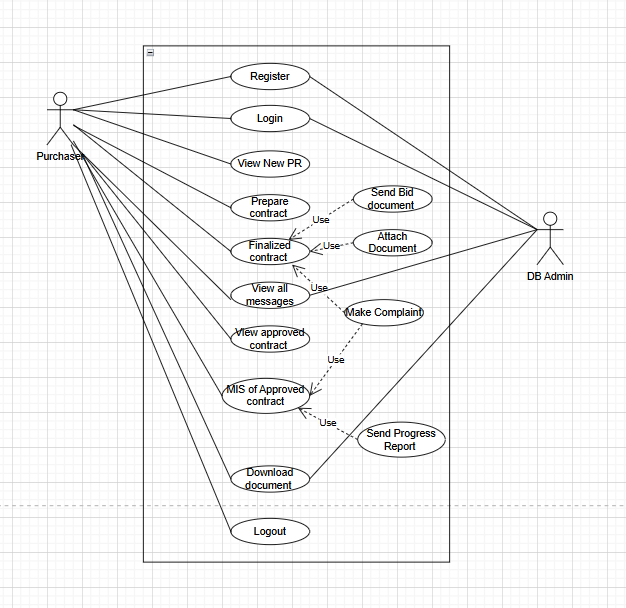
**Document 6- Please prepare a use case diagram, activity diagram and a use case specification document.**



**Use case Specification**

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 1 |
| Use Case Name | User Registering in the Highrise ERP System |
| Brief Description | This Use Case describes how a user registers in the Highrise ERP system to access purchase and procurement functionalities. |
| Actors | 1. Employees (Purchasing Staff, Procurement Managers) |
| 2. Database Administrator (Admin) |
| 3. System (Highrise ERP Application) |
| Pre-Conditions | 1. Active Internet connection |
| 2. Supported device (Laptop/Mobile) |
| 3. Valid Email Address and Mobile Number |
| 4. ERP system operational |
| Basic Flow | 1. User initiates registration by entering personal details: Name, Job Title, Company Registered ID, Email Address, and Password. |
| 2. The system sends an OTP to the provided email address for verification. |
| 3. User enters the OTP received in their email. |
| 4. System verifies OTP and checks the validity of the Company Registered ID and password. |
| 5. If the details are correct, the user account is created successfully. |
| 6. User gains access to the system and can start using purchase/procurement features. |
| Alternate Flow | 1. Invalid OTP: If incorrect OTP is entered, system displays: "Please enter the correct OTP." |
| 2. Company Registered ID Already Used: If the entered company registered ID is already in use, system displays: "Company Registered ID is already in use. Please enter a different ID." |
| 3. Email Address Already Used: If the email address is already registered, the system displays: "Email address is already in use. Please enter a different email address." |
| 4. Password Issues: If the entered password does not meet security criteria (e.g., insufficient length or complexity), the system displays: "Password does not meet the required criteria. Please enter a valid password." |
| 5. Server or Network Issue: If the system faces a server issue, system displays: "Server busy. Please try again later." |
| Post Condition | 1. User accounts are successfully created and stored in the database. |
| 2. User can access purchase and procurement features. |
| 3. System logs user details, timestamp of registration, and actions. |
| Supplemental Requirements | 1. Security: Personal data is stored securely, adhering to data protection regulations. |
| 2. Logging: All registration attempts, including successful and failed ones, are logged with timestamps and error messages. |
| 3. OTP Expiry: OTP expires after 5 minutes of being sent. |
| Assumptions | 1. User has access to a working email account and mobile number. |
| 2. User has a basic understanding of how to register for an online system. |
| 3. The company registered ID is unique and can be checked against existing records. |
| Constraints | 1. The OTP sent to the user’s email has a validity period of 5 minutes. |
| 2. User password must meet defined security criteria (e.g., minimum length, use of alphanumeric characters, special symbols). |
| 3. The system can only handle a certain number of concurrent registration requests at any given time, depending on server capacity. |
| Dependencies | 1. The system must have an email service configured to send OTPs. |
| 2. The user must have an active, valid email account. |
| 3. The company registered ID must be unique and linked to the user's company records. |
| Inputs | 1. Full Name, Job Title, Company Registered ID, Email Address, Password |
| 2. OTP sent to the user’s email |
| Outputs | 1. Registration confirmation message |
| 2. Success or error messages depending on the registration status |
| 3. Database entry with user’s registration details (Name, Job Title, Email, Company Registered ID) |
| Business Rules | 1. Company Registered IDs must be unique for each user. |
| 2. Email addresses must be valid and unique across the system. |
| 3. The system must ensure that passwords adhere to company security policies (e.g., minimum length of 8 characters, inclusion of special characters). |
| 4. OTP verification must be completed within 5 minutes of sending. |
| 5. All registration data must be securely stored and comply with relevant data protection laws. |

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 2 |
| Use Case Name | User Login to the Highrise ERP System |
| Brief Description | This Use Case describes how a registered user logs in to the Highrise ERP system to access purchase and procurement functionalities. |
| Actors | 1. Employees (Purchasing Staff, Procurement Managers) |
| 2. Database Administrator (Admin) |
| 3. System (Highrise ERP Application) |
| Pre-Conditions | 1. User must have a valid account in the Highrise ERP system. |
| 2. User must have an active Internet connection. |
| 3. User must be accessing the system from a supported device (Laptop/Mobile). |
| Basic Flow | 1. User navigates to the login page of the Highrise ERP system. |
| 2. User enters their registered Company ID and password. |
| 3. The system validates the entered credentials. |
| 4. If the credentials are correct, the user is logged into the system and is granted access to the dashboard and purchase/procurement features. |
| 5. System logs the login timestamp and details in the database. |
| 6. Use case ends with successful login. |
| Alternate Flow | 1. Invalid Credentials: If the entered Company ID or password is incorrect, the system displays: "Invalid Company ID” or password. Please try again." |
| 2. Account Locked: If the user exceeds the maximum number of failed login attempts, the system displays: "Your account is locked due to multiple failed login attempts. Please reset your password or contact support." |
| 3. Password Reset: If the user clicks on "Forgot Password," the system sends a password reset link to the registered email. The user can reset their password and log in again. |
| 4. Server Issue: If there is a server issue, the system displays: "Server busy. Please try again later." |
| Post Condition | 1. User successfully logs into the system and gains access to purchase and procurement features. |
| 2. System logs the user login details, including timestamp, successful or failed login attempt, and IP address. |
| Supplemental Requirements | 1. Security: The system must securely store user credentials, and passwords should be encrypted. |
| 2. Logging: The system should log login attempts (both successful and failed), including the user's IP address, time of login, and the result of the attempt (success/failure). |
| 3. Session Management: The system must handle user sessions securely, automatically logging out the user after a predefined period of inactivity (e.g., 30 minutes). |
| Assumptions | 1. The user has already registered with the system and has a valid account. |
| 2. The user knows their Company ID and password. |
| 3. The user has an active email address for account recovery (password reset). |
| Constraints | 1. The system must prevent brute force attacks by limiting the number of failed login attempts. |
| 2. User passwords must meet defined security criteria (e.g., minimum length, alphanumeric characters, and special symbols). |
| 3. The system must be able to handle concurrent login requests without performance degradation. |
| Dependencies | 1. The system must have an active database to validate the user credentials. |
| 2. The system must have an email service configured for password reset functionality. |
| 3. The ERP system must be online and operational for login to work. |
| Inputs | 1. Company ID |
| 2. Password |
| 3. Optionally, email address (for password reset) |
| Outputs | 1. Login success message or error message |
| 2. Access granted to the ERP system dashboard and features |
| 3. Successful or failed login attempt logged into the system |
| Business Rules | 1. Company IDs must be unique for each user and used for authentication. |
| 2. Passwords must meet the system's defined security criteria (e.g., at least 8 characters, including both uppercase and lowercase letters, numbers, and special characters). |
| 3. After 5 failed login attempts, the user account is temporarily locked, and a security alert is generated. |
| 4. Users must reset their password through email verification if they forget it. |
| 5. Session expiration occurs after 30 minutes of inactivity, requiring the user to log in again. |

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 3 |
| Use Case Name | Preparation of Contract Document in Highrise ERP |
| Brief Description | This Use Case describes how a user (typically a Procurement Manager or Purchasing Staff) prepares a contract document within the Highrise ERP system for suppliers or vendors as part of the procurement process. |
| Actors | 1. Procurement Manager |
| 2. Purchasing Staff |
| 3. Supplier/Vendor |
| 4. System (Highrise ERP Application) |
| Pre-Conditions | 1. The user must be logged into the Highrise ERP system. |
| 2. A purchase order or agreement must have been created and approved. |
| 3. Supplier or vendor details must be available in the system. |
| Basic Flow | 1. User navigates to the "Contract Preparation" section of the ERP system. |
| 2. The system displays a list of suppliers/vendors eligible for contract creation based on previous transactions or agreements. |
| 3. User selects the relevant supplier/vendor for the contract. |
| 4. The system auto-populates the contract template with key details, including supplier name, contact information, product/service description, agreed terms, payment conditions, delivery timelines, and any other contractual clauses based on the purchase order. |
| 5. User reviews and updates the contract terms as needed, adding any special conditions or clauses. |
| 6. User confirms and generates the final contract document in PDF or Word format. |
| 7. System stores the generated contract document in the system’s repository and attaches it to the corresponding purchase order record. |
| 8. The system sends a notification to the procurement team and the supplier/vendor for review and signature. |
| 9. Use case ends. |
| Alternate Flow | 1. Missing Supplier Details: If supplier/vendor details are incomplete, the system prompts the user: "Missing supplier/vendor information. Please complete the details before proceeding." |
| 2. Contract Template Not Available: If the appropriate contract template is missing or not configured, the system displays: "No contract template available. Please configure the template in system settings." |
| 3. Invalid Purchase Order: If the purchase order linked to the contract is invalid or not found, the system displays: "Invalid purchase order. Please select a valid purchase order for contract preparation." |
| 4. Server or Network Issue: If there is a server or network issue during document preparation, the system displays: "Server busy. Please try again later." |
| Post Condition | 1. A contract document is successfully generated and stored in the ERP system repository. |
| 2. The contract document is linked to the relevant purchase order and supplier/vendor record. |
| 3. Notification is sent to the supplier/vendor for contract review and signature. |
| Supplemental Requirements | 1. Document Security: The contract document should be encrypted and stored securely. |
| 2. Version Control: The system should track any updates to the contract document and maintain the history of the contract and maintain the history. |
| 3. Approval Workflow: The system may integrate an approval workflow where the contract document requires approval from senior management before it is sent to the supplier/vendor. |
| Assumptions | 1. The user has the necessary permission to prepare a contract document for the system. |
| 2. The required supplier/vendor and purchase order data is available in the system. |
| 3. The system has pre-configured contract templates for various types of agreements. |
| Constraints | 1. The contract document must adhere to the company's legal and procurement standards. |
| 2. The contract document should be generated in a format that is compatible with document management systems (e.g., PDF, Word). |
| 3. Only authorized personnel can prepare, modify, or approve contract documents. |
| Dependencies | 1. The system must have access to supplier/vendor records and purchase order information. |
| 2. The system must have pre-configured contract templates. |
| 3. The system must be integrated with an email or notification system to send the contract for review. |
| Inputs | 1. Supplier/vendor selection |
| 2. Purchase order details |
| 3. Pre-configured contract template |
| 4. User-modified contract terms and conditions |
| Outputs | 1. Generated contract document in PDF or Word format |
| 2. Confirmation of successful document generation |
| 3. Notification sent to supplier/vendor for review and signature |
| Business Rules | 1. The contract document must include all required legal and financial terms agreed upon in the purchase order. |
| 2. All contracts must be reviewed and approved by authorized personnel before they are sent to suppliers/vendors. |
| 3. The contract document must be securely stored in the system, with access restricted to authorized users only. |
| 4. The contract document must be versioned and tracked for auditing purposes. |
| 5. Only valid, approved purchase orders can be used for contract preparation. |

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 4 |
| Use Case Name | Finalized Contract with Attachment of Bid Documents and Other Documents |
| Brief Description | This Use Case describes the process of finalizing a contract document, attaching the relevant bid documents, and any additional required documents (e.g., legal agreements, certificates) in the Highrise ERP system. |
| Actors | 1. Procurement Manager |
| 2. Purchasing Staff |
| 3. Supplier/Vendor |
| 4. System (Highrise ERP Application) |
| Pre-Conditions | 1. The contract document has been generated and reviewed. |
| 2. The bidding documents and any additional documents (e.g., legal agreements, certificates, etc.) are available and ready for attachment. |
| 3. The user must be logged into the Highrise ERP system. |
| Basic Flow | 1. User navigates to the finalized contract section in the ERP system. |
| 2. The system displays the previously generated contract document linked to the supplier/vendor and purchase order. |
| 3. User selects the relevant bid documents (e.g., bid proposals, evaluation reports) and other documents (e.g., legal certificates, insurance documents) for attachment. |
| 4. The system allows the user to upload and attach the selected documents to the finalized contract. |
| 5. User reviews the attached documents to ensure completeness and accuracy. |
| 6. Once reviewed, the user confirms the attachment of all documents to the finalized contract. |
| 7. The system stores the finalized contract along with all attached documents in the system’s document repository. |
| 8. The system sends notifications to the procurement team and the supplier/vendor about the finalization and document attachment. |
| 9. Use case ends. |
| Alternate Flow | 1. Missing Documents: If any required bid or other documents are missing, the system displays: "One or more required documents are missing. Please upload the necessary files before proceeding." |
| 2. Invalid Document Format: If the uploaded document is in an unsupported format, the system displays: "Invalid document format. Please upload a valid file (e.g., PDF, DOCX)." |
| 3. Incorrect Document Attachments: If an incorrect document is attached, the system allows the user to remove and re-upload the correct file. |
| 4. File Size Exceeded: If the file size exceeds the allowed limit, the system displays: "File size exceeds the maximum limit. Please upload smaller files." |
| 5. Server or Network Issue: If the system faces a server or network issue during the document upload, the system displays: "Server busy. Please try again later." |
| Post Condition | 1. The contract document is finalized and stored in the ERP system repository with all attached bid documents and other supporting documents. |
| 2. The finalized contract, along with the attached documents, is linked to the corresponding purchase order and supplier/vendor record. |
| 3. Notifications are sent to the procurement team and the supplier/vendor to inform them of the finalized contract and attached documents. |
| Supplemental Requirements | 1. Document Security: The finalized contract and attached documents should be encrypted and stored securely to maintain confidentiality. |
| 2. Version Control: The system should track changes to the contract and attached documents and maintain a version history. |
| 3. Audit Trail: The system should maintain an audit trail of all document attachments, including timestamps, user IDs, and changes made. |
| Assumptions | 1. The user has the necessary permission to finalize and attach documents to the contract. |
| 2. All required bid and supporting documents are available in electronic form and ready for upload. |
| 3. The system supports multiple document formats (e.g., PDF, DOCX, XLSX). |
| Constraints | 1. The system must support file size limits for document attachments. |
| 2. Only authorized personnel should have the ability to finalize contracts and attach documents. |
| 3. The system must be able to handle multiple document uploads without performance degradation. |
| Dependencies | 1. The system must have access to the purchase order and contract records. |
| 2. The system must support the document upload feature and be able to store multiple document formats. |
| 3. The ERP system must be integrated with an email or notification service to notify stakeholders. |
| Inputs | 1. Bid documents (e.g., bid proposals, evaluation reports) |
| 2. Other supporting documents (e.g., legal certificates, insurance documents) |
| 3. Confirmation of document attachments |
| Outputs | 1. Finalized contract document with attached bid documents and other supporting documents |
| 2. Confirmation of successful attachment |
| 3. Notification sent to procurement team and supplier/vendor |
| Business Rules | 1. All required documents (bid proposals, evaluation reports, legal certificates, etc.) must be attached before the contract can be finalized. |
| 2. The system must ensure that only valid document formats (e.g., PDF, DOCX) are accepted for attachment. |
| 3. The finalized contract and attached documents must be securely stored and accessible only by authorized personnel. |
| 4. Document attachments must be tracked with version control and audit logs. |
| 5. All documents must be reviewed for completeness and accuracy before finalization. |

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 5 |
| Use Case Name | Viewing the Approved Contract in Highrise ERP |
| Brief Description | This Use Case describes how a user views the details of an approved contract in the Highrise ERP system, including the associated bid documents, terms, conditions, and other related documents. |
| Actors | 1. Procurement Manager |
| 2. Purchasing Staff |
| 3. Supplier/Vendor |
| 4. System (Highrise ERP Application) |
| Pre-Conditions | 1. The contract must have been approved and finalized in the system. |
| 2. The user must be logged into the Highrise ERP system. |
| 3. The user must have the appropriate permissions to view the approved contract. |
| Basic Flow | 1. User logs into the Highrise ERP system. |
| 2. User navigates to the "Contracts" or "Approved Contracts" section. |
| 3. The system displays a list of approved contracts. |
| 4. User searches or selects the relevant contract from the list. |
| 5. The system retrieves and displays the full contract details, including the terms, conditions, supplier/vendor information, and associated bid documents. |
| 6. User can view attached documents (e.g., bid proposals, legal documents) linked to the contract. |
| 7. If applicable, the system allows the user to download or print the contract and related documents. |
| 8. Use case ends. |
| Alternate Flow | 1. Contract Not Found: If the contract is not found in the system, the system displays: "No approved contracts found for your search criteria." |
| 2. Permission Denied: If the user does not have permission to view the contract, the system displays: "You do not have permission to view this contract." |
| 3. Missing Documents: If any bid or other related documents are missing, the system displays: "One or more related documents are missing. Please contact the procurement team." |
| 4. Server Issue: If there is a server or network issue, the system displays: "Server busy. Please try again later." |
| Post Condition | 1. The user successfully views the details of the approved contract, including any attached documents. |
| 2. No changes are made to the contract, ensuring it remains in its approved state. |
| Supplemental Requirements | 1. Security: The contract details and documents should be viewable only by authorized users based on their role or permissions. |
| 2. Document Access: The system should allow users to download or print the contract and attached documents if required. |
| 3. Audit Log: All actions related to viewing the contract (including who viewed it and when) should be logged on for audit purposes. |
| Assumptions | 1. The contract has already been approved and finalized in the system. |
| 2. The user has valid credentials and permission to access the contract. |
| 3. All relevant documents (e.g., bid documents, signed agreements) are properly attached to the contract in the system. |
| Constraints | 1. The system must be able to handle a large volume of contract data and related documents efficiently. |
| 2. Only users with appropriate permissions should be able to view sensitive contract details. |
| Dependencies | 1. The contract must have been approved and stored in the system. |
| 2. The system must be integrated with a document management system to retrieve and display contract attachments. |
| 3. The system must support file formats such as PDF, DOCX, and images for viewing and downloading contract documents. |
| Inputs | 1. Contract search criteria (e.g., contract ID, supplier name, purchase order number) |
| 2. User credentials (to validate permissions) |
| Outputs | 1. Displayed contract details (terms, conditions, bid documents, related documents) |
| 2. Option to download or print the contract and attachments |
| Business Rules | 1. Only approved contracts can be viewed; draft or pending contracts are not accessible. |
| 2. Contracts and related documents must be securely stored, and access must be controlled based on user roles. |
| 3. Any viewing or downloading of contract documents must be logged in the audit trial for future reference. |
| 4. Contracts must be displayed in their entirety, including any attached documents that were part of the approval process. |

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 6 |
| Use Case Name | MIS of Approved Contracts in Highrise ERP |
| Brief Description | This Use Case describes how users can access, generate, and view Management Information System (MIS) reports related to approved contracts in the Highrise ERP system. The MIS report provides an overview of contract performance, key metrics, and status updates. |
| Actors | 1. Procurement Manager |
| 2. Purchasing Staff |
| 3. Senior Management (e.g., CFO, CEO) |
| 4. System (Highrise ERP Application) |
| Pre-Conditions | 1. The user must be logged into the Highrise ERP system. |
| 2. The approved contracts must be entered into and stored in the system. |
| 3. The user must have appropriate permission to view MIS reports. |
| Basic Flow | 1. User logs into the Highrise ERP system. |
| 2. User navigates to the "MIS Reports" or "Contract Performance" section. |
| 3. The system displays a dashboard or list of available MIS reports related to approved contracts, including contract value, timelines, payment status, and supplier performance metrics. |
| 4. User selects the desired MIS report or generates a custom report based on specific criteria (e.g., date range, supplier/vendor, contract status). |
| 5. The system generates and displays the selected MIS report, including relevant data and charts (e.g., contract completion status, payment progress, delivery timelines). |
| 6. User reviews the MIS report to analyze contract performance, identify trends, or evaluate supplier/vendor compliance. |
| 7. The system provides options to export the report to Excel, PDF, or other formats, or to print the report. |
| 8. User may save the generated MIS report for future reference or analysis. |
| 9. Use case ends. |
| Alternate Flow | 1. No Approved Contracts: If no approved contracts are available, the system displays: "No approved contracts found for the selected criteria." |
| 2. No MIS Data Available: If the system does not have sufficient data for the requested MIS report, the system displays: "Insufficient data to generate the requested MIS report." |
| 3. Permission Denied: If the user does not have permission to view MIS reports, the system displays: "You do not have permission to access the MIS reports." |
| 4. Server or Network Issue: If there is a server or network issue, the system displays: "Server busy. Please try again later." |
| Post Condition | 1. The user has successfully viewed or generated the MIS report for approved contracts. |
| 2. The report has been exported, saved, or printed if required. |
| Supplemental Requirements | 1. Data Accuracy: The MIS reports must accurately reflect the data from approved contracts, including contract status, financial information, and other relevant metrics. |
| 2. Real-time Reporting: The system should support real-time updates of contract-related data to provide accurate and up-to-date reports. |
| 3. Data Visualization: The MIS report should include charts, graphs, or other visual elements to improve readability and analysis. |
| 4. Report Customization: The system should allow the user to filter and customize the MIS report by various parameters such as date range, supplier/vendor, and contract value. |
| Assumptions | 1. The user has appropriate permissions to generate and view MIS reports related to contracts. |
| 2. The contract-related data, such as payment status and performance metrics, is accurately updated in the system. |
| 3. The system has pre-configured templates for generating various types of MIS reports for contracts. |
| Constraints | 1. The system must be able to generate reports quickly, even if there is a large volume of data. |
| 2. Users should not be able to access sensitive contract data if they do not have appropriate permissions (role-based access control). |
| 3. The system should ensure the report data is presented in a clear and user-friendly format. |
| Dependencies | 1. The system must have access to accurate, up-to-date contract data (e.g., contract value, completion status, payment progress). |
| 2. The system must support the export and printing of reports in various formats (e.g., Excel, PDF). |
| 3. The system must have the required analytical capabilities (e.g., calculations, charts) to generate meaningful insights from the contract data. |
| Inputs | 1. Contract search criteria (e.g., contract ID, supplier/vendor, date range) |
| 2. User credentials (to validate permissions) |
| Outputs | 1. MIS report on approved contracts, showing relevant metrics and insights (e.g., contract status, financial progress, supplier performance). |
| 2. Option to export the MIS report to Excel, PDF, or print it. |
| Business Rules | 1. Only users with the appropriate permissions should have access to generate or view the MIS reports for contracts. |
| 2. The MIS reports must reflect accurate, up-to-date data on approved contracts. |
| 3. All MIS reports must be exportable in common formats like Excel and PDF for further analysis. |
| 4. Reports should include visual components like charts/graphs where appropriate to aid in decision-making. |

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 7 |
| Use Case Name | Downloading the Contract Document in Highrise ERP |
| Brief Description | This Use Case describes the process of downloading the finalized and approved contract document from the Highrise ERP system, including any associated attachments (e.g., bid documents, signed agreements). |
| Actors | 1. Procurement Manager |
| 2. Purchasing Staff |
| 3. Supplier/Vendor |
| 4. System (Highrise ERP Application) |
| Pre-Conditions | 1. The contract must be finalized and approved in the system. |
| 2. The user must be logged into the Highrise ERP system. |
| 3. The user must have appropriate permission to download the contract document. |
| Basic Flow | 1. User logs into the Highrise ERP system. |
| 2. User navigates to the "Contracts" or "Approved Contracts" section. |
| 3. The system displays a list of approved and finalized contracts. |
| 4. User searches or selects the relevant contract to download from the list. |
| 5. The system retrieves the selected contract document, including any associated bid documents or other attachments. |
| 6. User clicks the "Download" button or link for the selected contract. |
| 7. The system prompts the user to select a file format (e.g., PDF, DOCX) for the download. |
| 8. User confirms the download format, and the system starts the download. |
| 9. The contract document is downloaded to the user's device (e.g., laptop, desktop, mobile). |
| 10. Use case ends. |
| Alternate Flow | 1. No Approved Contracts: If no approved contracts are available, the system displays: "No approved contracts found for the selected criteria." |
| 2. Permission Denied: If the user does not have permission to download the contract, the system displays: "You do not have permission to download this contract." |
| 3. Document Not Found: If the selected contract document is not found in the system, the system displays: "The contract document is unavailable for download." |
| 4. File Format Error: If the selected contract document cannot be downloaded due to a format issue, the system displays: "There was an error downloading the document. Please try again later." |
| 5. Server or Network Issue: If there is a server or network issue during the download, the system displays: "Server busy. Please try again later." |
| Post Condition | 1. The contract document is successfully downloaded to the user's device in the selected format. |
| 2. The system logs the download activity for auditing purposes. |
| Supplemental Requirements | 1. Download Security: The system should ensure that contract documents are only downloadable by authorized users with the necessary permissions. |
| 2. File Integrity: The downloaded contract should not be corrupted, and the integrity of the document should be maintained. |
| 3. Audit Trail: The system should log the download activity (e.g., user ID, document ID, timestamp) for compliance and auditing purposes. |
| Assumptions | 1. The user has the appropriate permission to download contract documents. |
| 2. The contract document and any associated attachments are stored in the system in a downloadable format. |
| 3. The user has a compatible device and software to view the downloaded contract document (e.g., PDF viewer). |
| Constraints | 1. The system must handle large contract documents efficiently for download. |
| 2. The system should support various common document formats (e.g., PDF, DOCX). |
| 3. Only authorized users should be able to download contract documents. |
| Dependencies | 1. The contract document must be finalized and stored in the system. |
| 2. The system must support the download of contract documents in the desired formats (e.g., PDF, DOCX). |
| 3. The system must be integrated with the file storage service for smooth document retrieval and download. |
| Inputs | 1. Contract search criteria (e.g., contract ID, supplier/vendor name) |
| 2. User credentials (to validate permissions) |
| Outputs | 1. Downloaded contract document (e.g., PDF, DOCX) |
| 2. Audit log of the download activity |
| Business Rules | 1. Only approved and finalized contracts can be downloaded. |
| 2. The system must ensure that only authorized users with the appropriate permissions can download contract documents. |
| 3. All documents download activities should be logged for auditing and compliance. |
| 4. The system must maintain the integrity of the contract document, ensuring it is not corrupted during the download process. |

|  |  |
| --- | --- |
| Use Case Specification | Details |
| Use Case ID | 8 |
| Use Case Name | Logging Out from Highrise ERP |
| Brief Description | This Use Case describes the process of logging out from the Highrise ERP system. It ensures that the user’s session is securely terminated and that no unauthorized access occurs after the user has logged out. |
| Actors | 1. End User (e.g., Procurement Manager, Purchasing Staff, Admin) |
| 2. System (Highrise ERP Application) |
| Pre-Conditions | 1. The user must be logged into the Highrise ERP system. |
| 2. The user must have an active session. |
| Basic Flow | 1. User is logged into the Highrise ERP system. |
| 2. User clicks on the "Logout" button or selects "Logout" from the system menu. |
| 3. The system prompts the user with a confirmation message: "Are you sure you want to log out?" |
| 4. User confirms the logout action by clicking the "Confirm" or "OK" button. |
| 5. The system terminates the user’s session. |
| 6. The user is redirected to the system’s login page or landing page. |
| 7. The system clears the user's session data and authentication tokens. |
| 8. The system displays a "Logout successful" message (optional). |
| 9. Use case ends. |
| Alternate Flow | 1. Cancel Logout: If the user clicks "Cancel" instead of confirming, the system returns to the previous screen or dashboard, and the user remains logged in. |
| 2. Session Expiry: If the user’s session has expired due to inactivity, the system automatically logs the user out and displays the message: "Your session has expired. Please log in again." |
| 3. Server or Network Issue: If there is a server or network issue preventing the logout, the system displays: "Logout failed due to a network/server issue. Please try again later." |
| Post Condition | 1. The user’s session is terminated. |
| 2. The user is logged out of the Highrise ERP system, and their authentication tokens are invalidated. |
| 3. No unauthorized access can occur until the user logs in again. |
| Supplemental Requirements | 1. Session Timeout: The system should automatically log the user out after a specified period of inactivity (e.g., 30 minutes) to prevent unauthorized access. |
| 2. Security: The system must ensure that the user’s session is fully terminated to prevent session hijacking or unauthorized access. |
| 3. Audit Log: The system should log the logout activity, including the user ID, time of logout, and any errors encountered. |
| Assumptions | 1. The user has a valid session and is authorized to access the ERP system. |
| 2. The user is actively logged in and can see the logout option. |
| Constraints | 1. The logout process should be simple, quick, and secure. |
| 2. The system should ensure that the user’s session is properly cleared, and no sensitive data is retained after logout. |
| Dependencies | 1. The system must manage user sessions and track active sessions accurately. |
| 2. The system should provide a mechanism for handling user session timeouts or automatic logouts. |
| Inputs | 1. User confirmation to log out (via clicking "Logout" button or menu option). |
| Outputs | 1. Successful logout action and redirection to the login page or landing page. |
| 2. Session invalidation and clear session data. |
| 3. An audit log entry for the logout event (optional). |
| Business Rules | 1. Only authenticated users can log out of the system. |
| 2. The logout action should fully terminate the user’s session and clear any session-related data. |
| 3. The system must support automatic logout after a specified period of inactivity. |
| 4. The system must provide the user with an option to cancel the logout process if needed. |

**1.0 Activity Diagram – Registration**

A diagram of a flowchart

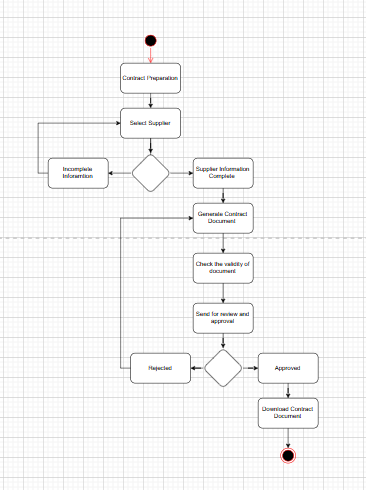
Description automatically generated

**2.0 Activity Diagram – Registration**

**A diagram of a computer program

Description automatically generated**

**3.0 Activity Diagram – Generate Contract Document**

****

**4.0 Activity Diagram – View Contract Document**

**A diagram of a document

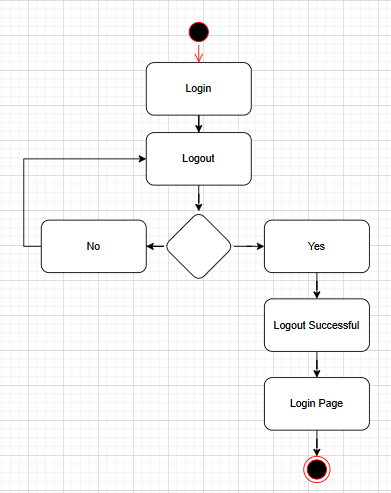
Description automatically generated**

**5.0 Activity Diagram – Generate MIS Report**

**A diagram of a report

Description automatically generated**

**6.0 Activity Diagram – ERP Log out**

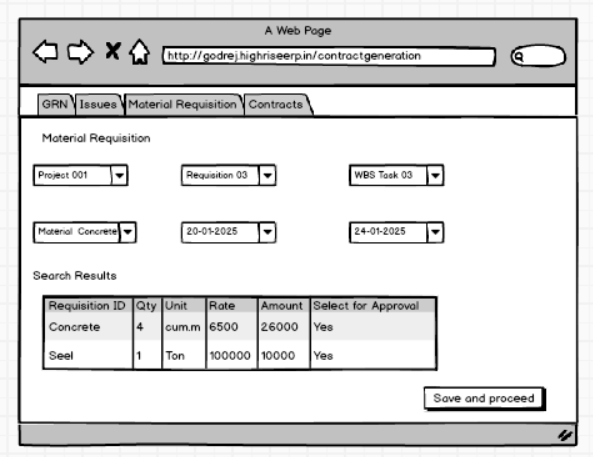
****

**Document 7- Screens and pages**

1. **Login and registration Page**

****

1. **PR Selection Page**

****

1. **Contract Preparation Page**

**A screenshot of a web page

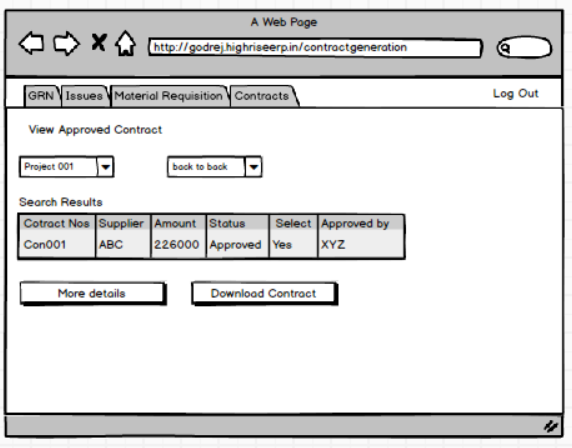
Description automatically generated**

1. **View Contract Status page**

**A screenshot of a computer screen

Description automatically generated**

1. **View Approved Contract Status**

****

1. **Log Out**

**A screenshot of a login page

Description automatically generated**

**Document 8- Tools-Visio and Axure**

Visio Usage

In Visio, I created several activity diagrams and UML diagrams to represent the various processes within the ERP system, such as user registration and contract document workflows. For instance:

* User Registration Workflow: I designed an activity diagram that mapped out the entire user registration process, from entering personal details (Name, Job Title, Company Registered ID, Email Address, and Password) to OTP verification, password validation, and account creation. This diagram provided a clear visualization of the sequence of actions and decision points in the process. I also included alternate flows, such as scenarios where an OTP is invalid or when the email address or company registered ID is already in use. This helped the team to identify possible exceptions and improve the system's error-handling mechanisms.
* Contract Document Workflow: I created another set of diagrams to represent the contract document creation, approval, and signing process within the ERP system. The diagram depicted the flow of actions, from creating the document, submitting it for approval, managing versioning, and ultimately sending it for digital signing. The activity diagram was crucial for illustrating complex interactions between different system components, such as the procurement team and the system’s document management features.

These diagrams played a vital role in helping the team understand the workflow visually, and they served as a reference throughout the project. Additionally, I used Visio to modify and update these diagrams as the project evolved, ensuring the entire team remained aligned on the system’s functionality.

Axure Usage

In Axure, I focused on creating interactive prototypes for critical ERP system functionalities, which were particularly useful for validating use cases such as contract management and user registration. For example:

* User Registration Prototype: I created an interactive prototype that simulated the user registration process. Stakeholders could interact with the UI by inputting details, receiving an OTP, and proceeding through the account validation process. This prototype was especially useful in gathering feedback on the user experience (UX) and ensuring that the design was intuitive and aligned with the business requirements. Feedback from the stakeholders led to improvements in the form field layouts, password strength indicators, and error messages, which improved the final user interface.
* Contract Document Management Prototype: I developed a detailed prototype to simulate the contract creation, review, and signing process. This included creating mock-ups of forms for entering contract details, uploading documents, and the approval flow between the procurement team and managers. The prototype allowed users to visualize and interact with the document management features, giving them a better understanding of how the system would work once fully developed. This early interaction helped surface usability concerns and made it easier to refine the design before development.

Overall Experience

Using Visio and Axure allowed me to bridge the gap between business requirements and technical design, making the process more collaborative. The activity diagrams in Visio helped break down complex workflows into easy-to-understand visual representations, ensuring that all stakeholders, including developers and business users, understood the system’s functionalities and business rules. Meanwhile, Axure served as an effective tool for prototyping, allowing the team to test and refine the user interface early in the process, which improved the overall user experience.

These tools were essential in communicating requirements, managing expectations, and ensuring the alignment of business objectives with technical execution. The ability to update and iterate on the designs quickly helped reduce errors and misalignments during the later stages of development, ultimately contributing to the success of the project.

**Document 9- BA experience**

1. Requirement Gathering:

* MOSCOW Technique: To prioritize requirements, I employed the MOSCOW technique, which helped us categorize features into Must-Have, Should-Have, Could-Have, and Won't-Have. For example, features like automated vendor contract document generation were categorized as "Must-Have", while advanced reporting features were considered "Could-Have".
* Point of Contacts (POCs): During this phase, the primary stakeholders were occasionally unavailable, so I worked with the client to identify alternative points of contact within their team to gather critical information and clarify specific details on business processes, such as how contract documents should be created and managed.
* Requirement Validation: I validated requirements using the FURPS technique, ensuring that each feature (e.g., real-time contract document tracking, vendor database integration) was Functional, Usable, Reliable, Performance-optimized, and Security-compliant.
* Prototyping: To refine and gather more specific requirements, I worked with the team to prototype key features, such as the automated contract document creation system, helping to better define user needs and expectations.

2. Requirement Analysis:

* UML Diagrams: I created UML use case diagrams to map the interactions within the ERP system. For example, I designed a "Create Contract Document" use case that outlined how a user would interact with the system, the validations required, and the resulting actions.
* Activity Diagrams: I used activity diagrams to describe key process flows like the contract document approval process. These diagrams clearly illustrate the sequential steps, decision points, and interactions, helping to visualize business processes such as vendor selection, contract generation, and approval routing.
* Feedback and Modifications: During team discussions, some members suggested modifications to the flow of the contract document approval process. As the BA, I incorporated these changes and updated the diagrams, ensuring alignment between all team members.
* BRS and SRS Documentation: I created the Business Requirements Specification (BRS) and System Requirements Specification (SRS) documents. These documents captured functional and non-functional requirements like automated contract document creation, password security policies for users, and the system's performance expectations (e.g., handling 1,000 concurrent requests).

3. Design:

* Test Case Preparation: Based on the use case diagrams, I prepared comprehensive test cases to validate functionalities like contract document creation, vendor approval, and contract lifecycle management. For instance, a test case for the "Create Contract Document" process included scenarios for valid contract creation, invalid vendor details, and missing contract terms.
* Solution Design Communication: I reviewed the solution design documents and shared these with the team. For example, the solution design for the contract document approval workflow outlined how a contract document would be reviewed, approved, and signed by stakeholders before being finalized.
* RTM (Requirements Traceability Matrix): To ensure that all business requirements were met, I maintained and updated the RTM. This document ensured that every requirement, like system scalability and contract document accuracy, was mapped to the corresponding design and testing elements.

4. Development:

* JAD Sessions: I organized Joint Application Development (JAD) sessions to clarify and finalize requirements. For example, we had a JAD session to finalize the workflow for contract document approvals, ensuring the process was both efficient and aligned with business rules.
* Clarifying Technical Queries: I acted as a liaison between the technical team and business stakeholders, answering queries related to business rules, system behavior, and process flows during coding. For example, I clarified the logic for how contract documents should be auto generated when vendor terms are met.
* Team Collaboration: I facilitated collaboration among team members, especially when technical team members had concerns or needed further clarification on process flows. In one case, there was confusion about the validation process for vendor data in contract documents, so I organized a meeting to walk through the process and address their concerns.

5. Testing:

* Test Case Execution: I prepared and executed test cases based on the use cases. For example, I tested the "Create Contract Document" functionality by simulating valid and invalid inputs (e.g., missing contract terms or incorrect vendor details) to ensure the system could handle different scenarios.
* Test Data Requests: I coordinated with stakeholders to ensure that test data (e.g., vendor lists, contract terms) was available for the testing phase. This ensured accurate validation of features like contract document creation, vendor communication, and approval processes.
* UAT (User Acceptance Testing) Preparation: I prepared the team for User Acceptance Testing (UAT), providing the necessary documentation and test data to ensure the business users could validate key processes such as contract document creation and vendor approval workflows.
* Signoff: After testing, I took the sign off on all test cases and ensured that all identified issues were addressed before moving forward to deployment.

6. Deployment:

* RTM and Project Closure Documentation: After the successful testing phase, I forwarded the final RTM to ensure that all requirements were met. This document was included in the project closure documentation to formalize the completion of the project.
* End-User Manuals: I coordinated the preparation of end-user manuals that covered key functionalities like contract document creation, vendor management, and contract approval. These manuals helped users understand how to navigate the ERP system and perform day-to-day procurement tasks.
* Training Sessions: I organized and facilitated training sessions to ensure that end-users were well-prepared to use the system. For example, I hosted a training session for the procurement staff to demonstrate how to create, review, and approve contract documents, ensuring they were familiar with the system’s functionality before the official rollout.