**Graduate Admission Tracking System**

**Document 1: Business Case Document Template**

* Why is this project initiated?

This project aims to address inefficiencies in the graduate admissions process and improve applicant experience. The Graduate Admission Tracking System will automate application processing, reduce errors, and enhance communication with real-time updates. It will also leverage data analytics to inform strategic decision-making, helping to attract and enroll top talent. By implementing this system, the institution seeks to maintain a competitive edge in the evolving landscape of higher education, aligning with its long-term strategic goals.

* What are the current problems?
* Manual Interventions: Excessive manual interventions increase the risk of errors, leading to a suboptimal experience for applicants and administrative staff.
* Processing Delays: The current system experiences delays in processing applications, leading to longer wait times for applicants.
* Technology: The existing technology is outdated and lacks compatibility with modern systems.
* Impact on Business: Errors and inaccuracies can affect the quality of service and the institution's reputation.
* How many problems could be solved with this project?
* Manual Interventions: Integrate automation to minimize manual interventions in the application processing workflow.
* Processing Delays: Implement streamlined processes to significantly reduce processing times, ensuring a quicker and more efficient application journey for prospective students.
* Technology: Integrate new features and scalability for future advancements.
* Resources Required
1. People Resources:
	* Project Team: Cross-functional team comprising individuals from IT, admissions, and project management (BA, PM, UI/UX, Front End, Quality & Testing engineers).
	* Key Stakeholders: Representatives from the academic departments, admissions office, and IT leadership.
2. Financial Resources:
	* Budget: Allocate funds for hardware, software, training, and services, ensuring they are within the set budget. Include an emergency fund for unforeseen circumstances.
3. Technology Resources:
	* Hardware, Software & Training: Invest in updated hardware, necessary software for application, and staff training.
4. Physical Resources:
* Office Space: Allocate workspace for the project team, fostering collaboration and communication.
* Organizational Change Required

Adoption of new technology necessitates assessing organizational changes. This includes training staff, updating procedures, and justifying the investment in ease of use, quality of information, speed of accessibility, and ease of support and maintenance.

* Time Frame to Recover ROI

Time Frame: The time frame to recover the initial investment is estimated to be 12 months.

Operational Efficiencies: Reducing processing times and associated costs.

Budget: The budget should be Rs. 50,00,000 and cover hardware, software, training, and services.

* How to identify stakeholders?

1. Internal Stakeholders

* Executive Leaders: Decision-makers and strategists who will receive regular updates and engage in alignment discussions to ensure the project meets strategic goals.
* Project Team: Members involved in the planning and execution of the project, responsible for day-to-day implementation tasks.
* IT Department: Technology experts handle system integration and ensure technical feasibility.
* Admissions Office: Staff responsible for current admissions processes who will ensure the new system integrates smoothly with existing workflows.

2. External Stakeholders

* Applicants: End-users of the system who will provide feedback and insights on their experience with the new admissions process.
* Regulatory Bodies: Ensure that the system complies with educational regulations and standards.
* Third-Party Vendors: Suppliers of software and technology who will provide updates on system features and support.

3. Interviewing Key Influencers

* Engage with project management experts and key stakeholders to identify individuals significantly influencing the project’s success and gather insights on their expectations and concerns.

4. Asking Key Questions

* Who are the project’s key stakeholders?
* Who is directly or indirectly involved in the project?
* Who benefits from the project's success?
* Who are the suppliers and partners?
* Who has the authority to approve or reject the project?
* Who might be impacted by the project’s outcomes?

**Document 2: Business Analyst Strategy**

* Business Analyst Strategy for Graduate Admissions Tracking System
1. Project Understanding
* Conduct thorough research and analysis to understand the current admissions system and identify pain points.
* Collaborate with key stakeholders, including faculty, administrative staff, and IT personnel, to gain insights into their expectations and challenges.
1. Stakeholder Identification
* Employ stakeholder identification techniques, such as reviewing the project charter, interviewing key sponsors, and analyzing enterprise environmental factors specific to the admissions process.
* Prioritize stakeholders based on their interests, impact, and influence on the project, including applicants, admission officers, and academic departments.
1. Requirements Elicitation
* Utilize elicitation techniques like interviews, workshops, and prototyping to gather comprehensive requirements, focusing on both functional (e.g., application submission, document verification) and non-functional aspects (e.g. system performance, user experience).
1. Change Management
* Identify potential areas for organizational change, such as new overflow of application review, and develop a change management plan.
* Work closely with the project team to ensure a smooth transition to the new admissions tracking system, minimizing disruption to ongoing admissions processes.
1. Use Case Specification
* Create detailed use case specifications for critical functionalities such as user login, application submission, review workflows, and decision-making processes.
* Validate use cases with stakeholders to ensure alignment with business objectives and compliance with institutional policies.
1. Collaboration with Project Team
* Foster strong communication and collaboration with project team members, including developers, testers, and system administrators.
* Facilitate regular meetings to discuss progress, address challenges, and ensure alignment with project goals.
1. Requirement Traceability
* Establish a robust requirement traceability matrix to link business requirements to project deliverables, ensuring that all admissions-related functionalities are adequately addressed.
* Ensure each project phase aligns with identified requirements and business objectives.
1. Communication and Documentation
* Maintain clear and concise documentation throughout the project lifecycle, including requirements specifications, meeting minutes, and status reports.
* Establish communication channels and protocols for effective information exchange among team members and stakeholders, including applicants, staff, and faculty.
1. Quality Assurance
* Implement quality assurance measures to validate that the delivered system meets specified requirements and complies with industry standards.
* Collaborate with testing engineers to develop comprehensive test cases, focusing on user experience, data integrity, and system security.
1. Continuous Improvement
* Embrace an iterative approach, allowing for continuous improvement based on feedback and changing business needs.
* Conduct regular retrospectives to assess project performance and implement lessons learned, ensuring the system evolves to meet future admissions challenges.
* Goals for the Graduate Admission Tracking System project include:
1. Operational Efficiency: Streamline and enhance the admissions process to reduce application processing times, minimize errors, and improve overall workflow efficiency.
2. Institutional Competitiveness: Position the institute as a leader in higher education by offering an innovative, efficient, and user-friendly admissions system that attracts top candidates.
3. Applicant Experience: Improve the overall experience for prospective students, ensuring a seamless, transparent, and efficient application journey.
4. Technology Integration: Implement advanced technologies to optimize the system’s performance, including robust server management, intuitive user interface design, and effective database integration for seamless functionality.
5. Compliance: Ensure adherence to educational regulations and best practices by utilizing compliant technologies and components, maintaining the integrity of the admissions process.
* Database Management

In the context of a business strategy for the Graduate Admissions Tracking System, effective database management is crucial for optimizing the admissions process and ensuring smooth operations. The following key components contribute to a robust database management strategy:

1. Data Centralization: Centralize and organize applicant and admissions data within a secure and scalable database structure. This ensures easy access and management of critical information across various departments, enhancing collaboration and efficiency.
2. Real-time Data Updates: Implement mechanisms for real-time data updates to ensure the system reflects the most current applicant information, status changes, and document submissions. This facilitates accurate decision-making and timely communication with applicants.
3. Data Security Measures: Incorporate robust data security measures, including encryption, role-based access controls, and regular audits, to protect sensitive applicant information and ensure compliance with educational data protection regulations.
4. Backup and Recovery: Establish regular backup and recovery procedures to safeguard against data loss or system failures. This ensures business continuity and minimizes the impact of unforeseen events, protecting the integrity of the admissions process.
5. Integration with External Systems: Enable seamless integration with external systems, such as standardized test databases, transcript verification services, and financial aid platforms, to enrich the system's database with relevant applicant information.
6. Historical Data Tracking: Incorporate features for tracking historical admissions data, allowing the institution to analyze trends over time, assess the effectiveness of recruitment strategies, and make informed decisions based on past performance.
7. Comprehensive Reporting: Provide tools for generating comprehensive reports and analytics based on the data stored in the system. These insights into applicant demographics, acceptance rates, and overall admissions performance empower decision-makers to refine their strategies and improve outcomes.
8. Compliance with Data Regulations: Ensure that database management practices align with educational data protection and privacy regulations, providing applicants and stakeholders with confidence in the security and privacy of their information.
* Marketing and Campaign Management

An effective marketing and campaign management features are essential for attracting and engaging prospective students. Here are key components that contribute to a robust marketing and campaign management strategy:

1. Integrated Marketing Tools: Provide tools that allow universities and colleges to seamlessly execute and monitor marketing campaigns directly within the admissions system. This can include email outreach, social media integration, and event promotion.
2. Campaign Planning and Execution: Enable the planning and execution of targeted marketing campaigns aimed at prospective students, including email newsletters, social media advertising, and online promotions to reach a broader audience of potential applicants.
3. Prospective Student Segmentation: Implement features that allow for the segmentation of prospective students based on criteria such as academic interests, geographical location, and engagement history. This enables the delivery of tailored marketing messages that resonate with specific groups.
4. Multi-channel Marketing: Support multi-channel marketing strategies by integrating with various communication platforms, such as email, social media, and SMS, ensuring consistent and effective outreach across all channels.
5. Campaign Analytics: Offer comprehensive analytics on the performance of marketing campaigns, including metrics such as engagement rates, application submissions, and conversion rates, enabling data-driven adjustments and improvements to strategies.
6. Personalization and Customization: Provide tools for personalizing marketing content based on individual prospective student profiles, such as academic interests or preferred campus locations, to enhance the relevance and impact of communications.
7. Lead Generation and Tracking: Integrate lead generation features within marketing campaigns to capture and track prospective student inquiries and interactions throughout the admissions process, helping to identify and nurture potential applicants.
8. Compliance Management: Ensure that all marketing materials and campaigns comply with relevant regulations and institutional policies, maintaining adherence to legal and ethical standards in student communications.
* Lead Management

Robust lead management capabilities are essential for effectively managing prospective students and enhancing the admissions process. Here are key components related to lead management that contribute to a successful strategy:

1. Lead Capture Forms: Implement user-friendly lead capture forms within the admissions system to gather essential information from prospective students, such as academic interests, contact details, and preferred programs.
2. Automated Lead Assignment: Utilize automated algorithms or predefined criteria to assign prospective student leads to the most suitable admissions counselors or departments, ensuring prompt and personalized follow-ups.
3. Lead Scoring: Integrate lead scoring mechanisms to prioritize prospective students based on their likelihood to apply, enabling admissions teams to focus on high-value candidates.
4. Lead Segmentation: Implement tools for segmenting leads based on factors such as academic background, geographic location, and engagement history, allowing for targeted outreach and personalized communication.
5. CRM Integration: Seamlessly integrate lead management functionalities with Customer Relationship Management (CRM) tools for a comprehensive view of prospective student interactions and histories, facilitating more informed engagement.
6. Automated Follow-ups: Set up automated follow-up communications at various stages of the prospective student journey, ensuring consistent and timely engagement throughout the admissions process.
7. Lead Nurturing Campaigns: Design and execute lead nurturing campaigns within the admissions system, providing relevant information and resources to prospective students to build trust and guide them through the application process.
8. Real-time Lead Analytics: Provide real-time analytics on lead performance, conversion rates, and engagement metrics, allowing for data-driven adjustments to lead management and outreach strategies.
9. Integration with Marketing Channels: Enable integration with various marketing channels, including email marketing, social media, and online advertising, to capture leads from diverse sources and expand the reach of admissions campaigns.
10. Personalization Tools: Incorporate personalization tools to tailor communication and offerings based on individual prospective student profiles, enhancing the likelihood of successful application submissions.
* Sales Automation

Incorporating admissions process automation features is crucial for streamlining operations and enhancing efficiency. Here are key components related to admissions process automation that contribute to an effective strategy:

1. Lead Management: Implement lead tracking and scoring mechanisms to effectively manage prospective students, ensuring a systematic approach to nurturing leads and guiding them through the admissions process.
2. Automated Communication: Utilize automated communication tools for personalized and timely interactions with prospective students. This includes automated emails, notifications, and updates throughout the application process, from initial inquiry to final admission.
3. Admissions Funnel Visualization: Provide a visual representation of the admissions funnel within the system, allowing teams to track the progression of prospective students from initial inquiry through application submission, review, and final decision.
4. Cross-program Promotion Tools: Integrate tools that identify opportunities for promoting related academic programs or courses based on student interests and profiles, enhancing enrollment in multiple programs.
5. Application Review and Decision Automation: Include features for automating the generation of application review summaries and decision letters, ensuring that admissions teams can swiftly provide accurate and timely feedback to applicants.
6. Integration with Marketing Automation: Seamlessly integrate with marketing automation tools to align outreach and admissions efforts, ensuring a cohesive approach to student engagement and conversion.
7. Performance Analytics: Provide detailed analytics on the admissions process, including conversion rates, application completions, and individual team member achievements, facilitating data-driven decision-making.
8. Mobile Admissions Enablement: Enable mobile access to admissions tools, allowing admissions teams to stay productive on the go, access real-time data, and respond promptly to student inquiries.
* Workflow Automation and Employee Management

An effective system should significantly enhance team efficiency and productivity by integrating workflow automation features. These features help streamline the admissions process, manage tasks, and optimize employee performance. Key components for such a system include:

1. Admissions Process Workflow Automation: Implement workflow automation to streamline the end-to-end admissions process, reducing manual intervention, minimizing errors, and accelerating application review and decision-making.
2. Automated Notifications: Integrate automated notifications for key milestones in the admissions process, such as application submission, document review completion, interview scheduling, and final decision updates. This keeps both applicants and internal teams informed.
3. Task Assignment and Tracking: Enable task assignment and tracking functionalities, allowing admissions officers and staff to efficiently manage and monitor their responsibilities within the admissions workflow.
4. Document Management: Provide a centralized document management system within the application, ensuring secure storage, easy retrieval, and version control for all applicant-related documents.
5. Collaboration Tools: Include collaboration tools to facilitate communication and coordination among team members involved in the admissions process, enhancing teamwork and efficiency.
6. Employee Performance Tracking: Implement features for tracking employee performance metrics, such as the number of applications processed, response times, and applicant satisfaction, to identify areas for improvement and recognize high-performing individuals.
7. Leave and Attendance Management: Integrate tools for leave and attendance management to streamline HR processes, ensuring accurate records and efficient workforce management.
8. Employee Records and Hierarchy: Maintain a comprehensive employee records system and organizational hierarchy within the application, providing clarity on roles, responsibilities, and reporting structures within the admissions team.
9. Security and Access Controls: Implement robust security measures and access controls to protect sensitive applicant and employee data, ensuring compliance with data protection regulations.
* Business Intelligence and Reporting

An effective Graduate Admissions Tracking System should incorporate robust business intelligence and reporting features to facilitate data-driven decision-making and strategic planning. Key components related to business intelligence and reporting include:

1. Data Analytics: The system should incorporate advanced data analytics capabilities, enabling institutions to gain insights into applicant behavior, admission trends, and overall performance metrics.
2. Real-time Reporting: Provide real-time reporting features that offer up-to-the-minute data on application statuses, admissions rates, and other critical metrics. This ensures that decision-makers have access to the latest information for making informed choices.
3. Customizable Dashboards: Allow users to create personalized dashboards with key performance indicators (KPIs) relevant to the admissions process, such as application volume, acceptance rates, and processing times.
4. Predictive Analytics: Incorporate predictive analytics tools to forecast application trends, assess the likelihood of acceptance, and identify potential areas for process improvement or increased applicant engagement.
5. Integration with External Data Sources: Facilitate seamless integration with external data sources, such as demographic databases or educational performance metrics, to enhance the depth and accuracy of analytics.
6. Compliance Reporting: Include features that support compliance reporting, ensuring that the system helps institutions meet regulatory requirements related to admissions data and applicant privacy.
7. Mobile Business Intelligence: Provide mobile access to business intelligence tools, allowing stakeholders to access critical insights on the go, ensuring flexibility in decision-making.
8. Security and Data Governance: Implement robust security measures and data governance protocols to protect sensitive applicant information and ensure compliance with data protection regulations.
* Deployment Environment

The deployment environment of a Graduate Admissions Tracking System is crucial for ensuring seamless operations, scalability, and security. Key components related to the deployment environment include:

1. Cloud Infrastructure: Opt for a cloud-based deployment model to leverage scalability, flexibility, and cost-efficiency. Cloud infrastructure allows for easy scaling based on user demand and provides accessibility from various locations, facilitating remote admissions management.
2. High Availability: Design the deployment environment to ensure high availability, minimizing downtime and ensuring continuous access to the admissions system. This is essential for meeting the needs of applicants and administrators, especially during peak application periods.
3. Security Measures: Implement robust security measures within the deployment environment, including encryption, secure communication protocols, and access controls. Security is paramount to protect sensitive applicant data and maintain compliance with privacy regulations.
4. Data Redundancy and Backups: Establish mechanisms for data redundancy and regular backups to prevent data loss in the event of system failures or disasters. This ensures data integrity and supports efficient recovery processes.
5. Performance Monitoring and Optimization: Incorporate tools for real-time performance monitoring to identify and address potential bottlenecks or issues promptly. Continuous optimization of the deployment environment enhances the overall performance of the admissions system.
6. Regulatory Compliance: Ensure that the deployment environment adheres to regulatory requirements specific to educational institutions, including data protection and privacy laws. Compliance is critical for maintaining the trust of applicants and regulatory authorities.
7. Disaster Recovery Planning: Develop and implement a comprehensive disaster recovery plan to mitigate the impact of unforeseen events. This includes strategies for data recovery, system restoration, and ensuring business continuity, safeguarding the admissions process against disruptions.
* SDLC Methodologies - Waterfall Model

The Waterfall Model is a sequential and linear approach to software development, where each phase depends on the completion of the previous one. It consists of distinct phases including planning, requirement analysis, design, implementation, testing, deployment, and maintenance.

### Planning

### In the planning phase of the Software Development Life Cycle (SDLC) for the Graduate Admissions Tracking System, key activities are centered around initiating and defining the project. The following steps are crucial:

* Stakeholder Identification: Identify and engage key stakeholders, such as university admissions officers, IT staff, and potential users (e.g., applicants and faculty). This ensures that all relevant perspectives are considered.
* Project Team Establishment: Assemble a project team with clearly defined roles, including project managers, business analysts, developers, and QA testers.
* Requirement Gathering: Conduct comprehensive requirement analysis by gathering input from stakeholders through meetings, interviews, and surveys. Document both functional and non-functional requirements.
* Feasibility Studies: Assess the technical capabilities and resources needed to implement the system, including evaluating existing systems and determining integration requirements.
* Risk Assessment: Identify potential risks, such as data security challenges, and prioritize them based on their potential impact on the project.
* Project Plan Development: Develop a detailed project plan outlining tasks, timelines, and resource allocation. Ensure the plan aligns with the university's academic calendar and admissions cycles.
* Technology Selection: Choose appropriate technologies, such as programming languages, databases, and frameworks, that align with the scalability and integration needs of the admissions system.
* Budgeting and Legal Compliance: Establish a budget that covers all project aspects, including development, testing, and deployment. Ensure compliance with educational data privacy regulations (e.g., FERPA).
* Communication Plan: Develop a communication plan to ensure effective information flow among team members and stakeholders.
* Documentation: Conclude the planning phase with thorough documentation, providing a clear foundation for the subsequent development stages.

### Requirement Analysis

### In the Requirement Analysis phase of the SDLC for the Graduate Admissions Tracking System, the focus is on gathering and defining detailed requirements. Key activities include:

* Stakeholder Collaboration:
	+ Engage with stakeholders including university admissions staff, IT teams, and applicants.
	+ Conduct workshops, interviews, and surveys to gather insights into their needs and expectations for the system.
* Requirements Elicitation:
	+ Gather and document both functional and non-functional requirements. Functional requirements might include features like applicant tracking, document management, and communication tools. Non-functional requirements could involve system performance, security, and user accessibility.
	+ Identify specific features and system behaviors expected by end-users, such as automated notifications, role-based access, and integration with other university systems.
* Regulatory Compliance:
	+ Identify and understand legal and regulatory requirements governing educational data, such as FERPA (Family Educational Rights and Privacy Act).
	+ Ensure that the system's functionalities align with these regulations, especially regarding data security and privacy.
* Use Cases and User Stories:
	+ Develop use cases and user stories to illustrate how users (e.g., applicants, admissions officers, faculty) will interact with the system.
	+ Define scenarios covering a range of user actions, such as submitting applications, reviewing documents, and managing admissions decisions, to capture diverse requirements.
* Prototyping:
	+ Create prototypes or mockups to visually represent the system’s user interface and workflows.
	+ Use these prototypes to validate requirements with stakeholders, ensuring everyone has a shared understanding of the system’s design and functionality.
* Requirement Documentation:
	+ Document all gathered requirements in a clear and structured manner.
	+ Create a Requirements Specification document that serves as a comprehensive reference for the development team, ensuring alignment with stakeholder needs.
* Prioritization:
	+ Prioritize requirements based on their importance and impact on the system. Focus on critical functionalities first, ensuring essential features are developed early in the process.
* Traceability Matrix:
	+ Establish a traceability matrix to link requirements to specific design elements, ensuring each requirement is addressed during development and testing phases.
* Change Management:
	+ Implement a change management process to handle modifications to requirements as needed. Ensure that changes are carefully evaluated for their impact on project scope, timelines, and costs.

### Design

### In the Design phase of the Software Development Life Cycle (SDLC) for the successful development of the Graduate Admissions Tracking System, several critical activities are carried out:

* Architectural Design:
	+ Develop the overall architecture of the Graduate Admissions Tracking System.
	+ Define the system's structure, components, and their interactions, ensuring scalability, security, and performance. This may include defining the relationships between applicant data, document management, and user roles.
* Database Design:
	+ Design the database schema to organize, store, and access applicant and admissions data.
	+ Define relationships between different data entities such as applicants, applications, supporting documents, and decisions, ensuring data integrity and efficient retrieval.
* User Interface (UI) Design:
	+ Create the user interface design based on the requirements gathered in earlier phases.
	+ Focus on developing an intuitive and user-friendly interface for applicants, admissions officers, and other stakeholders, ensuring a seamless user experience.
* System & Security Design:
	+ Develop a detailed system design, specifying each component and module, including workflows for application submission, review, and decision-making.
	+ Incorporate security measures to protect sensitive applicant information, such as encryption of personal data and secure access controls for different user roles.
* Network Design:
	+ Design the network architecture to support the communication needs of the Graduate Admissions Tracking System.
	+ Ensure the system can handle expected traffic, especially during peak application periods, and maintain optimal performance.
* Prototyping:
	+ Create prototypes or wireframes to visualize the designed interfaces and gather feedback from stakeholders, including applicants, faculty, and administrative staff.
	+ Refine designs based on user input to align the system’s functionality and aesthetics with stakeholder expectations.
* Design Documentation:
	+ Document the design decisions, architectural choices, and technical specifications.
	+ Create comprehensive design documents that serve as a reference for the development team and future maintenance efforts.
* Review and Validation:
	+ Conduct design reviews to ensure the proposed design aligns with the requirements and best practices.
	+ Validate the design with stakeholders to ensure it meets their needs and expectations.
	+ Provide training to the development team on any new tools or technologies introduced during the design phase to ensure successful implementation.

### Implementation (Coding Phase)

### In the Implementation (Coding) phase of the SDLC for the successful development of the Graduate Admissions Tracking System, the focus shifts to actual coding and building the system. Key activities include:

* Coding:
	+ Developers write code based on the design specifications, ensuring alignment with the planned architecture.
	+ Follow coding standards and best practices to ensure consistency, maintainability, and scalability of the system.
* Database Implementation:
	+ Implement the database according to the designed schema, ensuring that all data relationships and storage mechanisms are correctly set up.
	+ Ensure that data storage and retrieval processes align with the system’s performance requirements and are optimized for quick access.
* Integration of Components:
	+ Integrate individual components and modules to create a cohesive system, such as linking the user interface with the backend database and application processing logic.
	+ Ensure seamless communication between different parts of the system, particularly between application submission, document processing, and decision-making modules.
* Unit Testing:
	+ Conduct unit testing to verify the functionality of individual components or modules, such as testing the accuracy of the applicant tracking features.
	+ Identify and fix any bugs or issues at the unit level, ensuring each part of the system works as intended before integration.
* Documentation Updates:
	+ Update documentation to reflect any changes made during the coding phase, ensuring all modifications are well-documented.
	+ Maintain synchronization between the evolving codebase and the documentation to facilitate future maintenance and updates.
* Security Implementation:
	+ Implement security measures defined in the design phase, such as encryption of applicant data and secure login mechanisms.
	+ Address potential vulnerabilities, ensuring the system securely handles sensitive user information and adheres to relevant privacy regulations.
* Error Handling:
	+ Implement robust error-handling mechanisms to gracefully manage unexpected situations, such as application submission failures or data retrieval errors.
	+ Log errors for troubleshooting and debugging purposes, ensuring quick resolution of issues.
* Performance Optimization:
	+ Optimize code and database queries for performance, ensuring efficient execution and minimizing latency, especially during high-traffic periods like application deadlines.
	+ Address any bottlenecks identified during testing, ensuring the system meets performance expectations.
* Training and Knowledge Transfer:
	+ Provide training to the development team on any new tools or technologies introduced during the implementation phase.
	+ Facilitate knowledge transfer to ensure a shared understanding of the codebase, enabling effective collaboration and future development efforts.

### Testing

### In the Testing phase of the Software Development Life Cycle (SDLC) for the successful development of the Graduate Admissions Tracking System, comprehensive testing activities are undertaken to ensure the system's quality, reliability, and security. Key activities include:

* Unit Testing:
	+ Conduct unit testing to validate the functionality of individual components and modules, such as application forms, document upload features, and user role management.
	+ Identify and address any defects at the unit level to ensure each component functions correctly.
* System Testing:
	+ Test the entire Graduate Admissions Tracking System as a whole to validate end-to-end functionality.
	+ Ensure that all system components, such as applicant processing, decision-making workflows, and notifications, work together seamlessly.
* Functional Testing:
	+ Validate that the system meets specified functional requirements, including application submission, document review, and decision tracking.
	+ Test each feature to ensure it performs as expected according to the requirements.
* Non-Functional Testing:
	+ Conduct non-functional testing, including performance, security, and usability testing.
	+ Ensure the system's responsiveness, security measures, and user experience meet established standards.
* User Acceptance Testing (UAT):
	+ Collaborate with end-users, such as admissions staff and applicants, to conduct User Acceptance Testing.
	+ Validate that the system aligns with user expectations and business requirements, addressing any issues identified during testing.
* Security Testing:
	+ Conduct security testing to identify vulnerabilities and ensure the protection of sensitive applicant data and system integrity.
	+ Address any security concerns or weaknesses to safeguard against potential threats.
* Usability Testing:
	+ Evaluate the system's user interface and overall user experience.
	+ Identify areas for improvement to enhance the usability and satisfaction of users interacting with the system.
* Documentation Validation:
	+ Validate that all documentation, including user manuals and technical documents, is accurate and up-to-date.
	+ Ensure that documentation reflects the tested system and provides clear guidance for users and administrators.
* Bug Fixing and Retesting:
	+ Address any defects or issues identified during testing, making necessary fixes to ensure system stability.
	+ Conduct retesting to confirm that bug fixes are effective and that the system continues to meet quality standards.

### Deployment

### In the Deployment phase of the Software Development Life Cycle (SDLC) for the successful development of the Graduate Admissions Tracking System, the focus shifts to releasing the system into a production environment. Key activities include:

* Release Planning:
	+ Develop a detailed deployment plan, including timelines, contingency measures, and coordination with stakeholders.
	+ Ensure a smooth transition to the production environment, addressing any potential issues in advance.
* Environment Setup:
	+ Prepare the production environment, including servers, databases, and infrastructure, to support the Graduate Admissions Tracking System.
	+ Ensure that the production environment mirrors the testing environment to minimize potential issues during deployment.
* Data Migration:
	+ If applicable, migrate relevant data from testing or development databases to the production database.
	+ Validate data integrity during the migration process to ensure accurate and complete data transfer.
* Verification and Validation:
	+ Perform final verification and validation checks to ensure that the deployed system functions correctly in the production environment.
	+ Confirm that all configurations are accurate and compatible with the production setup.
* User Training:
	+ Provide training sessions for end-users and stakeholders, such as admissions staff and applicants, to familiarize them with the new system.
	+ Address any questions or concerns related to the deployment to ensure a smooth transition.
* Communication and Notifications:
	+ Communicate the deployment schedule to relevant stakeholders, including users and administrators.
	+ Notify them about upcoming changes, expected downtime, and any other relevant information.
* Go-Live:
	+ Execute the deployment plan and release the Graduate Admissions Tracking System to the production environment.
	+ Monitor the system closely during the initial period to address any immediate issues and ensure a successful launch.
* Post-Deployment Testing:
	+ Conduct additional testing in the production environment to confirm that all features work as expected under actual usage conditions.
	+ Validate the system's performance and stability following the go-live.
* Documentation Update:
	+ Update documentation to reflect any changes made during the deployment phase.
	+ Ensure that documentation remains accurate and provides ongoing reference for users and administrators.
* Feedback Collection:
	+ Collect feedback from end-users and stakeholders regarding their experience with the newly deployed system.
	+ Address any reported issues and gather insights for future improvements.
* Closure:
	+ Conclude the deployment phase by formally closing the deployment project.
	+ Document lessons learned and areas for improvement to inform future deployments and project management practices.

Choosing the Waterfall model for the successful development of the Graduate Admissions Tracking System can be justified based on several factors:

* Well-Defined Requirements:
	+ The Waterfall model is ideal when project requirements are clear, stable, and unlikely to change significantly. In the case of a Graduate Admissions Tracking System, where processes, regulatory compliance, and specific functionalities are critical, having fixed and clear requirements aligns well with the Waterfall model’s structured approach.
* Sequential and Phased Approach:
	+ The Waterfall model follows a linear and sequential progression through distinct phases such as requirements, design, implementation, testing, deployment, and maintenance. For a complex system like graduate admissions, this structured approach ensures that each phase is completed thoroughly before moving to the next, reducing the likelihood of errors and enhancing overall project control.
* Risk Management:
	+ The Waterfall model is perceived as less risky concerning scope creep and changes in requirements during development. This is advantageous for the Graduate Admissions Tracking System, where adherence to detailed requirements and regulations is crucial. The model’s clear documentation and understanding of requirements at the outset contribute to effective risk management.
* Documentation and Traceability:
	+ Waterfall emphasizes comprehensive documentation at each phase, ensuring that requirements, design decisions, and test cases are well-documented. This level of documentation is beneficial for audit trails, compliance reporting, and maintaining clear traceability of decisions, which is vital for the admissions process and for meeting institutional or regulatory standards.
* Regulatory Compliance:
	+ Graduate admissions systems often need to comply with institutional policies and external regulations. The Waterfall model allows for meticulous planning and execution, facilitating compliance with these standards. It ensures that all compliance requirements are addressed and validated before proceeding to subsequent phases.
* Client Involvement:
	+ The Waterfall model typically involves client or stakeholder reviews at the end of each phase. This structured involvement ensures that the delivered product aligns with client expectations and requirements, minimizing the chances of misunderstandings or misalignments with the needs of admissions staff and applicants.
* RACI Matrix

The RACI matrix, also known as the Responsibility Assignment Matrix or ARCI matrix, is a

valuable tool in project management that outlines the participation of individuals in

completing tasks and delivering project deliverables. Pronounced 'ray-see' and originally

called the 'Decision Rights Matrix' in the 1950s, it addresses the common challenge of

unclear roles and responsibilities within projects or business processes.

This matrix serves various purposes, including facilitating clarity in meetings, aiding in

onboarding processes, and managing large tasks. Its primary association, however, lies with

project management. The RACI matrix enhances teamwork, collaboration, and

communication by providing a simple and effective language to discuss and define roles and

responsibilities within an organization.

The RACI matrix, a key tool in project management, helps clarify roles and responsibilities for tasks and deliverables. It stands for Responsible, Accountable, Consulted, and Informed, and is used to ensure clear communication and effective collaboration within a project. Here’s a RACI matrix tailored for the Graduate Admissions Tracking System:

| Task | Project Sponsor | Head of Delivery | Project Manager | Business Analyst | Developer/Senior | Developer/Junior | Testers | Stakeholders |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement Gathering | A/I | C | R | R | I | I | I | C |
| Analysis | I | R | A | C | I | I | I | C |
| Development | I | C | A | C | R | R | I | I |
| Testing | I | C | I | I | I | I | R | I |
| Implementation | I | C | R | I | I | I | I | I |
| User Acceptance Testing (UAT) | I | C | R | C | I | I | R | A |

Stakeholder Analysis for Graduate Admissions Tracking System

* Primary Stakeholders:
	+ University Administration and Management: Decision-makers and sponsors of the project.
	+ Project Team (Business Analysts, Developers, Testers): Responsible for executing the project and ensuring it meets the university's requirements.
	+ Prospective Graduate Students: End-users of the admissions system who expect an efficient and user-friendly application process.
* Secondary Stakeholders:
	+ Regulatory Bodies (Educational Accrediting Agencies): Setting standards and regulations that the system must comply with.
	+ IT Security and Compliance Teams: Ensuring the system adheres to data protection and security requirements.
	+ University Admissions Staff: Implementing and using the system in daily operations.
* Tertiary Stakeholders:
	+ Third-Party Service Providers (if any): Providing tools or services that integrate with the admissions system.
	+ Competitors (Other Universities): Observing changes for competitive analysis and potential impacts.

### What Documents to Write for the Graduate Admissions Tracking System

* Project Vision Document:
	+ Purpose: Defines the high-level goals, objectives, and vision for the Graduate Admissions Tracking System project.
	+ Content: Includes project scope, objectives, stakeholders, high-level requirements, and success criteria.
* Business Analysis Plan:
	+ Purpose: Outlines the approach and activities for conducting business analysis throughout the Graduate Admissions Tracking System project.
	+ Content: Covers the scope of business analysis, stakeholder analysis, requirements gathering methods, and communication plan.
* Business Requirements Document (BRD):
	+ Purpose: Captures the high-level business needs and objectives that the Graduate Admissions Tracking System aims to address.
	+ Content: Includes business context, goals, objectives, functional and non-functional requirements, constraints, and assumptions.
* Functional Specification Document (FSD) / Functional Requirements Specification (FRS):
	+ Purpose: Describes how the Graduate Admissions Tracking System will function from a user's perspective.
	+ Content: Contains detailed functional requirements, use cases, user stories, data flow diagrams, and process flows.
* System Requirement Specification (SRS):
	+ Purpose: Details how the Graduate Admissions Tracking System will behave and be implemented.
	+ Content: Includes technical specifications, system architecture, hardware and software requirements, data models, and interfaces.
* Software Requirement Document (SRD):
	+ Purpose: A comprehensive document outlining software requirements for the Graduate Admissions Tracking System.
	+ Content: Provides detailed software specifications, algorithms, data structures, user interfaces, and database design.

### Process to Follow for Document Sign-Off

* Draft Document Submission:
	+ Share draft versions of documents (Project Vision, Business Analysis Plan, BRD, FSD/FRS, and SRS) with key stakeholders.
* Review and Feedback:
	+ Allow stakeholders time to review the documents thoroughly.
	+ Collect feedback through meetings, emails, or collaborative tools.
* Incorporate Feedback:
	+ Revise documents based on the feedback received, addressing any concerns or suggestions.
* Clarifications and Amendments:
	+ Address any additional questions or clarifications that stakeholders may have.
	+ Make necessary amendments in real-time if possible.
* Document Approval:
	+ Request formal approval during a meeting or through a designated approval process.
	+ Stakeholders signify their approval by providing written sign-off.
* Sign-Off Confirmation:
	+ Obtain signatures or written confirmation from key stakeholders, indicating their approval.
	+ Keep a record of the signed documents for documentation and audit purposes.
* Distribution of Final Documents:
	+ Distribute final approved documents to all relevant parties.
	+ Ensure all stakeholders have access to the approved versions.
* Communication:
	+ Communicate the formal sign-off and approval to the project team, ensuring everyone is aware of the finalized documents.

### How to Take Approvals from the Client

* Draft Document Submission:
	+ Share draft versions of relevant documents (Project Vision, Business Analysis Plan, BRD, FSD/FRS, and SRS) with the client.
* Client Review:
	+ Allow the client sufficient time to review the documents.
	+ Provide additional explanations or clarifications if requested by the client.
* Feedback Collection:
	+ Establish a feedback loop through meetings, emails, or collaborative tools to gather the client's input.
	+ Document all feedback and ensure it is clearly understood.
* Revision and Incorporation:
	+ Revise documents based on client feedback.
	+ Clearly document how each piece of feedback has been addressed.
* Approval Request:
	+ Request formal approval during a meeting or through a designated approval process.
	+ Communicate that their approval is crucial for the project to proceed.
* Sign-Off Confirmation:
	+ Obtain the client's signature or written confirmation, indicating formal approval.
	+ Ensure that all key stakeholders on the client side provide sign-off.
* Distribution of Final Documents:
	+ Distribute final approved documents to the client and any other relevant parties.
	+ Provide access to a secure repository for their records.
* Project Kickoff:
	+ Once approvals are obtained, proceed with the project according to the agreed-upon timelines and milestones.
* Regular Updates:
	+ Keep the client informed with regular project updates and progress reports.
	+ Address any changes or issues promptly and transparently.

### Communication Channels to Establish and Implement

* Project Management Software:
	+ Use tools like Jira, Trello, or Asana for task tracking, progress monitoring, and collaboration.
* Email:
	+ Use email for formal communication, document sharing, and updates.
	+ Send regular project updates, meeting agendas, and minutes via email.
* Meetings:
	+ Schedule regular meetings for status updates, issue resolution, and milestone discussions.
	+ Conduct kick-off meetings, progress reviews, and milestone celebrations.
* Video Conferencing:
	+ Use platforms like Zoom, Microsoft Teams, or Google Meet for virtual meetings.
	+ Schedule virtual meetings for important discussions or presentations.
* Collaboration Platforms:
	+ Implement tools like Microsoft Teams, Slack, or Workplace by Facebook for real-time messaging and file sharing.
	+ Foster a collaborative environment for team members to ask questions and share information.
* Document Management System:
	+ Set up a centralized document repository using tools like Google Drive, SharePoint, or Dropbox.
	+ Ensure all project documents are stored, organized, and easily accessible.
* Progress Reports:
	+ Develop a regular cadence for sending progress reports to stakeholders.
	+ Share key metrics, achievements, and upcoming milestones in these reports.
* Client Meetings:
	+ Schedule periodic meetings with the client to discuss project progress, address concerns, and gather feedback.
* Change Management Communication:
	+ Develop a structured communication plan for any changes in project scope, timelines, or requirements.
	+ Clearly communicate the impact of changes and steps being taken to address them.

### How to Handle Change Requests

* Change Request Submission:
	+ Establish a clear process for submitting change requests.
	+ Provide a standardized form or template for documenting change details.
* Change Request Documentation:
	+ Require detailed documentation for each change request, including the reason for the change, potential impacts, and proposed solutions.
	+ Encourage stakeholders to articulate the benefits of the proposed change.
* Change Review Board:
	+ Establish a Change Review Board (CRB) responsible for reviewing and approving/rejecting change requests.
	+ Include key stakeholders, project managers, and subject matter experts in the board.
* Initial Assessment:
	+ Conduct an initial assessment to understand the change request's scope and potential impacts on the project timeline, budget, and resources.
	+ Evaluate whether the change aligns with project goals and objectives.
* Impact Analysis:
	+ Perform a comprehensive impact analysis to assess how the change will affect project deliverables, timelines, costs, and risks.
	+ Identify dependencies and potential conflicts with existing tasks.
* Prioritization:
	+ Prioritize change requests based on urgency, impact, and alignment with project priorities.
	+ Consider the overall project timeline and critical milestones.
* Communication:
	+ Communicate the status of the change request to stakeholders.
	+ Clearly convey whether the change is approved, rejected, or pending further review.
* Approval/Rejection:
	+ Present the change request to the Change Review Board for approval or rejection.
	+ Clearly document the reasons for approval or rejection.
* Documentation Update:
	+ Update project documentation, including the project plan, requirements, and affected documents, based on the approved changes.
* Implementation:
	+ If approved, implement the change following established project management processes.
	+ Communicate the approved changes to the project team and relevant stakeholders.
* Monitoring:
	+ Monitor the impact of the implemented change on project performance.
	+ Track any deviations from the original plan and address them promptly.
* Closure:
	+ Officially close the change request, updating all relevant records and documentation.

###

### How to Update the Progress of the Project to Stakeholders

* Define Communication Plan:
	+ Develop a communication plan that outlines the frequency, format, and channels for project updates.
	+ Identify key stakeholders and their information needs.
* Regular Status Meetings:
	+ Conduct regular status meetings or check-ins with stakeholders.
	+ Provide updates on project milestones, achievements, and any challenges.
* Project Status Reports:
	+ Prepare and distribute project status reports at defined intervals (weekly, bi-weekly, or monthly).
	+ Include key performance indicators (KPIs), progress against milestones, and upcoming activities.
* Visual Dashboards:
	+ Create visual dashboards or project scorecards to represent project progress.
	+ Use charts, graphs, and other visual elements for easy comprehension.
* Email Updates:
	+ Send concise and informative email updates to stakeholders.
	+ Highlight key accomplishments, upcoming activities, and any risks or issues needing attention.
* Project Documentation:
	+ Maintain up-to-date project documentation, including project plans, timelines, and requirements.
	+ Share relevant documentation with stakeholders as needed.

### How to Take Sign-Off on the UAT - Client Project Acceptance Form

* Document Preparation:
	+ Prepare the UAT - Client Project Acceptance Form with all relevant details.
	+ Clearly outline the scope of UAT, criteria for acceptance, and any specific testing scenarios.
* Conduct UAT Session:
	+ Coordinate a UAT session with the client or their designated representatives.
	+ Provide access to the testing environment and necessary documentation.
* Detailed Test Cases:
	+ Ensure UAT test cases are detailed and cover all aspects of the system.
	+ Include scenarios that simulate real-world usage of the application.
* User Training:
	+ If required, provide training to users on how to navigate the system and perform UAT.
	+ Clarify the purpose and expected outcomes of UAT.
* Monitoring UAT Progress:
	+ Monitor UAT progress closely.
	+ Address any questions or issues raised by users promptly.
* UAT Completion Notification:
	+ Notify users when they have completed UAT tasks.
	+ Provide clear instructions on how to communicate their acceptance or any identified issues.
* Client Review Meeting:
	+ Schedule a meeting with the client to review UAT results.
	+ Present findings, including identified issues and their resolutions.
* Confirmation of Readiness:
	+ Obtain confirmation from the client that they are satisfied with UAT results and the resolution of any identified issues.
* Sign-Off Meeting:
	+ Schedule a sign-off meeting with client representatives.
	+ Present the finalized UAT - Client Project Acceptance Form for review and signature.
* Formal Acknowledgment:
	+ Ensure the client formally acknowledges their acceptance of the system by signing the UAT form
* Document Distribution:
	+ Distribute the signed UAT - Client Project Acceptance Form to all relevant stakeholders.
	+ Store a copy in the project documentation repository.
* Project Closure:
	+ Consider UAT sign-off a significant milestone in project closure.
	+ Begin the formal project closure process.
* UAT Signoff Process

By following a structured and collaborative approach, you ensure that the UAT signoff process for the Graduate Admissions Tracking System is smooth, transparent, and leads to the successful acceptance of the project by the client.

Project Sign-off Sheet

Version Date: 24/01/2025
Project Name: Graduate Admissions Tracking System

Project Goals for the Graduate Admissions Tracking System:

* Enhance Operational Efficiency:
Streamline and digitize the graduate admissions process to reduce manual efforts, minimize errors, and improve overall operational efficiency.
* Reduce Processing Times:
Implement a system that significantly reduces the time required for processing applications, ensuring a quicker and more seamless experience for prospective students.
* Minimize Downtime and Errors:
Introduce a reliable admissions tracking system that minimizes system downtime, related wait times, and errors, providing a more user-friendly and efficient platform.
* Improve User Experience:
Prioritize the enhancement of the overall user experience by offering a user-friendly and efficient online application tracking process for applicants and administrators alike.
* Maintain Competitive Edge:
Position the institution at the forefront of digital admissions by optimizing the tracking system, aligning with industry best practices, and maintaining a competitive edge in the education sector.

Project Manager: Mr. Venkat
Sponsor: Mr. Rohit

Start Date: 01/02/2025
Budget: 50,00,000
Planned Completion Date: 01/08/2025
Actual Completion Date: 01/09/2025
Variance: 5,00,000
Days Past the Planned Completion Date: 0

Project Deliverables for the Graduate Admissions Tracking System:

The project deliverables for the Graduate Admissions Tracking System encompass the development and implementation of a user-friendly admissions tracking system, comprehensive documentation of functional and technical requirements, testing protocols, and reports. Additionally, the project includes the creation of training materials for users and technical staff, a detailed financial breakdown, an implementation plan, and post-implementation support documentation. The formal acceptance through a UAT - Client Project Acceptance Form and a lessons-learned report contribute to a successful project closure, ensuring improved operational efficiency and a competitive edge for the institution in the digital education sector.

By signing this document, I acknowledge that I have delivered all the stated deliverables at the agreed-upon quality levels.

Project Manager Signature:
Date: 25/01/2025

By signing this document, I acknowledge that I have received all the stated deliverables at the agreed-upon quality levels.

Sponsor Signature:
Date: 25/01/2025

* Requirement Elicitation Techniques for the Graduate Admissions Tracking System



1. Interviews:

Conduct interviews with key stakeholders, including admissions officers, applicants, and IT staff, to understand their perspectives on the current admissions process and their expectations for improvement.

* User Needs and Requirements:
	+ What challenges do users currently face in the graduate admissions process?
	+ Can you describe the ideal user experience for tracking and managing graduate applications?
	+ Are there specific features or functionalities that users expect in an admissions tracking system?
* Process and Workflow:
	+ Walk me through the current admissions workflow in your organization. What are the key steps involved in reviewing and processing applications?
	+ Are there any specific compliance or regulatory requirements that need to be addressed in the workflow?
* System Integration:
	+ How does the admissions tracking system need to integrate with existing university systems? Are there third-party systems or services that should be integrated into the platform?
* Performance and Scalability:
	+ What are the expected application volumes for the admissions tracking system? Are there peak times or seasons that may impact system performance?
	+ How do you envision the scalability of the system as the number of applicants grows?
1. Workshops:

Organize workshops involving the admissions team to collaboratively identify pain points, inefficiencies, and potential solutions for streamlining the graduate admissions process.

* Process Mapping and Workflow:
	+ Let's create a visual representation of the end-to-end admissions workflow. What are the key stages?
	+ Where are the decision points, and what criteria are used for application approval or rejection?
	+ Are there bottlenecks or inefficiencies in the current process that we should address?
* Role Definition and Permissions:
	+ Define and clarify the roles involved in the admissions system (e.g., applicant, admissions officer, admin).
	+ Discuss the specific responsibilities and permissions associated with each user role. Are there role-based workflows or approvals that need to be incorporated into the system?
1. Prototyping:

Create interactive prototypes of the proposed admissions tracking system to gather feedback from end-users and stakeholders on the system's user interface and functionality.

* Creating a Prototype:
	+ Based on the identified features, develop a basic prototype. This prototype can be a simplified version of the user interface or a functional representation of a specific workflow within the system.
	+ The prototype allows stakeholders to visualize how the system will look and feel. Users can interact with the prototype to understand the flow of the application.
	+ Stakeholders, including end-users, have the opportunity to provide feedback on the prototype. This can include comments on the user interface, workflow, and overall user experience.
1. Document Analysis:

Review existing admissions documents, manuals, and reports to gain insights into the current system's workflows, pain points, and areas that need improvement.

* Reviewing Existing Documents:
	+ Business analysts examine various documents related to the admissions process.
* Identifying Stakeholders:
	+ Stakeholder lists and organizational charts are reviewed to identify key individuals and groups involved in or affected by the admissions tracking system.
* Analyzing Requirements Specifications:
	+ If there are existing requirements documents, business analysts analyze them to understand the functional and non-functional requirements that have been previously identified.
* Building a Knowledge Base:
	+ The information gathered through document analysis contributes to building a knowledge base for the project team.
* Feasibility Study

As a Business Analyst, my approach to handling change requests in the Graduate Admissions Tracking System is methodical and follows a clear process. Here’s a summarized version tailored to the system:

1. Document Change Requests:

* Action: Meticulously document any change requests that arise during the project lifecycle.
* Context: For the Graduate Admissions Tracking System, this could involve requests for additional features, modifications in workflows, or changes in data handling processes.

2. Evaluate Change Worthiness:

* Action: Assess the importance and relevance of the change, prioritizing it based on its alignment with the project’s objectives.
* Context: Prioritize changes that enhance user experience, improve system efficiency, or ensure compliance with academic regulations.

3. Approval from Project Manager:

* Action: Seek approval from the Project Manager for the proposed change.
* Context: If the change aligns with the goals of streamlining the admissions process, it is then escalated for formal consideration.

4. Approval Confirmation:

* Action: Obtain formal approval from the Project Manager and other relevant stakeholders via email, ensuring a traceable record.
* Context: This could involve approvals from the admissions department head or IT lead, depending on the nature of the change.

5. Determine Change Impact:

* Action: Collaborate with the Project Manager to assess whether the change is minor or major, ensuring clarity on its potential impacts.
* Context: For example, adding a new reporting feature might be a minor change, while overhauling the user interface could be a major one.

6. Maintain Delivery Timeline:

* Action: For major changes, ensure that the project’s delivery date and scope remain on track, managing expectations and risks accordingly.
* Context: This might involve negotiating deadlines or adjusting project milestones to accommodate significant changes.

7. Stakeholder Communication:

* Action: Facilitate communication with stakeholders to help them understand the importance of the change request and mitigate any potential negative impacts.
* Context: Clear communication is crucial, especially with faculty and administrative staff who rely on the system for managing admissions.

**Document 3: Functional Specifications**

Project Name: Graduate Admissions Tracking System (GATS)

| Project Name | GATS |
| --- | --- |
| Customer Name | Nexon University |
| Project Version  | V0.9 |
| Project Sponser | Mr Venkat |
| Project Manager | Mr Rohit |
| Project Initiation Date | 01/02/2025 |

This document will outline the detailed functional requirements of the Graduate Admissions Tracking System, ensuring that all system features align with the needs of the admissions process and contribute to a seamless and efficient user experience.

### **Functional Requirement Specifications for the Graduate Admissions Tracking System**

| **Req ID** | **Req Name** | **Req Description** | **Priority** |
| --- | --- | --- | --- |
| FR0001 | Login | Registered users (e.g., applicants, administrators) should be able to securely log in to their accounts. | 10 |
| FR0002 | Role-based Authorization | Users should be assigned roles (e.g., applicant, admissions officer, admin) based on their privileges. | 8 |
| FR0003 | Password Encryption | User passwords should be securely encrypted for storage and transmission. | 10 |
| FR0004 | Account Lockout | Implement account lockout functionality after multiple unsuccessful login attempts to enhance security. | 7 |
| FR0005 | Application Submission | Applicants should have the ability to submit admission applications online, providing necessary details and attaching supporting documents. | 10 |
| FR0006 | Application Form Validation | The system should validate the submitted application form to ensure completeness and accuracy of information. | 8 |
| FR0007 | Document Upload | Users should be able to securely upload and attach supporting documents (e.g., transcripts, identification) to their applications. | 9 |
| FR0008 | Application Status Tracking | Applicants and authorized personnel should be able to track the status of their applications in real-time. | 7 |
| FR0009 | Application Processing Workflow | Define a workflow for application processing, including stages such as application verification, review, and approval/rejection. | 10 |
| FR0010 | Application Verification | The system should have a stage for verifying the details provided in the application for accuracy and completeness. | 8 |
| FR0011 | Eligibility Checks | Conduct eligibility checks on applicants to assess their qualification for the program. | 8 |
| FR0012 | Approval/Rejection | Implement stages for approving or rejecting applications based on predefined criteria and assessments. | 10 |
| FR0013 | Document Management | Allow users to upload and manage documents related to applications securely. | 8 |
| FR0014 | Admission Decision Notification | Specify the process for notifying applicants of their admission decision (acceptance, waitlist, rejection). | 10 |
| FR0015 | Enrollment Management | Implement features for admitted students to view enrollment details and complete the enrollment process online. | 9 |
| FR0016 | User Support and Helpdesk | Implement a user support and helpdesk system to assist users with queries, issues, and provide guidance. | 8 |
| FR0017 | Security Measures | Specify security measures such as encryption, secure connections, and data protection to ensure the confidentiality of user information. | 9 |
| FR0018 | Integration with University Systems | If applicable, specify requirements for integration with other university systems for seamless data exchange. | 7 |
| FR0019 | Reporting and Analytics | Provide reporting features for administrators and managers to track the performance of the admission process. | 8 |

###

###

### **Document 4: Requirement Traceability Matrix for the Graduate Admissions Tracking System**

| **Req ID** | **Req Name** | **Req Description** | **Design** | **D1** | **T1** | **D2** | **T2** | **UAT** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| REQ001 | Login | Registered users should be able to securely log in to their accounts. | Yes | Yes | Yes | Yes | Yes | Yes |
| REQ002 | Role-based Authorization | Users should be assigned roles based on their privileges (e.g., applicant, admissions officer, admin). | Yes | Pending | No | Yes | Yes | Yes |
| REQ003 | Password Encryption | Users should change passwords every 3 months. | Yes | Pending | No | Yes | Yes | Yes |
| REQ004 | Account Lockout | Implement account lockout functionality after multiple unsuccessful login attempts to enhance security. | Yes | Pending | No | Yes | Yes | Yes |
| REQ005 | Application Submission | Applicants should have the ability to submit applications online, providing necessary details and attaching supporting documentation. | Yes | Yes | Yes | Yes | Yes | Yes |
| REQ006 | Application Form Validation | The system should validate the submitted application form to ensure completeness and accuracy of information. | Yes | No | No | Yes | Yes | Yes |
| REQ007 | Document Upload | Users should be able to securely upload and attach supporting documents (e.g., transcripts, identification) to their applications. | No | No | No | No | No | No |
| REQ008 | Application Status Tracking | Applicants and authorized personnel should be able to track the status of their applications in real-time. | Yes | Yes | Yes | Yes | Yes | Yes |
| REQ009 | Application Processing Workflow | Define a workflow for application processing, including stages such as application verification, review, and approval/rejection. | Yes | Yes | Yes | Yes | Yes | Yes |
| REQ010 | Application Verification | The system should have a stage for verifying the details provided in the application for accuracy and completeness. | Yes | No | No | Yes | Yes | Yes |
| REQ011 | Eligibility Checks | Conduct eligibility checks on applicants to assess their qualification for the program. | Yes | No | No | Yes | Yes | Yes |
| REQ012 | Approval/Rejection | Implement stages for approving or rejecting applications based on predefined criteria and assessments. | Yes | No | No | Yes | Yes | Yes |
| REQ013 | Document Management | Allow users to upload and manage documents related to applications securely. | Yes | Yes | Yes | Yes | Yes | Yes |
| REQ014 | Admission Decision Notification | Specify the criteria for notifying applicants of their admission decision. | Yes | No | No | Yes | Yes | Yes |
| REQ015 | Enrollment Management | Implement features for admitted students to view enrollment details and complete the enrollment process online. | Yes | Pending | No | Yes | Yes | Yes |
| REQ016 | User Support and Helpdesk | Implement a user support and helpdesk system to assist users with queries, issues, and provide guidance. | Yes | No | No | Yes | Yes | Yes |
| REQ017 | Security Measures | Specify security measures such as encryption, secure connections, and data protection to ensure the confidentiality of user information. | Yes | No | No | Yes | Yes | Yes |
| REQ018 | Integration with University Systems | If applicable, specify requirements for integration with other university systems for seamless data exchange. | Yes | Pending | No | Yes | Yes | Yes |
| REQ019 | Reporting and Analytics | Provide reporting features for administrators and managers to track the performance of the admission process. | Yes | Yes | Yes | Yes | Yes | Yes |

**Document 5 BRD Template**

**<GRADUATE ADMISSIONS TRACKING SYSTEM(GATS)>**

**<2025-147>**

**<V0.9>**

**<SWATI KURWADE>**

**Contents**

1. Document Revision
2. Approvals
3. RASCI Chart for This Document
 Codes Used in RASCI Chart
 RASCI Chart
4. Introduction
 4.1. Business Goals
 4.2. Business Objectives
 4.3. Business Rules
 4.4. Background
 4.5. Project Objective
 4.6. Project Scope
 4.6.1. In-Scope Functionality
 4.6.2. Out-of-Scope Functionality
5. Assumptions
6. Constraints
7. Risks
 Technological Risks
 Skills Risks
 Political Risks
 Business Risks
 Requirements Risks
 Other Risks
8. Business Process Overview
 8.1. Legacy System (AS-IS)
 8.2. Proposed Recommendations (TO-BE)
9. Business Requirements
10. Appendices
 10.1. List of Acronyms
 10.2. Glossary of Terms
 10.3. Related Documents

**1. Document Revisions**

| **Date** | **Version Number** | **Document Changes** |
| --- | --- | --- |
| 29/12/2024 | 0.1 | Initial Draft |
| 05/01/2025 | 0.3 | Updated Business Goals and Objectives |
| 12/01/2025 | 0.6 | Incorporated Stakeholder Feedback on Assumptions |
| 19/01/2025 | 0.8 | Modified Business Process Overview based on Team Input |
| 23/01/2025 | 0.9 | Reviewed and Updated Business Requirements |

**2. Approvals**

| **Role** | **Name** | **Title** | **Signature** | **Date** |
| --- | --- | --- | --- | --- |
| Project Sponsor | Dr. Venkat | Project Sponsor |  | 24/01/2025 |
| Business Owner | Ms. Sanika | Admissions Director |  | 24/01/2025 |
| Project Manager | Mr. Rohit | Project Manager |  | 24/01/2025 |
| System Architect | Ms. Richa | System Architect |  | 24/01/2025 |
| Development Lead | Ms. Diksha | Development Lead |  | 24/01/2025 |
| User Experience Lead | Ms. Kirti | User Experience Lead |  | 24/01/2025 |
| Quality Lead | Mr. Kunal | Quality Lead |  | 24/01/2025 |
| Content Lead | Mr. Aditya | Content Lead |  | 24/01/2025 |

**3. RACI Chart for This Document**

The RACI chart identifies the persons who need to be contacted whenever changes are made to this document. RACI stands for responsible, accountable, consulted, and informed. The codes used in the chart are described below:

| **Codes Used in RACI Chart** |
| --- |
| **A**: Accountable - Has ultimate signing authority for any changes to the document. |
| **R**: Responsible - Responsible for creating this document. |
| **C**: Consulted - Provides input (such as an interviewee). |
| **I**: Informed - Must be informed of any changes. |

**RACI Chart**

| **Task** | **Sponsor** | **PM** | **Developer** | **Tester** | **Stakeholders** | **BA** |
| --- | --- | --- | --- | --- | --- | --- |
| Requirements Gathering | A/I | A | C | I | R | R |
| Analysis | I | R | C | I | I | R |
| Development | I | I | R | C | I | C |
| Testing | I | I | I | R | C | I |
| Implementation | I | I | R | I | I | C |
| UAT | I | I | C | R | I | A |

**4. Introduction**

**4.1. Business Goals** To establish a streamlined Graduate Admissions Tracking System, enhancing application management, review efficiency, and providing a superior user experience for applicants and staff. This project focuses on digital transformation to improve operational effectiveness and transparency.

**4.2. Business Objectives**

* **Application Management:** Digitize and centralize the graduate admissions process to ensure seamless application handling and tracking.
* **Workflow Automation:** Introduce automated review and decision-making workflows to reduce manual intervention.
* **Role-Based Access and Security:** Ensure secure, role-specific access for applicants, reviewers, and administrators.
* **Data-Driven Insights:** Enable comprehensive reporting and analytics to optimize admissions strategies and decisions.

**4.3. Business Rules**

1. Applications must include all required documents before submission.
2. Only designated reviewers can approve or reject applications.
3. Automated reminders will notify stakeholders about pending actions or incomplete applications.
4. Admissions decisions must adhere to institutional policies and deadlines.

**4.4. Background** The current admissions process relies heavily on manual interventions, leading to inefficiencies, errors, and delays. Rising application volumes necessitate a scalable digital solution to handle admissions systematically while maintaining accuracy and compliance.

**4.5. Project Objective** Develop an intuitive Graduate Admissions Tracking System to:

* Enhance applicant experience with user-friendly interfaces.
* Provide robust tools for administrators and reviewers to manage applications.
* Ensure seamless integration with existing university systems.
* Deliver data insights for continuous improvement in the admissions process.

**4.6. Project Scope**

**4.6.1. In-Scope Functionality**

* User authentication and secure login mechanisms.
* Application submission, document uploads, and tracking.
* Automated workflows for application review and decisions.
* Real-time status updates and notifications.
* Role-based dashboards and access controls.
* Integration with existing student management systems.
* Comprehensive reporting and analytics tools.

**4.6.2. Out-of-Scope Functionality**

* Mobile-specific application features (beyond standard browser compatibility).
* Major updates to existing university-wide IT infrastructure.
* Development of third-party integrations for external recruitment services.

**5. Assumptions**

* Stakeholders will actively participate in requirements gathering and testing.
* Sufficient hardware and IT resources will be available to support deployment.
* The project team has the necessary technical expertise to execute deliverables.

**6. Constraints**

* Timeline: Project completion within 6 months.
* Budget: Restricted to 50,00,000.
* Regulatory compliance with institutional and regional data security laws.

**7. Risks**

#### **1. Technological Risks**

* **System Downtime**: Unplanned outages due to server or infrastructure issues.
* **Cybersecurity Threats**: Data breaches or unauthorized access exposing sensitive information.
* **Integration Challenges**: Difficulties integrating with existing university systems.
* **Data Loss or Corruption**: Database failures or lack of backups causing permanent data loss.
* **Scalability Issues**: Inability to handle peak loads or increased user traffic.
* **Outdated Technology**: Use of obsolete tools making upgrades difficult.
* **Poor Performance**: Slow system response impacting user experience.

#### **2. Skills Risks**

* **Inadequate Training**: Staff may lack proper training on the system.
* **Shortage of IT Expertise**: Limited technical resources for managing the system.
* **User Resistance**: Reluctance of staff or applicants to adopt the new system.
* **High Dependency on Key Individuals**: Over-reliance on a few critical team members.

#### **3. Political Risks**

* **Policy Changes**: Regulatory or institutional policy changes requiring system updates.
* **Stakeholder Conflicts**: Disputes between departments on system priorities.
* **Budget Cuts**: Reduced funding affecting system development or maintenance.
* **Legal Risks**: Non-compliance with data privacy regulations (e.g., GDPR, FERPA).

#### **4. Business Risks**

* **Budget Overruns**: Unexpected costs causing project delays.
* **Missed Deadlines**: System delays disrupting the admissions cycle.
* **Inefficient Processes**: Failure to streamline workflows effectively.
* **Lack of User Adoption**: Poor user interface or trust impacting adoption rates.
* **Reputation Damage**: Failures during critical periods harming institutional reputation.

#### **5. Other Risks**

* **Third-Party Dependency**: Reliance on vendors for hosting or support introducing risks.
* **Change Management**: Resistance to transitioning from manual to digital workflows.
* **Cultural Barriers**: Difficulty for diverse applicants understanding system requirements.
* **Environmental Risks**: Natural disasters affecting operations or data centers.

**8. Business Process Overview**

**8.1. Legacy System (AS-IS)**

* Predominantly manual workflows for managing applications and reviews.
* High dependency on paper-based submissions and communication.
* Challenges in tracking application statuses across departments.

**8.2. Proposed Recommendations (TO-BE)**

* A fully digital admissions system offering end-to-end application tracking.
* Automation of key steps in the admissions lifecycle, such as document validation and decision notifications.
* Centralized dashboards for administrators and streamlined communication tools.

**9. Business Requirements** Detailed functional and non-functional requirements will focus on:

* Applicant onboarding and application submission processes.
* Reviewer tools for assessing applications efficiently.
* Dashboards for real-time monitoring and analytics.
* Secure storage and retrieval of application data.

**10. Appendices**

**10.1. List of Acronyms**

* BA: Business Analyst
* PM: Project Manager
* LMS: Lead Management System
* UI: User Interface
* API: Application Programming Interface
* ITS: Information Technology Services
* RFP: Request for Proposal
* UAT: User Acceptance Testing

**10.2. Glossary of Terms**

* **Online Application Processing:** The digital system for managing admission applications and processing.
* **User Authentication**: The process of verifying and confirming the identity of users during login.
* **RACI Matrix**: A responsibility matrix that defines the roles and responsibilities of individuals in a project.
* **SDLC**: Software Development Life Cycle, a framework for structuring, planning, and controlling the process of developing an information system.
* **UAT**: User Acceptance Testing, the final phase of software development to ensure the system meets user requirements.
* **Application Workflow:** A series of steps from submission to decision-making in the admissions process.
* **Stakeholder:** Any individual or group with a vested interest in the admissions process.
* **Dashboard:** A user interface summarizing key metrics and tasks.

**10.3. Related Documents**

* System Design Document
* Use Case Specifications

**System Design Document**

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###



###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

###

### **Use Case Specifications for Graduate Admissions Tracking System**

| **Use Case ID** | **UC001** |
| --- | --- |
| **Use Case Name** | Login |
| **Created by** | Business Analyst |
| **Use Case Description** | This use case allows users (applicants, admissions officers, and admins) to securely log in to the Graduate Admissions Tracking System. |
| **Actors** | * **Primary Actor:** User (Applicant, Admissions Officer, Admin)
* **Secondary Actor:** System, Database
 |
| **Basic Flow** | 1. User navigates to the login page.
2. User enters their username and password.
3. System validates the credentials.
4. If valid, the system logs in the user and redirects them to their dashboard.
 |
| **Alternate Flow** | * **Forgot Password:**
	1. User clicks "Forgot Password."
	2. System prompts the user to enter their email.
	3. System sends a password reset link to the email.
 |
| **Exceptional Flow** | * **Invalid Credentials:**
	1. System displays an error message for invalid credentials.
	2. User is prompted to re-enter credentials.
 |
| **Pre-conditions** | * User must be registered in the system.
 |
| **Post-conditions:** | * User gains access to their dashboard.
 |
| **Assumptions** | * User has an active account.
 |
| **Constraints** | * The login process must be completed within 60 seconds of inactivity.
 |
| **Dependencies** | * System must connect to the database for user verification.
 |
| **Inputs and Outputs** | * **Inputs:** Username, password.
* **Outputs:** Dashboard access or error message.
 |
| **Business Rules** | * User must provide correct credentials.
 |
| **Miscellaneous Information** | * The system supports multi-factor authentication in future updates.
 |

| **Use Case ID** | **UC002** |
| --- | --- |
| **Use Case Name** | Submit Application |
| **Created by** | Business Analyst |
| **Use Case Description** | Applicants can submit their graduate application online, including providing personal details and uploading documents. |
| **Actors** | * **Primary Actor:** Applicant
* **Secondary Actor:** System, Database
 |
| **Basic Flow** | 1. Applicant logs in to the system.
2. Applicant navigates to the "Submit Application" section.
3. Applicant fills in the required details.
4. Applicant uploads supporting documents.
5. Applicant clicks "Submit."
6. System validates the information and confirms the submission.
 |
| **Alternate Flow** | * **Incomplete Submission:**
	1. System highlights missing or incorrect fields.
	2. Applicant corrects errors and re-submits.
 |
| **Exceptional Flow** | * **File Upload Error:**
	1. System rejects unsupported file types.
	2. Applicant is prompted to upload files in the correct format.
 |
| **Pre-conditions** | * Applicant must have an active account and be logged in.
 |
| **Post-conditions** | * Application is submitted and stored in the database.
 |
| **Assumptions** | * Applicant has all required documents ready.
 |
| **Constraints** | * File uploads must not exceed 10MB per document.
 |
| **Dependencies** | * System must validate and store the application data.
 |
| **Inputs and Outputs** | * **Inputs:** Applicant's personal details, uploaded documents.
* **Outputs:** Confirmation message.
 |
| **Business Rules** | * All required fields must be completed before submission.
 |
| **Miscellaneous Information** | * The system supports multiple file types for uploads.
 |

| **Use Case ID** | **UC003** |
| --- | --- |
| **Use Case Name** | Track Application Status |
| **Created by** | Business Analyst |
| **Use Case Description** | This use case allows applicants to track the real-time status of their submitted applications. |
| **Actors** | * **Primary Actor:** Applicant
* **Secondary Actor:** System
 |
| **Basic Flow** | 1. Applicant logs in to the system.
2. Applicant navigates to the "Application Status" page.
3. System displays the current status of the application (e.g., submitted, under review, accepted).
 |
| **Alternate Flow** | * **Multiple Applications:**
	1. Applicant selects a specific application from a list.
	2. System displays the status of the selected application.
 |
| **Exceptional Flow** | * **System Error:**
	1. If the system cannot retrieve the status, an error message is displayed.
	2. Applicant is advised to try again later.
 |
| **Pre-conditions** | * Applicant must have submitted at least one application.
 |
| **Post-conditions** | * Application status is displayed to the applicant.
 |
| **Assumptions** | * All applications have updated statuses in the database.
 |
| **Constraints** | * The system must handle concurrent status requests efficiently.
 |
| **Dependencies** | * Application data must be updated in the database.
 |
| **Inputs and Outputs** | * **Inputs:** Application ID or selection.
* **Outputs:** Application status.
 |
| **Business Rules** | * Status updates must be accurate and reflect the latest processing stage.
 |
| **Miscellaneous Information** | * The system allows viewing of historical statuses if needed.
 |

| **Use Case ID** | **UC004** |
| --- | --- |
| **Use Case Name** | Approve or Reject Applications |
| **Created by** | Business Analyst |
| **Use Case Description** | Admissions officers review and make decisions on submitted applications based on predefined criteria. |
| **Actors:** | * **Primary Actor:** Admissions Officer
* **Secondary Actor:** System
 |
| **Basic Flow:** | 1. Admissions officer logs in to the system.
2. Officer navigates to the "Application Review" section.
3. Officer selects an application for review.
4. Officer reviews the applicant's details and documents.
5. Officer approves or rejects the application.
6. System updates the application status accordingly.
 |
| **Alternate Flow:** | * **Request Additional Information:**
	1. Officer requests additional details or documents from the applicant.
	2. System notifies the applicant about the request.
 |
| **Exceptional Flow:** | * **System Timeout:**
	1. If the system times out during the review, progress is saved automatically.
	2. Officer resumes the review upon re-login.
 |
| **Pre-conditions:** | * The application must be submitted and verified.
 |
| **Post-conditions:** | * Application status is updated (approved, rejected, or pending additional details).
 |
| **Assumptions:** | * Officers have access to complete applicant information.
 |
| **Constraints:** | * The review process must be completed within predefined deadlines.
 |
| **Dependencies:** | * System must provide accurate and complete application details.
 |
| **Inputs and Outputs:** | * **Inputs:** Application details, documents.
* **Outputs:** Updated application status.
 |
| **Business Rules:** | * Applications must meet eligibility criteria for approval.
 |
| **Miscellaneous Information:** | * The system tracks review progress and timestamps decisions.
 |

| **Use Case ID:**  | **UC005** |
| --- | --- |
| **Use Case Name:**  | Document Upload |
| **Created by:**  | Business Analyst |
| **Use Case Description:**  | This use case allows applicants to upload supporting documents for their applications. |
| **Actors:** | * **Primary Actor:** Applicant
* **Secondary Actor:** System
 |
| **Basic Flow:** | 1. Applicant navigates to the "Upload Documents" section.
2. Applicant selects the document type and file from their device.
3. System verifies the file format and size.
4. System uploads and stores the document securely.
5. Applicant receives a confirmation message.
 |
| **Alternate Flow:** | * **Multiple Documents:**
	1. Applicant uploads multiple documents in one session.
	2. System processes each document sequentially.
 |
| **Exceptional Flow:** | * **Unsupported Format:**
	1. System rejects unsupported file formats.
	2. Applicant is prompted to upload a valid file.
 |
| **Pre-conditions:** | * Applicant must be logged in.
 |
| **Post-conditions:** | * Documents are securely stored and linked to the application.
 |
| **Assumptions:** | * Applicants have digital copies of their documents.
 |
| **Constraints:** | * File size must not exceed 10MB per document.
 |
| **Dependencies:** | * System must have sufficient storage and processing capacity.
 |
| **Inputs and Outputs:** | * **Inputs:** Document files, document type.
* **Outputs:** Confirmation message, stored documents.
 |
| **Business Rules:** | * Only required document types can be uploaded.
 |
| **Miscellaneous Information:** | * The system supports drag-and-drop functionality for file uploads.
 |

| **Use Case ID:**  | **UC006** |
| --- | --- |
| **Use Case Name:**  | Application Form Validation |
| **Created by:**  | Business Analyst |
| **Use Case Description:**  | Validates the application form data to ensure all required fields are completed accurately. |
| **Actors:** | * **Primary Actor:** System
* **Secondary Actor:** Applicant
 |
| **Basic Flow:** | 1. Applicant submits the application form.
2. System checks for completeness of all required fields.
3. System validates the format and content of the entered data.
4. If validation passes, the application form is marked as "Ready for Review."
 |
| **Alternate Flow:** | * **Incomplete Form:**
	1. If required fields are missing, the system highlights the incomplete sections.
	2. Applicant is prompted to complete the missing fields before resubmission.
 |
| **Exceptional Flow:** | * **Invalid Data:**
	1. If the format of certain fields is incorrect (e.g., invalid email or phone number), the system provides an error message.
	2. Applicant must correct the errors and resubmit the form.
 |
| **Pre-conditions:** | * Applicant must be logged in.
* The application form must be accessible.
 |
| **Post-conditions:** | * The application form is validated and marked as "Ready for Review" if all checks are passed.
 |
| **Assumptions:** | * Applicants have the necessary information to complete the form.
 |
| **Constraints:** | * Data validation must comply with institutional standards.
 |
| **Dependencies:** | * The database must be available for data validation checks.
 |
| **Inputs and Outputs:** | * **Inputs:** Application form data.
* **Outputs:** Validation status, error messages (if any).
 |
| **Business Rules:** | * All required fields must be completed before submission.
* Data must adhere to specified formats and constraints.
 |
| **Miscellaneous Information:** | * System supports real-time validation and auto-save functionality during form filling.
 |

| **Use Case ID:**  | **UC007** |
| --- | --- |
| **Use Case Name:**  | Payment Processing |
| **Created by:** | Business Analyst |
| **Use Case Description:** | Facilitates secure processing of application fees for applicants. |
| **Actors:** | * **Primary Actor:** Applicant
* **Secondary Actor:** Payment Gateway, System
 |
| **Basic Flow:** | 1. Applicant navigates to the "Payment" section of the application.
2. Applicant selects a payment method (e.g., credit card, debit card, online banking).
3. System redirects the applicant to the payment gateway.
4. Applicant enters payment details and confirms the transaction.
5. Payment gateway processes the transaction and sends a success or failure response to the system.
6. System updates the application status to "Payment Completed" if the transaction is successful.
7. Applicant receives a payment confirmation message and receipt.
 |
| **Alternate Flow:** | * **Payment Failure:**
	1. If the transaction fails, the system notifies the applicant with an error message.
	2. Applicant is prompted to retry the payment or use an alternate payment method.
 |
| **Exceptional Flow:** | * **Gateway Timeout:**
	1. If the payment gateway times out, the system retries the transaction up to three times.
	2. If the retries fail, the system notifies the applicant and logs the issue for review.
 |
| **Pre-conditions:** | * Applicant must have completed the application form.
* The payment gateway must be operational.
 |
| **Post-conditions:** | * Payment is successfully processed, and the application status is updated.
 |
| **Assumptions:** | * Applicants have a valid payment method.
 |
| **Constraints:** | * Payment must be completed within the application deadline.
 |
| **Dependencies:** | * Payment gateway service must be integrated and operational.
 |
| **Inputs and Outputs:** | * **Inputs:** Payment details (e.g., card number, expiry date).
* **Outputs:** Payment confirmation message, updated application status.
 |
| **Business Rules:** | * Payments must comply with financial regulations and institutional policies.
 |
| **Miscellaneous Information:** | * The system supports multiple payment methods for user convenience.
 |

| **Use Case ID:**  | **UC008** |
| --- | --- |
| **Use Case Name:**  | Notify Admission Decision |
| **Created by:** | Business Analyst |
| **Use Case Description:** | Notify applicants about their admission decisions (e.g., accepted, waitlisted, rejected) through email and system notifications. |
| **Actors:** | * **Primary Actor:** System
* **Secondary Actor:** Applicant
 |
| **Basic Flow:** | 1. Admission decision is finalized by the admissions officer.
2. System generates a notification for the applicant.
3. Notification is sent to the applicant's registered email and displayed on their dashboard.
 |
| **Alternate Flow:** | * **Notification Failure:**
	1. System retries sending the notification if the email fails initially.
 |
| **Exceptional Flow:** | * **Invalid Email Address:**
	1. System logs the error and flags the applicant's account for manual review.
 |
| **Pre-conditions:** | * Admission decision must be finalized.
 |
| **Post-conditions:** | * Applicant receives their admission decision.
 |
| **Assumptions:** | * Applicant’s email address is valid and active.
 |
| **Constraints:** | * Notification must be delivered within 1 hour of decision finalization.
 |
| **Dependencies:** | * System email service must be operational.
 |
| **Inputs and Outputs:** | * **Inputs:** Admission decision.
* **Outputs:** Notification to applicant.
 |
| **Business Rules:** | * Notifications must comply with institutional communication policies.
 |
| **Miscellaneous Information:** | * Notifications include next steps for accepted applicants.
 |

| **Use Case ID:** | **UC009** |
| --- | --- |
| **Use Case Name:** | Account Lockout |
| **Created by:** | Business Analyst |
| **Use Case Description:** | Enhance security by locking user accounts after multiple unsuccessful login attempts. |
| **Actors:** | * **Primary Actor:** System
* **Secondary Actor:** User
 |
| **Basic Flow:** | 1. User enters incorrect credentials.
2. System tracks the number of failed attempts.
3. After three failed attempts, the account is locked.
4. System notifies the user of the lockout and provides recovery options.
 |
| **Alternate Flow:** | * **Recovery Process:**
	1. User clicks on the account recovery link.
	2. System verifies the user’s identity through email or security questions.
 |
| **Exceptional Flow:** | * **Invalid Recovery Attempts:**
	1. System blocks further recovery attempts after three invalid attempts.
 |
| **Pre-conditions:** | * User must have an existing account.
 |
| **Post-conditions:** | * Account is locked and flagged for security.
 |
| **Assumptions:** | * Users will not intentionally exceed login attempts.
 |
| **Constraints:** | * Lockout duration is configurable by the admin.
 |
| **Dependencies:** | * System’s user authentication service.
 |
| **Inputs and Outputs:** | * **Inputs:** Login credentials.
* **Outputs:** Lockout notification.
 |
| **Business Rules:** | * Lockout rules must comply with security best practices.
 |
| **Miscellaneous Information:** | * Admins can manually unlock accounts.
 |

| **Use Case ID:** | **UC010** |
| --- | --- |
| **Use Case Name:**  | Application Verification |
| **Created by:**  | Business Analyst |
| **Use Case Description:**  | Verify the completeness and authenticity of submitted applications. |
| **Actors:** | * **Primary Actor:** Admissions Officer
* **Secondary Actor:** System
 |
| **Basic Flow:** | 1. Admissions officer reviews submitted applications.
2. System flags incomplete or suspicious entries.
3. Officer marks applications as "verified" or "requires additional details."
 |
| **Alternate Flow:** | * **Auto Verification:**
	1. System performs preliminary checks (e.g., mandatory fields, file formats).
 |
| **Exceptional Flow:** | * **False Verification:**
	1. Officer identifies and corrects errors flagged by the system.
 |
| **Pre-conditions:** | * Application must be submitted.
 |
| **Post-conditions:** | * Application verification status is updated.
 |
| **Assumptions:** | * Officers have access to all necessary tools and data.
 |
| **Constraints:** | * Verification process must be completed within 7 business days.
 |
| **Dependencies:** | * Applicant’s supporting documents.
 |
| **Inputs and Outputs:** | * **Inputs:** Submitted application.
* **Outputs:** Verification status.
 |
| **Business Rules:** | * Verified applications proceed to eligibility checks.
 |
| **Miscellaneous Information:** | * System maintains a log of verification actions.
 |

**Screens and pages**

Please follow the following steps to create the mock-ups:

1. Kindly use balsamic or Axure.
2. Always start with a home page of an application.
3. Take a feature and follow it to the end
4. Eg: Home page of Nexon University
5. Select Apply Now - Create a coursework identification page
6. On coursework identification page, you select from various parameters to determine the course best suited for you. If you are satisfied with the course, then Proceed with the choice - land on Application Form page.
7. Application Form - fill in all the required fields; Save or Save & Proceed
8. Land on Upload Document page - upload all the required document in specified fromat and size
9. Payment page - After successful submission of form and documents, page lands you on Application Tracking Page, thereafter Decision Notification page and once you get approval you proceed to payment gateway.
10. Apart from these sequential pages, there are Admin Dashboard, Logout Page, etc.

















