



STUDENT MANAGEMENT SYSTEM Software Requirements Specification

TERMS OF REFERENCE

for

STUDENT MANAGEMENT SYSTEM SOLUTION VENDORS/DEVELOPERS

Dean Foy
Director ICT
Information and Communications Department
P.O Box R113 | Honiara | Solomon Islands

Tel: +677 42837 | Mobile: +677 7441760

Email: Director.ICT@sinu.edu.sb | Web: www.sinu.edu.sb

SUMMARY OF FUNDAMENTAL FUNCTIONS & OPTIONAL SMS FUNCTIONS

Based on the required functionalities for the proposed Student Management System (SMS) at Solomon Islands National University (SINU), here is the categorized breakdown of these into the core fundamental and optional functionalities:

Core Fundamental Functionalities:

These are essential for the core operations of the university's academic and administrative management:

1. **Web based Interface** - Central access point for all system interactions.
2. **System Administration** - Essential for managing user roles, security, and system configurations.
3. **Organization Setup** - Fundamental for structuring university departments and faculties.
4. **Faculty Management** - Key for managing faculty details and responsibilities.
5. **Pre-enrolment Course Counselling and Entrance Examinations** - Supports student orientation and selection.
6. **International Students** - Manages international student specifics and international enrolments.
7. **Online Application System** - Crucial for handling admissions processes.
8. **Admission Management** - Central to processing and managing new student admissions.
9. **Online Enrolment & Course Registration** - Essential for student registration for courses.
10. **Fees Management** - Critical for handling tuition and other fees.
11. **Scholarship / Sponsorship Management** - Important for managing financial aid and scholarships.
12. **Program & Course Management** - Fundamental for curriculum management and course details.
13. **Time Table Management** - Essential for scheduling classes and resources.
14. **Student Attendance Management** - Crucial for tracking student attendance.
15. **Examinations & Mark Sheet Management** - Core to academic assessments and grading.
16. **Assignment Management** - Important for managing coursework and assignments.
17. **Library Reservation and Borrowed Books Module** - Enhances library services.
18. **Online Learning Management System** - Essential for delivering and managing online education.
19. **Completion Management (Graduation & Certificates)** - Critical for managing end-of-program certifications and graduations.
20. **Student Academic Records** - Fundamental for maintaining secure and accurate academic histories.

21. **Staff Portal** - Facilitates staff access to necessary tools and information.
22. **Messaging & Notifications** - Enhances communication and is a useful tool.
23. **Analytics and Reports** - Necessary for data-driven decision-making and reporting.
24. **Academic Staff Management** - Essential for managing academic staff details and roles.
25. **Hostel Management** - Manages student accommodations, useful but not critical.
26. **Student Dining Management** - Manages meal services, enhances student life.
27. **Student Administration Services** - Provides administrative support, optional enhancement.
28. **Certificates, Awards, & Document Management** - Important for recognition but not core system functionality.
29. **On-going Support & on-site full user training** - Critical for system adoption and categorized as fundamental as it pertains to post-implementation training and support.

Optional Functionalities:

These enhance the user experience or provide additional services that are not critical to the basic functioning of the university:

1. **Campaigns & Enquiries Management** - Useful for marketing and handling prospective student inquiries.
2. **Student Bookshop** - Convenient for students.
3. **Student Printing** - Service that facilitates student document handling.
4. **Learning Space Booking** - Useful for managing study spaces and resources.
5. **IT Lab Computer Booking** - Helps in managing IT resource allocation.
6. **Student Association & Affiliated Student Groups & Clubs** - Supports student life.
7. **Elections & Surveys** - Useful for student governance and feedback collection.
8. **Student Appointments with Staff System** - Facilitates student booking meetings with staff.
9. **Student Welfare & Counselling** - Important for student support and wellbeing.
10. **Thesis Management & Publication** - Supports graduate students, more specific to higher education levels.
11. **University News and Announcements** - Useful for communication but can be managed through other channels.
12. **Student Grievance Management** - Handles student complaints, important for student relations.
13. **Student Discipline Management** - Manages disciplinary actions, important for maintaining discipline and order.
14. **University Events** - Supports event management, enhances community engagement.

15. **Alumni Platform** - Engages former students, beneficial for fundraising and networking.
16. **Distance Flexible Education Management** - Expands reach to distance learners.
17. **University Bridging Programs** - Supports preparatory educational programs for university level study.
18. **Learning Space Management** - Helps in managing physical resources, optional enhancement.
19. **TVET Competency Based Education Management** - Specific to vocational training programs.
20. **Full Access and Ownership to Source Code and Database Code** - must have ownership and direct access to the source code and the database code (**Only for developers**).

These lists categorize the functionalities into those that are essential for CORE operations and compliance (fundamental) and those that provide additional, valuable services and OPTIONAL enhancements.

Please note that the functionalities listed in the ensuing pages are not classified under whether it is option or fundamental and must be confirmed with the above lists to determine this attribute.

Contents

1. Web based Interface.....	7
2. System Administration.....	12
3. Organization Setup.....	16
4. Faculty Management.....	20
5. Campaigns & Enquiries Management.....	23
6. Online Application System.....	27
7. Pre-enrolment Course Counselling and Entrance Examinations.....	33
8. Admission Management.....	36
9. Online Enrolment & Course Registration.....	39
10. Student Portal.....	45
11. Fees Management.....	50
12. Scholarship / Sponsorship Management.....	55
13. Program & Course Management.....	61
14. Time Table Management.....	67
15. Student Attendance Management.....	70
16. Disciplinary Management.....	75
17. Examinations & Mark Sheet Management.....	80
18. Assignment Management.....	86
19. Online Learning Management System.....	91
20. Completion Management (Graduation & Certificates).....	95
21. Academic Calendar Management.....	99
22. Library Reservation and Borrowed Books Module.....	103
23. Student Bookshop.....	107
24. Student Printing.....	111
25. Learning Space Booking.....	114
26. IT Lab Computer Booking.....	117
27. Student Association & Affiliated Student Groups & Clubs.....	121
28. Elections & Surveys.....	124
29. Certificates, Awards, & Document Management.....	127
30. Student Appointments with Staff System.....	131
31. Internal Staff Directory.....	133
32. Hostel Management.....	136
33. Student Dinning Management.....	139
34. Student Administration Services.....	142
35. Student Welfare & Counselling.....	145

36.	International Students	148
37.	Thesis Management & Publication	151
38.	University News and Announcements	154
39.	Student Grievance Management	157
40.	Student Discipline Management	159
41.	University Events	162
42.	Alumni Platform	165
43.	Messaging & Notifications	168
44.	Analytics and Reports	171
45.	Staff Portals	174
46.	Academic Staff Management	177
47.	Distance Flexible Education Management	180
48.	University Bridging Programs	184
49.	Learning Space Management	187
50.	TVET Competency Based Education Management	189
51.	Student Academic Records	192
52.	On-going Support & on-site full user training	195
53.	Full Access and Ownership to Source Code and Database Code (Only for Developers)	198

1. Web based Interface

1.1 *Cross Browser Compatibility:*

1.1.1 Requirement:

- Compatibility testing shall be conducted regularly to ensure consistent performance across different browsers and operating systems.

1.1.2 Specification:

- The system shall be thoroughly tested to ensure compatibility with a wide range of web browsers and versions. Compatibility testing shall cover major browsers including, but not limited to, Chrome, Firefox, Safari, and Edge across different operating systems. Any browser-specific issues or inconsistencies shall be addressed.

1.1.3 Verification:

- Implement automated browser compatibility tests as part of the continuous integration/continuous deployment (CI/CD) pipeline.
- Manual testing should also be conducted during the pre-release phase to capture any discrepancies that automated tests may not detect.

1.2 *Responsive Design:*

1.2.1 Requirement:

- The interface must support responsive design principles to ensure optimal viewing and interaction experiences across various devices including desktops, laptops, tablets, and smartphones.

1.2.2 Specification:

- The interface shall dynamically adjust layout and content presentation based on the user's

device screen size and orientation. Design elements such as font sizes, spacing, and button sizes shall adapt to provide a seamless user experience across devices. Content shall be optimized for readability and interaction on smaller screens.

1.2.3 Verification:

- Conduct device compatibility testing using emulators and physical devices to verify that the interface adjusts appropriately for different screen sizes and orientations. User acceptance testing should confirm ease of navigation and readability across all target devices.

1.3 *Accessibility:*

1.3.1 Requirement:

- UI/UX features must adhere to WCAG guidelines including colour contrast, font size, keyboard navigation, and screen reader compatibility.

1.3.2 Specification:

- Interface elements must comply with the latest Web Content Accessibility Guidelines (WCAG). This includes providing sufficient colour contrast, resizable text without loss of content or functionality, keyboard navigable interfaces, and proper ARIA (Accessible Rich Internet Applications) labels for assistive technologies.

1.3.3 Verification:

- Accessibility testing shall be performed using both automated tools and manual testing by accessibility experts. Regular audits by third-party accessibility consultants are recommended to ensure compliance with WCAG guidelines. User testing with participants who rely on assistive technologies

should also be conducted to gather real-world usability feedback.

1.4 Intuitive and User-Friendly Interface:

1.4.1 Requirement:

- The web-based interface should be intuitive and user-friendly to facilitate ease of navigation and interaction.

1.4.2 Specification:

- The interface shall include simple navigation menus, minimalistic design for reducing user cognitive load, and clear action buttons. Tooltips and modal dialogs will be used to provide contextual help and feedback. The design shall follow established UX best practices such as Fitts's Law and the Gestalt principles of visual perception.

1.4.3 Verification:

- Usability testing sessions will be conducted with real users to identify navigation issues and interaction difficulties. The system should also undergo heuristic evaluations by UX experts to ensure compliance with usability standards.

1.5 Consistent Layout and Design:

1.5.1 Requirement:

- The system must have a consistent layout and design across different pages and modules to ensure a cohesive user experience.

1.5.2 Specification:

- A standard UI template shall be used across the application, which includes a consistent colour scheme, typography, and layout structures. Reusable UI components and CSS style sheets will be enforced to maintain consistency.

1.5.3 Verification: Review sessions with the design team will be conducted periodically to ensure consistency. Automated UI tests will check for deviations from defined UI templates and styles.

1.6 *Clear Labelling and Visual Cues*

1.6.1 Requirement:

- The interface should have clear labelling and visual cues to guide users through various functionalities and tasks.

1.6.2 Specification:

- All interactive elements must be clearly labelled with concise, descriptive text. Visual cues such as icons, colour coding, and animations will be used to indicate interactivity and guide users in completing tasks. Essential information will be highlighted using visual emphasis techniques such as bolding or colour changes.

1.6.3 Verification:

- A/B testing will be implemented to determine the effectiveness of labels and visual cues. Feedback from these sessions will be used to refine the UI elements.

1.7 *Customizable Themes and Templates:*

1.7.1 Requirement:

- The interface should support customizable themes and templates to accommodate diverse user preferences.

1.7.2 Specification:

- The system shall allow users to select from a set of predefined themes that alter the visual appearance of the interface without affecting the layout or functionality. Themes will include options for different colour schemes, font

sizes, and contrast settings that support accessibility.

1.7.3 Verification:

- Theme customization will be tested to ensure it can be applied uniformly across all interface components without breaking the layout or functionality. User feedback will be collected to assess satisfaction with the available customization options.

2. System Administration

2.1 *User Authentication and Authorization:*

2.1.1 Requirement:

- The system should include robust user authentication mechanisms to verify the identity of users accessing the system.

2.1.2 Specification:

- Support for various authentication methods, including username/password, multi-factor authentication, and single sign-on (SSO) integration with external identity providers.
- Implementation of secure password policies, such as minimum length, complexity requirements, and expiration periods.
- Integration with LDAP (Lightweight Directory Access Protocol) or Active Directory for centralized user management and authentication.

2.1.3 Verification:

- Verify user authentication functionality through thorough testing, including validation of login credentials, authentication workflows, and error handling mechanisms.

2.2 *Role-Based Access Control (RBAC):*

2.2.1 Requirement:

- The system should enforce role-based access control to restrict user access to specific features and functionalities based on their roles and permissions.

2.2.2 Specification:

- Definition and management of user roles, each with a set of predefined permissions and access rights.
- Granular control over access permissions at the module, functionality, and data level to ensure least privilege access.

- Support for hierarchical roles and role inheritance to streamline permission management.

2.2.3 Verification:

- Conduct comprehensive testing to validate the effectiveness of RBAC implementation, including role assignment, permission enforcement, and access control scenarios.

2.3 *User Account Management:*

2.3.1 Requirement:

- The system should provide administrative tools for managing user accounts, including creation, modification, and deletion of user accounts.

2.3.2 Specification:

- User-friendly interface for administrators to manage user accounts, including search, filter, and sorting capabilities.
- Support for bulk user import/export functionality to facilitate user account management at scale.
- Provision of account activation/deactivation mechanisms to control user access based on status.

2.3.3 Verification:

- Validate user account management functionality through testing of account creation, modification, and deletion workflows, ensuring accuracy and reliability.

2.4 *Permission Management:*

2.4.1 Requirement:

- The system should enable administrators to assign and modify permissions for users and roles to control access to system features and data.

2.4.2 Specification:

- Granular control over permissions at the module, functionality, and data level, allowing administrators to tailor access rights to specific user roles.
- Role-based assignment of permissions to streamline permission management and ensure consistency.
- Audit trail functionality to track permission changes and maintain a history of permission assignments.

2.4.3 Verification:

- Conduct testing to verify the accuracy and effectiveness of permission assignment and enforcement mechanisms, including permission inheritance, role-based access control, and permission conflicts resolution.

2.5 *Administrative Configuration Tools*

2.5.1 Requirement:

- Administrative tools must be provided for system administrators to configure global settings and manage interface elements.

2.5.2 Specification:

- Admin tools shall include capabilities to modify system settings such as themes, default layouts, and accessibility options. Administrators will have the ability to add or modify content, configure feature toggles, and set permissions for different user roles.

2.5.3 Verification:

- Functional testing will be conducted to ensure that administrative tools are effective and secure. Tests will include scenario-based testing for configuring settings and managing interface elements, as well as security audits to prevent unauthorized access.

2.6 Audit Trail Functionality:

2.6.1 Requirement:

- The system should include audit trail functionality to record and track administrative actions, system changes, and user activities for accountability and compliance purposes.

2.6.2 Specification:

- Logging of critical system events, including user logins, access attempts, data modifications, and configuration changes.
- Capturing relevant metadata for each logged event, such as timestamp, user ID, IP address, and action performed.
- Storage of audit logs in a secure, tamper-evident format with restricted access to authorized personnel.

2.6.3 Verification:

- Validate audit trail functionality through testing of event logging, retrieval, and analysis capabilities, ensuring completeness, accuracy, and integrity of audit logs.

3. Organization Setup

3.1 Customizable Organizational Hierarchy:

3.1.1 Requirement:

- The system should support a customizable organizational hierarchy to reflect SINU's structure, allowing for flexibility and scalability.

3.1.2 Specification:

- Capability to define and configure organizational units, academic faculties, schools, departments, institutes, centers, and other relevant entities.
- Support for hierarchical relationships between organizational units, allowing for parent-child relationships and multi-level structures.
- Ability to customize the naming conventions, labels, and attributes of each organizational unit to align with SINU's terminology and naming conventions.

3.1.3 Verification:

- Validate the customization capabilities through testing, ensuring that administrators can create, modify, and delete organizational units as needed and that changes are reflected accurately in the system.

3.2 Campus Management:

3.2.1 Requirement:

- The system should provide functionality for defining and managing campuses, reflecting SINU's physical locations and campuses.

3.2.2 Specification:

- Capability to define and configure multiple campuses, each with its own unique attributes, such as location, facilities, and contact information.

- Support for campus-specific settings, such as academic calendars, course offerings, and administrative workflows.
- Integration with location-based services for mapping and navigation purposes.

3.2.3 Verification:

- Test the campus management functionality to ensure administrators can create, edit, and delete campuses, and that campus-specific settings are applied accurately throughout the system.

3.3 *Organizational Unit Management:*

3.3.1 Requirement:

- The system should allow administrators to define and manage various organizational units, including faculties, schools, departments, institutes, and centers.

3.3.2 Specification:

- Capability to create and configure organizational units with customizable attributes, such as name, description, and leadership roles.
- Support for hierarchical relationships between organizational units, allowing for nested structures and parent-child relationships.
- Ability to assign staff members, such as deans, heads of schools, and department chairs, to specific organizational units.

3.3.3 Verification:

- Validate organizational unit management functionality through testing, ensuring administrators can create, edit, and delete organizational units, assign leadership roles, and manage hierarchical relationships effectively.

3.4 *Staff Setup and Management:*

3.4.1 Requirement:

- The system should allow administrators to define and manage staff members, including academic and administrative staff, within each organizational unit.

3.4.2 Specification:

- Capability to define staff roles and positions, such as dean of faculty, head of school, head of department, and lecturers, under each organizational unit.
- Support for assigning staff members to specific roles within the organizational hierarchy, including leadership for workflow approval and teaching roles.
- Ability to manage staff profiles, including contact information, qualifications, and responsibilities.

3.4.3 Verification:

- Test staff setup and management functionality to ensure administrators can create, edit, and delete staff profiles, assign roles and permissions accurately, and manage staff assignments within the organizational hierarchy.

3.5 *Security Access Management:*

3.5.1 Requirement:

- The system should provide tools for system administrators to use the organizational hierarchy to map and manage security access to modules, functionalities, and permissions.

3.5.2 Specification:

- Capability to map user roles and permissions to the organizational hierarchy, allowing for role-based access control (RBAC) aligned with SINU's organizational structure.

- Support for defining access permissions at the organizational unit level, allowing administrators to grant or restrict access to specific modules, functionalities, and data based on organizational roles and responsibilities.
- Integration with user authentication and authorization mechanisms to enforce security access controls throughout the system.

3.5.3 Verification:

- Validate security access management functionality through testing, ensuring that administrators can configure role-based access controls aligned with the organizational hierarchy, assign permissions accurately, and enforce access restrictions consistently.

4. Faculty Management

4.1 Management of Academic Structures

4.1.1 Requirement:

- The system should allow administrators to define and manage Faculties, Schools, and Departments within the organizational academic structure.

4.1.2 Specification:

- The system shall provide an interface for administrators to create, edit, and delete academic units such as Faculties, Schools, and Departments. Each unit should be able to have a unique identifier, name, description, and associated administrative staff.

4.1.3 Verification:

- Functional testing will be conducted to ensure that academic units can be managed effectively. This will include tests for creating, editing, and deleting units, and verifying that changes are correctly reflected in the system.

4.2 Course Linking

4.2.1 Requirement:

- The system must have the capability to link all offered courses to their respective Faculty, School, and Department.

4.2.2 Specification:

- Courses shall be linked to academic units through a relational database structure where each course record includes references to its associated Faculty, School, and Department.
- Administrators must be able to assign and reassign courses to different units through the administrative interface.

4.2.3 Verification:

- Database integrity tests will check for correct relationships and constraints. User

acceptance testing will ensure that course linking is intuitive and meets administrative needs.

4.3 *Lecturer and Tutor Assignment*

4.3.1 Requirement:

- The system should have the capability to link all lecturers and tutors to a Faculty, School, and Department.

4.3.2 Specification:

- Lecturers and tutors shall be assigned to academic units within the system, with each staff member's profile including fields for associated Faculty, School, and Department. This will facilitate resource management and allocation.

4.3.3 Verification:

- Verification will involve testing the assignment and reassignment processes, ensuring that changes are reflected accurately and promptly in the system.

4.4 *Course Assignment to Departments*

4.4.1 Requirement:

- Support for creating and assigning courses to specific academic departments within the organizational hierarchy.

4.4.2 Specification:

- Administrators should be able to assign courses to departments using a drop-down menu or a selection interface within the course management system. Each course will have a primary department association, with the option to cross-list in other departments.

4.4.3 Verification:

- Testing will include scenarios where courses are assigned and reassigned to different

departments. Cross-listing functionality will also be tested for accuracy and ease of use.

4.5 *Creation and Management of Academic and Other Units*

4.5.1 Requirement:

- Ability to create and manage academic units such as Departments, Schools, Faculties, and other units like TVET, Centers, Institutes, Bridging Programs, and Remote Centers.

4.5.2 Specification:

- The system shall offer a flexible structure to accommodate a variety of academic and instructional units beyond traditional academic departments. This includes specialized units such as TVET and remote centers, each with customizable attributes and settings.

4.5.3 Verification:

- Systematic validation will check the creation, modification, and deletion capabilities for all types of academic units. The flexibility and extensibility of the system to accommodate various types of units will also be tested.

5. Campaigns & Enquiries Management

5.1 *Enrolment Campaign Management:*

5.1.1 Requirement:

- The system should provide functionality to create, manage, and track enrolment campaigns and marketing efforts targeted at both domestic and international students.

5.1.2 Specification:

- Capability to create and configure enrolment campaigns with customizable parameters such as campaign name, duration, target audience, and objectives.
- Support for defining campaign strategies, including marketing channels, messaging, and promotional materials.
- Integration with communication and outreach tools for executing campaign activities, such as email marketing, social media promotion, and advertising campaigns.
- Ability to track campaign performance metrics, including enrolment leads, conversions, and return on investment (ROI).

5.1.3 Verification:

- Validate enrolment campaign management functionality through testing, ensuring administrators can create and manage campaigns effectively, track campaign performance, and analyze results to optimize future campaigns.

5.2 *Integration with Social Media Platforms:*

5.2.1 Requirement:

- The system should integrate with social media platforms to facilitate targeted marketing and promotion of scholarships, course programs, and enrolment campaigns.

5.2.2 Specification:

- Integration with popular social media platforms, such as Facebook, Twitter, Instagram, LinkedIn, and YouTube, to publish and promote campaign content.
- Support for sharing campaign updates, scholarship announcements, program highlights, and other relevant information across social media channels.
- Capability to track engagement metrics, such as likes, shares, comments, and click-through rates, to measure the effectiveness of social media marketing efforts.

5.2.3 Verification:

- Validate social media integration functionality through testing, ensuring seamless connectivity with social media platforms and accurate tracking of campaign engagement metrics.

5.3 *Student Enquiries Management:*

5.3.1 Requirement:

- The system should have the ability to capture, manage, and track student enquiries, including communication history, to facilitate effective engagement and follow-up.

5.3.2 Specification:

- Capability to capture student enquiries through various channels, such as social media and website forms, email inquiries, phone calls, and in-person visits.
- Support for logging and recording student enquiries in a centralized database, including relevant details such as student information, enquiry type, date/time, and source.
- Integration with communication tools for responding to student enquiries, tracking correspondence, and maintaining communication history.

- Ability to categorize and prioritize enquiries based on urgency, importance, and subject matter to ensure timely and effective responses.

5.3.3 Verification:

- Test student enquiries management functionality to ensure enquiries are captured accurately, stored securely, and managed efficiently, with appropriate communication history tracking and follow-up processes in place.

5.4 *Communication History Tracking:*

5.4.1 Requirement:

- The system should maintain a comprehensive communication history for each student enquiry, documenting all interactions and correspondence for reference and follow-up purposes.

5.4.2 Specification:

- Logging and tracking of all communication activities related to student enquiries, including inbound and outbound messages, emails, phone calls, meetings, and notes.
- Capture and storage of communication details such as message content, sender/receiver information, timestamps, and status updates.
- Ability to view and access communication history for each student enquiry, providing a chronological record of all interactions and correspondence.
- Integration with communication tools and platforms for seamless recording and tracking of communication activities within the system.

5.4.3 Verification:

- Validate communication history tracking functionality through testing, ensuring that all communication activities related to student

enquiries are accurately captured, logged, and accessible for reference and follow-up purposes.

5.5 Reporting and Analytics:

5.5.1 Requirement:

- The system should provide reporting and analytics capabilities to monitor and analyse enrolment campaign performance, student enquiry metrics, and communication effectiveness.

5.5.2 Specification:

- Generation of standard and custom reports to track key metrics and performance indicators related to enrolment campaigns, such as lead generation, conversion rates, demographics, and campaign ROI.
- Visualization of data through charts, graphs, and dashboards to provide insights into enrolment trends, campaign effectiveness, and student engagement metrics.
- Analysis of communication history data to evaluate response times, resolution rates, communication channels effectiveness, and overall customer satisfaction.
- Exporting and sharing of reports and analytics findings with stakeholders, including administrators, marketing teams, and academic advisors.

5.5.3 Verification:

- Validate reporting and analytics functionality through testing, ensuring that administrators can generate, customize, and interpret reports and analytics findings to inform decision-making and optimize enrolment campaigns and student enquiry management processes.

6. Online Application System

6.1 *User-friendly Application Form:*

6.1.1 Requirement:

- The system should provide a user-friendly application form with customizable fields to streamline the application process for prospective students.

6.1.2 Specification:

- Design and implementation of an intuitive and easy-to-navigate application form with clear instructions and guidance for applicants.
- Customization options to tailor the application form fields based on the specific requirements of domestic and international applicants.
- Customization options to tailor the application form fields according to the specific requirements of different course programs and admission criteria.
- Validation rules to ensure data accuracy and completeness, including field-level validation and error messaging for invalid inputs.

6.1.3 Verification:

- Test the application form functionality to ensure ease of use, data validation, and adherence to usability standards, including testing on different devices and screen sizes.

6.2 *Selection of Applicant Type:*

6.2.1 Requirement:

- The system should be configured to allow applicants to select their applicant type (domestic or international) and customize their application forms accordingly.

6.2.2 Specification:

- Incorporation of an applicant type selection mechanism into the application form interface.

- Dynamic adjustment of application form fields and requirements based on the selected applicant type.
- Validation checks to ensure consistency and accuracy of applicant type selection throughout the application process.

6.2.3 Verification:

- Test applicant type selection functionality to ensure that application forms are customized correctly based on the selected applicant type and that validation checks are enforced as expected.

6.3 *Account Setup and Communication:*

6.3.1 Requirement:

- The system should allow potential applicants to set up an account using their private email as the primary means of communication, and provide email updates on changes in application status.

6.3.2 Specification:

- Capability for applicants to create a user account using their private email address, with options for password setup and account verification.
- Integration with email notification services to send automatic updates to applicants' primary email addresses on changes in application status, including acknowledgments, updates, and notifications.
- Support for opt-in/opt-out preferences for receiving email marketing communications, with options for managing subscription preferences.

6.3.3 Verification:

- Validate account setup and communication functionality through testing, ensuring that users can register accounts, verify email

addresses, and receive timely updates on application status changes.

6.4 Course Program Search and Selection:

6.4.1 Requirement:

- The system should include course program search and selection functions for all course programs offered at the university, its institutes, and centers.

6.4.2 Specification:

- Implementation of a search feature to allow applicants to search for course programs based on keywords, program names, disciplines, or other criteria.
- Integration with a comprehensive database of course programs, including descriptions, prerequisites, and availability status.
- Capability for applicants to select and add course programs to their application submissions, with options for multiple selections and preferences ranking.

6.4.3 Verification:

- Test course program search and selection functionality to ensure accurate search results, seamless integration with the course catalogue, and intuitive selection processes for applicants.

6.5 Multiple Course Preferences:

6.5.1 Requirement:

- The system should allow applicants to select course preferences for up to three choices, enabling flexibility in course selection and preference ranking.

6.5.2 Specification:

- Provision of options for applicants to indicate up to three preferred course programs or

study options within their application submissions.

- Support for ranking or prioritizing course preferences to indicate the applicant's order of preference if multiple options are available.
- Validation rules to ensure that applicants provide a minimum number of course preferences and avoid duplicate or conflicting selections.

6.5.3 Verification:

- Test multiple course preference functionality to ensure that applicants can indicate their preferred study options effectively and that preferences are captured accurately in the application submissions.

6.6 *Document Uploads and Attachments:*

6.6.1 Requirement:

- The system should support document uploads and attachments as part of the application submission process.

6.6.2 Specification:

- Capability for applicants to upload and attach supporting documents, such as transcripts, certificates, resumes, and letters of recommendation, to their application forms.
- Support for various file formats, including PDF, Word documents, images, and scanned copies, with size restrictions and validation checks.
- Secure storage and encryption of uploaded documents to protect applicant data and ensure compliance with data privacy regulations.

6.6.3 Verification:

- Test document upload and attachment functionality to ensure that applicants can upload, view, and manage attachments

effectively, and that uploaded documents are stored securely and accessible to administrative staff as needed.

6.7 *Hostel Accommodation and Dining Application:*

6.7.1 Requirement:

- The system should allow students the option to apply for on-campus hostel accommodation and dining as part of the application process.

6.7.2 Specification:

- Integration of hostel accommodation and dining application forms into the overall application process, allowing applicants to indicate their preferences and requirements.
- Inclusion of relevant fields and options for hostel room preferences, meal plans, dietary restrictions, and special requests.
- Seamless integration with the main application form and application status tracking system for administrative purposes.

6.7.3 Verification:

- Validate hostel accommodation and dining application functionality through testing, ensuring that applicants can submit their preferences and requirements accurately and that administrative staff can manage and track applications effectively.

6.8 *Application Status Tracking:*

6.8.1 Requirement:

- The system should provide application status tracking functionality for both applicants and administrative staff to monitor the progress of applications.

6.8.2 Specification:

- Integration with a centralized application status tracking system to provide real-time

updates on application progress and status changes.

- Capability for applicants to log in to their accounts and view the current status of their applications, including pending, under review, accepted, or rejected statuses.
- Administrative dashboard and reporting tools for staff to monitor application statuses, track application metrics, and generate status reports as needed.

6.8.3 Verification:

- Validate application status tracking functionality through testing, ensuring that applicants and staff can access up-to-date information on application statuses and that status updates are reflected accurately in the system.

7. Pre-enrolment Course Counselling and Entrance Examinations

7.1 *Booking System for Appointments:*

7.1.1 Requirement:

- The system should include a booking system to schedule appointments between students and faculty staff for pre-enrolment course counselling.

7.1.2 Specification:

- User-friendly interface for students to view available appointment slots, select preferred times, and book counselling sessions with faculty staff.
- Integration with faculty staff calendars to display real-time availability and prevent double bookings.
- Capability for faculty staff to manage their availability, set appointment hours, and block off unavailable time slots.
- Confirmation emails or notifications sent to both students and faculty staff upon successful booking, including appointment details and instructions.

7.1.3 Verification:

- Test booking system functionality to ensure that students can book appointments seamlessly, availability is accurately reflected, and notifications are sent and received as expected.

7.2 *Booking System for Entrance Examinations:*

7.2.1 Requirement:

- The system should provide a booking system for scheduling entrance examinations between applicants and respective faculty examiners, generating auto-generated exam candidate numbers.

7.2.2 Specification:

- Integration with the application system to identify applicants requiring entrance examinations and generate candidate numbers automatically upon scheduling.
- User-friendly interface for applicants to view available examination slots, select preferred dates, and book examination appointments.
- Integration with faculty examiners' calendars to display availability and prevent scheduling conflicts.
- Automated generation of unique candidate numbers for each examination booking for tracking and identification purposes.

7.2.3 Verification:

- Validate examination booking system functionality through testing, ensuring seamless scheduling, accurate candidate number generation, and integration with faculty calendars.

7.3 *Integration with Faculty Calendars:*

7.3.1 Requirement:

- The system should integrate with faculty calendars for examiner availability, venue booking, and availability tracking.

7.3.2 Specification:

- Integration with faculty staff calendars to display real-time availability for appointments and examinations.
- Capability to synchronize booking system data with faculty calendars to ensure availability updates are reflected instantly.
- Option for faculty staff to mark unavailable dates or times in their calendars, blocking off slots for appointments or examinations.

7.3.3 Verification:

- Test integration with faculty calendars to ensure synchronization of availability data, accurate display of available slots, and prevention of conflicts with existing appointments.

7.4 Automated Reminders and Notifications:

7.4.1 Requirement:

- The system should send automated reminders and notifications for appointments and entrance examination dates and venues.

7.4.2 Specification:

- Automated generation and sending of reminders and notifications to students, faculty staff, and examiners for scheduled appointments and examinations.
- Notifications sent via email, SMS, or in-app messages, depending on user preferences and contact information provided.
- Reminders sent in advance of scheduled appointments and examinations, with configurable lead times and frequency options.
- Include details such as date, time, location, and any additional instructions or requirements for the appointment or examination.

7.4.3 Verification:

- Validate reminder and notification functionality through testing, ensuring that reminders are sent promptly, accurately, and via preferred communication channels, and that recipients receive and acknowledge notifications as expected.

8. Admission Management

8.1 Streamlined Admission Process:

8.1.1 Requirement:

- The system should facilitate a streamlined admission process from application submission to offer generation.

8.1.2 Specification:

- Automated handling of application submissions, including data capture, validation, and storage.
- Integration with the online application system to seamlessly transfer application data to the admission management module.
- Workflow management tools to track the progress of each application through various stages of the admission process.
- Dashboard for administrators to monitor the overall status of admissions, identify bottlenecks, and track key metrics.

8.1.3 Verification:

- Validate the admission process workflow through testing, ensuring that applications progress smoothly through each stage and that data integrity is maintained throughout.

8.2 Offer Letter Generation:

8.2.1 Requirement:

- The system should have the capability to issue different types of offer letters, including confirmed offer letters, conditional offer letters, invitations for course counselling interviews, or invitations to sit for entrance examinations.

8.2.2 Specification:

- Template-based offer letter generation functionality to produce customized letters based on applicant status and criteria.

- Support for dynamic insertion of applicant-specific information, such as name, program details, admission conditions, and deadlines.
- Integration with applicant data to automatically populate offer letters with relevant information.
- Workflow automation for offer letter approval, generation, and distribution.

8.2.3 Verification:

- Test offer letter generation functionality to ensure that letters are generated accurately, contain all necessary information, and are delivered to applicants in a timely manner.

8.3 *Application Review and Interview Scheduling:*

8.3.1 Requirement:

- The system should support application review, interview scheduling, and decision-making workflows.

8.3.2 Specification:

- Capability to assign applications to reviewers or admission officers for evaluation and decision-making.
- Integration with faculty calendars and scheduling tools to facilitate interview scheduling and coordination.
- Workflow management features to track the progress of each application through review, interview scheduling, and final decision stages.
- Communication tools for notifying applicants of interview schedules, outcomes, and next steps in the admission process.

8.3.3 Verification:

- Validate application review and interview scheduling functionality through testing, ensuring that applications are reviewed efficiently, interviews are scheduled without

conflicts, and applicants receive timely updates on their application status.

9. Online Enrolment & Course Registration

9.1 *Enrolment Options:*

9.1.1 Requirement:

- The system should allow both online enrolment and in-person enrolment options for students.

9.1.2 Specification:

- Implementation of online enrolment functionality, allowing continuing students to enrol in courses through the web interface.
- Provision of in-person enrolment facilities on campus for students who prefer or require assistance with the enrolment process.
- Integration of both enrolment methods into the overall enrolment workflow to ensure consistency and accessibility for all students.

9.1.3 Verification:

- Validate enrolment options functionality through testing, ensuring that both online and in-person enrolment processes are available and functional.

9.2 *Fee Payment Options:*

9.2.1 Requirement:

- The system should allow students to select fee payment options during enrolment, including scholarship, full payment, or instalment fee payment options.

9.2.2 Specification:

- Integration of fee payment selection functionality into the enrolment process, allowing students to choose their preferred payment option.
- Support for different fee payment methods and arrangements, such as online payments, bank transfers, or payment plans.

- Validation checks to ensure that selected payment options are compatible with students' eligibility and financial status.

9.2.3 Verification:

- Test fee payment options functionality to ensure that students can select and confirm their preferred payment option during the enrolment process, with validation checks and error handling in place.

9.3 *Course Selection and Enrolment:*

9.3.1 Requirement:

- The system should provide an intuitive interface for continuing students to select and enrol in courses online, with fee payment receipt upload and confirmation feedback workflow as prerequisites for access.

9.3.2 Specification:

- User-friendly interface for continuing students to browse, search, and select courses for enrolment.
- Integration of fee payment receipt upload functionality as a prerequisite for accessing the course selection and enrolment interface.
- Workflow for verifying fee payment receipts and providing confirmation feedback to students before granting access to the online course enrolment functionality for continuing students.

9.3.3 Verification:

- Validate course selection and enrolment functionality through usability testing, ensuring ease of use, clarity of instructions, and smooth workflow transitions.

9.4 *Real-Time Course Availability and Prerequisite Checks:*

9.4.1 Requirement:

- The system should provide real-time course availability and prerequisite checks to assist students in selecting courses.

9.4.2 Specification:

- Integration of real-time course availability status into the course selection interface, indicating which courses are open for enrolment.
- Implementation of prerequisite checks to verify whether students meet the eligibility criteria for enrolling in specific courses.
- Dynamic updates to course availability and prerequisite status based on enrolment activity and changes to student records.

9.4.3 Verification:

- Test course availability and prerequisite checks functionality to ensure that students receive accurate and up-to-date information when selecting courses and that eligibility criteria are enforced correctly.

9.5 *Automated Conflict Resolution:*

9.5.1 Requirement:

- The system should have automated conflict resolution mechanisms for scheduling conflicts during course enrolment.

9.5.2 Specification:

- Implementation of conflict detection algorithms to identify scheduling conflicts, such as overlapping class times or exam schedules.
- Automatic resolution of conflicts through prioritization rules, alternative scheduling options, or manual intervention by administrative staff.
- Notification and feedback mechanisms to inform students of conflicts and resolution outcomes during the enrolment process.

9.5.3 Verification:

- Validate conflict resolution functionality through testing, ensuring that conflicts are detected and resolved accurately and that students are notified of any conflicts or changes to their enrolment status.

9.6 *Auto-Generated Student ID Numbers and Passwords:*

9.6.1 Requirement:

- The system should automatically generate student ID numbers and network access passwords for newly enrolled students.

9.6.2 Specification:

- Automatic generation of unique student ID numbers for each enrolled student, following predefined format and numbering conventions.
- Generation of network access passwords or login credentials for students to access online resources and systems.
- Secure storage and distribution of student ID numbers and passwords, with encryption and access controls to protect sensitive information.

9.6.3 Verification:

- Test student ID and password generation functionality to ensure that unique identifiers are generated accurately and securely for each enrolled student.

9.7 *Self-Service ID Photo Upload:*

9.7.1 Requirement:

- The system should allow students to upload their ID photos using a self-service interface, integrated with a camera.

9.7.2 Specification:

- Integration of a camera interface into the enrolment system, allowing students to capture and upload ID photos directly from their devices.
- User-friendly upload interface with instructions and guidance for capturing and submitting ID photos.
- Validation checks to ensure that uploaded photos meet predefined criteria, such as size, resolution, and format.

9.7.3 Verification:

- Validate ID photo upload functionality through testing, ensuring that students can capture and submit ID photos accurately and efficiently using the integrated camera interface.

9.8 *Enrolment Confirmation Letter Generation:*

9.8.1 Requirement:

- The system should generate enrolment registration confirmation letters for students, including student ID numbers and lists of enrolled courses.

9.8.2 Specification:

- Automated generation of enrolment confirmation letters upon successful completion of the enrolment process, including relevant student information and course details.
- Customizable letter templates with placeholders for student-specific data, such as name, ID number, enrolled courses, and academic program.
- Delivery options for distributing confirmation letters to students, including email, print, or download options.

9.8.3 Verification:

- Test confirmation letter generation functionality to ensure that letters are generated accurately and contain relevant student information and course details, with customizable templates and delivery options.

10. Student Portal

10.1 Personalized Dashboard:

10.1.1 Requirement:

- The system should provide a personalized dashboard for students to view academic records, schedules, and announcements.

10.1.2 Specification:

- Design and implementation of a customizable dashboard interface for each student, displaying relevant information such as current courses, grades, upcoming events, and academic deadlines.
- Integration with student information systems to retrieve and display real-time data, including academic records, schedules, and announcements.
- Customization options for students to personalize their dashboard layout, widgets, and preferences based on their individual needs and preferences.

10.1.3 Verification:

- Validate the personalized dashboard functionality through usability testing, ensuring that students can easily access and customize their dashboard to view relevant information effectively.

10.2 Access to Course Materials and Grades:

10.2.1 Requirement:

- The student portal should provide access to course materials, grades, enrolled courses, and financial information.

10.2.2 Specification:

- Integration with learning management systems (LMS) or course management systems (CMS) to provide seamless access to

course materials, assignments, lecture notes, and other educational resources.

- Display of grades and academic progress for each enrolled course, including assignment grades, exam scores, and overall course grades.
- Access to financial information, such as tuition fees, payment history, and financial aid status, within the student portal interface.

10.2.3 Verification:

- Test course material access and grade viewing functionality to ensure that students can access educational resources and monitor their academic progress accurately through the portal.

10.3 Additional Portal Features:

10.3.1 Requirement:

- The student should have access through their portal to various features and functionalities.

10.3.2 Specification:

- *Library books borrowed and resource reservation:* Integration with library systems to allow students to view borrowed books, reserve resources, and manage library accounts.
- *Student space and computer booking:* Capability for students to book study spaces and computer labs through the portal interface.
- *Course lecturer appointment:* Integration with appointment scheduling systems to facilitate booking appointments with course lecturers or academic advisors.
- *Attendance:* Display of attendance records for each enrolled course, including dates attended and overall attendance percentage.

- *Elections & Surveys*: Access to participate in university elections and surveys, view election results, and submit survey responses.
- *Printing Account*: Access to view printing account balance, print quotas, and transaction history, with options to add funds or manage printing settings.
- *Events*: Display of university events, workshops, seminars, and extracurricular activities, with options to RSVP or register for events.
- *Transcripts*: Access to view and download official transcripts and academic records, including GPA, course credits, and degree progress.
- *Internship and other student position application*: Capability to apply for internships, student positions, or volunteer opportunities through the portal interface.
- *Grievance reporting*: Functionality for students to report grievances, complaints, or issues to university authorities, with options for follow-up and resolution tracking.
- *Anonymous Whistle Blower tool*: Secure mechanism for students to report misconduct, fraud, or unethical behaviour anonymously, with protection of whistle-blower identity.
- *Academic & Personal counselling Appointment*: Integration with counselling services to schedule appointments for academic or personal counselling sessions.
- *Timetable*: Display of class schedules, exam timetables, and other academic calendars, with options to filter and customize views.
- *Academic Calendar*: Access to university academic calendar, including important dates, holidays, registration deadlines, and semester schedules.

- *Personalized Academic Dashboard:* Customized dashboard displaying academic performance metrics, progress towards degree completion, and personalized recommendations for academic success.
- *Messaging App:* In-built messaging app for communication between students, faculty, and staff, including university-wide announcements, reminders, and notifications.

10.3.3 Verification:

- Test each additional portal feature to ensure functionality, accessibility, and usability, with validation checks for data accuracy, security, and integration with relevant systems.

10.4 *In-built Messaging App:*

10.4.1 Requirement:

- The in-built messaging app should be the primary communication tool for interacting with faculty and staff, including receipt of university-wide announcements, fee payment reminders, assignment due dates, test/exam schedules, and notifications from various areas.

10.4.2 Specification:

- Implementation of a messaging app within the student portal interface, allowing students to send and receive messages to/from faculty, staff, and peers.
- Integration with university systems to deliver automated notifications and reminders for important events, deadlines, and announcements.
- Support for push notifications, email alerts, and in-app messaging features to ensure timely communication and engagement.

10.4.3 Verification:

- Test the messaging app functionality to ensure seamless communication between students, faculty, and staff, with validation checks for message delivery, notification accuracy, and user experience.

11. Fees Management

11.1 Management of Fee Structures, Payment Plans, and Invoicing:

11.1.1 Requirement:

- The system should manage fee structures, payment plans, and invoicing processes.

11.1.2 Specification:

- Capability to define and configure fee structures for various programs, courses, and services offered by the university, including tuition fees, registration fees, and other charges.
- Support for creating flexible payment plans, including installment options, deadlines, and payment schedules.
- Automated generation and issuance of invoices for fee payments, including itemized breakdowns of charges and payment instructions.

11.1.3 Verification:

- Test fee structure configuration, payment plan setup, and invoice generation functionality to ensure accuracy, flexibility, and compliance with university policies and regulations.

11.2 Integration with Payment Gateways:

11.2.1 Requirement:

- The system should integrate with payment gateways to facilitate online fee payments.

11.2.2 Specification:

- Integration with reputable payment gateway services to securely process online payments via credit/debit cards, e-wallets, and other electronic payment methods.
- Implementation of secure communication protocols (e.g., HTTPS) and encryption

mechanisms to protect sensitive payment information during transactions.

- Seamless redirection to the payment gateway interface for completing transactions, with real-time validation and confirmation of payment status.

11.2.3 Verification:

- Test payment gateway integration functionality to ensure seamless, secure, and reliable processing of online fee payments, with validation checks for transaction accuracy and error handling.

11.3 *Integration with Online Bank Payments and Mobile App Payments:*

11.2.4 Requirement:

- The system should integrate with online bank payments and mobile app payments for alternative fee payment options.

11.2.5 Specification:

- Integration with banking APIs or payment aggregators to facilitate direct bank transfers or online banking payments from students' bank accounts.
- Support for mobile app payment integrations, allowing students to initiate fee payments through dedicated university mobile applications or third-party payment apps.
- Verification checks to ensure compatibility with popular banking platforms and mobile payment services, including support for various devices and operating systems.

11.2.6 Verification:

- Test online bank payment and mobile app payment integrations to ensure compatibility, reliability, and ease of use, with validation checks for transaction processing and payment confirmation.

11.4 Integration with Messaging App for Notifications:

11.2.7 Requirement:

- The system should integrate with the messaging app to send notifications to students and sponsors for due dates and overdue payments.

11.2.8 Specification:

- Integration with the messaging app to deliver automated notifications, reminders, and alerts for fee payment deadlines, overdue payments, and outstanding balances.
- Customization options for configuring notification templates, delivery schedules, and recipient groups based on fee status, program enrolment, or sponsorship arrangements.
- Real-time synchronization between the fees management system and the messaging app to ensure timely and accurate delivery of notifications.

11.2.9 Verification:

- Test messaging app integration for fee notifications to ensure reliable delivery, message formatting, and synchronization with fee management data, with validation checks for message accuracy and delivery status.

11.5 Integration with University Finance System:

11.2.10 Requirement:

- The system should integrate with the University Finance System (e.g., Attache') for seamless financial data exchange and reconciliation.

11.2.11 Specification:

- Integration with the university's finance system to synchronize fee management data, including payment transactions, invoices, receipts, and financial reports.
- Mapping of data fields and records between the fees management system and the finance system to ensure consistency and accuracy of financial information.
- Automated data transfer and reconciliation processes to streamline accounting operations and financial reporting.

11.2.12 Verification:

- Test integration with the University Finance System to ensure smooth data exchange, data integrity, and financial accuracy between the fees management system and the finance system.

11.6 *Generation of Fee Statements and Receipts:*

11.2.13 Requirement:

- The system should generate fee statements and receipts for students' reference and record-keeping purposes.

11.2.14 Specification:

- Automated generation of fee statements detailing outstanding balances, payment history, and transaction summaries for each student.
- Provision of printable or downloadable fee receipts with unique transaction IDs, payment timestamps, and payment confirmation details.
- Customization options for fee statement formats, branding elements, and receipt layouts to align with university branding and compliance requirements.

11.2.15 Verification:

- Test fee statement and receipt generation functionality to ensure accuracy, completeness, and compliance with financial regulations and audit standards.

11.7 Generation of Analytical Reports for Fees Paid and Fees Owning:

11.2.16 Requirement:

- The system should generate analytical reports for fees paid and fees owing.

11.2.17 Specification:

- Predefined and customizable report templates for analysing fee payment trends, revenue streams, and outstanding balances across student accounts.
- Support for generating summary reports, detailed transaction logs, aging analyses, and other financial metrics to monitor fee collection and arrears.
- Integration with data visualization tools or dashboards for presenting fee management insights in a visually appealing and actionable format.

11.2.18 Verification:

- Validate analytical report generation functionality through testing, ensuring that reports are accurate, comprehensive, and accessible for financial analysis and decision-making purposes.

12. Scholarship / Sponsorship Management

12.1 Tracking and Management of Applications:

12.1.1 Requirement:

- The system should track and manage scholarship and sponsorship applications efficiently.

12.1.2 Specification:

- Implementation of a centralized database to store scholarship and sponsorship application details, including applicant information, program preferences, and supporting documents.
- User-friendly interface for students to submit scholarship applications online, with options for uploading required documents and monitoring application status.
- Administrative dashboard for staff to review, evaluate, and manage scholarship applications, with features for sorting, filtering, and prioritizing applications.

12.1.3 Verification:

- Test application tracking and management functionality to ensure that applications are processed accurately, securely, and transparently, with validation checks for data accuracy and application status updates.

12.2 Management of Domestic and International Scholarships:

12.2.1 Requirement:

- The system should manage both domestic and international scholarships.

12.2.2 Specification:

- Support for defining and administering various types of scholarships, including those available to domestic and international students.

- Customizable eligibility criteria and application requirements for each scholarship category, based on residency status, academic merit, financial need, or other qualifying factors.
- Integration with student information systems to identify eligible candidates and streamline scholarship award processes based on residency status and other criteria.

12.2.3 Verification:

- Validate scholarship management functionality for both domestic and international scholarships, ensuring that eligibility criteria are enforced correctly and scholarships are awarded fairly and transparently.

12.3 *Administration of Full and Partial Scholarships:*

12.3.1 Requirement:

- The system should administer both full and partial scholarships.

12.3.2 Specification:

- Capability to define and manage scholarship awards as either full scholarships covering all tuition and board and lodging fees or partial scholarships providing partial financial assistance.
- Calculation and allocation of scholarship amounts based on predefined funding limits, tuition costs, and other financial considerations.
- Transparent communication of scholarship terms, conditions, and funding amounts to recipients and relevant stakeholders.

12.3.3 Verification:

- Test scholarship administration functionality to ensure accurate calculation and allocation of scholarship amounts, with validation

checks for compliance with funding constraints and transparency in communication.

12.4 Administration of Student Stipends:

12.4.1 Requirement:

- The system should administer student stipends as part of scholarship packages.

12.4.2 Specification:

- Capability to include stipends or living allowances as components of scholarship packages, in addition to tuition and fee coverage.
- Calculation and disbursement of stipend amounts based on predefined criteria, such as cost of living, academic program, or duration of scholarship.
- Integration with financial systems to process stipend payments securely and efficiently, with options for direct deposit or electronic transfers.

12.4.3 Verification:

- Validate stipend administration functionality to ensure accurate calculation and timely disbursement of stipend amounts, with validation checks for eligibility and compliance with scholarship terms.

12.5 Automated Eligibility Checks and Disbursement Processes:

12.5.1 Requirement:

- The system should perform automated eligibility checks and disbursement processes for scholarships.

12.5.2 Specification:

- Implementation of automated eligibility criteria verification to assess applicants'

qualifications and determine eligibility for scholarships.

- Workflow automation for processing scholarship awards, including application review, eligibility verification, selection, and disbursement.
- Generation of automated notifications and updates to applicants regarding their scholarship status, eligibility outcomes, and disbursement details.

12.5.3 Verification:

- Test automated eligibility checks and disbursement processes to ensure accuracy, efficiency, and transparency in evaluating scholarship applications and awarding scholarships.

12.6 *Management of Sponsors:*

12.6.1 Requirement:

- The system should manage information related to sponsors providing scholarship funding.

12.6.2 Specification:

- Capability to create and maintain sponsor profiles within the system, including contact details, funding commitments, and sponsorship terms.
- Tracking features for monitoring sponsor contributions, disbursed funds, and compliance with sponsorship agreements.
- Integration with donor management systems or CRM platforms to synchronize sponsor data and streamline communication and reporting.

12.6.3 Verification:

- Test sponsor management functionality to ensure accurate recording and tracking of sponsor information, funding commitments,

and interactions, with validation checks for data integrity and synchronization.

12.7 *Capability to Promote Scholarships on Social Media Platforms and Student Portals:*

12.7.1 Requirement:

- The system should promote scholarships on social media platforms and through the messaging app on student portals.

12.7.2 Specification:

- Integration with social media platforms (e.g., Facebook, Twitter, LinkedIn) to publish scholarship announcements, updates, and application reminders to a wider audience.
- Embedding promotional banners, announcements, or links within the student portal interface to inform students about available scholarships and encourage application submissions.
- Automation features for scheduling and publishing scholarship promotions, including opening and closing dates for application periods.

12.7.3 Verification:

- Validate scholarship promotion features through testing, ensuring seamless integration with social media platforms and student portals, with validation checks for scheduling accuracy and message delivery.

12.8 *Reporting on Scholarship Utilization and Outcomes:*

12.8.1 Requirement:

- The system should provide reporting on scholarship utilization and outcomes.

12.8.2 Specification:

- Development of predefined and customizable reports to analyse scholarship utilization, distribution, and impact over time.
- Metrics tracking for scholarship recipients, including academic performance, retention rates, graduation rates, and post-graduation outcomes.
- Integration with data visualization tools or dashboards to present scholarship analytics in a visually accessible and actionable format for stakeholders.

12.8.3 Verification:

- Validate reporting functionality to ensure that reports provide insights into scholarship utilization, effectiveness, and outcomes, with validation checks for data accuracy, completeness, and relevance.

13. Program & Course Management

13.1 Central Repository for Undergraduate and Postgraduate Program and Course Information:

13.1.1 Requirement:

- The system should serve as a central repository for undergraduate and postgraduate program and course information, including descriptions and prerequisites.

13.1.2 Specification:

- Capability to store comprehensive details about undergraduate and postgraduate programs, such as program names, descriptions, credit requirements, and program outlines.
- Storage of course information within each program, including course titles, descriptions, learning objectives, prerequisites, credit points, and course outlines.
- Support for organizing program and course information hierarchically, allowing easy navigation and retrieval of relevant data.

13.1.3 Verification:

- Validate data storage and retrieval functionality through testing, ensuring that program and course information is accurately captured, organized, and accessible within the system.

13.2 Central Repository for Technical Vocational Education Training (TVET) Program and Course Information:

- #### **13.2.1 Requirement:**
- The system should serve as a central repository for Technical Vocational Education Training (TVET) program and course information, including descriptions and prerequisites.

13.2.2 Specification:

- Similar to the undergraduate and postgraduate programs, the system should store comprehensive details about TVET programs, including program names, descriptions, certification levels, and competency frameworks.
- Storage of course information within each TVET program, including course titles, descriptions, learning outcomes, competencies, prerequisites, and credit hours.
- Customization options for aligning TVET program and course structures with industry standards, accreditation requirements, and workforce demands.

13.2.3 Verification:

- Validate data storage and retrieval functionality specific to TVET programs and courses, ensuring that relevant information is accurately captured, structured, and maintained within the system.

13.3 *Central Repository for Pre-University Preparatory Program and Course Information:*

13.3.1 Requirement:

- The system should serve as a central repository for Pre-university preparatory program and course information, including descriptions and prerequisites.

13.3.2 Specification:

- Capability to store details about pre-university preparatory programs, including program names, descriptions, objectives, and target student demographics.
- Storage of course information within each preparatory program, including course titles, descriptions, learning objectives, prerequisites, and duration.

- Integration with academic advising or career counselling services to guide prospective students in selecting suitable preparatory programs based on their academic background and career goals.

13.3.3 Verification: Validate data storage and retrieval functionality for pre-university preparatory programs and courses, ensuring that relevant information is accurately recorded and accessible to students and academic advisors.

13.4 *Central Repository for Distance Flexible Learning Program and Course Information:*

13.4.1 Requirement:

- The system should serve as a central repository for Distance Flexible Learning program and course information, including descriptions and prerequisites.

13.4.2 Specification:

- Similar to other program types, the system should store comprehensive details about Distance Flexible Learning programs, including program names, descriptions, delivery modes, and learning outcomes.
- Storage of course information within each Distance Flexible Learning program, including course titles, descriptions, learning objectives, prerequisites, and instructional materials.
- Integration with learning management systems (LMS) or online course platforms to facilitate delivery of distance education courses, including multimedia resources, interactive activities, and assessments.

13.4.3 Verification:

- Validate data storage and retrieval functionality specific to Distance Flexible Learning programs and courses, ensuring

seamless integration with online learning platforms and adherence to instructional design standards.

13.5 *Capability for Course Administrators to Indicate Mode of Learning:*

13.5.1 Requirement:

- Course administrators should be able to indicate whether a program or course is available through Distance Flexible Learning mode or face-to-face mode on campus, linked to the Student Application portal for mode selection during application.

13.5.2 Specification:

- User interface for course administrators to specify the mode of delivery (e.g., online, blended, on-campus) for each program and course offering.
- Integration with the Student Application portal to display available modes of learning for each program and course during the application process, allowing applicants to make informed decisions.
- Real-time synchronization between program/course information and application portal to reflect updates or changes in delivery modes promptly.

13.5.3 Verification:

- Validate mode selection functionality through testing, ensuring that course administrators can update delivery modes accurately and that applicants can view and select preferred modes of learning during the application process.

13.6 *Version Control for Program and Course Updates:*

13.6.1 Requirement:

- The system should implement version control for program and course updates to track changes and maintain historical records.

13.6.2 Specification:

- Versioning functionality to track revisions, updates, and modifications made to program and course information over time.
- Capture of metadata for each version, including timestamps, authorship, and change descriptions, to provide a clear audit trail of modifications.
- Access controls and permissions management to restrict editing privileges and ensure data integrity, with designated roles for content creators, reviewers, and approvers.

13.6.3 Verification:

- Validate version control functionality through testing, ensuring that changes to program and course information are accurately recorded, tracked, and managed, with appropriate access controls and audit trails in place.

13.7 *Integration with Academic Catalogue and Scheduling Systems:*

13.7.1 Requirement:

- The system should integrate with academic catalogue and scheduling systems for seamless data exchange and coordination.

13.7.2 Specification:

- Data synchronization mechanisms to exchange program and course information between the SMS and academic catalogue systems, ensuring consistency and accuracy of course offerings and descriptions.

- Integration with scheduling systems to align course offerings with academic calendars, semester schedules, and classroom assignments, facilitating course planning and timetabling.
- Real-time updates and notifications to stakeholders (students, faculty, administrators) regarding changes to program or course availability, scheduling conflicts, or cancellations.

13.7.3 Verification:

- Validate integration with academic catalogue and scheduling systems through testing, ensuring smooth data exchange, synchronization, and communication between systems, with validation checks for data accuracy and timeliness.

14. Time Table Management

14.1 Automated Generation of Class Schedules:

14.1.1 Requirement:

- The system should automate the generation of class schedules based on course offerings, room allocation, and faculty availability.

14.1.2 Specification:

- Integration with course management systems to retrieve course offerings, including course codes, titles, sections, and enrolment capacities.
- Allocation of available rooms and facilities based on course requirements, capacity constraints, and scheduling preferences.
- Incorporation of faculty availability, teaching preferences, and workload considerations into the scheduling algorithm to optimize resource utilization and minimize conflicts.
- Automatic generation of class schedules for each academic term or semester, considering factors such as class duration, break times, and institutional policies.

14.1.3 Verification:

- Validate automated scheduling functionality through testing, ensuring that class schedules are generated accurately, efficiently, and in compliance with institutional requirements and constraints.

14.2 Flexible Scheduling Options:

14.2.1 Requirement:

- The system should offer flexible scheduling options to accommodate different teaching formats and modes of delivery.

14.2.2 Specification:

- Configuration settings to support various teaching formats, such as undergraduate, postgraduate, research, Technical Vocational Education Training (TVET), and University Preparatory Programs.
- Customization options for defining class structures, including lecture sessions, seminars, tutorials, labs, workshops, and fieldwork activities.
- Flexibility to accommodate different modes of delivery, such as online, blended (hybrid), and face-to-face instruction, based on course requirements and instructional objectives.

14.2.3 Verification:

- Test scheduling flexibility through scenario simulations, ensuring that the system can accommodate diverse teaching formats and delivery modes effectively, with validation checks for scheduling accuracy and adaptability.

14.3 *Ability to Publish and Distribute Timetables:*

14.3.1 Requirement:

- The system should have the ability to publish and distribute timetables to students and staff.

14.3.2 Specification:

- User-friendly interface for administrators to review and finalize class schedules before publication.
- Options for generating timetable reports in various formats (e.g., PDF, HTML) for easy viewing and distribution.
- Integration with student and staff portals to publish timetables online, providing access to registered students, faculty members, and administrative staff.

- Notification mechanisms to inform stakeholders (students, faculty) about timetable updates, changes, or cancellations via email, SMS, or push notifications.

14.3.3 Verification:

- Validate timetable publication and distribution functionality through testing, ensuring that timetables are accessible, accurate, and up-to-date for all stakeholders, with validation checks for data consistency and notification delivery.

15. Student Attendance Management

15.1 Recording and Tracking of Student Attendance:

15.1.1 Requirement:

- The system should record and track student attendance for classes, tutorials, and practicals.

15.1.2 Specification:

- Capability to capture attendance data for each session or event, including date, time, and student identification.
- Support for different types of attendance records, such as roll-call, barcode scanning, biometric identification, or RFID card swiping.
- Ability to track attendance for various academic activities, including lectures, tutorials, labs, and practicals.
- Storage of attendance records in a centralized database, associating each record with the respective student, course, and session details.

15.1.3 Verification:

- Validate attendance recording and tracking functionality through testing, ensuring accurate capture and storage of attendance data for different types of activities and events.

15.2 Support for Manual and Automated Attendance Capture Methods:

15.2.1 Requirement:

- The system should support both manual and automated attendance capture methods.

15.2.2 Specification:

- User interface for manual attendance recording, allowing instructors to mark

student attendance using digital forms or traditional paper-based methods.

- Integration with automated attendance capture systems, such as biometric scanners, RFID readers, or mobile applications, to streamline data collection and minimize manual effort.
- Flexibility to switch between manual and automated methods based on instructor preferences, class size, or environmental constraints.

15.2.3 Verification:

- Test both manual and automated attendance capture methods to ensure seamless integration and compatibility with the system, with validation checks for data accuracy and reliability.

15.3 *Support for Online Automated Attendance Capture Methods:*

15.3.1 Requirement:

- The system should support online automated attendance capture methods.

15.3.2 Specification:

- Integration with online attendance tracking tools or platforms, allowing students to log attendance remotely using web-based or mobile applications.
- Implementation of secure authentication mechanisms to verify student identities and prevent unauthorized attendance submissions.
- Real-time synchronization of online attendance data with the central attendance database, ensuring immediate updates and accuracy of records.

15.3.3 Verification:

- Validate online automated attendance capture methods through testing, ensuring secure and reliable data transmission, authentication, and synchronization with the central system.

15.4 *Integration with Course Grading and Evaluation Systems:*

15.4.1 *Requirement:*

- The system should integrate with course grading and evaluation systems.

15.4.2 *Specification:*

- Synchronization of attendance data with course management systems or grading platforms to incorporate attendance records into overall student assessment and evaluation processes.
- Calculation of attendance-based grades or participation scores as components of course assessments, in alignment with course syllabi and grading policies.
- Configuration options for defining attendance criteria, weightage, and grading scales within the grading system, allowing instructors to customize assessment parameters based on course requirements.

15.4.3 *Verification:*

- Validate integration with course grading and evaluation systems through testing, ensuring accurate calculation and incorporation of attendance data into student assessments and final grades.

15.5 *Integration with Student Portal:*

15.5.1 *Requirement:*

- The system should integrate with the student portal to display current attendance data.

15.5.2 Specification:

- Development of a dedicated attendance module within the student portal interface, allowing students to view their attendance records for enrolled courses.
- Secure functionality to personalized access to attendance information based on individual enrolment and course registrations.
- Real-time data synchronization between the attendance management system and the student portal to ensure up-to-date display of attendance records and statuses.

15.5.3 Verification:

- Validate integration with the student portal through testing, ensuring seamless navigation, authentication, and access to attendance data for registered students.

15.6 *Integration with Lecturer Portal:*

15.6.1 Requirement:

- The system should integrate with the lecturer portal to display course attendance data.

15.6.2 Specification:

- Implementation of an attendance dashboard or module within the lecturer portal interface, allowing instructors to view attendance records for their assigned courses.
- Role-based access controls to restrict attendance data visibility to authorized instructors and teaching staff, ensuring confidentiality and data privacy.
- Customizable views and filters for instructors to analyze attendance data by course, session, student, or other relevant parameters.

15.6.3 Verification:

- Validate integration with the lecturer portal through testing, ensuring secure authentication, role-based access, and user-friendly presentation of attendance data for instructors.

15.7 Report Generation by Course and Program, and by Mode of Delivery:

15.7.1 Requirement:

- The system should generate reports by course and program, and by mode of delivery, to display attendance data.

15.7.2 Specification:

- Predefined and customizable report templates for generating attendance reports by course, program, department, or academic unit, summarizing attendance data for analysis and review.
- Filtering options to generate reports based on specific criteria, such as date range, academic term, course instructor, or delivery mode (e.g., online, blended, face-to-face).
- Aggregation of attendance data at the program level to assess overall attendance rates, trends, and patterns across multiple courses and student cohorts.

15.7.3 Verification:

- Validate report generation functionality through testing, ensuring accurate and comprehensive presentation of attendance data by course, program, and mode of delivery, with validation checks for data integrity and report formatting.

16. Disciplinary Management

16.1 Recording and Tracking of Disciplinary Incidents and Actions:

16.1.1 Requirement:

- The system should record and track disciplinary incidents and actions.

16.1.2 Specification:

- Capability to capture details of disciplinary incidents, including date, time, location, nature of the offense, involved parties, witnesses, and evidence.
- Logging of disciplinary actions taken in response to incidents, such as warnings, sanctions, referrals to disciplinary boards, suspensions, or expulsions.
- Storage of incident and action data in a centralized database, associating each record with the respective student and incident details.

16.1.3 Verification:

- Validate recording and tracking functionality through testing, ensuring accurate capture and storage of disciplinary incident and action data in compliance with institutional policies and legal requirements.

16.2 Capability to Track and Manage Different Types of Disciplinary Incidents:

16.2.1 Requirement:

- The system should track and manage academic discipline, code of student conduct discipline, and criminal discipline incidents.

16.2.2 Specification:

- Segregation of disciplinary incidents into categories based on the type of offense, including academic misconduct (e.g., plagiarism, cheating), violations of the

student code of conduct, and criminal behaviour.

- Customization options for defining incident classifications, severity levels, and corresponding disciplinary procedures based on institutional policies and legal standards.
- Flexibility to associate each incident with relevant policy violations, sanctions, and follow-up actions specific to the type of discipline involved.

16.2.3 Verification:

- Test incident tracking and management capabilities for different types of disciplinary infractions, ensuring accurate categorization, classification, and processing of incidents according to established disciplinary protocols.

16.3 *Integration with Student Portal for Notices and Messaging:*

16.3.1 Requirement:

- The system should integrate with the Student Portal for related notices and messaging.

16.3.2 Specification:

- Development of a dedicated disciplinary module within the Student Portal interface, allowing students to view notices, updates, and communications related to disciplinary matters.
- Integration with messaging features to notify students about disciplinary incidents, investigations, outcomes, and any required actions or hearings.

16.3.3 Verification:

- Validate integration with the Student Portal through testing, ensuring seamless communication and notification functionalities for students regarding

disciplinary matters, with validation checks for data security and access permissions.

16.4 *Workflow for Investigation, Resolution, and Escalation of Disciplinary Issues:*

16.4.1 Requirement:

- The system should have a workflow for investigation, resolution, and escalation of disciplinary issues.

16.4.2 Specification:

- Definition of standardized workflows and procedures for handling disciplinary incidents, from initial reporting and investigation to resolution and escalation as necessary.
- Assignment of roles and responsibilities to stakeholders involved in the disciplinary process, including investigators, disciplinary officers, hearing boards, administrators, and legal advisors.
- Incorporation of escalation pathways for addressing serious or repeat offenses, with predefined steps for progressive disciplinary measures, such as warnings, counselling, disciplinary hearings, temporary suspension, and permanent termination.

16.4.3 Verification:

- Validate workflow functionality through testing, ensuring adherence to established disciplinary protocols and escalation procedures, with validation checks for workflow efficiency and compliance with legal and regulatory requirements.

16.5 *Confidentiality and Security Controls for Sensitive Disciplinary Records:*

16.5.1 Requirement:

- The system should have confidentiality and security controls for sensitive disciplinary records.

16.5.2 Specification:

- Implementation of access controls and permissions management to restrict access to disciplinary records based on role-based authentication and authorization.
- Encryption of sensitive disciplinary data to protect against unauthorized access, tampering, or disclosure.
- Logging and auditing mechanisms to track access to disciplinary records, including user actions, timestamps, and IP addresses, for accountability and compliance purposes.

16.5.3 Verification:

- Validate confidentiality and security controls through testing, ensuring robust protection of disciplinary records against unauthorized access, data breaches, and privacy violations, with validation checks for access controls and audit trail accuracy.

16.6 *Report Generation with Filtering Capability:*

16.6.1 Requirement:

- The system should generate reports with filtering capability for academic discipline, code of student conduct discipline, and criminal discipline.

16.6.2 Specification:

- Development of predefined and customizable report templates for generating disciplinary reports based on different criteria, such as incident type, severity, date range, and student demographics.
- Filtering options to generate reports specific to academic discipline, code of student conduct discipline, criminal discipline, or

combinations thereof, to facilitate analysis and review.

- Export functionality to generate reports in various formats (e.g., PDF, CSV) for dissemination to stakeholders, regulatory bodies, or legal authorities as required.

16.6.3 Verification:

- Validate report generation functionality through testing, ensuring flexibility and usability of filtering options for generating disciplinary reports tailored to specific needs and requirements.

17. Examinations & Mark Sheet Management

17.1 Administration of Examinations:

17.1.1 Requirement:

- The system should administer examinations, including scheduling, seating arrangements, and invigilation.

17.1.2 Specification:

- Interface for administrators to create examination schedules, specifying exam dates, times, locations, and seating arrangements.
- Capability to assign invigilators to examination venues, manage proctoring duties, and oversee exam administration.
- Flexibility to accommodate different types of examinations, including written tests, practical exams, oral assessments, and presentations.

17.1.3 Verification:

- Validate examination administration functionality through testing, ensuring accurate scheduling, seating assignments, and invigilation procedures according to institutional policies and academic regulations.

17.2 Automated Workflow for Test and Examination Question Preparation:

17.2.1 Requirement:

- The system should have an automated workflow for tests and examination question preparation and approval process in the faculty.

17.2.2 Specification:

- Workflow automation tools to streamline the process of test and examination question

creation, review, and approval by faculty members.

- Integration with collaborative authoring platforms or document management systems to facilitate online collaboration and version control for question development.
- Notification mechanisms to alert faculty members about pending tasks, deadlines, or revisions required during the question preparation process.

17.2.3 Verification:

- Validate workflow automation functionality through testing, ensuring efficient and transparent coordination of test and examination question preparation tasks among faculty members.

17.3 *Secure Storage and Distribution of Examination Papers:*

17.3.1 Requirement:

- The system should securely store and distribute examination papers.

17.3.2 Specification:

- Secure file storage infrastructure with access controls and encryption mechanisms to protect confidentiality and integrity of examination papers.
- Role-based permissions management to restrict access to examination papers based on user roles and responsibilities (e.g., administrators, faculty members, proctors).
- Secure distribution channels for delivering examination papers to designated examination venues or invigilators, ensuring secure transmission and delivery.

17.3.3 Verification:

- Validate secure storage and distribution functionality through testing, ensuring robust

protection of examination papers against unauthorized access, tampering, or leakage.

17.4 *Functionality for Supplementary Tests or Examinations:*

17.4.1 Requirement:

- The system should have functionality to administer supplementary tests or examinations due to absenteeism with valid reasons.

17.4.2 Specification:

- Interface for administrators to schedule supplementary tests or examinations for students who missed scheduled assessments due to valid reasons, such as illness or extenuating circumstances.
- Integration with student records and attendance data to verify eligibility and determine appropriate scheduling and assessment criteria for supplementary exams.
- Automated notifications to inform eligible students about supplementary exam dates, venues, and registration procedures.

17.4.3 Verification:

- Validate functionality for supplementary tests or examinations through testing, ensuring accurate scheduling, eligibility checks, and communication with affected students.

17.5 *Automated Grading and Generation of Mark Sheets:*

17.5.1 Requirement:

- The system should automate grading and generation of mark sheets, with the capability for Faculty Deans to set automated deadlines for marking and marks integration into student records.

17.5.2 Specification:

- Automated grading tools to streamline assessment and grading processes, including multiple-choice scoring, rubric-based grading, and calculation of weighted scores.
- Integration with mark sheet generation modules to produce standardized mark sheets or grade reports for each student, summarizing assessment results and final grades.
- Configuration options for Faculty Deans to set automated deadlines for grading submissions by faculty members and ensure timely processing of assessment data.

17.5.3 Verification:

- Validate automated grading and mark sheet generation functionality through testing, ensuring accuracy, consistency, and efficiency in grading processes and mark sheet generation.

17.6 *Integration with Portals:*

17.6.1 Requirement:

- The system should integrate with the Student portal, lecturer portal, tutor portal, Head of Department portal, Head of School portal, and Faculty Dean portal.

17.6.2 Specification:

- Seamless integration with various portals to facilitate access to examination-related information, such as exam schedules, venues, instructions, and results.
- Customized views and functionalities tailored to the roles and responsibilities of different portal users (e.g., students, faculty members, department heads, and deans).

- Real-time data synchronization between the examination management system and portal interfaces to ensure up-to-date access to examination-related information and updates.

17.6.3 Verification:

- Validate integration with portals through testing, ensuring smooth navigation, authentication, and data exchange between the examination management system and portal interfaces.

17.7 Confidentiality and Security Controls:

17.7.1 Requirement:

- The system should have confidentiality and security controls for sensitive examination records.

17.7.2 Specification:

- Implementation of access controls, encryption mechanisms, and audit trails to protect the confidentiality and security of examination-related data, including question papers, assessment records, and student grades.
- Role-based permissions management to restrict access to examination data based on user roles and responsibilities, ensuring that only authorized personnel can view, modify, or delete sensitive information.
- Regular security audits and vulnerability assessments to identify and mitigate potential risks or threats to examination data security.

17.7.3 Verification:

- Validate confidentiality and security controls through testing, ensuring robust protection of examination-related data against

unauthorized access, data breaches, or integrity violations.

18. Assignment Management

18.1 Assignment Submission and Grading Platform:

18.1.1 Requirement:

- The system should provide an assignment submission and grading platform for instructors and students.

18.1.2 Specification:

- User-friendly interface for students to submit assignments electronically, with options to upload files, enter text, or provide links to external resources.
- Integration with grading tools and rubrics to facilitate efficient evaluation and feedback by instructors.
- Integration with LMS.
- Real-time notifications to inform students about assignment submissions, grading updates, and feedback provided by instructors.

18.1.3 Verification:

- Validate submission and grading platform functionality through testing, ensuring smooth assignment submission, grading, and feedback processes for both students and instructors.

18.2 Automated Messaging for Approaching Due Dates:

18.2.1 Requirement:

- The system should send automated messages to the student portal for approaching due dates and confirmation of successful assignment submission.

18.2.2 Specification:

- Automated notifications to remind students of upcoming assignment deadlines via the student portal, email, or mobile app notifications.

- Confirmation messages sent to students upon successful submission of assignments, providing acknowledgment and assurance of receipt.
- Customizable notification settings for students to manage frequency and preferences for receiving reminders and confirmation messages.

18.2.3 Verification:

- Validate automated messaging functionality through testing, ensuring timely delivery and accuracy of notifications for assignment deadlines and submission confirmations.

18.3 *Capability for Multiple Assignment Submissions:*

18.3.1 Requirement:

- The system should allow students to delete and re-submit assignments multiple times before the deadline.

18.3.2 Specification:

- Module functionality to enable students to delete and resubmit assignments within the deadline period, allowing for iterative improvements or corrections.
- Validation checks to prevent students from resubmitting assignments if the existing version has not been deleted, with pop-up notifications prompting students to delete existing submissions before resubmitting.
- Enforcement of late submission rules for students who delete and resubmit assignments after the original deadline, with penalties applied according to university policies.

18.3.3 Verification:

- Validate multiple submission functionality through testing, ensuring seamless handling of assignment deletions, resubmissions, and

late submissions with accurate penalty calculations.

18.4 *Setting Due Date and Time for Assignments:*

18.4.1 Requirement:

- The system should allow lecturers to set due dates and times for assignments.

18.4.2 Specification:

- Interface for instructors to specify assignment due dates and times, providing clear instructions and expectations for students.
- Customization options to accommodate different assignment deadlines based on course requirements, instructional schedules, and assessment criteria.
- Support for time zone adjustments and deadline extensions as needed to accommodate students in different geographic locations or time zones.

18.4.3 Verification:

- Validate due date and time settings functionality through testing, ensuring flexibility and precision in assignment deadline management by instructors.

18.5 *Setting Late Submission Policies:*

18.5.1 Requirement:

- The system should allow administrators to set the number of days after which late assignments will be accepted and the penalties incurred.

18.5.2 Specification:

- Administrative controls to configure parameters for late assignment submissions, including the number of days allowed for late

- submissions and the corresponding percentage penalties for each day overdue.
- Enforcement mechanisms to automatically calculate penalty deductions based on late submission durations and apply them to student grades accordingly.
 - Lockdown of assignment submission module after the specified late submission deadline to prevent further late submissions and maintain fairness in assessment procedures.

18.5.3 Verification:

- Validate late submission policies functionality through testing, ensuring accurate calculation and application of penalties for late assignments in compliance with institutional policies.

18.6 *Support for Various File Formats and Submission Types:*

18.6.1 Requirement:

- The system should support various file formats and submission types for assignments.

18.6.2 Specification:

- Compatibility with a wide range of file formats commonly used for assignment submissions, including documents, spreadsheets, presentations, images, audio files, and video recordings.
- Integration with cloud storage platforms or document repositories to facilitate submission of large files or multimedia content.
- Flexibility to accommodate alternative submission types, such as URLs, online quizzes, code repositories, or multimedia portfolios, based on assignment requirements and instructor preferences.

18.6.3 Verification:

- Validate file format and submission type support through testing, ensuring seamless acceptance and processing of diverse assignment submissions with validation checks for file compatibility and integrity.

18.7 *Plagiarism Detection and Prevention Features:*

18.7.1 Requirement:

- The system should include plagiarism detection and prevention features.

18.7.2 Specification:

- Integration with plagiarism detection software or services to scan assignment submissions for similarities with existing sources and detect potential instances of academic dishonesty.
- Threshold settings to define acceptable plagiarism percentages and corresponding penalties for different levels of similarity detected in assignment submissions.
- Administrative controls to configure plagiarism detection parameters, including exclusion rules for citations, references, and common phrases, and set penalties for plagiarism violations.

18.7.3 Verification:

- Validate plagiarism detection and prevention features through testing, ensuring accurate identification of plagiarism instances, adherence to predefined thresholds, and consistent application of penalties for academic integrity violations.

19. Online Learning Management System

19.1 Integrated Platform for Delivering Course Materials, Assignments, Quizzes, and Assessments:

19.1.1 Requirement:

- The system should provide an integrated platform for delivering online course materials, assignments, quizzes, and assessments.

19.1.2 Specification:

- User-friendly interface for instructors to upload and organize course materials, including lecture slides, readings, videos, and supplementary resources.
- Functionality for students to access and view course materials, download resources, and interact with learning content asynchronously.
- Support for assignment submission, including file uploads, text submissions, and online assessments, with automatic grading and feedback features.
- Integration with assessment tools to create and administer quizzes, exams, and other types of assessments within the learning management system.

19.1.3 Verification:

- Validate platform functionality through testing, ensuring seamless delivery and management of course materials, assignments, quizzes, and assessments for both instructors and students.

19.2 Support for Multimedia Content, Discussion Forums, and Collaboration Tools:

19.2.1 Requirement:

- The system should support multimedia content, discussion forums, and collaboration tools.

19.2.2 Specification:

- Compatibility with various multimedia formats for embedding audio, video, images, and interactive media elements into course content and activities.
- Creation of discussion forums and interactive spaces where students can engage in asynchronous discussions, ask questions, and collaborate with peers and instructors.
- Integration with collaboration tools, such as wikis, blogs, virtual classrooms, and group workspaces, to facilitate teamwork, project-based learning, and knowledge sharing.

19.2.3 Verification:

- Validate multimedia and collaboration features through testing, ensuring seamless integration and usability of multimedia content, discussion forums, and collaboration tools within the learning environment.

19.3 *Integration with Online Meeting Platforms:*

19.3.1 Requirement:

- The system should support integration with online meeting platforms such as Microsoft Teams, Zoom, Google Meet, Slack, etc.

19.3.2 Specification:

- Integration with popular online meeting platforms to facilitate synchronous communication, virtual lectures, and live interactions between instructors and students.
- Seamless access to online meeting rooms or virtual classrooms directly from the learning management system interface, with single sign-on capabilities and calendar integration.

- Compatibility with features such as screen sharing, chat messaging, video conferencing, breakout rooms, and recording functionalities offered by the integrated meeting platforms.

19.3.3 Verification:

- Validate integration with online meeting platforms through testing, ensuring smooth connectivity, functionality, and user experience for conducting virtual lectures, discussions, and collaborative sessions.

19.4 *Tracking of Student Attendance, Progress, and Engagement:*

19.4.1 Requirement:

- The system should track student attendance, progress, and engagement.

19.4.2 Specification:

- Automated tracking of student attendance in online classes and virtual meetings, capturing login timestamps, session durations, and participation levels.
- Monitoring of student progress through course milestones, completion rates, assignment submissions, quiz scores, and other learning activities recorded within the system.
- Analysis of student engagement metrics, such as interaction frequency, discussion forum participation, resource access patterns, and response times to instructor feedback.

19.4.3 Verification:

- Validate attendance, progress, and engagement tracking functionality through testing, ensuring accurate data capture, reporting, and analysis of student

interactions and performance within the online learning environment.

20. Completion Management (Graduation & Certificates)

20.1 Workflow for Graduation Application, Review, and Approval:

20.1.1 Requirement:

- The system should provide a workflow for graduation application, review, and approval.

20.1.2 Specification:

- Interface for students to initiate graduation applications electronically, providing necessary information such as completion of academic requirements, major/minor declarations, and optional honours.
- Workflow automation tools to route graduation applications through predefined stages, including departmental review, faculty approval, registrar verification, and final clearance by academic authorities.
- Notifications to alert students about the status of their graduation applications, including pending reviews, missing requirements, and final approval decisions.

20.1.3 Verification:

- Validate workflow functionality through testing, ensuring smooth processing of graduation applications from submission to approval, with clear communication and transparent status tracking for students and administrative staff.

20.2 Generation and Distribution of Diplomas, Transcripts, and Certificates:

20.2.1 Requirement:

- The system should generate and distribute diplomas, transcripts, and certificates.

20.2.2 Specification:

- Automated generation of official graduation documents, including diplomas, transcripts,

degree certificates, and academic awards, based on approved graduation applications and academic records.

- Customization options for document formatting, content presentation, and institutional branding to comply with academic standards, accreditation requirements, and branding guidelines.
- Secure distribution channels for delivering official documents to graduating students, alumni, academic institutions, employers, and other authorized recipients, ensuring confidentiality, authenticity, and integrity.

20.2.3 Verification:

- Validate document generation and distribution functionality through testing, ensuring accurate and timely delivery of official graduation documents in compliance with academic regulations and accreditation standards.

20.3 *Compliance with Academic Regulations and Accreditation Requirements:*

20.3.1 Requirement:

- The system should comply with academic regulations and accreditation requirements.

20.3.2 Specification:

- Integration with academic policies, regulations, and accreditation standards to ensure alignment with institutional guidelines, industry benchmarks, and regulatory frameworks.
- Regular updates and maintenance to reflect changes in academic requirements, degree programs, curriculum structures, and accreditation criteria, ensuring ongoing compliance and quality assurance.

- Audit trails and version controls to track revisions, updates, and approvals related to academic regulations, accreditation reports, and compliance documentation.

20.3.3 Verification:

- Validate compliance functionality through testing, ensuring adherence to academic regulations and accreditation requirements, with comprehensive documentation, audit trails, and governance mechanisms in place.

20.4 *Integration with Student Portal for Graduation Application:*

20.4.1 Requirement:

- The system should integrate with the Student Portal for graduation application.

**20.4.2 Specification: **

- Seamless integration between the completion management module and the Student Portal to enable students to initiate and track graduation applications directly from their online accounts.
- Single sign-on functionality to provide secure access to graduation application features within the Student Portal interface, with consistent user experience and authentication mechanisms.
- Real-time data synchronization between the completion management module and the Student Portal, ensuring accurate reflection of graduation application status, academic records, and degree progress.

20.4.3 Verification:

- Validate integration functionality through testing, ensuring smooth navigation, data exchange, and user interaction between the completion management module and the Student Portal interface, with seamless

access to graduation application features
and status updates for students.

21. Academic Calendar Management

21.1 Centralized Calendar System:

21.1.1 Requirement:

- The system should provide a centralized calendar system for academic events, deadlines, and holidays.

21.1.2 Specification:

- User-friendly interface displaying a comprehensive view of academic events, including semester start and end dates, registration deadlines, examination periods, holidays, and special events.
- Customizable filters and categories to organize and differentiate between different types of events, such as academic, administrative, extracurricular, and public holidays.
- Support for recurring events, allowing administrators to schedule events that repeat on a regular basis, such as weekly classes or monthly meetings.

21.1.3 Verification:

- Validate calendar system functionality through testing, ensuring accurate display and management of academic events, deadlines, and holidays, with intuitive navigation and customizable views for users.

21.2 Incorporation of Semester, Trimester, and TVET Block Courses:

21.2.1 Requirement:

- The system should incorporate semester, trimester, and TVET block courses for outcome-based learning modules.

21.2.2 Specification:

- Integration with academic scheduling systems to populate the calendar with

course-related events, such as course start and end dates, class sessions, assessments, and project deadlines.

- Support for different academic terms, including traditional semesters, trimesters, and block scheduling formats used in Technical and Vocational Education and Training (TVET) programs.
- Flexibility to accommodate variations in course durations, academic calendars, and instructional formats across different programs and departments.

21.2.3 Verification:

- Validate course incorporation functionality through testing, ensuring accurate representation and synchronization of semester, trimester, and TVET block courses within the academic calendar, with support for diverse learning modules and scheduling patterns.

21.3 *Administrator Access for Calendar Management:*

21.3.1 Requirement:

- The system should provide administrator access to insert public holidays, important meetings, and important deadlines.

21.3.2 Specification:

- Secure login and role-based access control mechanisms to grant administrators privileges for calendar management tasks, such as adding, editing, or deleting events.
- User-friendly administrative interface with intuitive controls and form fields for entering event details, including event names, descriptions, dates, times, locations, and associated resources.

- Validation checks and error handling to prevent duplicate entries, overlapping events, and conflicting schedules.

21.3.3 Verification:

- Validate administrator access functionality through testing, ensuring smooth navigation, data entry, and validation of calendar events by authorized personnel, with robust security measures and data integrity safeguards in place.

21.4 *Synchronization with Personal Calendars and Mobile Devices:*

21.4.1 Requirement:

- The system should have the ability to synchronize with personal calendars and mobile devices.

21.4.2 Specification:

- Integration with popular calendar applications and platforms, such as Google Calendar, Apple Calendar, Microsoft Outlook, and mobile device calendars (iOS, Android), using standard protocols (e.g., CalDAV, iCal).
- Support for bidirectional synchronization, allowing users to import calendar events from the SMS into their personal calendars and vice versa, ensuring consistency and accessibility across multiple devices and platforms.

21.4.3 Verification:

- Validate synchronization functionality through testing, ensuring seamless data exchange and synchronization between the SMS calendar system and external calendar applications, with real-time updates and compatibility across different devices and operating systems.

21.5 Notification System for Upcoming Events and Deadlines:

21.5.1 Requirement:

- The system should have a notification system for upcoming events and deadlines directed to the student portal and staff portal.

21.5.2 Specification:

- Automated notification system to alert users about upcoming events, deadlines, and important announcements via the student portal and staff portal interfaces.
- Customizable notification settings for users to define preferences regarding event types, delivery channels (e.g., email, SMS, in-app notifications), and notification timing (e.g., advance reminders).
- Integration with the calendar system to trigger notifications based on event schedules, user roles, and subscription preferences, ensuring timely and relevant communications.

21.5.3 Verification:

- Validate notification system functionality through testing, ensuring accurate delivery and presentation of event notifications to users via the student portal and staff portal interfaces, with options for customization and personalization according to user preferences and roles.

22. Library Reservation and Borrowed Books Module

22.1 *Reservation System for Library Resources:*

22.1.1 Requirement:

- The system should provide a reservation system for library resources, including books, journals, and multimedia materials.

22.1.2 Specification:

- User-friendly interface for students and staff to search and reserve library resources online, including books, journals, DVDs, and other materials, through the library portal.
- Reservation management features allowing users to view availability, place holds, and schedule pickup times for reserved items.
- Real-time updating of reservation statuses and availability based on library inventory and circulation data.

22.1.3 Verification:

- Validate reservation system functionality through testing, ensuring smooth reservation processes, accurate availability tracking, and seamless integration with library inventory systems.

22.2 *Management of Borrowing, Renewal, and Return Processes:*

22.2.1 Requirement:

- The system should manage borrowing, renewal, and return processes for library resources.

22.2.2 Specification:

- Checkout functionality allowing users to borrow library resources for specified loan periods, with options for renewals and returns.

- Automated renewal reminders and notifications sent to users prior to due dates, providing instructions for renewal or return.
- Overdue item management features to track late returns, apply fines or penalties, and suspend borrowing privileges for users with outstanding items.

22.2.3 Verification:

- Validate borrowing, renewal, and return processes through testing, ensuring accurate tracking of loaned items, timely notifications, and enforcement of library policies for overdue materials.

22.3 *Integration with Koha Library Catalogue and Inventory Systems:*

22.3.1 Requirement:

- The system should integrate with Koha library catalogue and inventory systems.

22.3.2 Specification:

- Seamless integration with Koha library management software for cataloguing, circulation, and inventory control of library resources.
- Synchronization of library holdings, item availability, circulation statuses, and borrower information between the SMS and Koha systems.
- Cross-platform compatibility and data exchange protocols to facilitate real-time updates and information sharing between the two systems.

22.3.3 Verification:

- Validate integration with Koha systems through testing, ensuring seamless data synchronization, accurate inventory management, and consistent user experiences across platforms.

22.4 Integration with Messaging App for Reminder of Due Dates:

22.4.1 Requirement:

- The system should integrate with a messaging app for reminders of library resource due dates or overdue library books sent directly to the student portal.

22.4.2 Specification:

- Integration with a messaging app to send automated reminders and notifications to student or staff regarding upcoming due dates and overdue items directly to the student or staff portal.
- Personalized messages containing due dates, renewal instructions, and penalty information sent directly to users' messaging app accounts in their respective student/staff portal.
- Configuration options for users to opt in or out of receiving notifications, set notification preferences, and manage communication channels.

22.4.3 Verification:

- Validate messaging app integration through testing, ensuring reliable delivery of reminders, adherence to user preferences, and compatibility with messaging app platforms.

22.5 Administrator Rule Creation for Overdue Items:

22.5.1 Requirement:

- The system should allow administrators to create rules for certain number of overdue days with reminders before library resources are classified as lost and borrowing privileges are suspended.

22.5.2 Specification:

- Administrative dashboard with tools for configuring overdue item policies, including grace periods, fine structures, and escalation thresholds.
- Rule creation functionality allowing administrators to define parameters such as the number of days overdue, frequency of reminders, and actions to be taken for non-compliance.
- Automated enforcement of overdue item rules, including generation of notifications, suspension of borrowing privileges, and invoicing for lost or damaged items.

22.5.3 Verification:

- Validate rule creation functionality through testing, ensuring flexibility, accuracy, and effectiveness of overdue item management policies, with clear communication and enforcement mechanisms for users and administrators.

23. Student Bookshop

23.1 *Online Bookstore for Purchasing Textbooks and Course Materials:*

23.1.1 Requirement:

- The system should provide an online bookstore for purchasing textbooks and course materials.

23.1.2 Specification:

- User-friendly interface for browsing, searching, and purchasing textbooks, course materials, and other educational resources online.
- Catalogue management features allowing administrators to upload, update, and categorize books by course, subject, author, and edition.
- Secure payment gateway integration for processing online transactions, supporting various payment methods such as credit/debit cards, mobile wallets, and online banking.

23.1.3 Verification:

- Validate online bookstore functionality through testing, ensuring seamless browsing, purchasing, and payment experiences for users, with accurate inventory management and secure transaction processing.

23.2 *Inventory Management and Order Tracking:*

23.2.1 Requirement:

- The system should include inventory management and order tracking features.

23.2.2 Specification:

- Inventory tracking system to monitor stock levels, track sales, and manage

replenishment of books and course materials.

- Order management functionality allowing users to track the status of their orders, view order history, and receive notifications about order processing, shipping, and delivery.
- Administrative tools for managing orders, processing refunds, and handling returns or exchanges.

23.2.3 Verification:

- Validate inventory management and order tracking functionality through testing, ensuring accurate stock monitoring, timely order fulfilment, and transparent communication with users regarding order status and shipment updates.

23.3 *Integration with Course Syllabi and Reading Lists:*

23.3.1 Requirement:

- The system should integrate with course syllabi and reading lists.

23.3.2 Specification:

- Integration with academic course management systems or learning management platforms to access course syllabi, reading lists, and recommended textbooks.
- Synchronization of bookshop inventory with course requirements, enabling automatic recommendations and linkages between course materials and relevant textbooks.
- Cross-referencing features allowing users to view course-specific book lists, compare prices, and purchase required materials directly from the bookshop interface.

23.3.3 Verification:

- Validate integration with course syllabi and reading lists through testing, ensuring

seamless access to course materials, accurate alignment with academic requirements, and convenient purchasing options for users.

23.4 *Integration with Student and Staff Portal for Ordering and Payment:*

23.4.1 Requirement:

- The system should integrate with the student and staff portal for ordering and paying for books.

23.4.2 Specification:

- Seamless integration with the student and staff portal interfaces, providing direct access to the bookshop functionality from within the existing user accounts.
- Single sign-on functionality onto student and staff portals to ensure secure authentication and access control, allowing users to shop for books without the need for separate logins or account creation.
- Integration of shopping cart and checkout features within the portal environment, enabling users to add items to their cart, complete purchases, and track order status without leaving the portal interface.

23.4.3 Verification:

- Validate portal integration functionality through testing, ensuring smooth navigation, secure transactions, and cohesive user experiences across the bookshop and portal interfaces.

23.5 *Capability to Integrate with Finance System through APIs for Receipting and Order Payment Processing:*

23.5.1 Requirement:

- The system should have the capability to integrate with the finance system through APIs for ordering and payment processing.

23.5.2 Specification:

- API-based integration with the university's finance system to facilitate seamless data exchange and transaction processing between the bookshop and finance modules.
- Secure communication protocols and data encryption mechanisms to ensure the confidentiality and integrity of financial transactions and sensitive information.
- Real-time synchronization of order details, payment records, and financial transactions between the bookshop and finance systems, providing accurate accounting and reconciliation.

23.5.3 Verification:

- Validate API integration functionality through testing, ensuring reliable data exchange, transaction integrity, and compliance with financial regulations and security standards.

24. Student Printing

24.1 *Print Management System:*

24.1.1 Requirement:

- The system should provide a comprehensive print management solution for students to manage their printing tasks efficiently.

24.1.2 Specification:

- Implementation of a web-based interface that allows students to send print jobs, track print status, and manage print preferences.

24.1.3 Verification:

- Functionality will be verified through user acceptance testing, ensuring ease of use, reliability, and accurate tracking of print jobs.

24.2 *Print Credit Balance and Payment:*

24.2.1 Requirement:

- Students must be able to view and manage their print credit balances and make payments to recharge their accounts.

24.2.2 Specification:

- Integration with online payment gateways and campus payment systems to facilitate print credit purchases. The system should provide real-time balance updates and secure transaction processing.

24.2.3 Verification:

- Testing will include payment processing and balance updates to ensure transactions are secure and balances are updated promptly.

24.3 *Integration with Campus Printing Infrastructure:*

24.3.1 Requirement:

- The system should seamlessly integrate with the existing campus printing infrastructure to support various printers and printing options.

24.3.2 Specification:

- Use of standard protocols and APIs to connect with multiple printer models and types, supporting functionalities like black and white, colour printing, and different paper sizes.

24.3.3 Verification:

- Integration testing with campus printers to confirm compatibility and functionality across different printer types and models.

24.4 *Integration with Finance Receipting System:*

24.4.1 Requirement:

- Ensure that every print job and credit purchase is accurately logged and integrated with the finance system for proper accounting and auditing.

24.4.2 Specification:

- Automated data transfer between the print management system and the university's financial system using secure APIs, ensuring accurate and timely financial reporting.

24.4.3 Verification:

- Cross-verification of transaction logs in both the print management and financial systems to ensure accuracy and integrity of financial data.

24.5 *Secure Release Printing Options:*

24.5.1 Requirement:

- Provide secure options for releasing print jobs to protect student privacy and confidentiality.

24.5.2 Specification:

- Implementation of secure release options such as PIN or ID-based release, where

students must enter a secure identifier at the printer location to release their print jobs.

24.5.3 Verification:

- Testing of the secure release mechanisms under various scenarios to ensure that print jobs are released only to authorized users.

25. Learning Space Booking

25.1 Reservation System for Booking Spaces:

25.1.1 Requirement:

- Users must be able to book various types of learning spaces through a centralized system.

25.1.2 Specification:

- Development of a web-based booking platform that lists available spaces, such as meeting rooms, lecture halls, and event spaces, allowing users to book spaces according to their needs.

25.1.3 Verification:

- Functionality will be verified through user tests to ensure that the booking process is intuitive and that all types of rooms are bookable.

25.2 Availability Calendar and Scheduling Interface:

25.2.1 Requirement:

- The system should display an availability calendar for all bookable spaces, allowing users to see open time slots.

25.2.2 Specification:

- Integration of a dynamic calendar within the booking system that updates in real-time, showing availability based on existing bookings.

25.2.3 Verification:

- Testing will involve confirming the accuracy of the calendar display against known bookings and ensuring that the system updates immediately after a booking is made.

25.3 Automated Notifications and Reminders:

25.3.1 Requirement:

- Users should receive automated notifications and reminders about their bookings.

25.3.2 Specification:

- Implementation of an automated alert system that sends email and/or SMS notifications when a booking is made, and reminder notifications 24 hours and 1 hour before the scheduled time.

25.3.3 Verification:

- Verification through scheduled simulations to ensure timely and accurate delivery of notifications.

25.4 *Booking Limits:*

25.4.1 Requirement:

- Restrict bookings to a maximum of 2 hours per user per day and limit reservations to 1-hour slots.

25.4.2 Specification:

- System rules that automatically enforce booking restrictions, preventing users from exceeding the daily time allotment or booking outside the 1-hour increments.

25.4.3 Verification:

- Testing booking scenarios where users attempt to exceed their daily limits or book in non-standard increments to confirm system enforcement.

25.5 *Automated Reminder Notifications for End of Booking Period:*

25.5.1 Requirement:

- Users should receive multiple automated reminders leading up to the end of their booked time.

25.5.2 Specification:

- Setup of notifications to be sent 15 minutes and 5 minutes before the end of the booking period. Additionally, an on-screen pop-up should appear at the end of the booking period displaying the name of the next user (if any).

25.5.3 Verification:

- Functional testing to check the accuracy and timing of these reminders and the pop-up against the booking schedule.

26. IT Lab Computer Booking

26.1 ICT Administration Portal:

26.1.1 Requirement:

- An administration portal for the ICT Department to manage the number of ICT labs and available computers in each lab.

26.1.2 Specification:

- Develop an ICT administration portal that allows ICT staff to update and manage lab and computer inventory data.

26.1.3 Verification:

Admin portal functionality will be verified through administrative user tests to ensure that data updates are reflected accurately and in real time.

26.2 Reservation System for Computer Labs and Equipment:

26.2.1 Requirement:

- A system to reserve computer labs and associated equipment by students and faculty.

26.2.2 Specification:

- Implementation of a reservation system integrated into the SMS that supports booking of labs and equipment, displaying available times and specific equipment.

26.2.3 Verification:

- Testing will involve making reservations to ensure the system correctly allocates resources without double booking.

26.3 Real-Time Availability Tracking and Capacity Management:

26.3.1 Requirement:

- The system must accurately track real-time availability and manage the capacity of each computer lab.

26.3.2 Specification:

- Real-time tracking system that updates the availability status of labs and equipment as bookings are made or cancelled.

26.3.3 Verification:

- Functionality will be verified through dynamic testing to ensure that capacity and availability updates are immediate and visible to all users.

26.4 *Integration with User Authentication and Access Control Systems:*

26.4.1 Requirement:

- Ensure that the booking system is integrated with existing user authentication and access control systems to verify user identity and permissions.

26.4.2 Specification:

- Use secure APIs to integrate the reservation system with the university's central authentication service to manage user access based on defined roles.

26.4.3 Verification:

- Integration testing with the authentication system to confirm that only authorized users can make bookings.

26.5 *Integration to Student Portal:*

26.5.1 Requirement:

- Seamless integration of the booking system with the student portal.

26.5.2 Specification:

- Embed the IT lab booking system within the SMS student portal framework to provide a unified access point.

26.5.3 Verification:

- Test computer booking system to ensure that the booking module functions properly within the student portal without technical issues.

26.6 *Availability Calendar and Booking Interface:*

26.6.1 Requirement:

- The interface should include an availability calendar that restricts bookings to operational hours of the IT labs.

26.6.2 Specification:

- A user-friendly interface that displays an interactive calendar, showing available slots during the lab's operational hours, with functionality to prevent bookings during closed hours.

26.6.3 Verification:

- Testing the interface to ensure it only allows bookings during specified hours and accurately displays the available times.

26.7 *Automated Notifications and Reminders:*

26.7.1 Requirement:

- Send automated notifications and reminders about bookings to students' portals and emails.

26.7.2 Specification:

- Set up an automated messaging system that sends booking confirmations, reminders 24 hours before, and 1 hour before the booking starts to student portals and linked email addresses.

26.7.3 Verification:

- Mock booking scenarios to ensure notifications are sent at the correct times and are received both on the student portal and via email.

26.8 Booking in 1 Hour Slots:

26.8.1 Requirement:

- Users should be able to book computer labs in 1-hour slots with a maximum of 2 hours per day.

26.8.2 Specification:

- Booking system rules that limit users to select slots in 1-hour increments and restrict total daily booking time to 2 hours per user.

26.8.3 Verification:

- Simulating booking scenarios to ensure the system enforces the 1-hour increments and daily time limits.

26.9 Automated Reminder Notifications for End of Booking Period:

26.9.1 Requirement:

- Multiple automated reminders should be sent as the booked period is about to end, and a pop-up should display the name of the next user.

26.9.2 Specification:

- Setup of system triggers to send notifications at 15 minutes and 5 minutes before the end of the booking, and an on-screen pop-up at the end of the booking period.

26.9.3 Verification:

- Testing to check the timing and content of end-of-booking notifications and pop-up messages for accuracy and visibility.

27. Student Association & Affiliated Student Groups & Clubs

27.1 Platform for Managing Student Groups:

27.1.1 Requirement:

- The system should provide a platform for managing student sports, student organizations, student clubs, and student associations.

27.1.2 Specification:

- The platform will feature modules for each type of student group, allowing for the creation, management, and overview of groups. Each module will include functionalities for posting updates, managing memberships, scheduling events, and sharing resources.

27.1.3 Verification:

- Functionality will be verified through both unit testing of each module and integration testing to ensure seamless interaction between modules. User acceptance testing will be conducted with actual users to ensure the platform meets their needs.

27.2 Approval Process for Student Groups:

27.2.1 Requirement:

- There must be an approval process for new student sports, student organizations, student clubs, and student associations.

27.2.2 Specification:

- The system will include a workflow module for the approval process. This will allow students to submit applications for new groups, which administrators can review, approve, or reject. The system will track the status of each application and notify users of approval status.

27.2.3 Verification:

- The approval process will be tested by simulating various scenarios, including approvals, rejections, and requests for additional information. The efficiency and effectiveness of the notification system will also be tested.

27.3 *Integration with Student Portal and Election System:*

27.3.1 Requirement:

- Integration is required between the platform for managing student groups and the existing student portal and election system.

27.3.2 Specification:

- Integration points will allow students to access the student group platform directly from the student portal. The election system will be integrated to facilitate the election of officers within the groups. APIs will be used to ensure data consistency across systems.

27.3.3 Verification:

- Integration testing will be conducted to ensure that data flows correctly between the student group and the SMS platform, the student portal, and the election system. End-to-end testing will also verify that users can navigate between systems without issues.

27.4 *Membership Management, Event Planning, and Communication Tools:*

27.4.1 Requirement:

- The platform should include tools for membership management, event planning, and communications for student groups.

27.4.2 Specification:

- Membership management will allow group leaders to add or remove members and

manage roles within the group. Event planning tools will include calendars, scheduling features, and notification systems. Communication tools will consist of forums, messaging systems, and announcement boards.

27.4.3 Verification:

- Each feature will be tested individually to ensure functionality and ease of use. System integration testing will ensure that these tools work together seamlessly, and user feedback will be collected to gauge satisfaction.

27.5 *Funding Allocation and Budget Tracking:*

27.5.1 Requirement:

- The system must include functionality for funding allocation and budget tracking for student groups.

27.5.2 Specification:

- This feature will allow student group administrators to apply for funding, manage allocated funds, and track expenses.
- The system will provide reporting tools to visualize budget status and spending.
- Access controls will ensure that only authorized users can manage and view financial information.

27.5.3 Verification:

- The budget tracking system's accuracy will be tested using simulated financial data. Security testing will ensure that financial information is accessible only to authorized personnel. User tests will assess the usability of financial management tools.

28. Elections & Surveys

28.1 System for Conducting Elections and Surveys:

28.1.1 Requirement:

- The system must provide functionality for conducting both student and staff elections as well as surveys.

28.1.2 Specification:

- The system will include separate modules for elections and surveys. Each module will allow administrators to create, manage, and close elections or surveys. The system will support scheduling, participant management, and real-time monitoring of participation rates.

28.1.3 Verification:

- Functionality will be verified through comprehensive testing of each module, including scenario-based testing for election and survey cycles. User acceptance testing will gather feedback from potential administrators and participants to ensure the system meets their needs.

28.2 Ballot Creation and Voting Systems:

28.2.1 Requirement:

- The system should support ballot creation and different voting types such as first-past-the-post or preferential voting systems, along with result tabulation.

28.2.2 Specification:

- Administrators will have the ability to create ballots and select the voting system for each election. The system will automatically tabulate results based on the chosen voting method. It will include safeguards to ensure accurate counting and prevent tampering.

28.2.3 Verification:

- Testing will involve simulating elections with each voting type to ensure accurate tabulation and reporting. Security testing will verify that ballot integrity is maintained throughout the process.

28.3 Survey Creation and Management:

28.3.1 Requirement:

- The system must allow for survey creation, including various question and answer formats, and facilitate result tabulation.

28.3.2 Specification:

- The survey module will enable the creation of surveys with multiple question types (e.g., multiple-choice, Likert scale, open-ended). It will provide tools for distributing surveys, collecting responses, and tabulating results.

28.3.3 Verification:

- Functionality testing will include the creation of surveys with all supported question types and the collection of responses to ensure accurate data collection and aggregation.

28.4 Secure and Anonymous Voting:

28.4.1 Requirement:

- Support for authenticated one-time but anonymous and secure voting is required.

28.4.2 Specification:

- The voting system will authenticate users to ensure that only eligible participants can vote and that they can only vote once. However, the system will maintain the anonymity of votes to protect voter privacy. Security features will include data encryption and secure access controls.

28.4.3 Verification:

- Security testing will validate authentication mechanisms, check for vulnerabilities in vote anonymity, and ensure that the system prevents multiple submissions by the same user.

28.5 Data Analysis Tools for Surveys and Elections:

28.5.1 Requirement:

- Survey and election results should be supported by comprehensive data analysis tools.

28.5.2 Specification:

- The system will include analytics dashboards that provide visual summaries of election outcomes and survey responses. Tools will allow users to perform custom analyses, such as demographic breakdowns, trends over time, and cross-tabulation.

28.5.3 Verification:

- The effectiveness of data analysis tools will be assessed through tests that verify the accuracy of data presentation and the flexibility of the tools. Feedback from users will also be incorporated to improve usability and functionality.

29. Certificates, Awards, & Document Management

29.1 Repository for Storing and Managing Certificates and Documents:

29.1.1 Requirement:

- The system should provide a repository for storing and managing student certificates, transcripts, and other documents.

29.1.2 Specification:

- The system will include a digital document repository that allows for uploading, storing, and categorizing various types of student documents, including certificates and transcripts.
- Documents will be stored in a secure, searchable database with metadata tagging for easy retrieval.

29.1.3 Verification:

- Functionality will be verified through testing for document upload, retrieval, searchability, and categorization. Performance testing will ensure that the system can handle large volumes of documents without degradation of service.

29.2 Repository for Student Awards and Accolades:

29.2.1 Requirement:

- There must be a repository for storing and managing records of student awards, prizes, and other accolades.

29.2.2 Specification:

- This part of the repository will specifically handle records related to awards and accolades, allowing for entry, storage, and retrieval of information such as award type, award date, and recipient details. The system will provide templates for award

certificates that can be customized and printed.

29.2.3 Verification:

- Tests will be conducted to ensure that award information is accurately recorded and retrievable. The system will also be tested for its ability to customize and print award templates correctly.

29.3 *Secure Access Controls and Versioning:*

29.3.1 Requirement:

- The document management system must feature secure access controls and versioning for documents.

29.3.2 Specification:

- Access to documents will be governed by role-based access controls ensuring that only authorized users can view or edit sensitive documents.
- The system will maintain versions of documents to track changes over time and allow rollback to previous versions if necessary.

29.3.3 Verification:

- Security testing will check access controls under various user scenarios to prevent unauthorized access. Versioning functionality will be tested to ensure accurate version control and retrieval of previous document versions.

29.4 *Monitoring and Change Logs with Notifications:*

29.4.1 Requirement:

- The system should monitor and log changes to documents and provide messaging notifications for printing or when new

documents are created or existing documents are changed.

29.4.2 Specification:

- The system will automatically log all changes made to documents in a change log.
- It will provide real-time notifications to relevant users about key events such as the creation of new documents or changes to existing ones.
- Notifications can be sent via email or system messages.

29.4.3 Verification:

- Change logs and notification systems will be tested to ensure they capture and report all relevant document events accurately and in a timely manner. User feedback will be used to refine notification settings and methods.

29.5 *Integration with Authentication and Verification Systems:*

29.5.1 Requirement:

- The document management system must integrate with existing authentication and verification systems.

29.5.2 Specification:

- Integration with the university's authentication systems will ensure that document access is securely controlled. The system will also integrate with external verification systems to authenticate certificates and transcripts issued by the university.

29.5.3 Verification:

- Integration testing will ensure that authentication workflows are seamlessly connected with document access controls. Tests will also verify that external verification

systems correctly authenticate the documents issued by the system.

30. Student Appointments with Staff System

30.1 Booking System for Appointments:

30.1.1 Requirement:

- The system should provide a booking system for scheduling appointments with academic advisors, counsellors, and support services.

30.1.2 Specification:

- The booking system will allow students to view available times and book appointments online with academic advisors, counsellors, and other support staff.
- It will include features to select the type of service, choose a specific staff member if desired, and see available time slots in real time. Users will be able to book, modify, or cancel appointments through the system interface.

30.1.3 Verification:

- Functionality will be verified through testing the booking process under various scenarios, including booking, modifying, and cancelling appointments. The system's ability to update availability in real-time will also be tested to ensure accuracy.

30.2 Calendar and Timetable Integration:

30.2.1 Requirement:

- The appointment system must integrate with calendars and timetables for managing bookings and staff availability.

30.2.2 Specification:

- Integration with existing university calendar systems will ensure that all appointments are synchronized with individual and university schedules. This feature will allow for automatic blocking of unavailable times based on class schedules, public holidays, or

other commitments recorded in the university calendar.

30.2.3 Verification:

- Integration testing will ensure seamless synchronization between the booking system and the university's calendar and timetable systems. This includes verifying that appointments do not overlap with pre-existing commitments in the calendars of both students and staff.

30.3 *Automated Reminders and Notifications:*

30.3.1 Requirement:

- The system should send automated reminders and notifications for upcoming appointments and deadlines.

30.3.2 Specification:

- Automated reminders will be sent to both students and staff via email and/or SMS notifications prior to scheduled appointments. The system will allow users to set the preferred lead time for reminders (e.g., 24 hours before the appointment). Notifications for impending deadlines related to academic requirements or services will also be managed through this system.

30.3.3 Verification:

- The functionality of reminders and notifications will be tested to ensure timely and accurate delivery. Testing will also include verifying user settings for notification preferences and assessing the system's reliability in sending these communications as scheduled.

31. Internal Staff Directory

31.1 Comprehensive Staff Directory:

31.1.1 Requirement:

- The system must include a directory of faculty and staff members, featuring contact information and details about their organizational roles.

31.1.2 Specification:

- The directory will provide profiles for each staff and faculty member, which include name, department, contact information (phone number, email address), role or title, and a brief bio if available.
- This information will be maintained in a central database, accessible through the system interface.

31.1.3 Verification:

- Functionality will be verified through testing for accuracy and completeness of the information in the directory entries. Additionally, regular database audits will be conducted to ensure the information remains up-to-date and accurate.

31.2 Integration with Organizational Structure:

31.2.1 Requirement:

- The staff directory should be integrated with the organizational structure module of the SMS.

31.2.2 Specification:

- Integration will allow the directory to automatically update based on changes within the organizational structure, such as staff movements, role changes, or new hires. This ensures that the directory reflects the current state of the university's organizational hierarchy.

31.2.3 Verification:

- Integration testing will verify that changes in the organizational structure are accurately and promptly reflected in the staff directory. Scenarios involving updates to the organizational structure (e.g., promotions, department changes) will be specifically tested.

31.3 *Search and Filter Capabilities:*

31.3.1 Requirement:

- The directory should include search and filter capabilities to allow users to easily locate personnel.

31.3.2 Specification:

- Users will be able to search the directory using various criteria, such as name, department, or role.
- Advanced filtering options will also be available to narrow down results based on specific attributes like location, specialization, or recent additions.

31.3.3 Verification:

- Testing will involve multiple scenarios to ensure that search and filter functions return accurate and relevant results quickly and efficiently. User feedback will be gathered to improve usability and functionality of these features.

31.4 *Integration with Messaging and Communication Systems:*

31.4.1 Requirement:

- The staff directory should be integrated with the university's messaging and communication systems.

31.4.2 Specification:

- From the staff directory, users should be able to initiate communications such as emails or instant messages directly to a staff member's contact information. This integration facilitates easy and quick contact without needing to leave the directory interface.

31.4.3 Verification:

- Testing will ensure seamless integration with email and messaging systems, confirming that communications can be initiated directly from the directory entries. Security testing will also verify that only authorized users can access contact information and send messages.

32. Hostel Management

32.1 Management of Student Accommodation:

32.1.1 Requirement:

- The system should manage student accommodation, including hostel creation and naming, room capacities, and a room numbering system.

32.1.2 Specification:

- The system will allow administrators to create and manage hostels by defining hostel names, location details, and the number of rooms available.
- Each room will be uniquely numbered according to a systematic numbering protocol.

32.1.3 Verification:

- The functionality will be verified through the creation, editing, and deletion of hostel records in the system. Testing will include validating the uniqueness and consistency of room numbering across all hostels.

32.2 Room Assignments and Financial Transactions:

32.2.1 Requirement:

- The system should handle room assignments, leases, bond fees, and payments for student accommodation.

32.2.2 Specification:

- Administrators will be able to assign rooms to students and manage the duration of leases. The system will facilitate the collection of bond fees and rent payments, including the setting up of payment schedules and reminders.

32.2.3 Verification:

- Testing will involve simulating the room assignment process and financial

transactions to ensure accuracy in lease management and financial record-keeping. Integration tests will confirm that payment data flows correctly to financial reporting.

32.3 *Maintenance Requests and Facilities Management:*

32.3.1 Requirement:

- There must be a mechanism for managing maintenance requests and other facilities management tasks within hostels.

32.3.2 Specification:

- Students and staff can submit maintenance requests through the system. Each request will be tracked from submission to resolution, including assignment to maintenance staff and status updates. Reporting features will provide overviews of maintenance activities.

32.3.3 Verification:

- Functionality will be tested by submitting various maintenance requests and tracking their progress. User feedback will assess the ease of use and effectiveness of the maintenance management module.

32.4 *Integration with Student Records and Billing Systems:*

32.4.1 Requirement:

- Hostel management must integrate seamlessly with student records and billing systems.

32.4.2 Specification:

- Integration with student records ensures that student accommodation details are linked with their academic and personal records. Billing system integration allows for automatic billing of accommodation fees and bond payments.

32.4.3 Verification:

- Integration testing will ensure that data related to student accommodation is accurately reflected in their overall records and that financial transactions related to accommodation are processed and recorded correctly.

32.5 *Staff and Student Assignments in Hostels:*

32.5.1 Requirement:

- The system should facilitate the assignment of staff wardens, matrons, and student sub-warden/sub-matron positions, including applications and appointments.

32.5.2 Specification:

- The system will manage applications for student sub-warden and sub-matron roles and allow senior hostel staff to review and appoint candidates. It should also manage records of staff wardens and matrons assigned to each hostel.

32.5.3 Verification:

- Testing will include simulating the application process and the appointment workflows to ensure they operate transparently and efficiently. Feedback from staff involved in these processes will help refine the functionalities.

33. Student Dining Management

33.1 Dining Hall Management System:

33.1.1 Requirement:

- The system must provide a dining hall management system for student meal planning and payments.

33.1.2 Specification:

- The dining management module will facilitate the scheduling of meal times, track student meal plan usage, and manage payments for meals.
- This system will support various payment methods, including pre-paid meal plans, pay-per-meal options, and guest meal purchases.

33.1.3 Verification:

- Testing will involve simulations of meal plan selections, meal payments, and tracking usage to ensure the system accurately records transactions and balances. User acceptance testing with cafeteria staff and students will validate functionality and usability.

33.2 Menu Management and Publishing:

33.2.1 Requirement:

- The system should include menu management functionalities, enabling staff to create, manage, and publish dining hall menus.

33.2.2 Specification:

- Administrators will be able to enter, update, and publish menu items for specific meals throughout the day.
- The system will allow for the creation of special dietary menus and will enable menu planning in advance.

- A public-facing interface (e.g., via the student portal) will display the current menu to students.

33.2.3 Verification:

- Functionality will be tested by creating and modifying menus, with subsequent checks to ensure that changes are accurately reflected in the system and correctly published. Feedback from students and staff will assess the clarity and accuracy of published menus.

33.3 *Integration with Student Accounts and Meal Plans:*

33.3.1 Requirement:

- Integration is required with student accounts to manage meal plans effectively.

33.3.2 Specification:

- The system will integrate with student financial accounts, allowing for the management of meal plan subscriptions and deductions per meal consumed.
- It will support the addition of meal credits and monitor meal plan balances.

33.3.3 Verification:

- Integration testing will ensure seamless synchronization between the dining management system and student financial accounts. Tests will include scenarios where meal plans are selected, modified, and used, ensuring accurate reflection in student accounts.

33.4 *Integration with Student Records and Billing Systems:*

33.4.1 Requirement:

- The dining management system must integrate with student records and billing systems.

33.4.2 Specification:

- This integration ensures that charges related to dining services are correctly applied to student bills and that student meal plan information is up-to-date and consistent with student records.

33.4.3 Verification:

- Comprehensive testing will check that financial transactions related to dining services are accurately processed and reflected in the overall billing system. Cross-system checks will confirm that all student record updates are consistent across platforms.

34. Student Administration Services

34.1 Centralized Portal for Student Administration Services:

34.1.1 Requirement:

- The system must provide a centralized portal for accessing a range of student administration services including academic advising, course changes, academic counselling, withdrawal from studies, and graduation applications.

34.1.2 Specification:

- The portal will serve as a one-stop interface where students can access various administrative services. It will allow students to request changes to their academic schedules, apply for counselling, withdraw from courses or the university, and apply for graduation.
- Each service will be accessible via a user-friendly dashboard with clear navigation and instructions.

34.1.3 Verification:

- Functionality will be tested through user scenarios to ensure that each service can be accessed and used as intended. Usability testing with students will help verify that the interface is intuitive and meets their needs.

34.2 Integration with Student Portal:

34.2.1 Requirement:

- The student administration services portal must be integrated with the existing student portal.

34.2.2 Specification:

- Integration will ensure that students can seamlessly access the administration services portal from within the main student

portal. This includes single sign-on capabilities, where students use one set of credentials to access all related systems.

34.2.3 Verification:

- Integration testing will confirm that students can navigate between the student portal and the administration services portal without issues. Testing will also include verification of security measures like data encryption and authentication protocols.

34.3 *Integration with Appointment Booking System:*

34.3.1 Requirement:

- Integration is required with an appointment booking system for services that require scheduling, such as academic advising and counselling.

34.3.2 Specification:

- The administration services portal will link directly to the university's appointment booking system, allowing students to schedule, reschedule, or cancel appointments with advisors, counsellors, and other service providers.
- The system will display available times and allow for automatic confirmation of bookings.

34.3.3 Verification:

- Testing will ensure that the booking process is smooth and functional, with checks on the accuracy of the scheduling data and the reliability of the confirmation and notification systems.

34.4 *Integration with Student Records and Communication Platforms:*

34.4.1 Requirement:

- The student administration services must integrate with student records and communication platforms to ensure coherent data management and effective communication.

34.4.2 Specification:

- The integration will ensure that any changes made via the administration services portal (such as course changes or withdrawal) are reflected in real-time in the student's official records.
- Additionally, the system will use integrated communication tools to send notifications and updates to students about their requests or any required actions.

34.4.3 Verification:

- Cross-system integration tests will check for data consistency across student records after actions are taken in the administration services portal. Communication functionality will be tested to ensure messages are sent and received correctly and securely.

35. Student Welfare & Counselling

35.1 Support for Student Welfare Services:

35.1.1 Requirement:

- The system must support a range of student welfare services, including a campus clinic, mental health services, counselling, sports facilities, and chaplaincy.

35.1.2 Specification:

- The system will provide dedicated modules for each service area, allowing for the management of service offerings, staff, and resources. Each module will feature user interfaces for both students seeking services and staff providing those services.

35.1.3 Verification:

- Functionality will be tested to ensure that each service module operates correctly, providing accurate information and facilitating the management of services. Usability tests with actual users will help ensure the system meets the needs of both students and service providers.

35.2 Case Management and Referral Tracking:

35.2.1 Requirement:

- The system should include capabilities for case management and referral tracking with strong confidentiality and privacy safeguards.

35.2.2 Specification:

- The system will allow service providers to create, manage, and track individual student cases.
- It will include features for noting session details, tracking progress, and managing referrals to other services or specialists.
- Data security measures will be implemented to ensure all information is stored and

transmitted securely, adhering to privacy laws and regulations.

35.2.3 Verification:

- Testing will verify that case management and referral systems function correctly, securely handling sensitive information. Security testing will include data encryption and access controls to ensure that only authorized personnel can access sensitive data.

35.3 *Integration with Appointment System:*

35.3.1 Requirement:

- Student welfare services must be integrated with the university's appointment booking system.

35.3.2 Specification:

- This integration will allow students to book appointments for welfare services directly through the student portal. The appointment system will show real-time availability and allow for immediate booking confirmations.

35.3.3 Verification:

- Integration testing will ensure that the appointment booking system is seamlessly connected to the welfare services modules, with testing focused on reliability, data accuracy, and real-time performance.

35.4 *Resource and Facilities Booking:*

35.4.1 Requirement:

- The system must include functionality for booking resources and facilities associated with student welfare services, such as counselling rooms, sports facilities, and chaplaincy services.

35.4.2 Specification:

- Users will be able to view available facilities and resources and book them according to their needs.
- The system will manage schedules to avoid double bookings and ensure optimal utilization of resources.

35.4.3 Verification:

- Testing will confirm that the booking system accurately reflects resource availability and that it can handle concurrent bookings without conflicts. Feedback from facility managers will be used to refine the booking process.

35.5 *After-Hours Support and Hotline System:*

35.5.1 Requirement:

- There should be provision for after-hours support through a request system and a hotline for urgent needs.

35.5.2 Specification:

- The system will include a feature for students to request after-hours support, which could include a call-back or direct access to an emergency hotline. This feature will route requests to the appropriate service provider based on the nature of the request.

35.5.3 Verification:

- The after-hours support system will be tested to ensure it correctly routes requests and provides timely responses. The effectiveness of the hotline system will be assessed through scenario-based testing to ensure reliability during emergencies.

36. International Students

36.1 *International Student Admissions and Documentation:*

36.1.1 Requirement:

- The system must support international student admissions, including handling visas and immigration documentation.

36.1.2 Specification:

- The system will provide an interface for international students to submit applications, upload necessary documentation (e.g., passport copies, visa applications), and track the status of their admissions and visa processing.
- The system will also integrate with external databases for visa and immigration verification.

36.1.3 Verification:

- Functionality will be verified through testing to ensure accurate submission, storage, and retrieval of documentation. Integration testing with external immigration databases will ensure that data exchanges are secure and comply with data protection regulations.

36.2 *Orientation and Support Services for International Students:*

36.2.1 Requirement:

- The system should offer orientation and support services tailored specifically to international students.

36.2.2 Specification:

- A module will be developed within the system to manage orientation schedules, resources for living in the host country, cultural adaptation tips, and essential services information.

- This module will allow international students to register for orientation events and access resources anytime.

36.2.3 Verification:

- Testing will include simulations of student interactions with the orientation and support services module to assess user experience and information accessibility. Feedback from international students will be used to refine the content and functionality.

36.3 *Compliance with International Regulations:*

36.3.1 Requirement:

- The system must ensure compliance with international student regulations and reporting requirements.

36.3.2 Specification:

- The system will maintain records of all international students in compliance with local and international laws, including enrolment status, course load, and attendance.
- It will generate reports required by government bodies related to international student status and activities.

36.3.3 Verification:

- Compliance will be verified through regular audits of the system's records and reporting functions to ensure all required information is accurate and complete. Test scenarios will include generating reports for hypothetical audits.

36.4 *Academic and Support Services for International Students:*

36.4.1 Requirement:

- There must be support services and academic support systems in place specifically designed for international students.

36.4.2 Specification:

- The system will include links to academic resources, tutoring services, language support, and counselling specifically designed for the needs of international students.
- It will facilitate the creation of support groups and mentorship programs linking international students with peers and faculty.

36.4.3 Verification:

- The effectiveness of academic and support services will be assessed through feedback from international students using these services. Testing will focus on the accessibility of services and the adequacy of support provided.

37. Thesis Management & Publication

37.1 Thesis Submission, Review, and Publication Workflow:

- 37.1.1 Requirement:** The system must facilitate a workflow for thesis submission, review, and publication.
- 37.1.2 Specification:** The workflow will include stages for thesis submission by students, initial administrative checks, assignment to reviewers, review feedback, revisions by the student, final approval, and publication. The system will provide status updates to all parties involved at each stage of the process.
- 37.1.3 Verification:** The workflow functionality will be verified through testing each stage of the process for different user roles (student, administrator, and reviewer). Test scenarios will include submissions, revisions, approvals, rejections, and final publication to ensure the workflow operates smoothly and meets user needs.

37.2 Repository for Theses and Dissertations:

- 37.2.1 Requirement:**
- The system should provide a repository for storing and accessing student theses and dissertations.
- 37.2.2 Specification:**
- This repository will allow students to upload their theses and dissertations in various formats, tag them with metadata for easier retrieval (e.g., author, year, topic, advisor), and store them securely.
 - The repository will be searchable and accessible based on defined access controls.
- 37.2.3 Verification:**
- Testing will include uploading, tagging, searching, and retrieving theses and

dissertations to ensure all functions work as intended. Security testing will verify that access controls are enforced properly.

37.3 *Digital Preservation and Access Controls:*

37.3.1 Requirement:

- Support must be provided for digital preservation of theses and digital access controls.

37.3.2 Specification:

- The system will implement digital preservation standards to ensure the long-term accessibility and integrity of theses.
- Access controls will allow students to set permissions on who can view or download their work, with options ranging from open access to restricted access for selected users.

37.3.3 Verification:

- Digital preservation capabilities will be tested by simulating long-term storage and retrieval.
- Access control testing will involve scenarios with different user permissions to verify that access restrictions are upheld.

37.4 *Integration with Library System:*

37.4.1 Requirement:

- The thesis management system must integrate with the existing library system.

37.4.2 Specification:

- Integration with the library Koha system will ensure that theses and dissertations are catalogued in the university's broader library catalogue.
- This will include linking data entries between the systems and ensuring that search

functions in the library catalogue can retrieve thesis information.

37.4.3 Verification:

- Integration testing will check for data consistency and seamless operation between the thesis management system and the library system. This will include verifying that updates in the thesis system are reflected in the library catalogue and vice versa.

38. University News and Announcements

38.1 Platform for Publishing News and Announcements

38.1.1 Requirement:

- The system must provide a platform for publishing news, announcements, and updates relevant to university activities.

38.1.2 Specification:

- The platform will include a content management system (CMS) that allows authorized administrators to create, edit, publish, and archive news articles and announcements.
- The CMS will support multimedia content, such as images and videos, and provide templates for standardizing the presentation of information.

38.1.3 Verification:

- Functionality will be tested to ensure that the CMS allows for smooth creation and management of content.
- The system's ability to handle multimedia uploads and display them correctly in articles will also be tested. User acceptance testing will gather feedback from administrators on ease of use.

38.2 Integration into Staff and Student Portals:

38.2.1 Requirement:

- News, announcements, and updates should be integrated into both the staff and student portals.

38.2.2 Specification:

- A feed of the latest news and announcements will be displayed prominently on the dashboard of both staff and student portals.

- Users will have the ability to customize their feed to show the most relevant news to their interests and roles.

38.2.3 Verification:

- Integration testing will confirm that news updates are correctly and promptly displayed in both portals.
- Tests will also ensure that customization features function correctly and user settings are saved and applied consistently.

38.3 *Integration with University Website:*

38.3.1 Requirement:

- News, announcements, and updates must be integrated into the university website via an API.

38.3.2 Specification:

- An API will be developed to push selected news and announcements from the internal CMS to the university's public website. This API will allow for real-time updates and ensure that public-facing content is consistent with internal communications.

38.3.3 Verification:

- The API's functionality will be verified by checking for timely and accurate updates on the university website following changes in the CMS. Security testing will ensure that the API is resistant to unauthorized access and data breaches.

38.4 *Content Management System for Administrators:*

38.4.1 Requirement:

- Administrators need a content management system to create and manage articles.

38.4.2 Specification:

- The CMS will provide tools for creating articles, including text editors, image and video uploaders, and features for previewing articles before publication.
- It will also include workflow features for reviewing and approving content before it goes live, and archiving features for managing old posts.

38.4.3 Verification:

- Testing will involve creating, editing, and publishing articles to ensure the CMS tools are functional and user-friendly.
- Workflow tests will simulate the approval process, and archival functionality will be tested to confirm it effectively manages content lifecycle.

39. Student Grievance Management

39.1 System for Reporting and Managing Grievances:

39.1.1 Requirement:

- The system must provide a mechanism for students to report grievances and complaints from their student portal.

39.1.2 Specification:

- The grievance management system will include a user-friendly interface for students to submit grievances, categorize them (e.g., academic, administrative, harassment), and describe the issues in detail.
- Administrators will have access to a dashboard to review grievances, assign them to appropriate staff for resolution, and communicate with complainants.

39.1.3 Verification:

- The system's functionality will be tested to ensure ease of use in reporting grievances and efficiency in managing them through the administrative dashboard.
- User acceptance testing will involve students and administrative staff to ensure the system meets their needs.

39.2 Escalation Processes and Resolution Tracking:

39.2.1 Requirement:

- There must be defined escalation processes and mechanisms for tracking the resolution of grievances.

39.2.2 Specification:

- The system will automate the escalation of unresolved grievances based on predefined rules (e.g., time thresholds or severity levels).
- Each grievance will have a status indicator to track its progress through resolution stages,

and detailed logs will be maintained to record actions taken on each complaint.

39.2.3 Verification:

- Testing will simulate various scenarios to ensure that escalation processes trigger correctly and that the resolution tracking provides clear, timely updates.
- Compliance with institutional policies will also be verified.

39.3 *Anonymous Reporting Options and Confidentiality Safeguards:*

39.3.1 Requirement:

- The system should allow options for anonymous reporting of grievances and ensure confidentiality safeguards are in place.

39.3.2 Specification:

- A feature for anonymous submissions will be available, allowing students to report issues without revealing their identity.
- The system will enforce strict access controls and data encryption to protect the confidentiality of grievance details and the identities of all parties involved.

39.3.3 Verification:

- The functionality for anonymous reporting will be tested to ensure that it conceals the identity of the reporter effectively.
- Security testing will verify that all grievance data is adequately protected against unauthorized access.

40. Student Discipline Management

40.1 System for Managing Student Disciplinary Matters:

40.1.1 Requirement:

- The system must provide a robust mechanism for reporting and managing student disciplinary matters.

40.1.2 Specification:

- The disciplinary management system will include functionalities for staff and students to report incidents, categorize them based on severity and type (e.g., academic dishonesty, behavioural issues), and detail the incidents.
- Administrative staff will have access to a management dashboard to review cases, conduct investigations, and record resolutions.

40.1.3 Verification:

- Testing will involve scenarios to ensure the system can handle report submissions, categorization, and case management effectively.
- The system's usability will be evaluated through user acceptance testing with administrative staff responsible for disciplinary actions.

40.2 Link to Student Profile and Record:

40.2.1 Requirement:

- Disciplinary actions and reports must be linked to the student's profile and academic record.

40.2.2 Specification:

- Any disciplinary action or incident report will automatically update the student's profile and record within the system.
- This link will ensure that all relevant student information is accessible from the

disciplinary module for review and decision-making.

40.2.3 Verification:

- Integration testing will confirm that disciplinary updates are reflected accurately in student profiles and records.
- Consistency checks will ensure no data mismatches or errors occur between the disciplinary module and the main student database.

40.3 *Integration with Student Portal:*

40.3.1 Requirement:

- The disciplinary management system should be integrated into the student portal.

40.3.2 Specification:

- Students will have access to their disciplinary status and case history via the student portal.
- This integration will provide transparency and allow students to track the status of any disciplinary processes involving them, submit documentation, or appeal decisions.

40.3.3 Verification:

- Testing will ensure that students can view and interact with their disciplinary information securely and accurately within the student portal.
- Functionality tests will cover login, information retrieval, and data privacy.

40.4 *Escalation Processes and Resolution Tracking:*

40.4.1 Requirement:

- There must be defined escalation processes and mechanisms for tracking the resolution of disciplinary cases.

40.4.2 Specification:

- The system will automate the escalation of cases that are not resolved within predetermined timeframes or require higher authority intervention.
- Each case will feature status updates to track its progress from report to resolution, including any appeals.

40.4.3 Verification:

- Scenario testing will simulate various cases to verify that escalation processes are functioning as intended and that the system provides timely updates on case status.

40.5 *Anonymous Reporting and Confidentiality Safeguards:*

40.5.1 Requirement:

- The system should allow for anonymous reporting of disciplinary issues and ensure strict confidentiality safeguards.

40.5.2 Specification:

- An anonymous reporting feature will enable students or staff to report issues without disclosing their identity.
- The system will implement strong access controls, data encryption, and privacy protocols to protect the confidentiality of all parties involved in disciplinary matters.

40.5.3 Verification:

- Security testing will assess the effectiveness of anonymous reporting functionalities and confirm that all disciplinary information is securely stored and accessed only by authorized personnel.

41. University Events

41.1 Calendar of University Events:

41.1.1 Requirement:

- The system must include a comprehensive calendar that lists all university events, encompassing academic, social, and cultural activities.

41.1.2 Specification:

- The events calendar will provide a centralized platform where event details such as date, time, location, and description are displayed.
- It should allow for different views (e.g., daily, weekly, monthly) and include filters by event type or category.
- The calendar should be updateable by authorized personnel only.

41.1.3 Verification:

- Testing will ensure the calendar displays all entered events accurately and allows for different viewing options and filters.
- Changes to event details must reflect immediately and correctly across all user views.

41.2 Event Registration and Ticketing System:

41.2.1 Requirement:

- The system should facilitate event registration and manage ticketing for events that require it.

41.2.2 Specification:

- An integrated registration and ticketing module will handle online registrations, issue digital tickets, and manage attendee lists.
- It should support settings for event capacity, waitlists, and different ticket types (e.g., student, staff, and general public).

41.2.3 Verification:

- Functionality testing will include scenarios of registering for an event, ticket issuance, and waitlist management to ensure the system handles these processes efficiently.
- Load testing may also be conducted to ensure the system can handle high demand for popular events.

41.3 *Promotion and Marketing Tools for Event Organizers:*

41.3.1 Requirement:

- The system must provide tools for event organizers to promote and market their events.

41.3.2 Specification:

- The promotion tools will include options to create and distribute marketing materials such as emails, social media posts, and digital flyers within the system.
- Organizers should be able to target specific groups within the university, such as departments or interest groups.

41.3.3 Verification:

- Tests will confirm that promotional materials can be created, scheduled, and sent to the intended audiences.
- Feedback from organizers will assess usability and effectiveness of these marketing tools.

41.4 *Integration into Messaging and Notification System*

41.4.1 Requirement:

- Integration is required with the messaging and notification systems of the student and staff portals.

41.4.2 Specification:

- The events system will integrate seamlessly with the existing messaging and notification systems to alert students and staff about new events, changes to scheduled events, and reminders.
- Users should be able to customize their notification preferences.

41.4.3 Verification:

- Integration testing will ensure that notifications about events are properly pushed through the messaging system to users based on their preferences and actions.
- Tests will include verifying that notifications are timely and accurate.

41.5 *Integration with University Calendar:*

41.5.1 Requirement:

- Events should be integrated with the university's overall calendar system.

41.5.2 Specification:

- Event information from the dedicated events calendar will automatically sync with the university's main calendar system, ensuring that all users have access to up-to-date event information across all calendar platforms.

41.5.3 Verification:

- Integration testing will confirm that updates to the events calendar are correctly and promptly reflected in the university's main calendar. Tests will also check for conflicts and proper handling of overlapping events.

42. Alumni Platform

42.1 Alumni Engagement Platform:

42.1.1 Requirement:

- The system must provide a platform for engaging with alumni and maintaining comprehensive alumni records.

42.1.2 Specification:

- The platform will facilitate the creation and management of alumni profiles, including educational history, contact information, and current employment.
- It will support communication tools such as newsletters, announcements, and personalized messages to keep alumni engaged.

42.1.3 Verification:

- Functionality testing will verify that the platform supports profile management, data accuracy, and effective communication tools.
- Usability testing with actual users will ensure the platform meets their needs for engagement and information sharing.

42.2 Integration with Social Media Platforms:

42.2.1 Requirement:

- The alumni platform should integrate with various social media platforms to enhance engagement.

42.2.2 Specification:

- Integration features will allow content from the alumni platform to be shared directly to social media platforms such as Facebook, LinkedIn, and Twitter.
- It will also enable pulling content from these platforms to aggregate in the alumni news feed.

42.2.3 Verification:

- Testing will ensure seamless integration with social media for content sharing and aggregation. Security testing will verify that only authorized actions are possible through this integration to protect user data.

42.3 Alumni Profile and Contact Directory:

42.3.1 Requirement:

- There must be a comprehensive alumni profile and contact directory.

42.3.2 Specification:

- The directory will allow alumni to update their profiles with contact information, career updates, and other relevant details.
- Alumni will have control over privacy settings to manage what information is visible to other alumni or university staff.

42.3.3 Verification:

- Testing will confirm the functionality for profile updates, search capabilities, and privacy settings. The directory's performance in handling large volumes of data will also be tested.

42.4 Alumni Events, Fundraising Campaigns, and Involvement Opportunities:

42.4.1 Requirement:

- The platform should support the management of alumni events, fundraising campaigns, and other involvement opportunities.

42.4.2 Specification:

- Tools will be provided to create, manage, and promote alumni events and fundraising campaigns.

- This will include features for event registration, donation processing, and tracking engagement.
- Opportunities for alumni to volunteer or participate in university initiatives will also be managed through the platform.

42.4.3 Verification:

- Testing will involve creating events and campaigns to ensure these functionalities work as intended and are user-friendly.
- Transaction security and data integrity tests will ensure that all financial transactions are processed securely.

43. Messaging & Notifications

43.1 Communication Platform:

43.1.1 Requirement:

- The system must provide a communication platform for sending messages and notifications to students, faculty, and staff.

43.1.2 Specification:

- The communication platform will support the creation, sending, and management of messages and notifications.
- It will allow for both broad and targeted communications, facilitating different types of messages such as announcements, reminders, and urgent alerts.

43.1.3 Verification:

- Testing will ensure the platform can send and manage messages effectively, with functionality tests for composing, sending, and tracking messages. Load testing will verify system performance under heavy usage.

43.2 Integration with Student and Staff Portals:

43.2.1 Requirement:

- The messaging system should be integrated with the student and staff portals.

43.2.2 Specification:

- Integration will ensure that messages and notifications are accessible directly from the student and staff portals.
- Users will receive real-time notifications within the portal and have a dedicated area to view and manage their messages.

43.2.3 Verification:

- Integration testing will confirm that messages are properly displayed and managed within the portals. Usability testing with end users

will ensure that the integration is intuitive and meets their communication needs.

43.3 *University-Wide Messaging Access Control:*

43.3.1 Requirement:

- The system must support university-wide messaging, with this functionality limited to selected staff only.

43.3.2 Specification:

- Access controls will be implemented to restrict university-wide messaging capabilities to designated staff members. These controls will ensure that only authorized personnel can send broad communications, thereby preventing misuse.

43.3.3 Verification:

- Security testing will verify that access controls are effective and that unauthorized users cannot send university-wide messages. Scenario testing will also be used to confirm that authorized staff can easily send such messages when needed.

43.4 *Support for Multiple Notification Channels:*

43.4.1 Requirement:

- The system should support multiple notification channels, including email and in-app notifications.

43.4.2 Specification:

- Users will be able to select their preferred methods of communication, and the system will deliver messages accordingly.
- The system will ensure reliable delivery of notifications across all supported channels.

43.4.3 Verification:

- Functionality testing will assess the reliability and timeliness of message delivery on each

channel. Cross-channel consistency checks will ensure that messages appear the same across all platforms.

43.5 *Prioritization and Flagging Options:*

43.5.1 Requirement:

- Messaging functionality should include prioritization and flagging options, with messages highlighted as read and unread.

43.5.2 Specification:

- The messaging interface will allow users to flag important messages and prioritize them in their inbox.
- The system will automatically mark messages as read or unread based on user interaction, and users will have the option to manually mark messages as read or unread.

43.5.3 Verification:

- Testing will verify that prioritization and flagging functions are user-friendly and effective.
- Usability testing will assess how intuitive and helpful these features are for managing communications.

44. Analytics and Reports

44.1 Reporting and Analytics Dashboard:

44.1.1 Requirement:

- The system must provide a reporting and analytics dashboard for monitoring key performance indicators (KPIs) relevant to university operations.

44.1.2 Specification:

- The dashboard will display real-time data on various KPIs such as enrolment figures, faculty performance, student achievement, and financial health.
- It will feature an intuitive interface that allows users to quickly assess critical metrics.

44.1.3 Verification:

- Testing will ensure that the dashboard updates accurately in real-time and displays data correctly.
- Usability testing will verify that the dashboard is intuitive and meets the needs of its users for monitoring and decision-making.

44.2 Pre-built and Customizable Reports:

44.2.1 Requirement:

- The system should offer both pre-built and customizable reports covering enrolment, academic staff, students in faculty and courses, financial, and operational data.

44.2.2 Specification:

- Users will have access to a library of pre-built reports that can be used as-is or customized as needed.
- The system will allow users to select specific data fields, apply filters, and save custom report configurations for future use.

44.2.3 Verification:

- Functionality testing will assess the ease of accessing and customizing reports.
- Testing will also ensure that reports are accurate and reflect the latest data, with performance testing to assess response times for generating reports.

44.3 Data Visualization Tools:

44.3.1 Requirement:

- Data visualization tools are required for trend analysis and decision-making.

44.3.2 Specification:

- The system will include tools to create visual representations of data, such as graphs, charts, and heat maps.
- These tools will be capable of illustrating trends over time, comparisons, and correlations among various data sets.

44.3.3 Verification:

- Testing will verify that visualization tools are effective in representing different types of data accurately.
- User feedback will be used to refine visualization options and ensure they provide meaningful insights.

44.4 Report Generation and Download Capabilities:

44.4.1 Requirement:

- Functionality to generate and download reports, including visual graphs and artefacts, is necessary.

44.4.2 Specification:

- Users will be able to generate reports with the option to include graphical data representations. The system will provide options to download reports in various formats, such as PDF, Excel, or PowerPoint,

ensuring that reports are presentation-ready and can be easily shared.

44.4.3 Verification:

- Testing will include generating reports with various data sets and downloading them in all supported formats.
- The integrity of the downloaded files and the accuracy of the information they contain will be thoroughly tested.

45. Staff Portals

45.1 *Role-Based Access Control for Staff Portal:*

45.1.1 Requirement:

- The system must support role-based access control for staff user accounts to assign appropriate access levels based on staff roles.

45.1.2 Specification:

- The admin dashboard will facilitate the setup of staff user accounts and assignment of roles. These roles include:
 - *Vice Chancellor & Pro-Vice Chancellor Academic Portal*
 - *Dean of Faculty Portal*
 - *Head of School Portal*
 - *Head of Department Portal*
 - *Course Lecturer Portal*
 - *Tutor Portal*
 - *Non-Academic Staff Portal*
- Each role will have access tailored to the needs and responsibilities associated with that position, ensuring that staff can access only the information and functionalities pertinent to their role.

45.1.3 Verification:

- Testing will involve verifying that each role-based account has appropriate access as defined.
- Security testing will ensure that access restrictions are enforced properly, preventing unauthorized access to sensitive information.

45.2 *Detailed Requirements for Each Role:*

45.2.1 *Vice Chancellor & Pro-Vice Chancellor Academic Portal*

Specification:

- High-level access to strategic and operational data across the university, including financial reports, academic performance, and faculty statistics.

Verification:

- Ensure that high-level data and reports are accessible and that the data integrity is maintained.

45.2.2 Dean of Faculty Portal

Specification:

- Access to faculty-specific academic performance, faculty staff records, and student progression statistics.

Verification:

- Validate access to detailed faculty reports and the ability to manage faculty-specific content.

45.2.3 Head of School Portal

Specification:

- Access to school-level data, including course performance, enrolment statistics, and lecturer performance within the school.

Verification:

- Confirm access to school-specific analytics and administrative functionalities.

45.2.4 Head of Department Portal

Specification:

- Department-specific access to manage and view departmental course details, staff performance, and student academic records.

Verification:

- Ensure department heads can efficiently manage departmental resources and data.

45.2.5 Course Lecturer Portal

Specification:

- Access to manage course content, view student enrolments, set and mark assignments, and communicate with enrolled students.

Verification:

- Test that lecturers can perform all course-related tasks without accessing other course data.

45.2.6 Tutor Portal

Specification:

- Access to specific course sections to manage tutoring sessions, track student attendance, and assist with grading.

Verification:

- Verify tutors have access only to necessary functionalities to support their tutoring roles.

45.2.7 Non-Academic Staff Portal

Specification:

- Access to non-academic modules such as human resources, finance, and facilities management, depending on the specific non-academic role.

Verification:

- Ensure non-academic staff access is restricted to relevant operational functionalities.

46. Academic Staff Management

46.1 Human Resources Management for Academic Staff:

46.1.1 Requirement:

- The system must include a human resources management system for managing academic staff profiles, contracts, and academic responsibilities.

46.1.2 Specification:

- The HR module will maintain comprehensive profiles that include personal information, academic qualifications, contract details, and specific academic responsibilities.
- It will provide functionalities for updating profiles and contract terms as necessary.

46.1.3 Verification:

- Testing will verify the accuracy and security of the data entry and retrieval processes for staff profiles and contracts.
- Additionally, user acceptance testing will ensure that the system meets the needs of HR administrators.

46.2 Assignment of Academic Staff:

46.2.1 Requirement:

- The system should provide a platform for assigning academic staff to Faculties, Schools, Departments, and courses.

46.2.2 Specification:

- This feature will enable administrators to assign staff to various academic units and specific courses based on their expertise and contractual obligations.
- The system will track these assignments and allow for adjustments and reassignments as needed.

46.2.3 Verification:

- Testing will ensure that staff assignments are correctly reflected in the system and can be easily updated.
- Integration testing will verify that these assignments are accurately displayed across all relevant modules of the system.

46.3 *Leave, Absenteeism, and Cover Management:*

46.3.1 Requirement:

- The system must manage leave, absenteeism, and cover for academic staff.

46.3.2 Specification:

- The module will allow staff to submit leave requests, which can be approved or denied by supervisors.
- The system will track absenteeism and facilitate the arrangement of cover for absent staff members.

46.3.3 Verification:

- Functionality testing will confirm that leave management processes operate smoothly and that cover management is effectively coordinated.
- Scenario testing will include various types of leave scenarios to ensure system flexibility and reliability.

46.4 *Academic Performance Evaluation and Professional Development:*

46.4.1 Requirement:

- The system should support the evaluation of academic performance and track professional development activities.

46.4.2 Specification:

- This feature will include tools for performance reviews, feedback collection, and professional development tracking.

- It will allow both staff and supervisors to access performance reports and plan development activities.

46.4.3 Verification:

- Testing will involve verifying that performance evaluations can be conducted and recorded accurately, and that professional development activities are tracked and reported in the system.
- Feedback from academic staff will be used to refine these features.

46.5 *Assignment to Centers, Institutes, and Research Projects:*

46.5.1 Requirement:

- The platform should enable the assignment of academic staff to Centers, Institutes, and Research Projects.

46.5.2 Specification:

- Administrators will be able to assign staff to various research and development projects or institutional centers based on their skills and research interests.
- This functionality will support the management of project teams and facilitate resource allocation.

46.5.3 Verification:

- Testing will check that staff can be assigned to projects and centers as required, and that these assignments are managed efficiently within the system. This includes verifying that project assignments do not conflict with other academic duties.

47. Distance Flexible Education Management

47.1 Online Learning Management System (LMS):

47.1.1 Requirement:

- The system must include an Online Learning Management System platform for delivering distance and flexible education programs to remote university centers.

47.1.2 Specification:

- The LMS will support the creation, management, and delivery of online courses, incorporating multimedia content such as videos, presentations, and interactive learning tools.
- It will be accessible from multiple devices to accommodate the needs of remote learners.

47.1.3 Verification:

- Testing will ensure that the LMS operates smoothly across different devices and internet connectivity conditions.
- Usability testing with remote learners will verify that the platform is user-friendly and meets educational needs.

47.2 Support for Online Courses and Virtual Classrooms:

47.2.1 Requirement:

- The platform should support online courses, virtual classrooms, and multimedia content delivery.

47.2.2 Specification:

- Virtual classroom functionalities will include live video streaming, interactive whiteboards, and real-time communication tools such as chat and forums.
- The platform will be able to host and stream multimedia content efficiently.

47.2.3 Verification:

- Functionality tests will assess the reliability and quality of video streams and interactive tools. Performance testing will evaluate the system's capability to handle simultaneous connections without degradation of service.

47.3 *Timed Exam Delivery:*

47.3.1 Requirement:

- The system must support the delivery of timed exams.

47.3.2 Specification:

- The LMS will facilitate the creation and administration of timed exams, with features to control exam durations and ensure that exams are only accessible during specified times.

47.3.3 Verification:

- Testing will involve simulating timed exam scenarios to ensure that time limits are enforced accurately and that exams cannot be accessed outside the scheduled times.

47.4 *Assignment Delivery and Submission:*

47.4.1 Requirement:

- The platform should support assignment delivery and allow students to upload assignments in various document formats.

47.4.2 Specification:

- Instructors will be able to distribute assignments through the platform.
- Students will be able to submit completed assignments in formats such as DOCX, PDF, and multimedia files.
- The system will confirm submissions and provide a digital receipt to students.

47.4.3 Verification:

- Testing will confirm that all supported file types can be uploaded and downloaded correctly, and that submission confirmations are reliably generated.

47.5 Assessed Assessment Feedback:

47.5.1 Requirement:

- The system must support the provision of assessed assessment feedback to students.

47.5.2 Specification:

- Instructors will be able to provide detailed feedback on student submissions directly through the platform. This feedback can include text comments, attached files, or even multimedia feedback.

47.5.3 Verification:

- Functionality testing will ensure that feedback mechanisms are user-friendly and that feedback is accurately recorded and communicated to students.

47.6 Monitoring and Attendance Support Tools:

47.6.1 Requirement:

- Monitoring and attendance tools are needed for remote learners and instructors.

47.6.2 Specification:

- The platform will include tools to monitor student engagement and track attendance in virtual classroom sessions.
- Tools will generate reports on attendance and participation, which can be accessed by instructors and administrators.

47.6.3 Verification:

- Testing will verify that monitoring tools accurately track and report attendance and engagement, with additional testing to

ensure that data privacy and security are maintained.

48. University Bridging Programs

48.1 *Management and Delivery Platform for Bridging Courses:*

48.1.1 Requirement:

- The system must provide a platform to manage and deliver pre-university bridging courses to students.

48.1.2 Specification:

- This platform will facilitate the creation, organization, and delivery of bridging course content, including lectures, readings, and other educational resources.
- The platform will be user-friendly to accommodate students new to university-level studies.

48.1.3 Verification:

- Testing will verify that the platform can efficiently manage and deliver course content, ensuring that resources are easily accessible and navigable by students.
- Usability testing will involve prospective students to ensure the interface is intuitive.

48.2 *Curriculum Design and Delivery Tools:*

48.2.1 Requirement:

- The system should include tools for curriculum design and delivery.

48.2.2 Specification:

- Tools will allow educators to design course curriculums, including the ability to set learning objectives, course outlines, and schedules.
- The system will support multimedia content, interactive tools, and other dynamic educational resources to enhance learning.

48.2.3 Verification:

- Testing will ensure these tools facilitate effective curriculum design and that changes to the curriculum can be implemented quickly and reflected accurately across the platform.

48.3 *Assessment and Progress Tracking Features:*

48.3.1 Requirement:

- The platform must include features for assessment and progress tracking.

48.3.2 Specification:

- The system will provide functionalities for creating and administering quizzes, assignments, and exams.
- It will track student progress through these assessments and generate reports on individual and cohort performance, identifying areas where students need additional support.

48.3.3 Verification:

- Testing will check that assessments are correctly administered and scored, and that progress tracking accurately reflects student performance and engagement.
- Testing will also involve validating the reliability of the reporting functions.

48.4 *Integration with Distance Flexible Education Management Platform and LMS:*

48.4.1 Requirement:

- Integration is required with the Distance Flexible Education Management Platform and Learning Management System.

48.4.2 Specification:

- The bridging course platform will seamlessly integrate with the university's existing Distance Flexible Education Management Platform and LMS, ensuring that bridging

course students have access to the same tools and resources as other students.

- This includes access to virtual classrooms, resource libraries, and student support services.

48.4.3 Verification:

- Integration testing will confirm that the bridging courses function as a component of the larger educational systems, with particular attention to data consistency and user experience continuity across platforms.

49. Learning Space Management

49.1 *Management and Assignment Platform for Learning Spaces:*

49.1.1 Requirement:

- The system must provide a platform to manage and assign university learning spaces including classrooms, lecture theatres, laboratories, and computer labs to courses.

49.1.2 Specification:

- The platform will allow administrators to allocate and schedule learning spaces based on course requirements, student enrolment numbers, and specific needs of the courses (e.g., special equipment in laboratories). It will include a calendar view for easy visualization of space utilization.

49.1.3 Verification:

- Functionality testing will ensure the platform accurately assigns and schedules learning spaces without conflicts. Testing will also involve stress testing the system to handle changes in course schedules and room assignments efficiently.

49.2 *Monitoring and Management of Learning Space Maintenance:*

49.2.1 Requirement:

- The system should monitor and manage maintenance issues related to learning spaces, including the management of furniture.

49.2.2 Specification:

- A module within the platform will track maintenance requests and issues related to each learning space, such as broken furniture or equipment malfunctions. It will

allow staff to report issues, track the status of repairs, and schedule regular maintenance checks.

49.2.3 Verification:

- Testing will verify that maintenance issues can be reported, tracked, and resolved within the system.
- This will include checking that notifications are sent to maintenance staff and updates are reflected in real-time.

49.3 *Integration with Messaging App:*

49.3.1 Requirement:

- The management platform must integrate with a messaging app to facilitate communication related to learning space assignments and maintenance.

49.3.2 Specification:

- Integration with a messaging app will enable direct communication between course coordinators, facility managers, and maintenance staff.
- This integration will support notifications about room assignments, changes, and maintenance updates.

49.3.3 Verification:

- Integration testing will ensure that messages related to learning space management are promptly sent and received across the system. Functionality tests will also assess the reliability and timeliness of notifications.

50. TVET Competency Based Education Management

50.1 Management and Delivery Platform for TVET Training Courses:

50.1.1 Requirement:

- The system must provide a platform to manage and deliver TVET (Technical and Vocational Education and Training) training courses to students.

50.1.2 Specification:

- This platform will facilitate the setup, administration, and delivery of TVET courses, specifically designed to handle the unique structure of competency-based training.
- It will support modular training delivered in 10-week blocks, allowing for flexible entry and exit points for students.

50.1.3 Verification:

- Testing will ensure that the platform can create, schedule, and manage courses in 10-week blocks.
- The functionality will be verified through user testing with administrators and educators to ensure it meets the logistical needs of competency-based training.

50.2 Competency-Based Training Delivery:

50.2.1 Requirement:

- The platform should deliver competency-based training, focusing on practical skills and assessments.

50.2.2 Specification:

- The system will support the delivery of training that assesses students based on their ability to perform specific competencies. It will include tools to create and manage

competency checklists and practical assessments.

50.2.3 Verification:

- Functionality testing will verify that the system can handle the creation and tracking of competency-based assessments.
- This includes ensuring that assessments are correctly linked to specific competencies and that student progress can be accurately monitored.

50.3 *Curriculum Design and Delivery Tools:*

50.3.1 Requirement: Curriculum design and delivery tools are required for TVET courses.

50.3.2 Specification: Tools within the system will allow for the creation and modification of curriculum that is tailored to competency-based education. These tools will support multimedia content, interactive simulations, and other resources suited for vocational training.

50.3.3 Verification: Testing will assess the effectiveness and usability of curriculum design tools, ensuring they are suitable for creating vocational training materials that meet educational standards.

50.4 *Assessment and Progress Tracking Features:*

50.4.1 Requirement:

- The platform must include robust assessment and progress tracking features.

50.4.2 Specification:

- The system will feature tools to create, administer, and grade assessments aligned with TVET competencies.
- Progress tracking functionalities will provide detailed reports on student achievements and areas needing improvement, accessible to both students and educators.

50.4.3 Verification:

- Testing will involve creating and grading assessments and tracking student progress through the system. This will include verifying the accuracy of progress reports and the system's ability to update and reflect student performance in real-time.

50.5 *Integration with Learning Management System:*

50.5.1 Requirement:

- Integration is required with the existing Learning Management System (LMS).

50.5.2 Specification:

- The TVET management platform will integrate seamlessly with the university's LMS to ensure that all learning resources, student data, and course materials are synchronized and accessible through a single system.
- This integration will support a unified approach to course management and student learning experiences.

50.5.3 Verification:

- Integration testing will confirm that data flows correctly between the TVET platform and the LMS, ensuring that course information, student records, and assessment data are consistent and up-to-date across systems.

51. Student Academic Records

51.1 Secured and Scalable Database Storage:

51.1.1 Requirement:

- The system must provide highly secured and scalable database storage for storing all academic records of students.

51.1.2 Specification:

- The database will be designed to handle large volumes of data securely and efficiently.
- It must be capable of scaling to accommodate growing numbers of student records over time.
- Security measures will include data encryption, regular security audits, and compliance with relevant data protection regulations.

51.1.3 Verification:

- Performance and security testing will verify that the database can handle the expected load and safeguard data privacy.
- Scalability tests will assess the database's ability to grow without performance degradation.

51.2 Immutability of Records:

51.2.1 Requirement:

- Academic records must be immutable once stored after graduation, with the system keeping logs of all access and changes.

51.2.2 Specification:

- Once a student graduates, their academic record will be locked for editing and marked as immutable in the system.
- The system will maintain detailed logs of all access and any changes made before the

record is locked, including the identity of the user and the timestamp of the action.

51.2.3 Verification:

- Testing will ensure that records become immutable upon graduation. Audit trail functionalities will be tested to verify that all accesses and modifications are logged accurately and are tamper-proof.

51.3 *Retrieval of Student Academic Records:*

51.3.1 Requirement:

- Student academic records must be retrievable based on student requests.

51.3.2 Specification:

- Students will have the ability to request copies of their academic records through the system.
- This feature will include identity verification processes to ensure that records are only provided to the rightful owner or authorized parties.

51.3.3 Verification:

- Functionality testing will check the process of requesting and retrieving academic records, ensuring that it is straightforward yet secure. Tests will also verify that unauthorized access attempts are correctly denied.

51.4 *Integration with Messaging App:*

51.4.1 Requirement:

- Integrate with a messaging app for real-time stamped notifications of any access and changes to records.

51.4.2 Specification:

- The system will integrate with a university-approved messaging app to send real-time notifications to students and relevant staff

when their academic records are accessed or modified. Notifications will include a timestamp and details of the action taken.

51.4.3 Verification:

- Integration testing will confirm that notifications are sent in real-time and contain accurate information. Testing will also assess the reliability and security of the notification system.

51.5 *Role-Based Access:*

51.5.1 Requirement:

- Access to student academic records must be restricted to approved staff based on their roles.

51.5.2 Specification:

- Role-based access controls (RBAC) will be implemented to ensure that only staff members with authorized roles can access or modify student academic records.
- Access levels will be defined based on job responsibilities and the principle of least privilege.

51.5.3 Verification:

- Security testing will evaluate the effectiveness of the RBAC system, verifying that each role has appropriate access and that unauthorized access attempts by other roles are blocked.

52. On-going Support & on-site full user training

52.1 Full System User Documentation and Customization Documentation:

52.1.1 Requirement:

- The system must provide comprehensive user documentation and customization documentation.

52.1.2 Specification:

- The documentation will cover all system functionalities, including step-by-step guides for users, technical documentation for system customization, and troubleshooting information.
- The documentation will be provided in both digital and printed formats.

52.1.3 Verification:

- Testing will involve a review of the documentation by end-users and technical staff to ensure clarity, accuracy, and completeness.
- Feedback will be used to make necessary revisions to ensure that all aspects of the system are well-documented and understandable.

52.2 Provision of Security Certificate and Build Frameworks:

52.2.1 Requirement:

- Provide security certificates and details on the build frameworks utilized for the system and database.

52.2.2 Specification:

- Documentation will include information on security certifications that the system complies with, as well as detailed descriptions of the software and hardware

frameworks used in building the system and database.

- This will also include any compliance adherence related to data security and privacy.

52.2.3 Verification:

- Compliance testing will verify that all provided security certificates and framework details are current and meet stated security standards.
- Documentation review will ensure that information is accurately presented and useful for technical audits.

52.3 *Full Onsite User Training and User Acceptance Testing:*

52.3.1 Requirement:

- Provide full onsite user training and conduct user acceptance testing after installation.

52.3.2 Specification:

- Comprehensive onsite training sessions will be conducted for all user levels, including administrative staff and system managers.
- Training will include hands-on sessions, interactive Q&A, and scenario-based learning.
- After training, user acceptance testing will be conducted to ensure that users are comfortable with all functionalities of the system.

52.3.3 Verification:

- Training effectiveness will be evaluated through feedback forms and performance in user acceptance tests.
- Adjustments will be made based on user feedback to ensure comprehensive understanding and usability of the system.

52.4 Ongoing User Support Based on SLA:

52.4.1 Requirement:

- Provide ongoing user support based on a predefined Service Level Agreement (SLA).

52.4.2 Specification:

- Ongoing support will include helpdesk services, regular system updates, and troubleshooting support.
- The SLA will detail response times, support hours, and escalation procedures.
- Support will be available through multiple channels, including phone, email, and a web-based ticketing system.

52.4.3 Verification:

- Regular audits will be conducted to ensure that the support services meet or exceed the standards set out in the SLA.
- Customer satisfaction surveys will be used to assess the quality of support and identify areas for improvement.

53. Full Access and Ownership to Source Code and Database Code (Only for Developers**)**

53.1 Documentation:

53.1.1 Requirement:

- Comprehensive documentation must be provided for the source code and database, including customization options.

53.1.2 Specification:

- The documentation shall include detailed descriptions of the system architecture, codebase structure, database schema, API references, and user guides for system customization and extension.
- This documentation should be accessible in both digital and printed formats and be updated with every system update or configuration change.

53.1.3 Verification:

- Review by SINU IT staff to confirm that the documentation is complete and comprehensible.
- Documentation accuracy and usefulness will be evaluated based on feedback from system administrators and developers during the initial setup and any subsequent customizations.

53.2 Source Code Level Access & Database Level Access:

53.2.1 Requirement:

- SINU must have direct access to the source code and the database of the SMS.

53.2.2 Specification:

- SINU will have access rights to view, modify, and manage the source code and the underlying database.
- This includes access to all repositories, development tools, and database management systems involved in the SMS.

53.2.3 Verification:

- Provision of credentials and access rights to SINU designated personnel, followed by a walkthrough and verification session to ensure that access is functional and meets SINU's operational requirements.

53.3 *Full System Ownership to SINU*

53.3.1 Requirement:

- Full ownership of the SMS, including all its components, must be transferred to SINU.

53.3.2 Specification:

- SINU shall own all rights to the software system, including source code, databases, documentation, and associated intellectual property.
- The contract will stipulate transfer of ownership and include clauses that ensure SINU retains control over any customizations or derivatives of the original system.

53.3.3 Verification:

- Legal review of ownership transfer documents and contracts to ensure that they accurately reflect the agreed terms. Confirmation of ownership rights in all applicable registries and databases.

53.4 *Options to Provide On-Going Development Support Based on SLA:*

53.4.1 Requirement:

- Ongoing user support should be provided based on a Service Level Agreement (SLA).

53.4.2 Specification:

- The SLA will outline the scope of support services, which may include software updates, maintenance, technical support, training, and troubleshooting. The SLA will

detail response times, support hours, issue escalation procedures, and any other relevant service metrics.

53.4.3 Verification:

- Regular SLA compliance reviews will be conducted to ensure the support services are being provided as agreed.
- User feedback will be collected periodically to assess satisfaction with the support services and identify areas for improvement.

These specifications provide an overview of the requirements for each module of the Student Management System (SMS) for Solomon Islands National University (SINU).