functional requirements must pass unit tests
- Integration tests verify end-to-end workflows
- Form validation tests ensure data integrity
- Security tests verify authentication and authorization
- Performance tests validate response times

---

8. Future Enhancements

8.1 Potential Features

- Email notifications for status changes
- Bulk application processing
- Advanced reporting and analytics
- Document version control
- Integration with payment systems
- Multi-level approval workflow
- Student portal with additional features

8.2 Scalability Considerations

- Database optimization for large volumes
- Cloud storage integration
- Load balancing for high traffic
- Caching mechanisms for better performance

---

9. Glossary

Term	Definition
Application ID	Unique identifier assigned to each student application
CGPA	Cumulative Grade Point Average
PDF	Portable Document Format
Session	Authenticated user session in the web application
Status	Current state of an application (Pending, Approved, Rejected)
Upload	Process of transferring files from client to server
Validation	Process of checking data integrity and format
