**Project Design Phase-II**

**Solution Requirements (Functional & Non-functional)**

|  |  |
| --- | --- |
| Date | 20 June 2025 |
| Team ID | LTVIP2025TMID28652 |
| Project Name | Educational Organisation Using ServiceNow |
| Maximum Marks | 4 Marks |

For educational organizations using ServiceNow, functional requirements include features for student lifecycle management, course delivery, resource management, and reporting. Non-functional requirements focus on performance, security, usability, and scalability of the ServiceNow platform. These requirements ensure the system effectively supports educational processes while maintaining reliability and user satisfaction

**Functional Requirements:**

**• Student Lifecycle Management:**

**• Admissions:** Manage applications, track progress, and automate acceptance/rejection processes.

**• Enrollment:** Handle course registration, waitlists, and student records management.

**• Student Information:** Maintain comprehensive student profiles with academic history, contact details, and financial information**.**

**• Graduation:** Manage graduation applications, degree audits, and diploma issuance.

**• Alumni Management:** Track alumni information, engagement, and communication.

**• Course Delivery:**

**• Course Catalog:** Create, manage, and publish course information, descriptions, and schedules.

**• Learning Management System (LMS) Integration**: Integrate with existing LMS platforms for content delivery, assignments, and assessments.

**• Virtual Classroom Support:** Enable online learning through integrations with video conferencing tools.

**• Scheduling:** Manage room bookings, class schedules, and instructor availability.

**• Resource Management:**

**• Asset Management:** Track and manage physical and digital assets (e.g., computers, software, library resources).

**• Facility Management:** Manage building maintenance, space allocation, and resource utilization.

**• Inventory Management:** Track and manage supplies and materials used in educational programs.

**• Reporting & Analytics:**

**• Standard Reports:** Generate reports on student performance, resource utilization, and financial data.

**• Custom Reports:** Allow users to create custom reports based on specific needs.

**• Data Visualization:** Present data in an accessible and understandable format.

**• Other Functional Requirements:**

**• Knowledge Management:** Create and manage a knowledge base for frequently asked questions and common issues.

**• Incident Management:** Handle IT support requests and track resolution progress.

**• Change Management:** Manage changes to the system and educational processes.

**• Service Catalog:** Provide a catalog of services available to students and staff.

**Non-Functional Requirements:**

**• Performance:**

• Response Time: Ensure fast loading times for pages and efficient processing of transactions.

• Scalability: Ability to handle increasing numbers of users, courses, and data without performance degradation.

• Availability: Maintain high uptime and ensure the system is accessible when needed.

**• Security:**

• Access Control: Implement role-based access control to restrict access to sensitive information.

• Data Encryption: Protect sensitive data through encryption both in transit and at rest.

• Compliance: Adhere to relevant security standards and regulations (e.g., FERPA, GDPR).

**• Usability:**

• User-Friendly Interface: Provide an intuitive and easy-to-navigate interface for all users.

• Accessibility: Ensure the system is accessible to users with disabilities.

• Mobile-Friendly: Provide a responsive design that works well on different devices.

**• Reliability:**

• Backup and Recovery: Implement robust backup and recovery procedures to minimize data loss.

• Disaster Recovery: Have a plan in place to recover from major disruptions

• Error Handling: Provide clear error messages and guidance to users.

**• Integration:**

• API Integration: Allow for seamless integration with other systems (e.g., student information systems, financial systems).

• Data Integration: Ensure data consistency and accuracy across integrated systems.

**• Maintainability:**

• Modularity: Design the system to be easily maintained and updated. [39, 39, 40]

• Documentation: Provide clear and comprehensive documentation for administrators and users.

• Support: Provide ongoing support and maintenance to ensure the system continues to function effectively