**Requirement Analysis for Online Admission Process**

# Introduction

This document outlines the requirement analysis for developing an online admission process system. The goal is to create a streamlined, automated system that improves efficiency and user experience for prospective students and administrative staff.

# 2. Aim

The aim of the online admission process system is to automate and enhance the application process for prospective students. This system will reduce administrative overhead, increase efficiency, and provide a seamless experience for both applicants and admissions staff.

# 3. Scope

**Inclusions:**

* **Application Submission:** Allows applicants to submit applications online.
* **Document Upload:** Facilitates the upload of required documents such as transcripts and certificates.
* **Application Review:** Provides tools for staff to review and manage applications.
* **Communication:** Automated notifications about application status and requirements.
* **Integration:** Compatibility with existing student information systems.

**Exclusions:**

* Post-admission activities such as course registration are not covered in this system.

# 4. Objectives

1. **Automate the Application Process:** Reduce manual data entry and processing.
2. **Enhance User Experience:** Provide a user-friendly interface for applicants and staff.
3. **Improve Data Accuracy:** Minimize errors through automated validation and integration.
4. **Increase Efficiency:** Streamline the review and decision-making process.

# 5. Purpose

The purpose is to modernize the admissions process by leveraging technology to create an efficient, transparent, and user-friendly online system for handling student applications.

# 6. Goals

1. **User-Friendly Interface:** Design an intuitive and accessible interface.
2. **Timely Processing:** Ensure applications are processed within 10 business days.
3. **Data Security:** Implement robust security measures to protect sensitive information.
4. **Scalability:** Build a system that can accommodate increasing numbers of applications.

# 7. Stakeholder Analysis

* **Applicants:** Individuals applying for admission who need a straightforward, easy-to-use system.
* **Admissions Staff:** Personnel who will review, process, and manage applications.
* **IT Support:** Technical team responsible for system maintenance and support.
* **University Administration:** Decision-makers interested in system performance and integration.

# 8. Functional Requirements Specification

* **User Registration:** Applicants must be able to register and create accounts.
* **Application Form:** The system should support various data fields with validation rules.
* **Document Management:** Allow uploading, storing, and retrieving application documents.
* **Review Workflow:** Provide functionality for processing and reviewing applications.

# 9. Non-Functional Requirements Specification

* **Performance:** Handle up to 1000 simultaneous users.
* **Security:** Data encryption and robust user authentication.
* **Usability:** Interface should be accessible and easy to navigate.

# 10. Regulatory and Compliance Requirements

* **Data Privacy:** Adhere to GDPR or local data protection laws.
* **Accessibility:** Follow WCAG guidelines to ensure accessibility for users with disabilities.
* **Legal Compliance:** Meet all legal requirements relevant to educational institutions.

# 11. Integration Requirements

* **Existing Systems:** Integrate with the university’s Student Information System (SIS).
* **Payment Gateway:** If applicable, integrate with a payment gateway for handling application fees.
* **Email Systems:** Synchronize with email servers for notifications.

# 12. User Experience (UX) Design Requirements

* **Navigation:** Simple menus and clear navigation paths.
* **Error Handling:** Informative error messages and help options.
* **Feedback Mechanisms:** Provide feedback on submissions and actions.

# 13. Acceptance Criteria

* **Functionality:** All specified features must work as intended.
* **Performance:** Meet performance benchmarks under load conditions.
* **User Satisfaction:** Positive feedback from user acceptance testing.

# 14. Change Management and Training

* **Change Management:** Process for handling system updates and modifications.
* **Training:** Develop and deliver training materials and sessions for users and administrators.

# 15. Maintenance and Support Requirements

* **Support:** Establish a helpdesk or ticketing system for user support.
* **Maintenance:** Schedule regular updates, patches, and performance monitoring.

# 16. Risk Analysis

* **Technical Risks:** Potential issues with technology or system integration.
* **Operational Risks:** Risks related to user adoption and process changes.
* **Mitigation Plans:** Strategies to address and mitigate identified risks.

# 17. Literature Review

* **Studies and Best Practices:** Research on the impact and best practices of online admission systems.
* **Case Studies:** Examples of successful implementations and their outcomes.

**18. Hardware and Software Requirements**

**Hardware:**

* **Server:** Quad-core 3.0 GHz, 16 GB RAM, 500 GB SSD, 100 Mbps internet.
* **Client:** Modern web browsers, internet connection.

**Software:**

* **OS:** Linux-based or Windows Server.
* **Web Server:** Apache or Nginx.
* **Database:** MySQL or PostgreSQL.
* **Application Framework:** Django, Ruby on Rails, or similar.
* **Security:** SSL/TLS certificates.

# 19. Documentation

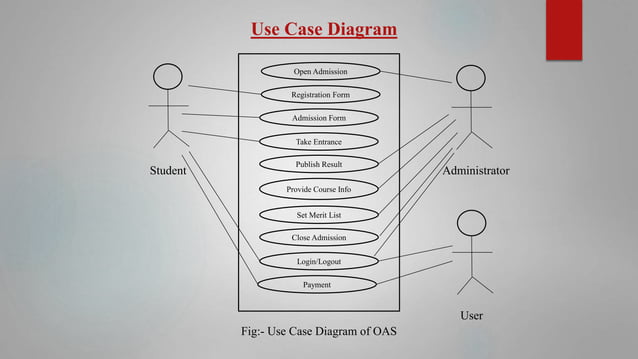
* **System Requirements Document:** Detailed functional and non-functional requirements.
* **User Manuals:** Instructions for applicants and staff.
* **Technical Specifications:** Documentation for developers.
* **Testing Plan:** Strategies for testing, including unit, integration, and user acceptance tests.

**Use Case Diagram**

A Use Case Diagram is a type of Unified Modeling Language (UML) diagram that represents the interaction between actors (users or external systems) and a system under consideration to accomplish specific goals. It provides a high-level view of the system’s functionality by illustrating the various ways users can interact with it.

**Key Components of Use Case Diagram:**

* **Actors:** These are the entities that interact with the system. They can be users, other systems, or hardware devices.
* **Use Cases:** These are the specific actions or services the system performs in response to interactions with the actors. They are typically represented by ovals.
* **System Boundary:** This is a rectangle that defines the scope of the system, showing what is inside and outside the system.
* **Relationships:** These depict how actors and use cases interact with each other. Common relationships include associations, generalizations, and dependencies



**Class Diagram**

A class diagram is a type of diagram used in software engineering to visually represent the structure and relationships of classes within a system. It’s a fundamental part of the Unified Modeling Language (UML), which is a standardized modeling language used to design and document software systems.

In a class diagram, classes are depicted as boxes divided into three compartments:

* **Class Name:** The top compartment contains the name of the class.
* **Attributes:** The middle compartment lists the attributes (or properties) of the class.
* **Methods:** The bottom compartment lists the methods (or functions) that the class can perform

