Requirements SRS

Functional Requirements

Student Management

1. **Register students**:
    - Enter basic student details such as name, ID number, email address, address, and phone number.
    - Assign an academic supervisor for each new student.
2. **Update student information**:
    - Provide an interface for updating personal (such as address and phone number) and academic (such as academic supervisor and courses) student information.
3. **View Student Grades**:
    - Enable students to view their grades via a customised interface with course and grade details.
4. **Enrol students in courses**:
    - Provide a course registration interface that displays all available courses with information such as schedule, location, and number of available places.
    - Check availability of places before completing the enrolment process.
5. **Paying tuition fees**:
    - View details of tuition fees due, including the amount due and the due date.
    - Support tuition payment via a secure payment gateway, with the payment status automatically updated after completion of the process.

Course Management

1. **Add new courses**:
    - Add new courses with course name, description, number of hours, schedule, location, and number of slots available.
2. **Update course information**:
    - Update available course information such as schedule, location, and number of hours.
3. **View All Courses**:
    - View a comprehensive list of all available courses with the ability to search and filter by academic department or grade levels.
4. **Check Course Availability**:
    - Check course availability before completing the enrolment process.
5. **Enter and update grades**:
    - Enter student grades for courses and update them as needed, ensuring that grades can be viewed for students.

Library Management

1. **Add new books**:
    - Add new books to the system with details such as title, author, status (available/loaned), and ID number.
2. **Update book information**:
    - Update available book information such as status, availability, and identify the borrowing student in case of borrowing.
3. **Borrow books**:
    - Provide students with a book borrowing interface that displays all available books with the ability to search and filter.
    - Check the availability of the book before completing the borrowing process.
4. **Return books**:
    - Provide an interface to return borrowed books and automatically update the status of the book to ‘Available’.
5. **View all books**:
    - Display a comprehensive list of all available books with the ability to search and filter by title or author.

Department Management

1. **Add new departments**:
    - Add new academic departments with details such as department name and department head.
2. **Update Department Information**:
    - Update available academic department information such as department name and department head.
3. **Assign Department Heads**:
    - Assign and update academic department heads as needed.

Fee Management

1. **View Tuition Fees**:
    - View details of tuition fees owed by students with the ability to view invoices and pay online.
2. **Send Invoices**:
    - Generate and send invoices to students via email or through the system, with details of the amount due and due date.
3. **Update Payment Status**:
    - Update payment status after receipt of tuition fees, with the ability to verify online payment and automatically update the status.
4. **Generate invoices**:
    - Generate new invoices for students with details of the fees due, due date and any additional fees.

Non-Functional Requirements

Performance

1. **System responsiveness**:

    - The system should respond to user requests within 2-3 seconds for most operations.
2. **Endurance**:
    - The system must be able to process at least 2000 requests per hour without performance degradation.

Security

1. **User authentication**:
    - The system must include strict user authentication procedures to ensure only authorized access.
2. **Data encryption**:
    - Sensitive data such as student information and tuition fees should be encrypted during transmission and storage using advanced encryption techniques.

Usability

1. **User Interface**:
    - The user interface should be simple and easy to use, with clear instructions for users.
2. **Hardware Compatibility**:
    - The system must be compatible with all types of devices (computers, tablets, smartphones).

Maintainability

1. **System documentation**:
    - Comprehensive documentation should be provided for all parts of the system, including source code, design diagrams, and user manuals.
2. **Scalability**:
    - The system should be designed so that it can be easily extended and upgraded to add new features or improve performance.

Reliability

1. **Failure recovery**:
    - The system must be able to properly recover from failures without loss of data.
2. **Availability**:
    - The system must be available at least 99.9% of the time.

Portability

1. **System compatibility**:
    - The system must be able to run on different operating platforms (e.g. Windows, Linux, macOS) without requiring major modifications.
2. **Data migration**:

- The system should be able to easily import and export data between different systems.